

UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA



TRABAJO DE GRADO

PROPUESTA DE UN MANUAL DE PRÁCTICAS PARA EL DESARROLLO DE LA
PLATAFORMA ARDUINO Y NODEMCU EN UN PROTOTIPO SIMULADO Y
PRÁCTICO PARA DIFERENTES ÁREAS DEL DEPARTAMENTO DE INGENIERÍA Y
ARQUITECTURA DE LA FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE DE LA
UNIVERSIDAD DE EL SALVADOR, C.A

PARA OPTAR AL GRADO DE
INGENIERA DE SISTEMAS INFORMÁTICOS

PRESENTADO POR

ANA MARÍA GARCÍA DE LA CRUZ

RAQUEL NOEMI RUÍZ AYALA

DOCENTE ASESOR

MAESTRO JOSÉ FRANCISCO ANDALUZ GUZMÁN

MAYO, 2020

SANTA ANA, EL SALVADOR, CENTROAMÉRICA

UNIVERSIDAD DE EL SALVADOR

AUTORIDADES



M.Sc. ROGER ARMANDO ARIAS ALVARADO
RECTOR

DR. RAÚL ERNESTO AZCÚNAGA LÓPEZ
VICERRECTOR ACADEMICO

ING. JUAN ROSA QUINTANILLA QUINTANILLA
VICERRECTOR ADMINISTRATIVO

ING. FRANCISCO ANTONIO ALARCÓN SANDOVAL
SECRETARIO GENERAL

LICDO. LUIS ANTONIO MEJÍA LIPE
DEFENSOR DE LOS DERECHOS UNIVERSITARIOS

LICDO. RAFAEL HUMBERTO PEÑA MARIN
FISCAL GENERAL

FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE

AUTORIDADES



M.Ed. ROBERTO CARLOS SIGÜENZA CAMPOS
DECANO

M.Ed. RINA CLARIBEL BOLAÑOS DE ZOMETA
VICEDECANA

LICDO. JAIME ERNESTO SERMEÑO DE LA PEÑA
SECRETARIO

ING. DOUGLAS GARCÍA RODEZNO
JEFE DEL DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA

Agradecimientos

A Dios todopoderoso, por permitirme la vida y darme las fuerzas necesarias para seguir adelante y no dejar que los obstáculos me detuvieran, la salud para luchar con energía y alcanzar mis metas, la sabiduría para lograr uno de mis sueños, por guiar mis pasos cada día y llegar al éxito.

A mi familia, por creer siempre que era posible terminar y lograr este triunfo, especialmente a mis padres por su apoyo incondicional a cada momento, por todo su amor, comprensión, sacrificios y enseñanzas, consejos, el tiempo que me han dedicado a lo largo de mi vida, así como a mis hermanos, por haber sido un ejemplo a seguir para mí.

A nuestro asesor de tesis, Ing. José Francisco Andaluz Guzmán, por su esfuerzo y dedicación, quien, con sus conocimientos, experiencia, paciencia y motivación ha logrado que pueda terminar mis estudios. También agradezco a los docentes que han aportado enseñanza en mi formación y por instruirme en el camino de la perseverancia.

A mi compañera Raquel Noemi por brindarme su confianza durante el proceso de la carrera y en este proyecto de tesis, por estudiar juntas y en armonía, por la ayuda que mutuamente hemos tenido y nos ha permitido culminar este trabajo de Grado, por su valiosa amistad, y por acompañarme en los momentos difíciles.

A mis amigos, los cuales me han brindado su apoyo, ánimo, han sido como un hermano en tiempos de angustia y han formado parte de mi vida.

Ana María García de la Cruz

Agradecimientos

Agradezco a Dios por darme la vida y por ser mi padre en todo el recorrido de este anhelado sueño y me ha brindado sabiduría apoyándome en los momentos más difíciles de mi vida, animándome a seguir adelante, a no darme por vencida, por otorgarme de buena salud y energía para culminar esta meta, además por darme la oportunidad de conocer personas que han sido de apoyo en el periodo de estudio.

Agradezco a mi familia, especialmente a mi mamá Guadalupe quien siempre ha sido mi motivación de seguir adelante dándome aliento en cada etapa de mi estudio y quien siempre me ha apoyado incondicionalmente en todos los aspectos de mi vida, por todo el esfuerzo, consejos, dedicación y paciencia, siempre será mi heroína, así como también agradecer a mi abuela Virginia por todo el apoyo brindado, paciencia, cuidados, y todos sus consejos los recuerdo con mucha nostalgia, dedico este logro en su memoria que aunque ya no esté presente conmigo sé que ella anhelaba verme triunfante. Amo a mis dos heroínas mi mamá y mi abuela.

A nuestro asesor de tesis Ing. José Francisco Andaluz Guzmán por todo su apoyo incondicional, sus conocimientos, su experiencia y paciencia para culminar mis estudios con éxito. También agradecer a docentes que han sido parte de todo el periodo de mi carrera profesional, sus consejos, por su aportación de conocimientos y enseñanzas.

A mis amigos, quienes han formado parte esencial de mi vida, compartir vivencias, experiencias y sobre todo por ser mi sostén en momentos muy duros y por darme palabras de aliento, son un regalo de Dios, gracias les doy por estar conmigo en todo momento.

También agradecer a mi compañera de tesis Ana María, quien ha sido una persona muy especial desde que la conocí, aprecio su linda amistad y por haber compartido tanto buenos como malos momentos y siempre haberme brindado su apoyo en todos aspectos de mi vida y sobre todo por permitirme ser la persona con quien pueda cumplir este éxito académico.

Raquel Noemi Ruíz Ayala

Índice

Introducción.....	xiv
Capítulo I: Generalidades	16
1.1 Antecedentes.....	17
1.1.2 Antecedentes del estudio	18
1.2 Planteamiento del problema	20
1.3 Objetivos.....	21
1.3.1 Objetivo General.....	21
1.3.2 Objetivos Específicos	21
1.4 Justificación	22
1.5 Alcances.....	23
1.6 Limitantes	24
Capítulo II: Marco Teórico.....	25
2.1. Principios básicos de Electrónica	26
2.3 Programación en lenguaje c.....	42
2.4 Arduino.....	43
2.5 NodeMCU	72
2.6 Aplicaciones con Arduino	86
2.7 Aplicaciones con NodeMCU.....	88
2.8 Aspectos fundamentales de una guía práctica	90
Capítulo III: Metodología de la Investigación.....	97
3.1 Tipo de estudio	98
3.2 Fuentes de información	98
3.2.1 Fuentes primarias.....	98

3.2.2 Fuentes secundarias	99
3.3 Técnicas de recolección de datos	99
3.3.1 Encuesta.....	99
3.3.2 Entrevista.....	100
3.3.3 Investigación documental	100
3.4 Instrumentos de recolección de datos.....	100
3.4.1 Cuestionario para encuesta	100
3.4.2 Modelo de preguntas para entrevistas	101
3.5 Determinación de la población y la muestra	101
3.6.1 Aplicación de técnicas e instrumentos de recolección de datos	102
3.6.2 Descripción de la condición actual de la población estudiada	103
3.7 Análisis e interpretación de los datos	104
3.8 Conclusiones y recomendaciones.....	104
Capítulo IV: Manual de Guías Prácticas utilizando Arduino y NodeMCU	105
4.1 Introducción.....	106
4.2 Objetivos.....	106
4.3 Entrenador Arduino y NodeMCU	106
4.4 Prácticas con Arduino Mega	120
4.5 Ejercicios con NodeMCU.....	190
Capítulo V: Manual de Usuario Arduino y NodeMCU.....	230
5.1 Introducción.....	231
5.2 Manual de usuario para Arduino	231
5.3 Introducción al manual NodeMCU	239
5.4 Objetivos del manual	239
5.5 Manual de usuario para NodeMCU.....	240

5.6 Simulador Fritzing.....	254
Conclusiones.....	265
Recomendaciones	266
Referencias Bibliográficas.....	267
Anexos.....	269

Índice de Figuras

Figura 1. Organigrama del Departamento de Ingeniería y Arquitectura.....	18
Figura 2. Ejemplos de entradas y salidas electrónicas.....	29
Figura 3. Colores de resistencias	32
Figura 4. Comparación de placas arduino.	45
Figura 5. Comparación de placa NodeMCU.	73
Figura 6. Comparación de placa NodeMCU vs Lolin.	74
Figura 7. Comparación de placa Amica vs Lolin V3.	75
Figura 8. Diagrama de conexión del módulo del sensor de lluvia al arduino	87
Figura 9. Proyecto HomeSmart	89
Figura 10. Entorno de Arduino.....	109
Figura 11. Gestor de librerías	110
Figura 12. Diagrama de sensor PIR Ultrasónico	112
Figura 13. Diagrama con botones RGB y resistencias.	112
Figura 14. Diagrama ESP32.....	113
Figura 15. Diagrama de motores	114
Figura 16. La placa con sus resistencias en uso y botones RGB.....	114
Figura 17. Servo y buzzer.....	115
Figura 18. Placa Nano	115
Figura 19. Sensor Ultrasónico	115
Figura 20. Plancha para diseño circuito impreso.....	116
Figura 21. Circuito impreso.....	116
Figura 22. Placa o superficie	117
Figura 23. PCB diseñada	117
Figura 24. Sensor de movimiento.....	118
Figura 25. Motores	118
Figura 26. Pantalla LCD.....	119
Figura 27. Elementos electrónicos	119
Figura 28. Esquema de circuito encendido y apagado led	123
Figura 29. Diagrama de conexión de circuito de led con pulsador	127
Figura 30. Diagrama de circuito de apagar y encender 3 leds.....	131

Figura 31. Diagrama de circuito de led y un pulsador.....	136
Figura 32. Esquema de circuito con led	136
Figura 33. Diagrama del circuito con sensor piezoeléctrico	141
Figura 34. Motores	176
Figura 35. Ultrasónico PIR.....	176
Figura 36. RGB	177
Figura 37. Esp 32.....	177
Figura 38. Vista de sitio web shiftr.io	191
Figura 39. Vista de sitio web nodejs.org	192
Figura 40. Ventana de instalación de node.....	192
Figura 41. Ventana de términos de licencia	193
Figura 42. Ventana de progreso de instalación.....	193
Figura 43. Ventana de finalización de instalación de node	193
Figura 44. Ventana de cmd.....	194
Figura 45. Pantalla de inicio	194
Figura 46. Ventana de node.....	195
Figura 47. Ventana de cmd con visualización de órdenes enviadas al esp32	195
Figura 48. Pantalla de aplicación Telegram	196
Figura 49. IDE arduino con opción para importar librería.....	196
Figura 50. Ventana de Gestor de tarjetas.....	197
Figura 51. Búsqueda de librería mqtt	197
Figura 52. IDE Arduino.....	198
Figura 53. Placa NodeMCU	207
Figura 54. Diagrama del circuito con conexiones	208
Figura 55. Ejecución del programa de parpadeo de led	210
Figura 56. Esquema de funcionalidad de placa NodeMCU	212
Figura 57. Diagrama de circuito control de led con servidor Access point.....	215
Figura 58. Visualización de página html con funciones.....	217
Figura 59. Diagrama de circuito con baterías.....	225
Figura 60. Diagrama de circuito Deep sleep	227
Figura 61. Placa Arduino Uno.....	232

Figura 62. Arduino Duecimila.....	232
Figura 63. Ventana de descarga de software arduino.....	233
Figura 64. Ventana de términos de licencia	234
Figura 65. Ventana de Opciones de instalación	234
Figura 66. Ventana de ubicación de destino.....	234
Figura 67. Ventana de progreso de instalación.....	235
Figura 68. Placa arduino con conexión usb.....	235
Figura 69. Menú para ejemplo de arduino.....	236
Figura 70. Ejemplo en el IDE arduino.....	237
Figura 71. Ventana de selección de puerto.....	237
Figura 72. Ventana de selección de placa.....	238
Figura 73. Visualización de placa arduino con led parpadeando	238
Figura 74. Ventana de búsqueda para variables de entorno	241
Figura 75. Ventana propiedades del sistema	241
Figura 76. Ventana de variables de entorno	242
Figura 77. Edición de variables de entorno	242
Figura 78. Edición de variables seleccionando botón nuevo.....	243
Figura 79. Ventana de inicio Windows	244
Figura 80. Ventana símbolo del sistema.....	244
Figura 81. Verificación de instalación de pip en ventana símbolo del sistema.....	245
Figura 82. Ventana administrador de dispositivos	246
Figura 83. Instalación satisfactoria.....	247
Figura 84. Visualización del firm cambiado	247
Figura 85. Programa serial teraterm	248
Figura 86. Ventana de terminal	248
Figura 87. Ventana de terminal	249
Figura 88. Ventana teraterm con operación matemática	249
Figura 89. Ventana Git Gui	250
Figura 90. Ventana clone repository	251
Figura 91. Ventana herramientas de aplicación.....	252
Figura 92. Verificación de sistema descargado	252

Figura 93. Ventana de menú herramientas	253
Figura 94. Vista Inicial de Fritzing al iniciar lo por primera vez	255
Figura 95. Vista de Protoboard (Placa de Prototipos)	256
Figura 96. Vista de Esquema.....	256
Figura 97. Vista de PCB	257
Figura 98. Vista intercambiador de vistas	258
Figura 99. Librería de Piezas.....	259
Figura 100. Inspector de Piezas.....	260
Figura 101. Historias de Acciones.....	261
Figura 102. Navegador	261

Índice de Tablas

Tabla 1. Test de Procesos de Librerías Arduino.....	48
Tabla 2.Descripción de microprocesador eeprom	49
Tabla 3. Funciones de Librería Eeprom	49
Tabla 4.Descripción de Librerías Arduino (Sd y File).....	50
Tabla 5. Biblioteca Ethernet	52
Tabla 6. Descripción de librería firmata.....	54
Tabla 7. Descripción de librería LCD crystal.....	55
Tabla 8. Descripción de la librería Servo	56
Tabla 9. Descripción de la librería stepper	57
Tabla 10. Descripción de periféricos	58
Tabla 11. Descripción de librería spi.....	59
Tabla 12. Modo de polaridad.....	60
Tabla 13. Descripción de interface de dos hilos.....	61
Tabla 14. Funciones de librería Wire	61
Tabla 15. Descripción de librería software serial	63
Tabla 16. Especificaciones del ESP8266	76
Tabla 17. Características de los pines.....	78
Tabla 18. Actividades para elaborar una guía	92

Introducción

Hoy en día, se vive en un mundo tecnológico, y nada de lo que se hace puede desligarse de ello, la tecnología es muy relevante en todos los aspectos rutinarios, en la forma como compramos y como estudiamos.

Se puede observar que la característica principal de este siglo XXI es el constante dinamismo y el ritmo acelerado de cambio en todas las áreas de la sociedad.

Debido a la constante evolución de la tecnología, se ha contribuido de manera significativa al progreso humano y en la consecución a una mejor calidad de vida. En este nuevo contexto dinámico, con información enriquecida y en constante cambio tecnológico se ha modificado la forma de desenvolverse en el nuevo entorno y se exige de nuevos conocimientos y habilidades. Por tanto, en los sistemas educativos de todo el mundo se desafía que la tecnología sea una herramienta para proporcionar a los alumnos los conocimientos necesarios para poder desempeñarse en el siglo XXI.

Es por ello que todos los conocimientos y habilidades que los estudiantes adquieran durante su educación se verán favorecidos, si además aprenden el buen uso de las tecnologías para sacar el mayor provecho de ello.

Una de las nuevas tecnologías que ha puesto a disposición una herramienta y aporta múltiples beneficios es la plataforma Arduino, la cual facilita el uso de la electrónica junto con un entorno de desarrollo para poder emplearlo en proyectos multidisciplinarios en la vida cotidiana. De esta manera se considera que es de suma importancia el proponer esta herramienta para innovar en el método enseñanza aprendizaje en diversas cátedras del departamento de Ingeniería y Arquitectura de la Facultad Multidisciplinaria de Occidente (FMOcc) de la Universidad de El Salvador.

Este anteproyecto está formado por los antecedentes de la investigación y el estudio, los cuales ofrecen un amplio panorama del estudio y exponen información que es relevante para que en apartados posteriores se justifique la necesidad encontrada de contar con un manual de prácticas para la materia de Sistemas Digitales de la carrera de Ingeniería de Sistemas Informáticos de la FMOcc. Además, se describe el contenido capitular por desarrollar a lo

largo del estudio y la metodología a aplicar, la cual representa la forma en cómo se estará abordando la investigación y la manera en que sistemáticamente irá avanzando.

Dentro de la metodología de la investigación a desarrollar se describen técnicas e instrumentos de recolección de datos tales como la encuesta y entrevistas. Estas técnicas e instrumentos se utilizarán con el propósito que sirvan como medios de recopilación de información primaria que sea útil para la investigación y que ayude a cumplir con los objetivos y alcances del estudio.

Capítulo I: Generalidades

1.1 Antecedentes

1.1.1 Antecedentes de la institución

La Universidad de El Salvador (UES), fue fundada el 16 de febrero de 1841, permitiendo con ello la Enseñanza Superior Universitaria en nuestro país; conforme el paso del tiempo, la demanda de estudiantes fue haciéndose mayor, debido a esto surge la necesidad de ampliar las alternativas para aspirantes de todas las zonas del país; es así como se da origen a los Centros Universitarios Regionales de Occidente, Oriente y paracentral El Centro Universitario de Occidente nació como una proyección de la Universidad de El Salvador, ante la necesidad de fomentar y difundir la enseñanza superior universitaria en la Zona Occidental del país.

Misión

Ubicarnos en la perspectiva de formar profesionales con alta calidad académica, moral, científica y humana; capaz de incidir en la solución de los problemas de la región occidental y de El Salvador en general.

Visión

Construir una Facultad líder en educación universitaria cuyas actividades académicas, científicas, tecnológicas y de proyección social tengan un real y significativo impacto en el desarrollo de una alta calidad educativa y, consecuentemente, en las condiciones de vida de la población y valores socialmente positivos como: la democracia, la solidaridad, la justicia, la honestidad, la tolerancia, y la investigación científica al servicio de la población.

Organigrama de la Facultad Multidisciplinaria de Occidente (FMOcc)

Como en toda entidad u organización es necesario tener una estructura jerárquica bien definida para su buen funcionamiento o administración, es por ello que la FMOcc ha definido su organigrama tal como se muestra en la siguiente figura.

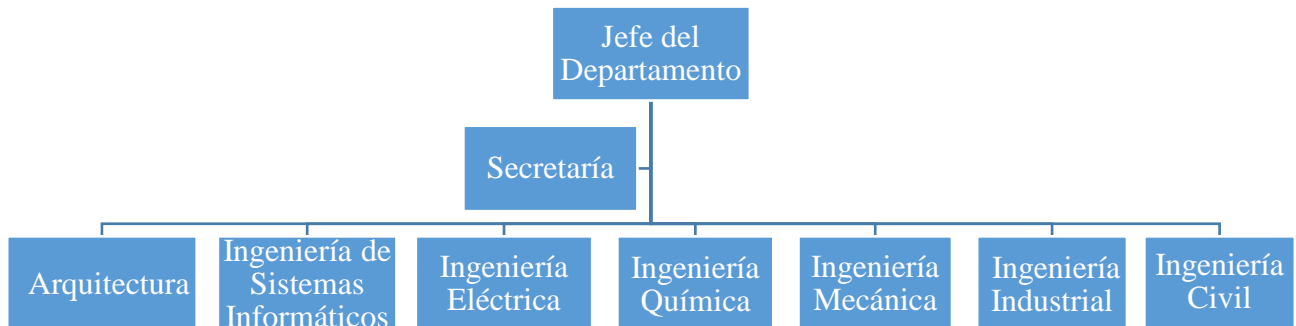


Figura 1. Organigrama del Departamento de Ingeniería y Arquitectura

1.1.2 Antecedentes del estudio

Arduino fue inventado en el año 2005 por el entonces estudiante del instituto IVRAE Massimo Banzi, quien, en un principio, pensaba en hacer Arduino por una necesidad de aprendizaje para los estudiantes de computación y electrónica.

Inicialmente fue un proyecto creado no solo para economizar la creación de proyectos escolares dentro del instituto, sino que, además, Banzi tenía la intención de ayudar a su escuela a evitar la quiebra de la misma con las ganancias que produciría vendiendo sus placas dentro del campus. (arduinodhtics, 2015)

Posteriormente Google colaboro en el desarrollo del Kit Android ADK (AccesoryDevelopment Kit), una placa Arduino capaz de comunicarse directamente con teléfonos inteligentes bajo el sistema operativo Android para que el teléfono controle luces, motores y sensores conectados al Arduino. (Lema11, 2016)

El firmware NodeMCU fue creado poco después de aparecer el ESP8266, el 30 de diciembre de 2013. Unos meses después, en octubre de 2014 se publicó la primera versión del firmware

NodeMCU en Github. Dos meses más tarde se publicaba la primera placa de desarrollo NodeMCU, denominada devkit v0.9, siendo también Open Hardware. (esp8266, 2018)

Con el paso del tiempo y la aparición de otras alternativas para programar ESP8266, como (especialmente) con C++ con el entorno del Arduino y otras como Micro Python, el interés en LUA ha disminuido considerablemente. (esp8266, 2018)

Básicamente, la placa de desarrollo NodeMCU está basada en el ESP12E y expone las funcionalidades y capacidad del mismo. Pero, además, añade las siguientes ventajas propias de placas de desarrollo:

- Puerto micro USB y conversor Serie-USB
- Programación sencilla a través del Micro-USB
- Alimentación a través del USB
- Terminales (pines) para facilitar la conexión
- LED y botón de reset integrados (esp8266, 2018)

Finalmente a mediados del 2016, salió una mejora del ESP8266 con 1 MB de flash integrada en el propio chip, llamado ESP8285, que fue el último modelo que desarrolló el fabricante antes de la salida al mercado del chip ESP32, en septiembre de 2016. (Herranz, 2019)

Domótica y control industrial

Con los conocimientos de: control de relés, sensores, envío y recepción de información en red, llegamos necesariamente a lo que muchos estábamos pensando desde el principio. La domótica es un campo en el que dispositivos como el ESP8266 facilitan el acceso a los aficionados y a las empresas que inician su andadura. (Martín, 2019).

1.2 Planteamiento del problema

Actualmente en el departamento de Ingeniería y Arquitectura, la carrera de Ingeniería de Sistemas Informáticos no cuenta con un laboratorio para desarrollar prácticas físicas o simuladas sobre Arduino y NodeMCU, en donde los alumnos puedan verificar la comprobación de la teoría.

De esta manera surge la necesidad de proponer una guía de prácticas con la plataforma Arduino y NodeMCU, que permita incursionar en aplicaciones tecnológicas como la automatización y la robótica en el Departamento de Ingeniería y Arquitectura; ya que, de esta manera no se ha construido un laboratorio para este tipo de prácticas.

1.3 Objetivos

1.3.1 Objetivo General

Elaborar un manual de prácticas para la plataforma Arduino y NodeMCU, las cuales se podrán desarrollar en un entrenador, para mejorar la enseñanza - aprendizaje en la carrera de Ingeniería de Sistemas Informáticos: caso práctico en la materia de Sistemas Digitales, Arquitectura de Computadoras y Microprogramación.

1.3.2 Objetivos Específicos

Determinar la aplicabilidad de la plataforma Arduino y NodeMCU en la materia de Sistemas Digitales de la carrera de Ingeniería de Sistemas Informáticos.

Valorar el diagnóstico realizado por medio de encuestas y entrevistas para aportar conclusiones fehacientes.

Elaborar un manual como propuesta para las prácticas de la plataforma Arduino y NodeMCU.

Construir un manual de usuario para la implementación de las prácticas, manejo de la tecnología y recursos electrónicos.

Fabricar el entrenador digital para el desarrollo de guías prácticas en la plataforma Arduino y NodeMCU en la materia de Sistemas digitales de la carrera de Ingeniería de Sistemas Informáticos del Departamento de Ingeniería y Arquitectura.

1.4 Justificación

Hoy en día, la tecnología ha tenido un avance a gran escala, tanto que lo podemos concentrar en cualquier campo que nos imaginemos, por lo tanto, el ser humano se ve en la necesidad de tener que dejar atrás el miedo y someterse a un mundo cada vez más innovado. La tecnología es una de las áreas donde se ha tenido un mayor auge, ahora ya se dispone de una gran cantidad de modalidades para mejorar la enseñanza de los docentes a los estudiantes. Teniendo bases en conocimientos tecnológicos por materias impartidas en los primeros años que son propias de la carrera de Ingeniería de Sistemas Informáticos, se propone que en la materia de Sistemas Digitales se pueda llevar la teoría a la práctica en nuevas tecnologías.

Las tecnologías actuales relacionadas en este contexto están agilizando, optimizando y perfeccionando algunas tareas que se realizan diariamente, entre una de esas tecnologías de acceso a las mejoras en las áreas de enseñanza práctica está la plataforma Arduino y NodeMCU. En el cual se invertirá económicamente porque los resultados son eficientes y confiables.

Arduino y NodeMCU es una buena manera de aprender sobre elementos electrónicos y programación, pero también aprender lo básico sobre algoritmos y sintaxis creando robots de Arduino. El código que se usa para programar placas Arduino y NodeMCU es bastante fácil para entender la lógica usada en lenguajes de programación que son más avanzados, por eso se considera de gran importancia.

El tiempo que se invertirá en realizar la investigación y aprender a desarrollar prácticas con esta nueva tecnología es adecuado para proponer soluciones a esta problemática.

1.5 Alcances

La investigación que se realizará se basará en el ámbito tecnológico y actual del uso de plataformas de hardware libre que es una placa Arduino y NodeMCU, específicamente en la Facultad Multidisciplinaria de Occidente del Departamento de Ingeniería y Arquitectura. La información obtenida se usará para el desarrollo de una propuesta de un manual de guías prácticas que beneficiará a los docentes y estudiantes.

La propuesta consistirá en:

Un manual de guías prácticas que orientarán a los docentes y estudiantes a hacer uso de elementos tecnológicos y electrónicos para aplicarlos en seguridad, iluminación, aplicaciones industriales, comerciales y familiares.

Un manual de usuario que contenga detalladamente cómo desarrollar una guía, los recursos necesarios y el medio requerido y que sea útil en el método de enseñanza y aprendizaje.

Un entrenador para el desarrollo de las prácticas propuestas en el manual que se elaborará para que contribuya a mejorar la calidad de enseñanza práctica de los docentes hacia los estudiantes de la carrera de Ingeniería y Arquitectura.

1.6 Limitantes

La experiencia en la innovación tecnológica es relativamente nueva y sobre todo en la Facultad Multidisciplinaria de Occidente, Departamento de Ingeniería y Arquitectura de la Universidad de El Salvador. Y puede haber cambios tecnológicos que pueden incidir de forma considerable cuando se esté elaborando la propuesta de los manuales de guías prácticas y de usuario.

La información recabada por medio de los métodos de investigación puede ser dudosa o falsa debido a un desinterés de los participantes de la población estudiada, así como también el tiempo de retardo al enviar los materiales solicitados al lugar de destino.

Capítulo II: Marco Teórico

2.1. Principios básicos de Electrónica

Electrónica es la que estudia y emplea sistemas cuyo funcionamiento se basa en la conducción y el control del flujo de los electrones u otras partículas cargadas eléctricamente.

El diseño y la gran construcción de circuitos electrónicos para resolver problemas prácticos forman parte de la electrónica y de los campos de la ingeniería electrónica, electromecánica y la informática en el diseño de software para su control.

La electrónica desarrolla en la actualidad una gran variedad de tareas. Los principales usos de los circuitos electrónicos son el control, el procesado, la distribución de información, la conversión y la distribución de la energía eléctrica. Estos dos usos implican la creación o la detección de campos electromagnéticos y corrientes eléctricas.

Ley de Ohm

Al analizar los principios básicos de la electrónica se tienen leyes tales como la de ohm que analiza:

La corriente (I) a través de un conductor o semiconductor en un sentido. La unidad de medida de ésta es el amperio (A). Una corriente eléctrica, puesto que se trata de un movimiento de cargas, produce un campo magnético, un fenómeno que puede aprovecharse en el electroimán, este es el principio de funcionamiento de un motor. (robótica, 2010)

El instrumento usado para medir la intensidad de la corriente eléctrica es el galvanómetro que, calibrado en amperios que circula por un conductor eléctrico es directamente proporcional al voltaje (V) e inversamente proporcional a la resistencia (R).

Voltaje es una magnitud física que impulsa a los electrones a lo largo de un conductor (por ejemplo: un cable) en un circuito eléctrico cerrado, provocando el flujo de una corriente eléctrica. Su unidad es el Voltio (V). El instrumento usado para medir el voltaje se conoce como voltímetro.

Voltaje DC es el flujo continuo de electrones a través de un conductor entre dos puntos de distinto potencial. En la corriente continua las cargas eléctricas circulan siempre en la misma

dirección, es continua la corriente mantiene siempre la misma polaridad. En la norma sistemática europea el color negro corresponde al negativo y el rojo al positivo o sencillamente se simboliza para el positivo con VCC, +, VSS y para el negativo con 0V, GND muchos aparatos necesitan corriente continua para funcionar, sobre todos los que llevan electrónica (equipos audiovisuales, computadores, etc.), para ello se utilizan fuentes de alimentación. Lo puedes encontrar en la batería, pilas, salida de los cargadores de computador. (robótica, 2010)

Voltaje AC es la corriente eléctrica en la que la magnitud y dirección varían cíclicamente. La forma de onda de la corriente alterna más comúnmente utilizada es la de una onda seno.

El voltaje AC es el que llega a la toma de electricidad de los hogares y a las empresas, es muy común encontrarla en las tomas de corriente donde se conectan nuestros electrodomésticos. Sin embargo, las señales de audio y de radio transmitidas por los cables eléctricos son también ejemplos de corriente alterna. En estos usos, el fin más importante suele ser la transmisión y recuperación de la información codificada (o modulada) sobre la señal de la AC.

Resistencia es la propiedad física mediante la cual todos los materiales tienden a oponerse al flujo de la corriente. La unidad de este parámetro es el Ohmio (Ω). Puedes encontrar resistencias en los calefactores eléctricos, tarjetas electrónicas, estufas son muy útiles para limitar el paso de la corriente.

Luego de haber analizado las variables electrónicas se procede a analizar los sistemas electrónicos. Un sistema electrónico es un conjunto de circuitos que interactúan entre sí para obtener un resultado. Una forma de entender los sistemas electrónicos consiste en dividirlos en:

Las entradas o Inputs: Son sensores (o transductores) electrónicos o mecánicos que toman las señales (en forma de temperatura, presión, humedad, contacto, luz, movimiento, pH etc.) del mundo físico y las convierten en señales de corriente o voltaje.

Por ejemplo: un sensor de temperatura, un pulsador, una fotocelda, un potenciómetro, un sensor de movimiento entre muchos más.

Las salidas u Outputs, son actuadores u otros dispositivos (también transductores) que convierten las señales de corriente o voltaje en señales físicamente útiles como movimiento, luz, sonido, fuerza, rotación entre otros. (robótica, 2010) Por ejemplo: un motor que gire, un LED o sistema de luces que se encienda automáticamente cuando esté oscureciendo, un buzzer que genere diversos tonos.

El procesamiento de señal Se realiza mediante circuitos de procesamiento de señales generalmente conocidos como microcontroladores. Consisten en piezas electrónicas conectadas juntas para manipular, interpretar y transformar las señales de voltaje y corriente provenientes de los sensores (Entradas) y tomar la respectiva decisión para generar acciones en las salidas.

Las señales electrónicas, las entradas y salidas de un sistema electrónico serán consideradas como las señales variables. En electrónica se trabaja con variables que se toman en forma de voltaje o corriente, éstas se pueden denominar comúnmente señales.

Las señales primordialmente pueden ser de dos tipos descritos a continuación:

Variable digital también llamadas variables discretas. Se caracterizan por tener dos estados diferenciados y por lo tanto se pueden llamar binarias. Siendo estas variables más fáciles de tratar (en lógica serían los valores Verdadero (V) y Falso (F) o podrían ser 1 o 0 respectivamente).

Un ejemplo de una señal digital es el interruptor del timbre de tu casa, porque este interruptor tiene dos estados pulsado y sin pulsar.

Variable análoga son aquellas que pueden tomar un número infinito de valores comprendidos entre dos límites. La mayoría de los fenómenos de la vida real son señales de este tipo (sonido, temperatura, voz, video, etc.) (robótica, 2010)

Un ejemplo de sistema electrónico analógico es un parlante, que se emplea para amplificar el sonido de forma que éste sea oído por una gran audiencia. Las ondas de sonido que son analógicas en su origen, son capturadas por un micrófono y convertidas en una pequeña variación analógica de tensión denominada señal de audio. (robótica, 2010)

Entrada/salida digital: Una señal digital es un tipo de señal generada por algún tipo de fenómeno electromagnético en que cada signo que codifica el contenido de la misma puede ser analizado en término de algunas magnitudes que representan valores discretos, en lugar de valores dentro de un cierto rango. Por ejemplo, el interruptor de la luz sólo puede tomar dos valores o estados: abierto o cerrado, o la misma lámpara: encendida o apagada.

Los sistemas digitales, como por ejemplo un microcontrolador, usan la lógica de dos estados representados por dos niveles de tensión eléctrica, uno alto, H y otro bajo, L (de High y Low, respectivamente, en inglés). Por abstracción, dichos estados se sustituyen por ceros y unos, lo que facilita la aplicación de la lógica y la aritmética binaria. Si el nivel alto se representa por 1 y el bajo por 0, se habla de lógica positiva y en caso contrario de lógica negativa.




		
Entrada	Salida	Entrada
Pulsador	Led	Red switch

Figura 2. Ejemplos de entradas y salidas electrónicas

Entrada/salida análoga: Una señal eléctrica analógica es aquella en la que los valores de la tensión o voltaje varían constantemente y pueden tomar cualquier valor. En el caso de la corriente alterna, la señal analógica incrementa su valor con signo eléctrico positivo (+) durante medio ciclo y disminuye a continuación con signo eléctrico negativo (-) en el medio ciclo siguiente.

Un sistema de control (como un microcontrolador) no tiene capacidad alguna para trabajar con señales analógicas, de modo que necesita convertirlas en señales digitales para poder trabajar con ellas. La señal digital obtenida de una analógica tiene dos propiedades fundamentales:

Valores. Qué valor en voltios define 0 y 1. En nuestro caso es tecnología TTL (0 – 5V)

Resolución analógica. nº de bits que usamos para representar con una notación digital una señal analógica

Divisor de voltaje es una configuración de circuito eléctrico que reparte el voltaje de una fuente (V_{in}) entre una o más resistencias (R_1 , R_2) conectadas en serie (una a continuación de otra). (robótica, 2010)

Convertidor análogo-digital. Un convertidor (o convertidor) analógico-digital (CAD), (o también ADC del inglés "Analog-to-Digital Converter") es un dispositivo electrónico capaz de convertir una entrada analógica de voltaje en un valor binario. Se utiliza en equipos electrónicos como computadores, grabadores de sonido y de vídeo, y equipos de telecomunicaciones. La señal analógica, que varía de forma continua en el tiempo, se conecta a la entrada del dispositivo y se somete a un muestreo a una velocidad fija, obteniéndose así una señal digital a la salida del mismo.

Modulación por ancho del pulso PWM (también conocida como PWM, siglas en inglés de pulse-width modulation) de una señal o fuente de energía es una técnica en la que se modifica el ciclo de trabajo de una señal periódica (una senoidal o una cuadrada, por ejemplo), ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

Por ejemplo: si le aplicamos PWM a un LED podemos variar su intensidad de brillo y si le aplicamos un PWM a un motor DC logramos variar la velocidad del mismo con la característica de mantener su par (fuerza) constante.

Comunicación serial es una interfaz de comunicaciones de datos digitales, frecuentemente utilizada por computadores y periféricos, donde la información es transmitida bit a bit enviando un solo bit a la vez. Uno de sus usos es monitorear a través de la pantalla del computador el estado del periférico conectado, por ejemplo: al pulsar la letra A en el teclado se debe encender un LED conectado de manera remota al computador.

Componentes Electrónicos son los que unen sus fuerzas para lograr aplicaciones fantásticas como por ejemplo el televisor o el computador, por dentro de ellos se encontrara tarjetas con resistencias, condensadores, circuitos integrados, transistores entre otros. (robótica, 2010)

Microcontrolador (abreviado μC , UC o MCU) es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Un microcontrolador incluye en su interior las tres unidades funcionales principales: unidad central de procesamiento (CPU), memoria y periféricos de entrada y salida (I/O).

Para que pueda controlar algún proceso es necesario crear y luego grabar en la memoria EEPROM del microcontrolador algún programa, el cual puede ser escrito en lenguaje ensamblador u otro lenguaje para microcontroladores; debe ser codificado en sistema numérico hexadecimal que es finalmente el sistema que hace trabajar al microcontrolador cuando éste es alimentado con el voltaje adecuado y asociado a dispositivos analógicos y discretos para su funcionamiento.

Los microcontroladores representan la inmensa mayoría de los chips vendidos, se tiene que distribuir seguramente entre los electrodomésticos del hogar una o dos docenas de microcontroladores. Pueden encontrarse en casi cualquier dispositivo electrónico como automóviles, lavadoras, hornos microondas, teléfonos, Arduino.

Los microcontroladores utilizan la mayoría para recibir señales de dispositivos de entrada/salida, con la gran ventaja de que se puede prescindir de cualquier otra circuitería externa.

Los puertos de E/S (entrada/salida o I/O) en el microcontrolador, se agrupan en puertos de 8 bits de longitud, lo que permite leer datos del exterior o escribir en ellos desde el interior del microcontrolador, el destino es el trabajo con dispositivos simples como relés, LED, motores, fotoceldas, pulsadores o cualquier otra cosa que se le ocurra al programador. (robótica, 2010)

Protoboard es una placa reutilizable usada para construir prototipos de circuitos electrónicos sin soldadura. Compuestas por bloques de plástico perforados y numerosas láminas delgadas de una aleación de cobre, estaño y fósforo.

Resistencia es un material formado por carbón y otros elementos resistivos para disminuir la corriente que pasa. Se opone al paso de la corriente. La corriente máxima en un resistor viene condicionada por la máxima potencia que puede disipar su cuerpo. Esta potencia se puede

identificar visualmente a partir del diámetro sin que sea necesaria otra indicación. Los valores más comunes son 0,25 W, 0,5 W y 1 W. (robótica, 2010)

El valor de la resistencia eléctrica se obtiene leyendo las cifras como un número de una, dos o tres cifras; se multiplica por el multiplicador y se obtiene el resultado en Ohmios (Ω).

Color de la banda	Valor de la 1ª cifra significativa	Valor de la 2ª cifra significativa	Multiplicador	Tolerancia
Negro	-	0	1	-
Marrón	1	1	10	$\pm 1\%$
Rojo	2	2	100	$\pm 2\%$
Naranja	3	3	1 000	-
Amarillo	4	4	10 000	$\pm 4\%$
Verde	5	5	100 000	$\pm 0,5\%$
Azul	6	6	1 000 000	$\pm 0,25\%$
Violeta	7	7	10000000	$\pm 0,1\%$
Gris	8	8	100000000	$\pm 0,05\%$
Blanco	9	9	1000000000	-
Dorado	-	-	0,1	$\pm 5\%$
Plateado	-	-	0,01	$\pm 10\%$
Ninguno	-	-	-	$\pm 20\%$

Figura 3. Colores de resistencias

Fuente: (robótica, 2010)

Un diodo es un componente electrónico de dos terminales que permite la circulación de la corriente eléctrica a través de él en un solo sentido. Tiene dos partes: el cátodo y el ánodo.

Transistor es un dispositivo electrónico semiconductor que cumple funciones de amplificador, oscilador, conmutador o rectificador. Tiene tres partes: la base (B), el emisor (E) y colector (C).

Actualmente se encuentran prácticamente en todos los aparatos domésticos de uso diario: radios, televisores, grabadoras, reproductores de audio y video, hornos de microondas, lavadoras, automóviles, equipos de refrigeración, alarmas, relojes de cuarzo, ordenadores, calculadoras, impresoras, lámparas fluorescentes, equipos de rayos X, tomógrafos, ecógrafos, reproductores mp3, teléfonos celulares, entre otros. (robótica, 2010)

Condensador o capacitor es un dispositivo utilizado en electrónica, capaz de almacenar energía sustentando un campo eléctrico. Está formado por un par de superficies conductoras, generalmente en forma de láminas o placas, en situación de influencia total separadas por un material dieléctrico o por el vacío. Las placas, sometidas a una diferencia de potencial, adquieren una determinada carga eléctrica, positiva en una de ellas y negativa en la otra.

Un led (Diodo emisor de luz, también "diodo luminoso") es un diodo semiconductor que emite luz. Se usan como indicadores en muchos dispositivos, y cada vez con mucha más frecuencia en iluminación. Los leds presentan muchas ventajas sobre las fuentes de luz incandescente como un consumo de energía mucho menor, mayor tiempo de vida, menor tamaño, gran durabilidad y fiabilidad.

El led tiene una polaridad, un orden de conexión, y al conectarlo al revés se puede quemar, revisa los dibujos de la parte superior para conocer a que corresponde el positivo y el negativo.

LED RGB es un LED que incorpora en su mismo encapsulado tres Leds, es RGB porque R (red, rojo), G (green, verde) y B (blue, azul) así se pueden formar miles de colores ajustando de manera individual cada color. Los tres Leds están unidos por el negativo o cátodo. (robótica, 2010)

Pulsador es un botón o pulsador es utilizado para activar alguna función. Los botones son por lo general activados al ser pulsados, normalmente con un dedo. Un botón de un dispositivo electrónico funciona por lo general como un interruptor eléctrico, es decir en su interior tiene dos contactos, si es un dispositivo NA (normalmente abierto) o NC (normalmente cerrado), con lo que al pulsarlo se activará la función inversa de la que en ese momento este realizando.

Reed switch es un interruptor eléctrico activado por un campo magnético, por ejemplo: con un imán. Cuando los contactos están normalmente abiertos se cierran en la presencia de un campo magnético; cuando están normalmente cerrados se abren en presencia de un campo magnético. Un uso muy extendido se puede encontrar en los sensores de las puertas y ventanas de las alarmas anti-robo, el imán va unido a la puerta y el reed switch al marco.

Potenciómetro es una resistencia cuyo valor de resistencia es variable. De esta manera, indirectamente, se puede controlar la intensidad de corriente que fluye por un circuito si se conecta en paralelo, o controlar el voltaje al conectarlo en serie. Son adecuados para su uso como elemento de control en los aparatos electrónicos. El usuario acciona sobre ellos para variar los parámetros normales de funcionamiento. Por ejemplo, el volumen de un radio.

Fotocelda es una resistencia, cuyo valor en ohmios varía ante las variaciones de la luz incidente. Una fotocelda presenta un bajo valor de su resistencia ante la presencia de luz y un alto valor de resistencia ante la ausencia de luz.

Pueden encontrarse en muchos artículos de consumo, como por ejemplo en cámaras, medidores de luz, relojes con radio, alarmas de seguridad o sistemas de encendido y apagado del alumbrado público de las calles. (robótica, 2010)

El zumbador o buzzer en inglés, es un transductor electro-acústico que produce un sonido o zumbido continuo o intermitente de un mismo tono. Sirve como mecanismo de señalización o aviso, y son utilizados en múltiples sistemas como en automóviles o en electrodomésticos. Inicialmente este dispositivo estaba basado en un sistema electromecánico que era similar a una campana eléctrica, pero sin el badajo metálico, el cual imitaba el sonido de una campana.

Su construcción consta de dos elementos, un electroimán y una lámina metálica de acero. El zumbador puede ser conectado a circuitos integrados especiales para así lograr distintos tonos. Cuando se acciona, la corriente pasa por la bobina del electroimán y produce un campo magnético variable que hace vibrar la lámina de acero sobre la armadura.

Motor DC el motor de corriente continua (DC) es una máquina que convierte la energía eléctrica en mecánica, provocando un movimiento rotatorio. Esta máquina de corriente continua es una de las más versátiles. Su fácil control de posición, paro y velocidad la han convertido en una de las mejores opciones en aplicaciones de control y automatización de procesos. Por ejemplo, los puedes encontrar en la tracción de los carros de juguetes de pilas o en las llantas de los robots. (robótica, 2010)

2.2 Tecnología de los Sensores

Las nuevas tecnologías están facilitando que cada vez haya más sensores a nuestro alrededor, capaces de procesar enormes cantidades de datos para ayudar a mejorar el funcionamiento de las fábricas, el control de los procesos productivos, el mantenimiento de las cosechas, o incluso para detectar terremotos.

Los sensores son cada vez más comunes en nuestra vida diaria. Un coche, por ejemplo, utiliza docenas de ellos para permitirnos controlar sus funciones básicas. Sin embargo, este tipo de sensores están muy limitados, puesto que, colocados estáticamente en un lugar, adolecen de la capacidad de analizar o actuar sobre los datos que detectan, y simplemente, su misión se limita a enviar las mediciones que han registrado a un procesador central.

Sensores de luz visible es un sensor fotoeléctrico o fotocélula es un dispositivo electrónico que responde al cambio en la intensidad de la luz. Estos sensores requieren de un componente emisor que genera la luz, y un componente receptor que percibe la luz generada por el emisor.

Fotorresistores son los sensores de luz, tal como su nombre indica, son sensores que permiten detectar la presencia de luz en el entorno. A veces se les llama “celdas CdS” (por el material con el que suelen estar fabricados, sulfuro de cadmio) o también “fotorresistores” y LDRs (del inglés “Light Dependent Resistor”), ya que básicamente se componen de una resistencia que cambia su valor dependiendo de la cantidad de luz que esté incidiendo sobre su superficie. Concretamente, reducen su resistencia a medida que reciben más intensidad de luz. (Torrente, 2013)

Suelen ser pequeños, baratos y fáciles de usar; por esto aparecen mucho en juguetes y dispositivos domésticos en general. Pero son imprecisos: cada fotorresistor reacciona de forma diferente a otro, aunque hayan sido fabricados en la misma tongada. Es por eso que no deberían ser usados para determinar niveles exactos de intensidad de luz, sino más bien para determinar variaciones en ella, las cuales pueden provenir de la propia luz ambiente (“amanece o anochece”) o bien de la presencia de algún obstáculo que bloquee la recepción de alguna luz incidente. También se podría tener un sistema de fotorresistores y comparar así cuál de ellos recibe más luz en un determinado momento (para construir por ejemplo un robot

seguidor de caminos pintados de blanco en el suelo o de focos de luz, entre otras muchas aplicaciones).

Otro dato que hay que saber es que su tiempo de respuesta típico está en el orden de una décima de segundo. Esto quiere decir que la variación de su valor resistivo tiene ese retardo respecto los cambios de luz. Por tanto, en circunstancias donde la señal luminosa varía con rapidez su uso no es muy indicado. (Torrente, 2013)

A la hora de adquirir un fotorresistor hay que tener en cuenta además otra serie de factores: aparte del tamaño y precio, sobre todo hay que mirar también la resistencia máxima y mínima que pueden llegar a ofrecer. Estos datos lo podremos obtener del datasheet que ofrece el fabricante. De hecho, en el datasheet no solo podemos consultar estos dos datos extremos sino también todos los valores intermedios de resistencia, gracias a un conjunto de gráficas que nos indican cómo varía de forma continua (y generalmente logarítmica) el valor resistivo del fotorresistor en función de la cantidad de luz recibida, medida en unidades lux.

Los sensores digitales TSL2561 son más precisos que los fotorresistores (ya que permiten lecturas exactas, medidas en unidades lux) y su sensibilidad puede ser configurada dependiendo de la intensidad de luz con la que se trabaje en ese momento (intensidad cuyo rango admitido es además mucho más amplio que el de los fotorresistores). Además, el TSL2561 concretamente detecta, además de todo el espectro visible, también la luz infrarroja; pudiéndose configurar para medir separadamente la luz visible, la luz infrarroja o ambos. Este chip se alimenta con un voltaje de entre 2,7 V y 3,6 V y funciona como mucho a 0,5 mA, por lo que es ideal para sistemas de bajo consumo. Su sistema de comunicación con el exterior es el protocolo I2C, por lo que en la plaquita breakout en la que se comercializa, además de los contactos de alimentación y tierra, aparecen los contactos “SDA” (a conectar al pin analógico nº 4 de Arduino) y “SCL” (a conectar al pin analógico nº 5 de Arduino).

El sensor analógico TEMENT6000, distribuido en forma de plaquita breakout por Sparkfun (producto nº 8688). Este sensor tiene la ventaja de ser mucho más preciso que un fotorresistor (reacciona mejor a cambios de iluminación en un rango mayor) sin añadir más complejidad a nuestros circuitos. Está adaptado a la sensibilidad del ojo humano, por lo que no reacciona ante la luz infrarroja o ultravioleta. (Torrente, 2013)

Sensores de luz infrarroja

Al analizar los sensores podemos mencionar lo siguiente:

Fotodiodos y fototransistores, un fotodiodo es un dispositivo que, cuando es excitado por la luz, produce en el circuito una circulación de corriente proporcional (y medible). De esta manera, pueden hacerse servir como sensores de luz, aunque, si bien es cierto que existen fotodiodos especialmente sensibles a la luz visible, la gran mayoría lo es sobre todo a la luz infrarroja. Se pueden adquirir en cualquier distribuidor de componentes básicos, tales como Mouser o Jameco, por poner un par de ellos. Ejemplos de dispositivos concretos que nos pueden venir bien son (el código es del fabricante) el TEFD4300F, el BPV22F, el BPV10NF o el SFH235FA.

Hay que tener en cuenta que, a pesar de tener un comportamiento en apariencia similar a los LDRs, una diferencia muy importante respecto estos (además de la sensibilidad a otras longitudes de onda) es el tiempo de respuesta a los cambios de oscuridad a iluminación, y viceversa, que en los fotodiodos es mucho menor. (Torrente, 2013)

Sensores de temperatura

Termistores es un resistor que cambia su resistencia con la temperatura. Técnicamente, todos los resistores son termistores ya que su resistencia siempre cambia ligeramente con la temperatura, pero este cambio es usualmente muy pequeño y difícil de medir. Los termistores están fabricados de manera que su resistencia cambia drásticamente, de tal manera que pueden cambiar 100 ohmios o más por grado centígrado.

Hay dos tipos de termistores, los llamados NTC (del inglés “negative temperature coefficient”) y los PTC (de “positive temperature coefficient”). En los primeros, a medida que aumenta la temperatura, decrece su resistencia; en los segundos, a medida que aumenta la temperatura, aumenta su resistencia. (Torrente, 2013)

El chip analógico TMP36. Este chip utiliza una tecnología de estado sólido para medir la temperatura: a medida que la temperatura crece, la caída de potencial entre la base y el emisor

de un transistor incrementa también una cantidad conocida. Amplificando este cambio de voltaje, se genera una señal analógica que es directamente proporcional a la temperatura. Este tipo de sensores son precisos, no se desgastan, no necesitan calibración, pueden trabajar bajo condiciones climáticas diversas, son bastante baratos y fáciles de usar. En Adafruit se distribuye con código de producto 165 y en Sparkfun con código 10988.

Otros detalles técnicos: su rango de temperatura va desde los $-4\text{ }^{\circ}\text{C}$ hasta los $125\text{ }^{\circ}\text{C}$ y su rango de voltaje de salida empieza desde $0,1\text{ V}$ (a los $-40\text{ }^{\circ}\text{C}$) y aumenta 10 mV por cada grado centígrado hasta llegar a los $1,75\text{ V}$ (a los $125\text{ }^{\circ}\text{C}$). Por otro lado, para que su circuitería interna funcione, necesita estar alimentado por una fuente de entre $2,7\text{ V}$ y $5,5\text{ V}$ y $0,05\text{ mA}$.

El chip digital DS18B20 y el protocolo 1-Wire

El termómetro digital DS18B20 es un chip que utiliza el protocolo 1-Wire. Es muy popular debido a su bajo precio y facilidad de uso. Es capaz de medir temperaturas en un rango de $-10\text{ }^{\circ}\text{C}$ hasta $85\text{ }^{\circ}\text{C}$ con una precisión de $\pm 0,5\text{ }^{\circ}\text{C}$. (Torrente, 2013)

La plaquita breakout TMP421. Modern Device distribuye una plaquita que incluye el chip analógico TMP421. (Torrente, 2013)

Debemos conectarla a los pines analógicos nº 2, 3, 4 y 5 de la placa Arduino. El primer servirá para alimentar la placa, el segundo para conectarla a tierra, y los dos últimos sirven para establecer el canal I2C a través del cual se realizará la comunicación entre ambos dispositivos. Puede medir temperaturas desde $-40\text{ }^{\circ}\text{C}$ hasta $125\text{ }^{\circ}\text{C}$ con un error de $\pm 1\text{ }^{\circ}\text{C}$.

Sensores de humedad

El sensor DHT22/RHT03 también se le puede encontrar con el nombre de AM2302. Este sensor es muy básico y lento (solo se pueden obtener datos como mínimo cada 2 segundos), pero es de bajo coste y muy manejable para obtener datos básicos en proyectos caseros.

Sus características técnicas más destacables son: se puede alimentar con un voltaje de entre 3 V y 5 V y $2,5\text{ mA}$ como máximo, puede medir un rango de temperaturas entre -40 y $125\text{ }^{\circ}\text{C}$ con una precisión de $\pm 0,5\text{ }^{\circ}\text{C}$ y un rango de humedad entre 0 y 100% con una precisión del $2-5\%$. Está formado básicamente por un sensor de humedad capacitivo y un termistor.

Sensores de distancia

El sensor digital de distancia Ping de Parallax es capaz de medir distancias entre aproximadamente 3 cm y 3 m. Esto lo consigue enviando un ultrasonido (es decir, un sonido de una frecuencia demasiado elevada para poder ser escuchado por el oído humano) a través de un transductor (uno de los cilindros que se aprecian en la figura lateral) y espera a que este ultrasonido rebote sobre un objeto y vuelva, retorno que es detectado por el otro transductor.

El sensor SRF05 otro sensor digital que utiliza el método de contar el tiempo transcurrido entre la emisión de un pulso ultrasónico y su posterior recepción para medir distancias es el SRF05 de Devantech. Es capaz de medir distancias entre 3 cm y 3 m a un ritmo de hasta 20 veces por segundo.

El sensor HC-SR04 este sensor es muy parecido a los anteriores. Dispone de cuatro pines: “VCC” (se ha de conectar a una fuente de 5 V), “Trig” (responsable de enviar el pulso ultrasónico; por tanto, se deberá conectar a un pin de salida digital de la placa Arduino), “Echo” (responsable de recibir el eco de ese pulso; luego se deberá conectar a un pin de entrada digital de la placa Arduino) y “GND” (a tierra).

El sensor LV-EZ0 Otro sensor de distancia que utiliza ultrasonidos es el sensor LV-EZ0 de Maxbotix. No obstante, a diferencia de los anteriores, el LV-EZO es un sensor analógico. Por ello, para usarlo con nuestra placa Arduino deberemos conectar (además del pin “+5 V” a la alimentación de 5V proporcionada por la placa Arduino y del pin “GND” a la tierra común) el pin etiquetado como “AN” a una entrada analógica de nuestra placa Arduino.

Sensor de inclinación

Los sensores de inclinación son pequeños, baratos, y fáciles de usar. Pueden trabajar con voltajes de hasta 24 V e intensidades de 5 mA. Constan de una cavidad y de una masa libre conductiva en su interior (como por ejemplo una bola metálica rodante); un extremo de la cavidad tiene dos polos conductivos de manera que cuando el sensor se orienta con este extremo hacia abajo, la masa rueda hacia los polos y los cierra. Por tanto, estos sensores actúan como interruptores, dejando o no pasar la corriente según la inclinación del circuito.

Aunque no son tan precisos o flexibles como un completo acelerómetro, pueden detectar orientación o movimiento fácilmente.

Sensores de movimiento

La piroelectricidad es la capacidad que tienen ciertos materiales para generar un cierto voltaje cuando sufren un cambio de temperatura. Pero ojo, si su temperatura (sea alta o baja) se mantiene constante, ese voltaje poco a poco irá desapareciendo.

¿Y esto qué tiene que ver con el movimiento? Los sensores PIR básicamente se componen de dos sensores piroeléctricos de infrarrojos. Y todos los objetos emiten radiación infrarroja, estando además demostrado que cuanto más caliente está un objeto, más radiación de este tipo emite. (Torrente, 2013)

El sensor ePIR

En Sparkfun (entre otros sitios) se puede adquirir con código de producto 9587 un sensor algo diferente llamado “ePIR”, del fabricante Zilog. La diferencia más importante entre este componente y los sensores PIR vistos anteriormente está en que el primero incluye además un microcontrolador propio dentro de su encapsulado. Esto permite una mayor flexibilidad a la hora de controlar el sensor y de gestionar los datos obtenidos. (Torrente, 2013)

Concretamente, se puede establecer comunicación con este componente de dos formas diferentes: en “modo hardware” y en “modo serie”. En el “modo hardware”, se puede ajustar la sensibilidad del sensor (es decir, a partir de qué valor detectado se considera movimiento) o el retardo (es decir, cuánto tiempo se esperará el sensor después de la detección de movimiento para volver a continuar detectando otro nuevo), entre otros parámetros.

Sensores de contacto

En los sensores de contacto se pueden mencionar los siguientes:

Sensores de fuerza estos sensores (también llamados FSRs, del inglés, “Force Sensitive Resistor”) permiten detectar fuerza. Son básicamente un resistor que cambia su resistencia dependiendo de la fuerza a la que es sometido (concretamente, su resistencia disminuye a mayor fuerza recibida). Estos sensores son muy baratos y fáciles de usar, pero no demasiado

precisos: una misma medida puede variar entre un sensor y otro hasta un 10%. Así que lo que uno puede esperar de un FSR es conseguir medir rangos de respuesta; es decir: aunque los FSRs sirven para detectar peso, son mala elección para detectar la cantidad exacta de este.

Sensores de flexión son unos sensores parecidos a los FSR son los sensores de flexión (en inglés llamados “flex sensors” o “bend sensors”). Estos sensores están compuestos por una tira resistiva flexible solo en una dirección. Su resistencia cambia según cuánto sea arqueada: si están en equilibrio (es decir, sin combarse) su resistencia es mínima y cuanto más se flexiona más resistencia ofrece. (Torrente, 2013)

Sensores de golpes se clasifican así debido a su constitución eléctrica interna, los zumbadores también pueden utilizarse, además de como emisores de sonidos, como sensores de golpes. El mecanismo es justo a la inversa del convencional: los golpes (suaves) recibidos por el zumbador se traducen en vibración de su lámina interna, la cual genera una serie de pulsos eléctricos que pueden ser leídos por una placa Arduino. De esta manera, podemos diseñar circuitos que respondan al tacto y que distingan incluso la presión ejercida. Como el zumbador es un dispositivo analógico, según lo fuerte que se golpee, la señal leída por la placa Arduino será de menor o mayor intensidad. (Torrente, 2013)

Sensores de sonido

En realidad, un sensor de sonido no es más que un sensor de presión que convierte las ondas de presión de aire (las ondas sonoras) en señales eléctricas de tipo analógico; es decir, un micrófono.

Existen muchos tipos de micrófonos según el mecanismo físico que utilizan para realizar esa conversión: los de tipo “inductivo” (también llamados “dinámicos”), los “de condensador”, los piezoeléctricos, etc. Dependiendo del tipo, unos tendrán una mejor respuesta a un rango determinado de frecuencias de sonido que otros (es decir, que serán más “fieles” a la onda original), unos tendrán una mayor sensibilidad que otros (es decir, que ya generarán un determinado voltaje a menores variaciones de volumen detectadas), unos comenzarán a distorsionar a menores volúmenes que otros (es decir, que ofrecerán una THD menor para un determinado voltaje), unos serán más resistentes y duraderos que otros, etc.

2.3 Programación en lenguaje c

La programación es un gran recurso que nos permite crear diversas secuencias de pasos lógicos que van a satisfacer nuestras necesidades y las de nuestros sistemas. Programar es todo un arte que requiere de una gran habilidad lógica y concentración por parte del programador.

Concepto de programación es el proceso de diseñar, escribir, probar, depurar y mantener el código fuente de programas computacionales. El código fuente es escrito en un lenguaje de programación. El propósito de la programación es crear programas que exhiban un comportamiento deseado.

El proceso de escribir código requiere frecuentemente conocimientos en varias áreas distintas, además del dominio del lenguaje a utilizar, algoritmos especializados y lógica formal. Programar involucra áreas como el análisis y diseño de la aplicación.

Para facilitar el trabajo de programación, los primeros científicos que trabajaban en el área decidieron reemplazar las instrucciones, secuencias de unos y ceros, por palabras o letras provenientes del inglés, codificándolas así y creando un lenguaje de mayor nivel, que se conoce como Assembly o lenguaje ensamblador. (robótica, 2010)

Lenguaje de programación es un idioma artificial diseñado para expresar operaciones que pueden ser llevadas a cabo por máquinas como los computadores. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana.

Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila y se mantiene el código fuente de un programa informático se le llama programación. (robótica, 2010)

2.4 Arduino

Es una placa de hardware libre que incorpora un microcontrolador reprogramable y una serie de pines hembra (los cuales están unidos internamente a las patillas de E/S del microcontrolador) que permiten conectar allí de forma muy sencilla y cómoda diferentes sensores y actuadores.

Arduino es una placa de “hardware libre” y también “un entorno y lenguaje de programación (es decir, software) libre”. ¿Pero qué significa aquí la palabra “libre” exactamente? Según la Free Software Foundation (<http://www.fsf.org>), organización encargada de fomentar el uso y desarrollo del software libre a nivel mundial, un software para ser considerado libre ha de ofrecer a cualquier persona u organización cuatro libertades básicas e imprescindibles:

Libertad 0: la libertad de usar el programa con cualquier propósito y en cualquier sistema informático.

Libertad 1: la libertad de estudiar cómo funciona internamente el programa, y adaptarlo a las necesidades particulares. El acceso al código fuente es un requisito previo para esto.

Libertad 2: la libertad de distribuir copias.

Libertad 3: la libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad

El objetivo del hardware libre es, por lo tanto, facilitar y acercar la electrónica, la robótica y en definitiva la tecnología actual a la gente, no de una manera pasiva, meramente consumista, sino de manera activa, involucrando al usuario final para que entienda y obtenga más valor de la tecnología actual e incluso ofreciéndole la posibilidad de participar en la creación de futuras tecnologías. (Torrente, 2013)

Básicamente, el hardware abierto significa tener la posibilidad de mirar qué es lo que hay dentro de las cosas, y que eso sea éticamente correcto. Permite, en definitiva, mejorar la educación de las personas. Por eso el concepto de software y hardware libre es tan importante, no solo para el mundo de la informática y de la electrónica, sino para la vida en general.

Arduino se compone de dos partes principales: el tablero de Arduino, que es la pieza de hardware en la que trabajas cuando construyes tus objetos; Y el Arduino IDE, el software que ejecuta en su computadora. Utilice el IDE para crear un boceto un pequeño programa de computadora) que cargue en la tarjeta Arduino. El boceto le dice al consejo qué hacer. (Massimo Banzi, 2009)

Con Arduino se pueden realizar multitud de proyectos de rango muy variado: desde robótica hasta domótica, pasando por monitorización de sensores ambientales, sistemas de navegación, telemática, etc. Realmente, las posibilidades de esta plataforma para el desarrollo de productos electrónicos son prácticamente infinitas y tan solo están limitadas por nuestra imaginación.

Ventajas de Arduino

Existen muchas otras placas de diferentes fabricantes que, aunque incorporan diferentes modelos de microcontroladores, son comparables y ofrecen una funcionalidad más o menos similar a la de las placas Arduino. Todas ellas también vienen acompañadas de un entorno de desarrollo agradable y cómodo y de un lenguaje de programación sencillo y completo. No obstante, la plataforma Arduino (hardware + software) ofrece una serie de ventajas:

Arduino es libre y extensible: esto quiere decir que cualquiera que desee ampliar y mejorar tanto el diseño hardware de las placas como el entorno de desarrollo software y el propio lenguaje de programación, puede hacerlo sin problemas. Esto permite que exista un rico “ecosistema” de extensiones, tanto de variantes de placas no oficiales como de librerías software de terceros, que pueden adaptarse mejor a nuestras necesidades concretas. (Torrente, 2013)











										
Fabricante	Arduino	Arduino	Arduino	Arduino	Arduino	Arduino	Arduino	Netduino	Texas Instruments	Fundación Raspberry Pi
Modelo	Pro Mini	Nano	Uno	Mega / Mega 2560	Leonardo	Micro	Due	Netduino 2	Stellaris Launchpad LM4F120	Raspberry Pi Mod.B
Microcontrolador	AVR Atmega 168 ó 328 8bits	AVR ATmega 168 ó 328 8bits	AVR ATmega 328 8bits	AVR ATmega2560 8bits	AVR ATmega 32u4 8bits	AVR ATmega 32u4 8bits	ARM SAM3X8E Cortex-M3 32bits	ARM STM32F2 STM32F2 Cortex-M3 32bits	ARM LM4F120HSQR Cortex-M4 32bits	ARM Broadcom BCM2835
Frecuencia	16Mhz	16Mhz	16Mhz	16Mhz	16Mhz	16Mhz	84Mhz	120Mhz	80Mhz	700Mhz
Memoria RAM	2KIB	2KIB	2KIB	8KIB	2.5KIB	2.5KIB	96KIB (64+32KIB)	60KIB	32KIB	512MIB
Memoria EEPROM	1KIB	1KIB	1KIB	4KIB	1KIB	1KIB	0	0	-	-
Memoria FLASH	16 ó 32KIB	16 ó 32KIB	32KIB	128 ó 256KIB	32KIB	32KIB	512KIB	192KIB	256KIB	-
Pines digitales entradas/salidas	14/14	14/14	14/14	54/54	20/20	20/20	54/54	20/20	43/43	8/8
Tensión/corriente pines digitales	3.3v ó 5v 40mA	5v 40mA	5v 40mA	5v 40mA	5v 40mA	5v 40mA	3.3v 3~15mA (130mA entre todos)	3.3v~5v 25mA (125mA entre todos)	5v	-
Pines analógicos entradas/salidas	6/0	8/0	6/0	16/0	12/0	12/0	12/2	6/0	-	-
Tensión/resolución pines analógicos	3.3v ó 5v 10bits (1024 valores)	5v 10bits (1024 valores)	5v 10bits (1024 valores)	5v 10bits (1024 valores)	5v 10bits (1024 valores)	5v 10bits (1024 valores)	3.3v 12bits (4096 valores)	5v 12bits (4096 valores)	-	-
Pines con interrupción externa	2	2	2	6	2	2	-	-	-	-
Pines PWM	6	6	6	15	7	7	12	6	-	-
Conexiones Serial / UART	1	1	1	4	1	1	4	4	8	Si
Conexiones I2C / TWI	1	1	1	1	1	1	2	1	4	Si
Conexiones ISP / ICSP	1	1	1	1	1	1	1	1	-	Si
Conexión USB	No (necesita adaptador externo)	Si	Si, USB-B	Si, USB-B	Si, Nativa, MicroUSB	Si, Nativa, MicroUSB	Si, Nativa, MicroUSB	Si, Nativa, MicroUSB	Si, Nativa, MicroUSB	Si, MicroUSB
Conexión USB de depuración	No	No	No	No	No	No	Si, MicroUSB	Si, MicroUSB	Si, MicroUSB	-
Conexión Bluetooth	No	No	No	No	No	No	No	No	No	-
Conexión WiFi	No	No	No	No	No	No	No	No	No	-
Conexión Ethernet	No	No	No	No	No	No	No	No	No	Si
Conexión USB Host	No	No	No	No	No	No	Si	No	Si	Si
Almacenamiento por SD	No	No	No	No	No	No	No	No	No	Si
Corriente en el pin de 5v	-	500mA	500~800mA	500~800mA	500~800mA	500mA	800mA	-	-	-
Corriente en el pin de 3.3v	-	50mA	50mA	50mA	50mA	50mA	800mA	-	-	-
Voltaje de alimentación por el USB	3.3v ó 5v (sin usb)	5v	5v	5v	5v	5v	5v	5v	5v	5v
Voltaje de alimentación recomendado por el Jack	3.35 -12 V (modelo 3.3v) ó 5 - 12 V (modelo 5v)	7~12v	7~12v	7~12v	7~12v	7~12v	7~12v	7.5~9v	-	-
Voltaje de alimentación limite por el Jack	-	6~20v	6~20v	6~20v	6~20v	6~20v	6~20v	-	-	-
Precio oficial	15+gi	-	20€+gi	40€+gi	18€+gi	18€+gi	39€+gi	~35\$+gi	13\$+gi	~43\$+gi
Precio BBB	~4€	~9€	~10€	~12€	11€~	~16€	~38€	25~30€	~15€	~35€

Figura 4. Comparación de placas arduino.

Fuente: (Torrento Peromingo, 2014)

Librerías de arduino.

Las

librerías sirven para proveer funcionalidad extra (manipulando datos, interactuando con hardware, etc.) pero también para facilitar el desarrollo de nuestros proyectos, porque están diseñadas para que a la hora de escribir nuestro programa no tengamos que hacer “la faena

sucia” de conocer todos los detalles técnicos sobre el manejo de un determinado hardware, o ser un experto en la configuración de determinado componente, ya que las librerías ocultan esa complejidad. (Torrente, 2013)

Además, la organización del lenguaje en librerías permite que los programas resulten más pequeños y sean escritos de una forma más flexible y modular, ya que solo están disponibles en cada momento las instrucciones ofrecidas por las librerías utilizadas en ese instante.

Librería de base (“core library”). La biblioteca de la base forma parte del entorno de desarrollo del arduino (ide). El propósito de esta librería es ocultar mucha de la complejidad que tradicionalmente implica el trabajar con el microprocesador. (Macho, 2020)

La mayoría de los proyectos leen datos de una de las entradas o escriben datos en una de las salidas. La librería core hace que estas tareas comunes sean simples ejecutar. Por ejemplo, para leer el valor de un pin digital basta con usar la función digitalread (por otra parte, el nombre es fácil de recordar).

Librerías estándar. Al descargar e instalar en tu ordenador el entorno de desarrollo del Arduino (ide), se incluyeron algunas librerías estándar con la instalación. Las librerías estándar son aquellas que el equipo de desarrollo de Arduino estimó que serían necesarias para los desarrollos de muchos usuarios.

Para incluir una librería basta con añadir una declaración #include al comienzo del sketch. Por ejemplo, si quieres incluir la librería de cristal líquido, que se utiliza para exhibir datos en una pantalla lcd, basta con incluir la siguiente declaración al principio de tu sketch:

```
#include <liquidcrystal.h>
```

A continuación, veremos las diferentes librerías estándar:

Librería arduinotestsuite

Librería eeprom

Librería sd

Librería ethernet

Librería firmata

Librería liquidcrystal

Librería servo

Librería stepper

Librería spi

Librería wire

Librería softwareserial

Programación en un entorno de test: arduinotestsuite

La manera en que los programas y aplicaciones para Arduino se escriben está sujeta a una continua evolución y mejora. La innovación más reciente consiste en un entorno de programación llamado test driven development (tdd, desarrollo bajo test).

En tdd, un proyecto de software se divide en pequeños módulos, y antes de escribir un módulo dado, estableces las pruebas (tests) que verifican el funcionamiento de cada función del módulo.

Cuando finalmente escribes el módulo, se ejecuta bajo un entorno de test que comprueba que los resultados obtenidos por el módulo son los descritos en el entorno de test que definiste previamente. Si no es así corriges los errores y vuelves a ejecutar el módulo en el entorno de test.

Cuando un módulo ha pasado los tests, desarrollas el siguiente módulo de la misma manera y así hasta que el proyecto haya sido completado. Si más tarde tienes que modificar uno de los módulos lo ejecutas bajo el entorno de test hasta que funcione de acuerdo con las especificaciones.

La librería arduinotestsuite proporciona métodos estándar y funciones que se pueden utilizar para probar sus sketches antes de cargarlos en el Arduino. De esta manera nos aseguramos de que el sketch funcione según lo previsto antes de usarlo en el mundo real con costosos dispositivos que pueden ser dañados si se utilizan incorrectamente.

La librería está todavía en fase de desarrollo, pero tiene actualmente tests para los procesos listados a continuación:

Tabla 1. Test de Procesos de Librerías Arduino

Test	Descripción	
Ats_begin	inicia el proceso de test	
Ats_end	termina el proceso de test	
Ats_test_digitalpin	testea un pin de entrada digital dado	
Ats_test_pwm	testea la salida pwm	
Ats_test_analoginput	testea la entrada analogical	
Ats_test_eeprom	testea la eeprom	
Ats_testserialloopback	testea los pins rx y tx de una puerta serie	
Ats_getfreememory	devuelve el número de posiciones libres de memoria en el arduino	Fuente: (Macho, 2020)

La memoria eeprom es un tipo de memoria de sólo lectura que puede ser borrada y reescrita mediante la aplicación de una tensión eléctrica y al mismo tiempo conserva los datos almacenados incluso tras desconectar la alimentación de la placa del microprocesador.

Tabla 2.Descripción de microprocesador eeprom

Microprocesador	eeprom memory (bytes)
Atmega8 and atmega168	512
Atmega328	1,024
Atmega1280 and atmega2560	4,096

Fuente: (Macho, 2020)

La cantidad de eeprom en un Arduino depende del microprocesador de base que muestra la cantidad (en octetos o bytes) de memoria eeprom disponible en diversos Arduinos.

Nota: el Arduino uno basado en el chip atmega328; tiene 1.024 bytes de memoria eeprom.

La librería eeprom proporciona sketches que permiten el acceso a la eeprom del arduino e incluye dos funciones, read y write, tal y como se muestra a continuación. (Macho, 2020)

Tabla 3. Funciones de Librería Eeprom

Function	Description
Read	lee el valor de un byte almacenado en una posición de la eeprom
Write	escribe un valor en una posición de la eeprom

Fuente: (Macho, 2020)

Nota: cualquier dato escrito en la eeprom se conserva incluso cuando el Arduino se apaga.

Un buen uso de la librería eeprom puede ser el almacenamiento de aquellos datos que serán necesarios entre dos sesiones de uso del arduino (contadores, constantes necesarias para el uso de ciertos dispositivos periféricos, como una pantalla lcd, etc.).

Si el volumen de datos que quieres almacenar entre sesión y sesión es considerable, debes de usar una memoria sd.

Almacenando datos en una tarjeta sd

El Arduino no está dotado con memoria ram ni memoria masiva para almacenar datos en grandes cantidades.

Nota: la librería utiliza mucha memoria de programa, lo cual limita el tamaño de tus sketches, por lo tanto, se recomienda el uso de esta librería solamente con un Arduino equipado de un procesador atmega328p. (Macho, 2020) .Cuando se incluye esta librería en un sketch, la librería sd proporciona una gama de funciones básicas que permiten a tu Arduino interactuar con las tarjetas sd:

Tabla

Clase	función	descripción
-------	---------	-------------

4.Descripción de Librerías Arduino (Sd y File)

	<code>begin</code>	inicializa la librería y la tarjeta sd
	<code>Exists</code>	verifica la existencia de un fichero o directorio en la tarjeta
Sd	<code>Mkdir</code>	crea un directorio en la tarjeta
	<code>Rmdir</code>	suprime un directorio en la tarjeta
	<code>Remove</code>	suprime un fichero de la tarjeta
	<code>Open</code>	abre un fichero en la tarjeta
	<code>available</code>	verifica que quedan bytes por leer en el fichero
	<code>Close</code>	Cierra un fichero y se asegura que los datos escritos en él han sido salvaguardados en la tarjeta.
	<code>Seek</code>	Busca una posición en un fichero
File	<code>Position</code>	Devuelve la posición actual dentro del fichero.
	<code>Size</code>	Devuelve el tamaño del fichero.
	<code>Print</code>	Escribe datos en un fichero abierto.
	<code>Println</code>	Escribe datos en un fichero y añade un carácter “newline” al final.
	<code>Write</code>	Escribe datos en un fichero
	<code>Read</code>	Lee un octeto (byte) de un fichero abierto.

Arduino usa un bus serie para comunicar con sus periféricos (el bus spi del que hablaremos más adelante) y en particular con la tarjeta sd (pins digitales 11, 12, y 13 en un Arduino estándar y 50, 51, y 52 en un mega).

Además, el pin 10 se usa generalmente para seleccionar el dispositivo de tarjeta en un Arduino estándar y el 53 en un mega. Podemos usar un pin diferente para esta función usando la función `sd.begin`.

Nota: antes de usar una tarjeta sd en el Arduino, debemos formatearla en formato fat16 o fat32 usando tu ordenador y el lector de tarjetas.

Existen microsdshields disponibles en el mercado. Asimismo, está disponible una gama de shields con conectores integrados microsdcard, incluyendo el último shield oficial ethernet y shields de registro de datos.

Conectándose con ethernet

Si quieres que tus aplicaciones Arduino sean accesibles a otras personas de nuestro entorno (o a través de internet) vas a necesitar la librería ethernet que te simplificará el acceso a funciones tcp/ip y te permitirá comunicar tu Arduino con internet o la red doméstica.

Esta librería está diseñada para interactuar con placas basadas en wiznet w5100. La placa oficial más reciente de Arduino lleva incorporada también un conector microsdcard que resulta muy apropiado para el registro y almacenamiento continuo de datos. (Macho, 2020)

De esta manera puedes, por ejemplo, utilizar el Arduino para tomar lecturas de datos en una ubicación remota y publicar estos datos en una página web básica mientras los almacena en un microsdcard que podría ser recuperado y analizado más adelante.

La biblioteca de ethernet es muy extensa y permite configurar el Arduino como servidor recibiendo conexiones de clientes, o como cliente, que se va conectando con uno o varios servidores. En seguida se muestran algunas de las funciones de la librería. (Macho, 2020)

Tabla 5. Biblioteca Ethernet

Class	function	description
Ethernet	begin	inicializa la librería y configura los parámetros de red
	Localip	devuelve la dirección ip local
	Dnsserverip	devuelve la dirección dns del servidor
Server	server	crea un servidor

	Begin	comienza a escuchar posibles peticiones de conexión
	Available	localiza un cliente para el que tiene datos disponibles para su lectura
	Write	escribe datos a clientes; tipo: byte o char
	Print	escribe datos a clientes; tipo puede ser: byte, char, int, long, o string
	Println	escribe datos a clientes, seguidos por un carácter “newline”
	client	crea un cliente
	Connected	devuelve true si el cliente está conectado al servidor
	Connect	conecta a la dirección ip y a la puerta especificado
Client	Write	Escribe datos a un servidor conectado.
	Print	escribe datos a un servidor, tipo puede ser: byte, char, int, long, o string
	Println	escribe datos a un servidor seguido por un carácter “newline”
	Available	devuelve el número de bytes que están listos para su lectura en el servidor
	Read	lee el siguiente octeto (byte) desde el servidor
Servidor	Flush	desestima los bytes que están en espera de ser leídos por el cliente
	Stop	se desconecta del servidor

Fuente: (Macho, 2020)

Comunicaciones serie con firmata

Firmata es un protocolo de comunicaciones que permite al ordenador (tu pc o laptop) comunicar con uno o varios Arduinos y controlar sus microprocesadores desde el ordenador.

La librería firmata proporciona los protocolos de comunicación serie (métodos) que permitirán comunicar al pc con el Arduino. Usando firmata podrás controlar servos, motores, pantallas, leds, etc desde tu pc a través de uno o varios Arduinos. En seguida se muestran los protocolos o métodos más usuales de firmata:

Tabla 6. Descripción de librería firmata

	Método	descripción
Común	begin	inicializa la librería firmata
	Printversion	envía versión del protocolo al pc
	Setfirmwareversion	establece la versión del firmware
Enviar mensajes	sendanalog	envía un mensaje analógico
	Senddigitalportpair	envía el valor de un pin digital
	Sendsysex	envía un comando con un array de bytes
	Sendstring	envía un string al pc
Recibir mensajes	available	comprueba que hay mensajes en el buffer de entrada
	Processinput	procesa los mensajes entrants
	Attach	asocia una función a un cierto

El protocolo firmata esta evolucionado constantemente; en el siguiente enlace <http://firmata.org/wiki/> se pueden obtener las actualizaciones más recientes sobre el protocolo. (Macho, 2020)

Visualizando datos usando la librería liquidcrystal

Muy frecuentemente queremos visualizar la información generada no en la pantalla del pc, sino en una pequeña pantalla lcd de 2 filas de 6 caracteres para mostrar la información a los usuarios de la aplicación (sin necesidad de disponer de un pc).

La herramienta que necesitamos para poder usar estas pantallas desde nuestro sketch es la librería liquidcrystal. En seguida se enumeran algunas de las funciones disponibles en la librería. (Macho, 2020)

Tabla 7. Descripción de librería LCD crystal

Función	Descripción
Begin	establece las dimensiones en filas y columnas de la pantalla lcd
Liquidcrystal	Inicializa la librería y determina los pins usados para comunicar con la pantalla lcd.

Print	visualiza datos en la pantalla lcd
Clear	borra los contenidos de la pantalla lcd
Setcursor	posiciona el cursor sobre la pantalla

Fuente: (Macho, 2020)

Controlando un motor servo

Los motores servo se usan generalmente para controlar con precisión movimientos dentro de nuestros modelos (por ejemplo en radio-control, las aletas de un aeroplano, o la dirección de un volante o timón, etc.).

Son ideales para los proyectos que necesitan el movimiento exacto, como por ejemplo evitar obstáculos o colocar una pieza en un ángulo pre-determinado en aplicaciones robóticas, etc.

La librería servo permite al Arduino controlar hasta 12 motores servos con un Arduino estándar, y 48 con un mega. En seguida se muestran las principales funciones proporcionadas por la biblioteca servo. (Macho, 2020)

Tabla 8. Descripción de la librería Servo

Función	Descripción
Attach	Asigna el servo a un pin.
Attached	verifica que el servo está conectado al pin
Detach	Desconecta el servo del pin.
Read	Lee el ángulo del servo.
Write	Hace girar el eje del servo al ángulo especificado. Entre 0 y 180 en un servo clásico. En un servo de rotación continua establece la velocidad de rotación.
Writemicroseconds	Gira el eje del servo un arco expresado en microsegundos.

Fuente: (Macho, 2020)

Nota: el uso de la librería servo en un arduino estándar deshabilita la función analogwrite en los pines 9 y 10 del pwm. En el mega, se deshabilita analogwrite en los pins 11 y 12 si estamos controlando más de 12 servos.

Controlando el giro de un motor de pasos

Un motor de pasos gira su eje en pasos. La especificación de un motor de pasos viene dada generalmente en pasos, así que un motor con una especificación de 200 pasos necesitaría 200 pequeños giros angulares para completar una revolución. (Macho, 2020)

La especificación se da a veces grados que se pueden convertir fácilmente en pasos dividiendo una revolución (360 grados), por el número de grados dados para cada paso en la especificación. Para un motor de pasos con una especificación de 1,5 grados calculamos el número de pasos por la revolución como sigue:

$$360 \text{ grados} / 1,5 \text{ grados por paso} = 240 \text{ pasos}$$

La librería stepper

La librería atepper nos proporciona la manera de controlar motores de pasos unipolares y bipolares con un Arduino.

Usando la librería podemos establecer la velocidad de la rotación del motor el número de pasos que queremos dar y la dirección de estos pasos. A continuación, se muestran las principales funciones de la librería stepper:

Tabla 9. Descripción de la librería stepper

Función	Descripción
Stepper	Inicializa la librería stepper y establece el número de pasos por revolución.
Setspeed	establece la velocidad de rotación del motor en revoluciones por minute (rpm)
Step	Gira el motor el número de pasos indicado. Un número positivo gira en un sentido y uno negativo en el contrario.

Fuente: (Macho, 2020)

Comunicando con periféricos sp

El bus serie para periféricos (denominado con las siglas spi en inglés), también llamado el bus de 4 hilos (four-wire bus) es un protocolo de comunicaciones síncrono usado para comunicar en distancias cortas con periféricos externos como sensores, incluyendo los sensores de temperatura, sensores de presión, convertidores analógico- digital, pantallas táctiles, controladores de videojuegos, y pantallas de visualización. Arduino usa también el bus spi para comunicar con tarjetas de memoria sd.

Este protocolo tiene un solo master, el Arduino, y uno o más dispositivos esclavos. Debido a la falta de un estándar formal, han surgido diferentes implementaciones del protocolo spi según los diversos fabricantes de dispositivos periféricos, es decir, que probablemente tendrás que recurrir a las especificaciones del periférico en cuestión para manejarlo desde el Arduino.

El protocolo utiliza cuatro hilos, de los cuales tres son comunes a cada dispositivo y el cuarto que es específico (no compartido) para cada periférico. Sus designaciones se muestran a continuación:

Tabla 10. Descripción de periféricos

Designación	descripción	pin arduino	pin mega
Miso	master in slave out, envío datos al master	12	50
Mosi	master out slave in, envío datos al esclavo	11	51
Sck	reloj serie	13	52
Ss	selección de esclavo	normalmente10	53

Fuente: (Macho, 2020)

Cada esclavo tiene un hilo “selectwire” específico para él, pero comparte los otros tres hilos del bus con los otros periféricos. El pin 10 (53 en el mega) se usa generalmente como línea de

selección de esclavo (“slave select”), pero los otros tres pueden ser elegidos durante la configuración. (Macho, 2020)

La librería spi proporciona las funciones que nos permitirán actuar sobre los periféricos spi tal y como se muestra a continuación:

Tabla 11. Descripción de librería spi

Función	descripción
Begin	inicializa el bus spi y pone los pins mosi y sck en baja y el ss en alta
End	desactiva el bus spi
Setbitorder	establece el orden en el cual se cargan los bits en el bus
Setclockdivider	establece el divisor de reloj spi como una fracción del reloj del sistema
Setdatamode	establece el modo de datos en el bus spi
Transfer	transfiere un byte al bus

Fuente: (Macho, 2020)

Veamos más en detalle alguno de estas funciones:

Setbitorder: establece el orden en el que los datos son puestos en el bus. Hay dos posibilidades: empezar por el bit de mayor peso (most significant bit, msb) o por el de menor peso (least significant bit, lsb). Las especificaciones del periférico concreto te dirán cuál de estas dos opciones debes de usar.

Setclockdivider: establece la velocidad a la que opera el bus spi como una fracción del periodo del reloj del sistema. El divisor por defecto es 4, lo cual reduce la velocidad del bus spi a un cuarto del periodo del reloj del sistema; otros ratios posibles: son 2, 8, 16, 32, 64, y 128. (Macho, 2020)

Setdatamode: controla el modo de transmisión de datos entre el periférico esclavo y el master. Existen tres modos de transmisión que vienen dados por cuando se ponen los datos en el bus:

cuando el reloj del sistema pasa a alta, cuando pasa a baja (estos dos modos se denominan “fase de reloj” (clock phase). El tercer modo es cuando el reloj está estable en el momento de poner los datos en el bus. Este es el modo de “polaridad de reloj” (clock polarity). A continuación, se resumen estos modos de transmisión.

Modo polaridad del reloj (cpol) fase del reloj (cpha)

Tabla 12. Modo de polaridad

0	0	0
1	0	1
2	1	0
3	1	1

Fuente: (Macho, 2020)

Todo esto parece un poco complicado, sin embargo; verás que leyendo las especificaciones del periférico y siguiendo sus instrucciones con cuidado podrás comunicar con el bus spi sin problemas.

Comunicando con la interface de dos hilos (i2c)

Esta interface comúnmente llamado interface de dos hilos (twi, two wire interface) se usa para comunicar a baja velocidad con una amplia gama de dispositivos y componentes (incluyendo relojes de tiempo real).

Es muy indicado para aplicaciones de registro de datos (logging applications), pantallas lcd, sensores ultrasónicos para medir distancias y potenciómetros digitales cuya resistencia pueda ser leída o regulada por control remoto. (Macho, 2020)

I2c se usa en los mandos Nintendo, el wii motion plus, y wii nunchuks.

Solo se necesitan dos pins para la interface con el bus i2c (ver tabla 13).

Tabla 13. Descripción de interface de dos hilos

Arduino estándar	mega	
Sda data line	analog input pin 4	digital pin 20
Scl clock line	analog input pin 5	digital pin 21

Fuente: (Macho, 2020)

Con la librería wire

El Arduino puede actuar como master o esclavo. Lo más frecuente es que Arduino haga de master e interactúe con uno o varios esclavos en el bus i2c; cada esclavo tiene una dirección única que lo distingue en la red del bus. Se pueden concatenar hasta un máximo de 112 dispositivos en el bus. El Arduino puede también operar como dispositivo esclavo interaccionado con un master. (Macho, 2020)

Se listan las funciones más importantes de la librería wire:

Tabla 14. Funciones de librería Wire

Function	Description
Begin	Inicializa la librería wire y conecta el arduino al bus i2c como master o esclavo.
Requestfrom	Pide datos del esclavo al master.
Begintransmission	Prepara la transmisión de datos.
Send	Envía datos del esclavo al master o pone en cola bytes para la transmisión de master a esclavo.
Endtransmission	Finaliza la transmisión (comenzada con begintransmission) de datos a un esclavo y envía los datos puestos en cola de envío por send.
Available	Devuelve el número de bytes disponible para su recepción con receive. Debe de ser usado en el dispositivo master tras una llamada requestfrom de

un esclavo dentro de la función activada por onreceive.

Receive	Requestfrom o desde el master a un esclavo.
Onreceive	Determina la función que se ha de activar cuando el esclavo recibe una transmisión desde el master.
Onrequest	Determina la función que se ha de activar cuando el master pide datos al dispositivo esclavo.

Fuente: (Macho, 2020)

Creando más puertas serie con software serial

Muchos de los proyectos que podemos realizar con el arduino requieren al menos una puerta serie. Los dispositivos gps, las pantallas lcd y otros dispositivos necesitan conectarse a una puerta serie. Esta puerta consiste en dos conexiones: una rx para recibir mensajes y una tx para enviarlos. (Macho, 2020)

La utilidad de esta puerta radica en su simplicidad. En el pasado todos los ordenadores tenían una o varias puertas serie. De hecho, el Arduino original usaba una puerta serie para conectarse al ordenador y, aunque actualmente la conexión se realiza a través de un usb, lo hace a través de una emulación de puerta serie (que vista desde el ordenador es una conexión serie).

Los Arduino uno y duemilanove tienen una puerta serie conectado a los pines digitales 0 y 1, pero si tu proyecto necesita conectarse a más dispositivos serie (por ejemplo: un gps y un display lcd) puedes hacer dos cosas: comprarte el Arduino mega (o compatible) que tiene 4 puertas serie o usar la librería softwareserial que se distribuye con el entorno de desarrollo del arduino (ide). (Macho, 2020)

La librería original solo proporcionaba una puerta software adicional además de la puerta hardware de serie y su velocidad estaba limitada a 9600 baudios. Estas limitaciones fueron superadas por la nueva librería newsoftserial escrita por mikal hart que fue incorporada por el

equipo de desarrollo del arduino a medidos de 2011 como la nueva softwareserial. En seguida se muestran las funciones incluidas en esta librería:

Tabla 15. Descripción de librería software serial

Función	descripción
Begin	activa la puerta y establece la velocidad de transmisión en baudios
Available	comienza a usar la puerta
Istlistening	devuelve la puerta active en este momento
Listen	escucha a esa puerta y lo activa
End	termina el uso de esa puerta
Read	lee datos de la puerta
Write	escribe datos en la puerta

Esta nueva librería puede crear varias puertas serie software que pueden comunicar a velocidades de hasta 115,000 baudios. Pero toda esta funcionalidad adicional tiene un precio: el arduino sólo puede escuchar o recibir datos de una puerta serie software a la vez (sin embargo, sí que puede transmitir por varias simultáneamente).

Cuando uses la librería con más de una puerta serie software tienes que planificar cuidadosamente tu sketch y el orden en el que recibirás los datos.

Consideremos un ejemplo: quieres conectar un gps y un termómetro usando puertas serie software. Los dispositivos gps suelen enviar sus datos en paquetes espaciados un segundo entre sí, de manera que tu sketch puede comenzar escuchando la puerta serie software y una vez ha recibido el primer paquete de datos pasar a escuchar la otra puerta software, procesar sus datos y devolver la escucha al gps. (Macho, 2020)

Este sketch muestra cómo funciona esto en la práctica:

```
#include <softwareserial.h>

Softwareserial gpsport(2, 3);

Softwareserial thermport(4, 5);

Void setup()

    { gpsport.begin(9600);

      Thermport.begin(9600);

    }

Void loop()

    {

      Gpsport.listen();

      While (gpsport.available() > 0)

        { char inbyte = gpsport.read();

          }

      Thermport.listen();

      While (thermport.available() > 0)

        { char inbyte = thermport.read();

          } }

}
```


Como se puede ver la librería software serial es una contribución fantástica a tu entorno de desarrollo, pero se tiene que tener cuidado al usarla con más de una puerta software.

Esto cierra el capítulo de librerías estándar. Se puede preguntar, qué sucede si quieres usar periféricos o dispositivos que no están cubiertos por las librerías estándar o de base (core).

Casi con seguridad existe una librería escrita por otro usuario que se puede adecuar a un proyecto. (Macho, 2020)

Las “contributed libraries”

Estas librerías son contribuciones de los usuarios del arduino que no se distribuyen como parte del entorno de desarrollo ide. Puedes encontrar muchas de estas librerías listadas en el sitio web Arduino.

Algunas de estas librerías son extensiones de las librerías estándar que añaden algunas funciones suplementarias, y con el tiempo, si estas funciones resultan ser útiles, el equipo de desarrollo puede añadirlas a las librerías estándar o incluso a la librería de base. Otras “contributed libraries” están diseñadas para funcionar con dispositivos específicos como consolas de juegos.

Dado que estas librerías no forman parte del ide vas a necesitar hacer una serie de operaciones previas para poder trabajar con ellas. En el siguiente capítulo explicamos cómo proceder.

Instalando una nueva librería en ide

Para añadir una nueva librería al entorno de desarrollo ide basta con realizar las siguientes operaciones:

Bájate la librería (que va a venir normalmente como un fichero zip).

Se debe instalar en el ide copiando los ficheros extraídos del fichero zip al directorio establecido por defecto para las librerías. Si dicho directorio no existe tienes que crearlo.

Si el ide está ya funcionando tienes que reiniciarlo. Tras reiniciar el ide la librería debiera estar disponible para inclusión en tus sketches.

Añade la librería a tu sketch seleccionando sketch > import library del menú

Nota: una vez añadida la librería al ide, queda disponible para su uso con futuros proyectos, de la misma manera que las librerías estándar. Una vez importada una librería en un sketch, todas sus funciones están disponibles en el sketch. (Macho, 2020)

Utilización de librerías

Una vez creada la librería y almacenados sus ficheros en el directorio apropiado, puedes usarla en tus sketches y distribuirla de manera que otros usuarios puedan también trabajar con ella.

Utilización de una librería en un sketch

El siguiente sketch muestra como la librería detmet sería usada desde un sketch, con el Arduino conectado al detector de metales equipado de un sensor metdet:

```
#include "metdet.h"          // (1) incluye librería metdet  
Metdet mymetdet           // (2) inicializa instancia de la clase metdet
```

```
Void setup()
```

```
{  
    // Inicializa el sensor metdet  
    Mymetdet.begin();  
}
```

```
Void loop()
```

```
{  
    // Actualiza lectura del sensor  
    Mymetdet.update()  
}
```

Notas: es necesario incluir la librería metdet (1) antes de inicializar cualquier instancia de la clase metdet (2). El sensor metdet se inicializa durante la fase setup y se actualiza en el bucle loop del sketch. (Macho, 2020)

Distribución de una librería

Si quieres compartir tus librerías es aconsejable empaquetarlas en un fichero zip con los ficheros de la librería (.h y .cpp) junto con algunos sketches de ejemplo. Los ejemplos deben de ubicarse en un subdirectorio de ejemplos:

Libraries/metdet/examples/metdet.ino

Cuerpo de un programa en Arduino

Arduino se programa en el lenguaje de alto nivel C/C++ y generalmente tiene los siguientes componentes para elaborar el algoritmo:

- Estructuras
- Variables
- Operadores matemáticos, lógicos y booleanos
- Estructuras de control (Condicionales y ciclos)
- Funciones
- Estructuras, y variables para programar el Arduino

Estructuras

Son dos funciones principales que debe tener todo programa en Arduino:

```
setup(){  
}
```

Código de configuración inicial, solo se ejecuta una vez. (robótica, 2010)

```
loop(){  
}
```

Esta función se ejecuta luego del setup(), se mantiene ejecutándose hasta que se des- energice o desconecte el Arduino.

Variables

Es un dato o conjunto de datos que cambia su valor con la ejecución del programa

Booleano

True ó false

Boolean encendido=true;

Entero

Valor entero

int conta=5;

Carácter

Almacena un ASCII

char letra='a';

Operadores booleanos

Usados generalmente dentro del condicional If (robótica, 2010)

&& (y)

|| (o)

! (negación)

Operadores de comparación

Usados generalmente dentro del condicional If y sobre el For y While

== (igual a)

!= (diferente de)

< (menor que)

> (mayor que)

<= (menor o igual)

>= (mayor o igual)

Operadores matemáticos

Se aplican al manejo de variables, condicionales y ciclos. (robótica, 2010)

= (asignar)

% (módulo)

+ (suma)

- (resta)

* (multiplicación)

/ (división)

Estructuras de control

Son instrucciones que nos permiten tomar decisiones y hacer diversas repeticiones de acuerdo a unos parámetros, dentro de las más importantes podemos destacar: (robótica, 2010)

If

Switch/case

For

While

Condicionales

Ideales para tomar decisiones luego de evaluar las condiciones lógicas:

= (asignar)

% (módulo)

+ (suma)

- (resta)

* (multiplicación)

/ (división)

int valor = valor +5

If

Switch/case

For

While

Ciclos

Ideales para repetir lo que se encuentre dentro de ellos: (robótica, 2010)

If (Si)

```
    if (entrada < 500)
    {
// acción A
} else
{
// acción B
}
```

For (por)

```
for( int a=0; a>10; a++ )
{
// acción a repetir
}

While (mientras)
while ( var < 200) {
// acción a repetir
```

Switch/case (Casos)

```
switch (var) {
case 1:
// acción A
```

```
break;
case 2:
// acción B
break;
default:
// acción C
}
```

Funciones

Una función es un conjunto de líneas de código que realizan una tarea específica y puede retornar un valor. Las funciones pueden tomar parámetros que modifiquen su funcionamiento. Las funciones son utilizadas para descomponer grandes problemas en tareas simples y para implementar operaciones que son comúnmente utilizadas durante un programa y de esta manera reducir la cantidad de código. (robótica, 2010)

Cuando una función es invocada se le pasa el control a la misma, una vez que ésta finalizó con su tarea el control es devuelto al punto desde el cual la función fue llamada.

Funciones digitales

Orientas a revisar el estado y la configuración de las entradas y salidas digitales

pinMode()

Permite configurar un pin

pinMode(pin,modo)

pinMode (13,OUTPUT);

pinMode (a,INPUT);

digitalRead()

Leer un pin digital (0 ó 1)

`digitalRead(pin)`

`int a = digitalRead (13);`

`digitalWrite()`

Escribir un pin digital con 1 ó 0

`digitalWrite(pin,estado)`

`digitalWrite (13,HIGH);`

`digitalWrite (13,LOW);`

Funciones análogas

Ideales para la lectura y escritura de valores análogos

`analogRead()`

Leer un valor análogo 0 a 1023

`analogRead(pin)`

`int a = analogRead (A0);`

`analogWrite() —> PWM`

Escribir un valor análogo 0 a 255

`analogWrite(pin,valor de PWM)`

`analogWrite (9, 134); (robótica, 2010)`

2.5 NodeMCU

NodeMCU (Unidad Microcontroladora de Nodos) es un entorno de desarrollo de hardware y software de código abierto que está construido en torno de un sistema en chip (SoC) barato llamado ESP8266. (Yuan, 2017)

Tipos de NodeMCU

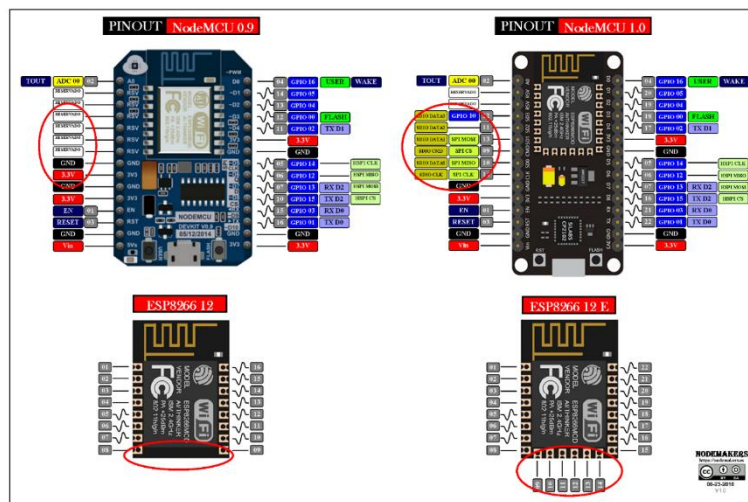
Para empezar a comparar las placas NodeMCU que hay en el mercado, en primer lugar, debemos saber que actualmente se está comercializando la segunda generación (1.0), conocida como V2 o V3 en función del fabricante. Estas placas integran el procesador Esp8266-12e. La primera generación (0.9), conocida como V1, integraba el modelo Esp8266-12. (No, 2016)

Esta evolución ha permitido mejoras sobre el diseño inicial, aunque su esencia se ha mantenido. Los cambios vienen de los seis pines adicionales de que dispone el ESP8266-12E respecto a su predecesora, siendo los siguientes:

Un nuevo bus de comunicación SPI (CS0, MIS0, MOSI y SCLK) cuatro pines.

Dos nuevos GPIO (GPIO09 y GPIO10).

De manera complementaria, los seis pines, en conjunto, permiten iniciar el procesador desde una tarjeta SD –modo SDIO- (CMD, CLK, DATA0, DATA1, DATA 2 y DATA3).



En la imagen se puede realizar una comparativa de la funcionalidad de las placas de la primera y segunda generación. También se puede apreciar que la placa de la segunda generación ha reducido su anchura. Figura 5. Comparación de placa NodeMCU.

Recuperado de <https://www.esploradores.com/comparacion-de-placas-nodemcu/>

Amica (como empresa diseñadora de las placas) y DOIT/SmartArduino se han mantenido en este estándar de fabricación, mientras que LoLin/WeMos ha mejorado ligeramente la placa (renombrándola como V3). Los cambios que ha realizado son los siguientes:

Ha puesto un nuevo pin de tierra en un pin reservado.

Ha puesto un nuevo pin conectado con la tensión de alimentación del USB (+5 V) en un pin reservado.

En la imagen se pueden apreciar los cambios.

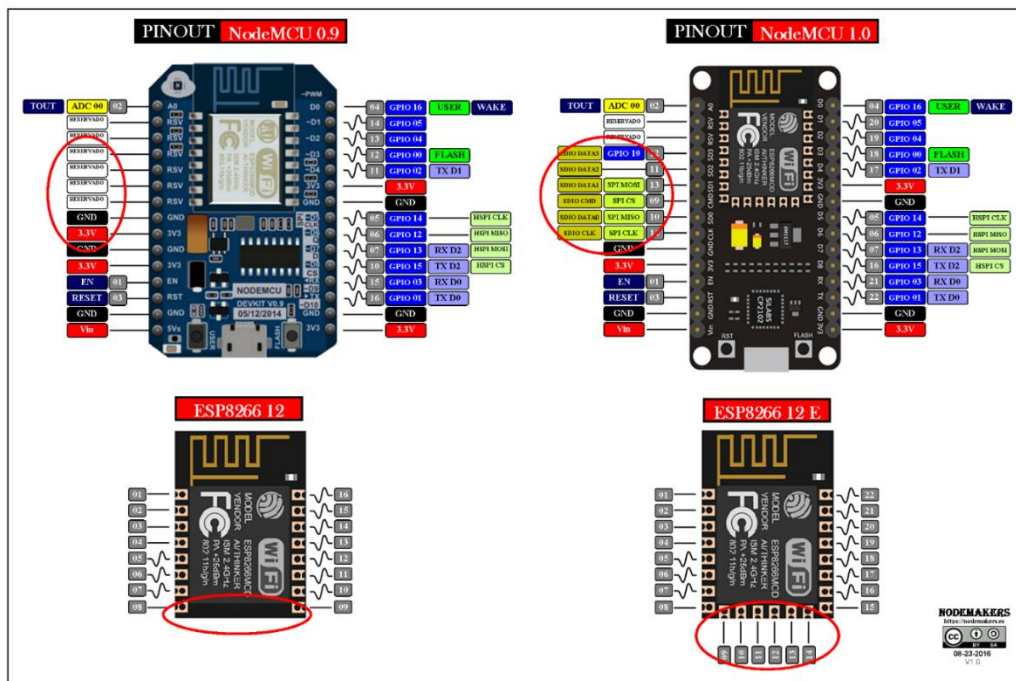


Figura 6. Comparación de placa NodeMCU vs Lolin.

Recuperado de <https://www.esploradores.com/comparacion-de-placas-nodemcu/>

Claramente también se ve que la placa fabricada por LoLin/WeMos es más grande que la de Amica y DoIt/SmartArduino. Esto dificulta enormemente su uso (al igual que ocurría con las placas de la primera generación), debido a que la placa LoLin/WeMos es una placa de

pruebas (protoboard), no deja espacio para la conexión de cables. Las alternativas son las de utilizar la placa base de la que dispone el fabricante o juntar dos placas de pruebas. (No, 2016)

En la actualidad se espera la tercera generación de placas NodeMCU que integrarán el nuevo procesador ESP32. Este procesador reemplaza al ESP8266-12E y su fabricación comenzó el 01/09/2016.

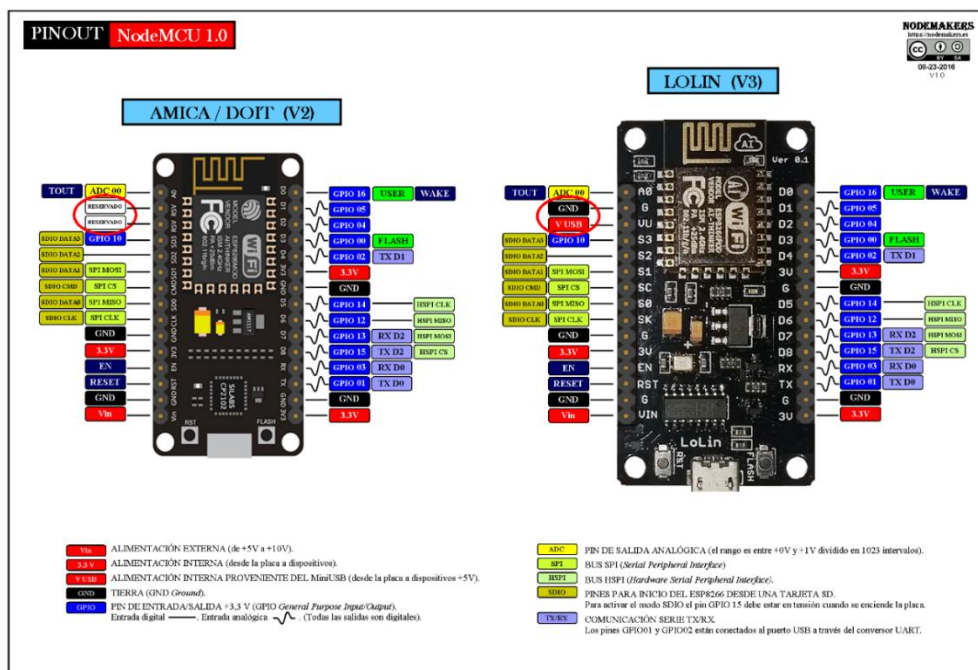


Figura 7. Comparación de placa Amica vs LoLin V3.

Recuperado de <https://www.esploradores.com/comparacion-de-placas-nodemcu/>

Las características técnicas principales son realmente impresionantes en cuanto a velocidad de procesamiento con 600 DMIPS, conectividad WiFi (con protocolos de seguridad mejorados) y BlueTooth (clásico y BLE), 32 puertos GPIO PWM (entrada y salida) con buses 3 x UART, 3 x SPI, 2 x I2S, 12 x ADC, 22 x DAC, 2 x I2C e interface para tarjetas SD. También dispone de amplificador de bajo ruido, sensor Hall y 10 x sensores capacitivos. (No, 2016)

Esp8266 es el nombre de un micro controlador diseñado por Espressif Systems, una compañía china con sede en Shanghai. El volumen de producción de este micro controlador no empezó hasta principios de 2014.

El ESP8266 se anuncia a sí mismo como una solución autónoma de redes WiFi que se ofrece como un puente entre los microcontroladores que hasta ahora existían hasta los MCU con WiFi, siendo además capaz de ejecutar aplicaciones independientes.

Si se fuera a usar un ESP8266 salido de fábrica probablemente no se sabría qué hacer. Y es gracias a que los fabricantes los construyen encima de circuitos impresos y placas prefabricadas, que estos quedan listos para nuestro uso. Por eso se da lugar a varias versiones de ESP8266 (Ej.: ESP-01, ESP-02...) pero todas con el mismo procesador, lo que las diferencian son el número de pines GPIO expuestos, la cantidad de memoria flash, las dimensiones, la forma de exponer los pines, y otras consideraciones varias relativas a su construcción. Pero desde una perspectiva de programación todas son iguales. (Ubach Granados, 2017)

Tabla 16. Especificaciones del ESP8266

Voltaje	3.3 V
Consumo de corriente	10 μ A – 170 mA
Memoria Flash	16 MB máx. (512 k normal)
Procesador	Tensilica L106 32 bit
Velocidad del procesador	80 – 160 MHz
GPIOs	17
Analógico a digital	1 entrada con 10 bit de resolución (1024 valores)
Soporte de 802.11	b/g/n/d/e/i/k/r
Máximas conexiones simultáneas	5

Fuente: (Ubach Granados, 2017)

Procesador, el system on a chip (SoC) ESP9266EX usa un microcontrolador Tensilica Xtensa L106, que es un procesador de 32 bit con instrucciones de 16 bit.

El SoC describe la tendencia cada vez más frecuente de usar tecnologías de fabricación que integran todos o gran parte de los módulos que componen un computador o cualquier otro sistema informático o electrónico en un único circuito integrado o chip. (Ubach Granados, 2017)

Arquitectura, tiene una arquitectura de Harvard, con lo cual la CPU puede tanto leer una instrucción como realizar un acceso a la memoria de datos al mismo tiempo, incluso sin una memoria caché. En consecuencia, una arquitectura de computadores Harvard puede ser más rápida para un circuito complejo, debido a que la instrucción obtiene acceso a datos y no compite por una única vía de memoria.

Variantes, como bien se ha comentado con anterioridad el ESP8266 tiene siempre el mismo procesador, pero su versión varía a la hora de construirlo sobre una placa impresa ya que sus características de construcción difieren en diferentes aspectos; los cuales se pincelan a continuación.

ESP-12

GPIOs: 9 GPIOs + ADC

Dimensiones: 24x16 mm

Antena: Impresa

De todos los modelos de ESP8266, para este proyecto se utilizará un ESP-12.

En el proyecto se ha usado un módulo NodeMCU del que se hablará posteriormente, este módulo utiliza un ESP-12 como base, con la ventaja de que viene con un conector USB a UART incorporado en una misma placa.

El módulo NodeMCU se utilizará por comodidad, pero además a modo de testeo y didacta se ha utilizado y cableado un ESP-12 como se explica a continuación. Ya que uno de los

objetivos del presente trabajo es aprender sobre el ESP8266, y no sobre un módulo prefabricado.

El ESP-12 es la versión actual más popular y flexible disponible hoy en día, siendo esta una de las que expone la mayor cantidad de pines GPIO para su uso.

El ESP-12 tiene un LED azul montado en la parte superior derecha que parpadea cuando hay tráfico UART. (Ubach Granados, 2017)

Tabla 17. Características de los pines

Nombre	Descripción
VCC	3.3 V
GPIO 13	También usado como MOSI
GPIO 12	También usado como MISO
GPIO 14	También usado como Clock

Fuente: (Ubach Granados, 2017)

Convertor ADC, la Conversión de analógico a digital es la capacidad de leer un nivel de voltaje de un pin entre 0 y algún valor máximo, para convertir ese voltaje analógico en una representación digital.

La variación de la tensión aplicada al pin cambiará el valor leído. El ESP8266 tiene un convertidor analógico a digital incorporado con una resolución de 1024 valores distintos.

Es decir que a 0 voltios producirá un valor digital de 0 mientras que al voltaje máximo produce un valor digital de 1023, los rangos de voltaje entre 0 y el voltaje máximo producirá un valor digital correspondientemente escalado. (Ubach Granados, 2017)

Para leer el valor digital del voltaje analógico, la función llamada `system_adc_read ()` debe ser llamada, que en el caso de ESP8266Basic es tan simple como el comando `io(ai)`.

El pin físico en el ESP8266 del cual se lee el voltaje se llama TOUT y no sirve para ningún otro propósito.

El rango de entrada en el pin ADC es de 0V a 1V, esto implica que el voltaje de entrada a la ADC no puede ser la tensión máxima utilizada para alimentar el propio ESP8266 (3.3V). Así que se tendrá que usar un circuito divisor de tensión.

Como se muestra a continuación una resistencia una resistencia R2 de 10k, con una resistencia R1 de 22k es suficiente para el montaje. (Ubach Granados, 2017)

Librerías en NodeMCU

WiFi(librería ESP8266WiFi)

La librería ESP8266WiFi ha sido desarrollada basándose en el SDK ESP8266, utilizando la nomenclatura convencional y la filosofía de la funcionalidad de la Librería de la Shield Arduino WiFi. Con el tiempo, la riqueza de las funciones WiFi portadas desde el SDK ESP8266 a esta librería superaron a las APIs de la librería de la Shield WiFi y se hizo evidente que tenemos que proporcionar documentación por separado sobre lo que es nuevo y extra. (Grokhotkov, 2017)

Ticker

Es una librería para llamar a funciones repetidas cada cierto periodo de tiempo. Dos ejemplos incluidos.

Actualmente no se recomienda realizar operaciones de bloqueo de IO (network, serial, file) desde una llamada a la función Ticker. En su lugar, establece una bandera dentro de la llamada a ticker y chequea por esta bandera desde dentro de la función loop.

Aquí hay una librería que simplifica el uso de Ticker y evita el reset WDT: TickerScheduler

EEPROM

Este es un poco diferente de la clase estándar EEPROM. Necesitas llamar a EEPROM.begin(size) antes de iniciar a leer o escribir, size es el número de bytes que desea utilizar. Size puede estar entre 4 y 4096 bytes.

EEPROM.write no escriba al flash inmediatamente, en su lugar llame EEPROM.commit() cada vez que desee guardar cambios en el flash. EEPROM.end() también guardará y liberará la copia RAM de contenido EEPROM.

La librería EEPROM utiliza un sector del flash justo después de SPIFFS.

I2C (librería Wire) actualmente soporta el modo maestro hasta aproximadamente 450 KHz. antes de utilizar I2C, los pines para SDA y SCL necesitan ser establecidos llamando Wire.begin(int sda, int scl), p.ej. Wire.begin(0, 2) en ESP-01, de lo contrario, los pines por defecto son 4(SDA) y 5(SCL). (Grokhotkov, 2017)

SPIsoporta por completo la API Arduino SPI incluyendo las transacciones, incluyendo la fase de ajuste (CPHA). El ajuste de Clock polarity (CPOL) no está soportado, todavía (SPI_MODE2 y SPI_MODE3 no funciona).

Los pines SPI usuales son:

MOSI = GPIO13

MISO = GPIO12

SCLK = GPIO14

Hay un modo extendido donde puedes intercambiar los pines normales a los pines hardware SPI0. Esto se activa llamando SPI.pins(6, 7, 8, 0) antes de llamar a SPI.begin(). Los pines cambiarían a:

MOSI = SD1

MISO = SD0

SCLK = CLK

HWCS = GPIO0

De este modo se liberan los pines SPI con el controlador que lee el código del programa desde la flash y se controla por un árbitro de hardware (la flash tiene siempre alta prioridad). (Grokhotkov, 2017)De este modo el CS será controlado por hardware y como ya no puede

manejar la línea CS con un GPIO nunca sabrá realmente cuándo el árbitro le otorgará acceso al bus, por este motivo debe dejar que maneje el CS automáticamente.

SoftwareSerial hecha por Peter Lerup (@plerup) soporta velocidades hasta 115200 y múltiples instancias de SoftwareSerial. Ver <https://github.com/plerup/espsoftwareserial> si desea sugerir una mejora o abrir un issue relacionado con SoftwareSerial.

APIs específicas de ESP

Algunas APIs específicas de ESP relacionadas a deep sleep, RTC y memoria flash están disponibles en el objeto ESP.

ESP.deepSleep (microseconds, mode) pondrá el chip en deep sleep (sueño profundo ahorro de energía). mode puede

ser WAKE_RF_DEFAULT, WAKE_RFCAL, WAKE_NO_RFCAL, WAKE_RF_DISABLE D. (Se necesita unir GPIO16 a RST para despertar del deep sleep). El chip puede dormir como mucho ESP.deepSleepMax() microsegundos.

ESP.deepSleepInstant(microseconds, mode) funciona similar a ESP.deepSleep pero duerme instantáneamente sin esperar a que el WiFi se apague. (Grokhotkov, 2017)

ESP.rtcUserMemoryWrite(offset, &data, sizeof(data)) y ESP.rtcUserMemoryRead(offset, &data, sizeof(data)) permite a los datos ser almacenados y recuperados de la memoria de usuario RTC del chip, respectivamente. offset se mide en bloques de 4 bytes y puede variar de 0 a 127 bloques (el tamaño total de la memoria RTC es de 512 bytes). data debe estar alineado con 4 bytes. Los datos almacenados pueden conservarse entre ciclos de sueño profundo, pero pueden perderse después de apagar y encender el chip. Los datos almacenados en los primeros 32 bloques se perderán después de realizar una actualización de OTA, ya que son utilizados por el núcleo interno. (Grokhotkov, 2017)

ESP.restart() reinicia la CPU.

ESP.getResetReason() devuelve un String conteniendo la última razón de reset en un formato legible por un humano.

ESP.getFreeHeap() devuelve el tamaño libre de la pila.

ESP.getHeapFragmentation() devuelve la métrica de fragmentación (0% está limpio, más de ~50% no es inofensivo).

ESP.getMaxFreeBlockSize() devuelve el bloque de ram asignable máximo con respecto a la fragmentación de la pila.

ESP.getChipId() devuelve el ID del chip ESP8266 como un 32-bit integer.

ESP.getCoreVersion() devuelve un String con la versión del core.

ESP.getSdkVersion() devuelve la versión del SDK como un char.

ESP.getCpuFreqMHz() devuelve la frecuencia de la CPU en MHz como un unsigned 8-bit integer.

ESP.getSketchSize() devuelve el tamaño del actual sketch como un unsigned 32-bit integer.

ESP.getFreeSketchSpace() devuelve el espacio libre de sketch como un unsigned 32-bit integer.

ESP.getSketchMD5() devuelve una String con el MD5 (en minúscula) del actual sketch sketch.

ESP.getFlashChipId() devuelve el ID del chip flash como un 32-bit integer.

ESP.getFlashChipSize() devuelve el tamaño del chip flash, en bytes, como lo ve el SDK (puede ser menor que el tamaño real).

ESP.getFlashChipRealSize() devuelve el tamaño real del chip, en bytes, basado en el ID del chip flash. (Grokhotkov, 2017)

ESP.getFlashChipSpeed(void) devuelve la frecuencia del chip flash, en Hz.

ESP.getCycleCount() devuelve la cuenta de ciclos de instrucciones de la CPU desde el arranque como un unsigned 32-bit. Esto es útil para tiempos precisos de acciones muy cortas, como bit banging. (Grokhotkov, 2017)

ESP.getVcc() puede usarse para medir el voltaje suministrado. ESP necesita reconfigurar el ADC al inicio para poder tener esta característica disponible. Añade la siguiente línea en lo alto de tu sketch para utilizar getVcc:

```
ADC_MODE(ADC_VCC);
```

El pin TOUT debe estar desconectado en este modo.

Nota: por defecto ADC está configurado para leer del pin TOUT pin utilizando analogRead(A0), y ESP.getVcc() no está disponible.

Respondedor mDNS y DNS-SD (librería ESP8266mDNS)

Permite al sketch responder a llamadas multicast DNS para nombres de dominios como «foo.local», y llamadas DNS-SD (descubrimiento de servicios)

Respondedor SSDP (ESP8266SSDP)

SSDP es otro protocolo de servicio de descubrimiento, suportado en Windows. Ver ejemplo incluido para referencia.

Servidor DNS (librería DNSServer)

Implementa un servidor simple DNS que puede usarse en ambos modos STA y AP. Actualmente el servidor DNS soporta solo un dominio (para otros dominios responde con NXDOMAIN o un código de estatus personalizado). Con esto, los clientes pueden abrir un servidor web corriendo en el ESP8266 utilizando un nombre de dominio, en vez de una dirección IP.

Servo

Esta biblioteca permite la capacidad de controlar motores servo RC (hobby). Admite hasta 24 servos en cualquier pin de salida disponible. Por definición, los primeros 12 servos usarán Timer0 y actualmente esto no interferirá con ningún otro soporte. Los conteos de servos superiores a 12 utilizarán Timer1 y las funciones que lo utilizan se verán afectadas. Si bien muchos servomotores RC aceptarán el pin de datos IO de 3.3V de un ESP8266, la mayoría no podrá funcionar a 3.3v y requerirá otra fuente de alimentación que coincida con sus

especificaciones. Asegúrese de conectar los cables entre el ESP8266 y la fuente de alimentación del servomotor. (Grokhotkov, 2017)

Librería mejorada EEPROM para ESP (ESP_EEPROM)

Una biblioteca mejorada para la EEPROM de ESPxxxx. Utiliza la memoria flash de acuerdo con la biblioteca estándar ESP EEPROM, pero reduce el reflash, por lo que reduce el desgaste y mejora el rendimiento de commit().

Como las acciones en el flash deben detener las interrupciones, un reflash de la EEPROM podría afectar notoriamente cualquier cosa usando PWM, etc.

Otras librerías (no incluidas con el IDE)

Las bibliotecas que no dependen del acceso a bajo nivel a los registros AVR deberían funcionar bien. Aquí hay algunas bibliotecas que se verificó que funcionan:

Adafruit_ILI9341 - Adafruit ILI9341 para el ESP8266

arduinoVNC - Cliente VNC para Arduino

arduinoWebSockets - Servidor y cliente WebSocket compatible con ESP8266 (RFC6455)

aREST - Manejador de la librería REST API.

Blynk - IoT framework sencillo para Makers

DallasTemperature

DHT-sensor-library - Librería Arduino para el sensor DHT11/DHT22 de temperatura y humedad. Descarga la última librería v1.1.1 y no serán necesarios cambios. Las versiones antiguas deben inicializar el DHT como sigue: `DHT dht(DHTPIN, DHTTYPE, 15)`

DimSwitch - Control electrónico regulable de balastos para luces de tubo fluorescentes remotamente como si se usara un interruptor de pared.

Encoder - Librería Arduino para encoders rotatorios. Versión 1.4 soporta ESP8266.

esp8266_mdns - Llamadas y respuestas mDNS en esp8266. O dicho de otro modo: Un cliente mDNS o librería de cliente Bonjour para el

Como programar Nodemcu con Arduino

Después de configurar el ESP8266 con el firmware de NodeMCU, verificaremos el IDE (Integrated Development Environment) necesario para el desarrollo de NodeMCU. NodeMCU con ESPlorer IDE los scripts Lua se utilizan generalmente para codificar el NodeMCU. Lua es un lenguaje de programación de código abierto, ligero e integrable construido sobre el lenguaje de programación C. (Guirado Roda, 2014)

NodeMCU con Arduino IDE He aquí otra forma de desarrollar NodeMCU con un IDE bien conocido, es decir, Arduino IDE. También podemos desarrollar aplicaciones en NodeMCU usando el entorno de desarrollo de Arduino.

Esto hace que sea más fácil para los desarrolladores de Arduino que aprender un nuevo lenguaje e IDE para NodeMCU. Instalar el IDE actual de Arduino en el nivel 1.8 o superior. La versión actual se encuentra en el sitio web de Arduino. Iniciaremos Arduino y abriremos la ventana de Preferencias.

Ingresaremos a este link: https://arduino.esp8266.com/stable/package_esp8266com_index.json en el campo Additional Board Manager URLs. Se podrá agregar múltiples URLs, separándolas con comas. Después de completar Cable de datos del teléfono móvil. Se utiliza para conectar el ESP8266 MCU NODE con el PC. Después de instalar los controladores si es necesario. Comprobaremos qué número está asignado a nuestra tarjeta. Abriremos el IDE de Arduino. Administrador de placas abiertas desde Herramientas > Plataforma Modelos esp8266 y seleccionaremos la placa NodeMCU 1.0 (Módulo ESP-12E) desde Herramientas. Cargar usando: Serial Frecuencia de CPU: 80Mhz Tamaño del flash: 4M Velocidad de subida: 115200 PUERTO: Seleccione Asignar sólo puerto.

Ejemplo de carga:

```
//In NODEMCU pin Number D0,D1,D2,...Dn.
```

```
#define LED D0 void setup() {  
  
pinMode(LED, OUTPUT); }  
  
void loop() { digitalWrite(LED, HIGH);  
  
delay(1000); digitalWrite(LED, LOW);  
  
delay(1000);} 
```

2.6 Aplicaciones con Arduino

Sensores de lluvia para Arduino, estos sensores se denominan a menudo de forma diferente: de lluvia, de agua, de nieve, pluviómetro, etc. Si bien hay varios modelos, siempre se trata del mismo sensor, una placa con un circuito impreso en forma de serpentina y dos conectores. El sensor se conecta muy fácilmente al Arduino, el código para trabajar con estos sensores es simple y el precio es bajo, lo que los hace ideales para proyectos sencillos en Arduino Uno, Mega, Nano. (proyecto, 2020)

Cómo funciona el sensor de lluvia este te permite detectar y responder a las gotas de humedad, por ejemplo, activando una alarma. Estos sistemas se utilizan mucho en domótica, la industria automotriz, los sistemas hogareños de riego y en otras áreas de nuestra vida cotidiana.

El módulo del sensor consta de dos partes:

Placa PCB con superficie niquelada, con resistencia a la oxidación y conductividad eléctrica. De hecho, el sensor es una simple resistencia variable, cerrada por el agua en diferentes lugares, que causa un cambio en la resistencia.

La segunda parte del sensor es un comparador dual (normalmente LM393, pero son posibles las variantes LM293 y LM193). Su tarea principal es convertir el valor del sensor en una señal analógica de 0 a 5 voltios.

También se puede conseguir el sensor con un relé de 12 voltios en lugar del comparador dual. En este caso se podría utilizar directamente sin intervención de una placa arduino, abriendo o cerrando el relé ante la presencia de agua.

El sensor está alimentado por 5V, que se puede arrancar fácilmente desde cualquier placa Arduino. Normalmente, el módulo sensor tiene dos salidas:

Analógico. El valor recibido por el controlador oscilará entre 0 y 1023. Donde 0 – todo está inundado o hay un aguacero, el sensor está muy húmedo, 1023 – tiempo seco, el sensor está seco (en algunos sensores hay valores opuestos, 1023 – humedad máxima, 0 – sequedad máxima).

Digital. Salida de alto (5V) o bajo voltaje si se excede algún umbral. El nivel de umbral es regulado por una resistencia de corte. (proyecto, 2020)

Conexión del detector de lluvia al arduino

Para conectar el sensor al Arduino necesitarás una placa (UNO, Mega, Nano o cualquier otra) y el propio sensor. Si deseas comprobar la intensidad de la precipitación, se recomienda colocar el sensor en ángulo y no en posición horizontal, para que las gotas acumuladas fluyan hacia abajo.

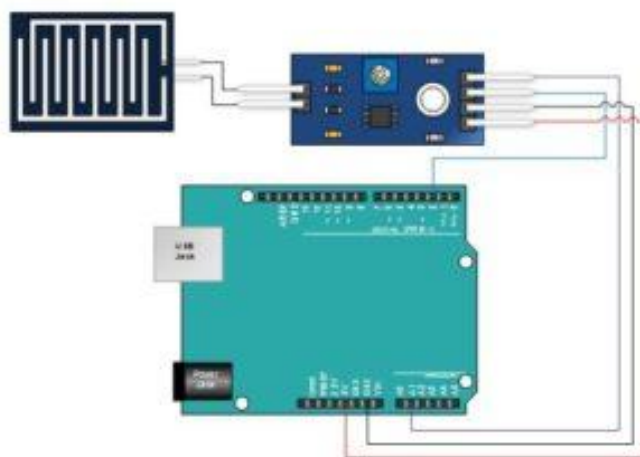


Figura 8. Diagrama de conexión del módulo del sensor de lluvia al arduino

(Proyecto, 2020)

VCC (power input) – debe ser el mismo para el circuito Arduino conectado en voltaje y corriente. Es decir, en este caso, 5V;

GND – conexión a tierra;

AO – salida analógica;

DO – salida digital.

La salida analógica se conecta a un pin analógico de microcontrolador, por ejemplo al A1. La salida digital se conecta a uno de los pines digitales. El voltaje puede ser suministrado desde la salida de 5V de la placa Arduino y la tierra está conectada a GND. (proyecto, 2020)

Detector Gases con Arduino son uno de los sensores más utilizados en industrias y comercios (y cada vez más también en nuestros hogares). Hace que el lugar de trabajo o nuestro hogar sea un poco más seguro e inteligente de lo normal al añadirle la capacidad de detectar gases peligrosos. Esto puede salvar vidas. (Proyecto, 2020)

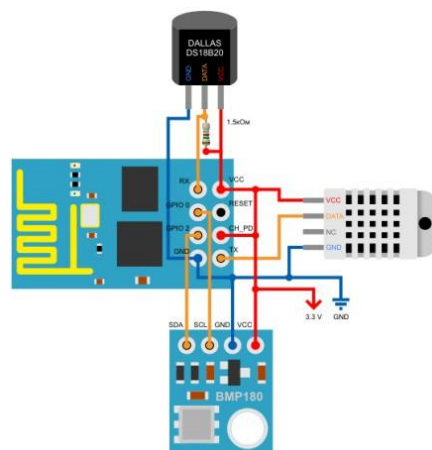
Sensores de Polvo y Partículas en el Aire, al respirar inhalamos los gases, vapores y partículas que hay en suspensión en el aire. La exposición a estas partículas o “polvo” puede causar efectos graves sobre la salud.

Un sensor de partículas de aire es básicamente un detector de polvo que podremos utilizar por ejemplo en alguna industria o línea de producción donde necesitemos que no haya partículas en suspensión. (arduino, 2020)

2.7 Aplicaciones con NodeMCU

HomesSmart, comenzó como un módulo para sensores para sistemas domóticos y de monitorización

sistema
conectar varios
sensores eran los
húmeda, etc.



meteorológica, integrables en el domótico MajorDomo. Permitía sensores a un ESP-01. Estos típicos de temperatura, presión,

Figura 9. .Proyecto HomeSmart

No es un proyecto de código abierto y además de una versión gratuita tiene una versión de pago que añade muchas más funcionalidades. Su creador cobra una pequeña cantidad (bastante justa a decir verdad) por cada dispositivo instalado con la versión completa del mismo. El precio es de \$2 por cada ESP8266 en el que se instale su firmware.

Implementación de comunicación USB mediante software, este es otro de los fantásticos proyectos de Charles Lohr. Entre las muchas características del ESP8266 no está la de soportar comunicación USB. Charles se propuso implementar la comunicación USB sólo mediante software. (admin, 2019)

Como prueba de concepto, Charles, desarrolló un emulador de ratón y teclado que se controla desde una página web alojada en la memoria flash del propio microcontrolador. Este emulador, programado sobre un ESP8285 montado sobre una placa diseñada a medida para el proyecto, ocupa poco más que el propio conector USB.

Servidor NTP sin conexión a Internet, cuando el ESP8266 está conectado a Internet, es muy cómodo y sencillo utilizar el protocolo NTP para sincronizar la hora interna. Cuando no existe la posibilidad de conectarse a Internet, es necesario utilizar otras fuentes como GPS, relojes de tiempo real (RTC) o receptores de señales de radio DCF77. (admin, 2019)

Este proyecto con ESP8266 permite que obtenga la hora real mediante un receptor GPS, por ejemplo, y otros dispositivos se conecten a él para sincronizarse utilizando el protocolo estándar NTP.

Sistema de posicionamiento en interiores SubPos

Una carencia del sistema GPS a la que estamos acostumbrados es la de no funcionar en interiores o, si lo hace, es cerca de ventanas y con una mala precisión. Por ello han surgido varios intentos de soluciones de posicionamiento en interiores. Uno de estos sistemas es SubPos, cuya pretensión es establecerse como un estándar.

SubPos utiliza la señal WiFi para transmitir la posición. Sin embargo, no es necesario que el dispositivo esté conectado a ninguna red, ni siquiera a Internet. Solamente basta con escuchar las redes WiFi que hay alrededor.

SubPos codifica en los mensajes de anuncio de presencia (tramas beacon) de puntos de acceso WiFi su propia posición, que debe configurarse de forma manual. Estos dispositivos, llamados nodos SubPos, se comportan como balizas que transmiten continuamente su posición. Al no necesitar ninguna comunicación entre nodos o de estos con Internet, el sistema no requiere ninguna infraestructura.

Una de estas balizas o nodos SubPos, no es más que un ESP8266 con un firmware que se puede descargar de GitHub. Se configura como punto de acceso y permite la configuración de su propia ubicación de forma muy sencilla.

El equipo que se conecta al sistema SubPos tampoco requiere de nada especial. Solamente es necesario un software que interprete los anuncios de posición de las balizas y mida el nivel de potencia con el que se reciben. (admin, 2019)

2.8 Aspectos fundamentales de una guía práctica

Introducción

El propósito de elaborar un manual de prácticas es lograr que los docentes planifiquen y organicen eficazmente su participación en el proceso educativo. Los elementos que se deben considerar en el diseño son racionalidad, viabilidad, utilidad y claridad, todos ellos para facilitar la instrumentación de cada actividad práctica. Esto resultará en un material didáctico que apoyará mejor el proceso enseñanza y aprendizaje. (Aleman Suárez & Mata Mendoza, 2006)

El manual de prácticas podrá ser utilizado en el proceso de enseñanza y aprendizaje como un medio didáctico, junto con los recursos materiales y educativos, lo que en conjunto puede cumplir diversas funciones. Entre las más frecuentes están:

Proporcionar explícitamente información del tema en estudio, de sus métodos y procedimientos

¿Qué se entiende por una práctica?

Una práctica de laboratorio, taller o campo es una actividad didáctica basada en una experiencia en la que se cuestionan los conocimientos y habilidades de una o más disciplinas. Se pone en juego un conjunto de conceptos, procedimientos, métodos y tecnologías que permiten su ejecución. (Alemán Suárez & Mata Mendoza, 2006)

Por tanto, es preciso que en el proceso de enseñanza y aprendizaje se le conceda gran importancia a las partes experimental y vivencial que acompañan el desarrollo de un curso teórico y práctico. Las prácticas de laboratorio, taller o campo deben diseñarse con un nivel pedagógico adecuado. (Alemán Suárez & Mata Mendoza, 2006)

En el diseño de una práctica de laboratorio, taller o campo para una asignatura se recomienda considerar los siguientes aspectos:

Revisión del objetivo general y del contenido de la asignatura.

Consulta de cuando menos dos libros o artículos científicos acerca del problema que se plantea resolver, mismos que deben ser referidos en la bibliografía del manual de prácticas.

Planificación del número adecuado de prácticas y de horas destinadas a esta actividad dentro del programa de la asignatura.

Selección y enunciado de los apartados que permitan describir la práctica, como son introducción, objetivo, referencias del tema en estudio, instrucciones generales, metodología, material y equipo, cuestionario, resultados, análisis y discusión.

Planificación, para cada actividad práctica, del tiempo que ocupará cada una de ellas contemplando un espacio para discutir sus resultados.

Bibliografía recomendada, la cual deberá estar disponible en las sesiones de laboratorio o de campo.

Evaluación: se deben formular de manera explícita los criterios para determinar el grado en que el estudiante ha alcanzado el objetivo de la actividad, lo que incluye el formato para el reporte escrito propuesto y la fecha de entrega. (Alemán Suárez & Mata Mendoza, 2006)

Introducción general del manual

Para mayor claridad es aconsejable elaborar un mapa que explicita la estructura del sistema de prácticas. Para ello es necesario describir en qué consiste el sistema, su orientación, duración total, su seriación y en cuál semana del semestre será impartida la práctica. Las actividades se pueden representar con un diagrama de relaciones u otro tipo de gráficos.

Tabla 18. Actividades para elaborar una guía

UNIDAD	PRÁCTICA O PRÁCTICAS PROGRAMADAS	ÁMBITOS DE DESARROLLO	TIEMPO DE DESARROLLO
Tema según programa de asignatura.	Nombre y número de las prácticas relacionadas con el tema, y su secuencia.	Tipo de actividad práctica de campo, laboratorio y taller.	Hora para cada práctica y semana del semestre en que se realizará.

Fuente: (Alemán Suárez & Mata Mendoza, 2006)

Elementos de una práctica

A continuación se describen los apartados que frecuentemente integran una práctica: introducción, objetivos, métodos, dinámicas de trabajo, materiales de apoyo y recursos con los que se formará el alumno (aprendizaje de las habilidades y actitudes), criterios coherentes de desempeño. Por otro lado, es necesario tener presente que las prácticas de laboratorio deben de estar coordinadas con las clases de teoría. Sin embargo, varias circunstancias hacen que esto no siempre sea posible a causa de la distribución horaria, el número de horas disponibles

para el laboratorio, número de alumnos y la disponibilidad económica para comprar suficientes equipos y material.

Título de la práctica, experimento o proyecto

En este apartado deberá expresarse el nombre de la práctica, del experimento o proyecto. El título deberá ser sugerente, atractivo y relacionado con el tema o problema en estudio.

Introducción

Integrar una explicación de los aspectos teóricos de la disciplina que, en particular, son necesarios. En este apartado se anotan los conceptos teóricos que sustentan el experimento propuesto: teorías, leyes, métodos, técnicas y estrategias en las que se apoya. También se mencionan los antecedentes de la situación actual; es decir del problema que se está resolviendo, las técnicas usadas en el desarrollo de la práctica o proyecto experimental, según el caso, y todos aquellos datos e información que permiten llevar a efecto el experimento. Este apartado debe ser breve, concreto y suficiente, y estar apoyado con las referencias bibliográficas utilizadas para el desarrollo de la práctica.

Objetivo de la práctica

El objetivo de la práctica señala la finalidad del experimento o actividad específica. Está directamente relacionado con la demostración o comprobación práctica que se va a llevar a efecto o con la transformación de situaciones planteadas desde un principio. En la redacción puede proponerse un solo objetivo general o bien, en ocasiones, desglosar diferentes niveles de éste; es decir, objetivos particulares o hasta específicos, mismos que pueden presentarse como incisos del objetivo general.

Los objetivos expresan las destrezas o conductas que debe obtener el estudiante. Tales habilidades pueden ser del orden cognoscitivo (definiciones, descripciones); afectivo (atención, aceptación) o; psicomotor, manipulación de equipos y materiales (Alemán Suárez & Mata Mendoza, 2006)

Medición y valoración, llamado también patrón de rendimiento o estándares de evaluación, se refiere a los niveles cuantitativos, a las medidas o a las valoraciones. El objetivo, por tanto, puede presentar criterios de medición.

Metodología, este apartado describe el proceso técnico o los pasos a seguir para el desarrollo del experimento. Se permite para ello utilizar diagramas, gráficas u otro tipo de representaciones. Lo importante es presentar claramente la secuencia en la formulación y desarrollo de la experiencia en el laboratorio o en campo.

Recursos materiales y equipo, este rubro especifica todo lo requerido en cuanto al tipo de equipos, materiales (reactivos, didácticos y referenciales), tecnologías, instrumental, herramientas, instalaciones, software y personal, tanto para la etapa de experimentación como para la reproducción, a futuro, del problema en estudio. No deberá escapar ningún detalle correspondiente al experimento en cuestión. (Alemán Suárez & Mata Mendoza, 2006)

Se procurará que cada equipo sea manejado por un número pequeño de alumnos, según el tipo de prácticas.

Descripción del desarrollo de la práctica, se debe explicar al estudiante los pasos que debe seguir para realizar las actividades en el laboratorio y los detalles para pasar de una parte a otra en cada acción planeada. Así como también, la relación de estos procedimientos con el uso adecuado de la maquinaria, del equipo y del instrumental, acorde con las necesidades de la disciplina y al tema de referencia.

Al sistema educativo moderno (globalizado, de la sociedad del conocimiento) se le plantea el reto de formar personas altamente preparadas y con flexibilidad mental para adaptarse a los cambios por la introducción de nuevas tecnologías. Estamos en un momento en que la idea de que una carrera profesional es suficiente para toda la vida ha cambiado. De aquí la importancia de tener conocimientos consolidados, lo que puede propiciarse con las asignaturas básicas dentro de un programa educativo. (Alemán Suárez & Mata Mendoza, 2006)

Evaluación, un objetivo fundamental de la evaluación es acopiar información pertinente para conocer la eficacia de la acción, la cual no depende sólo del alumno sino de un cúmulo de componentes de naturaleza variada: la adecuación de lo que se pretende respecto de la

capacidad y actitudes de los estudiantes, el ritmo de aprendizaje, los medios de que se dispone, los momentos elegidos, la relación del profesor con los alumnos dentro del ambiente de aprendizaje. (Alemán Suárez & Mata Mendoza, 2006)

Bibliografía

En este punto se indica la bibliografía básica y complementaria con la que fueron elaborados los contenidos de la práctica. Se recomienda consultar las principales revistas que prioritariamente publican trabajos experimentales específicos del área en estudio, así como libros de reciente publicación sobre la temática. En algunos casos es preciso incluir un breve comentario sobre cada revista porque cada una de ellas en particular ofrece algún rasgo distintivo, aunque en conjunto representen la totalidad de la investigación actual.

Resultados y conclusiones, el resultado de la práctica se traducirá en un breve informe, a partir del conjunto de datos que los alumnos obtuvieron durante el desarrollo de la actividad. O bien, con la participación de los estudiantes, plantear y exponer un proyecto que integre los planteamientos teóricos y metodológicos desarrollados a lo largo del curso, todo ello en busca de generar soluciones a problemas reales. (Alemán Suárez & Mata Mendoza, 2006)

Conclusiones: Comprenden las aportaciones personales o los juicios de valor propuestos a partir de los resultados de la práctica o del experimento, o bien de las acciones derivadas de todo el proceso de experimentación. En algunos casos incluyen recomendaciones para futuros experimentos relacionados con el tema. También, de manera adicional, se puede agregar un pequeño cuestionario, tres a siete preguntas, para verificar los resultados y el tipo de interpretación que el grupo de alumnos ha realizado a partir de la discusión de los mismos.

Recomendaciones: En este apartado se presenta la serie de observaciones adicionales que deben considerarse en el diseño de una práctica de laboratorio, taller o campo, como por ejemplo: normatividad, condiciones de trabajo, manejo de los recursos, preparación previa de la actividad práctica, toma de datos y análisis de los mismos. (Alemán Suárez & Mata Mendoza, 2006)

Capítulo III: Metodología de la Investigación

3.1 Tipo de estudio

Esta fase se centra en describir la manera en que se llevará a cabo el estudio en función al propósito por el cual se realiza, así en este caso se trazarán los principales procedimientos a seguir para lograr proponer un manual de guías prácticas en las materias de Sistemas Digitales, Arquitectura de Computadoras y Microprogramación.

Para ello, se ha definido que el estudio se iniciará como una investigación exploratoria, dado que se analizará un problema el cuál ha sido poco estudiado y en el cual se asume cierto grado de desconocimiento en los estudiantes de Ingeniería de Sistemas Informáticos. El estudio exploratorio permitirá hacer un acercamiento a los problemas relacionados con el tema en estudio, y proveer una referencia general de la temática a través de la recopilación, registro y análisis de información.

Sin embargo, el estudio no puede terminar como exploratorio, debido a que no arrojaría los resultados deseados, por lo que posteriormente la investigación será descriptiva, ya que el objetivo de este tipo de investigación es establecer una descripción completa de un fenómeno, utilizando técnicas específicas de recolección de la información, entre las cuáles se aplicarán las entrevistas y la encuesta. En consecuencia, se buscará identificar la aplicabilidad de la plataforma Arduino y NodeMCU.

3.2 Fuentes de información

Son los diversos tipos de instrumentos y documentos que contienen datos útiles para satisfacer una demanda de información o conocimiento. La recopilación de información útil relativa al tema en estudio es de suma importancia para el cumplimiento de los objetivos mismo, es por ello, que se utilizarán las fuentes primarias y secundarias de información descritas en los apartados posteriores.

3.2.1 Fuentes primarias

Son las fuentes que brindarán información original, proveniente directamente del área en la cual se realiza estudio. Para obtener dicha información se hará uso de ciertas técnicas de

recolección de datos, las cuales son: entrevistas y una encuesta a los estudiantes de la carrera de Ingeniería de Sistemas Informáticos.

3.2.2 Fuentes secundarias

Contienen información elaborada, producto del análisis, extracción o reorganización que refiere a documentos primarios. Reside en la recolección y selección de toda aquella información que tuviera relación con conocimientos previos de los estudiantes en la materia de Sistemas Digitales, Microprogramación y Arquitectura de Computadoras.

3.3 Técnicas de recolección de datos

3.3.1 Encuesta

Es una técnica de investigación que permitirá conocer información de un hecho a través de la percepción y experiencia de los estudiantes del tema en estudio. La cual será realizada a través de un cuestionario previamente elaborado, con el cual se espera recolectar información útil, para que de esta manera se describa mejor el diagnóstico de la problemática.

Por lo que la muestra vendría determinada por la siguiente fórmula:

$$n = \frac{N \cdot Z^2 \cdot \sigma^2}{(N - 1) \cdot e^2 + Z^2 \cdot \sigma^2}$$

En donde:

- **n** = Tamaño de la muestra poblacional de estudiantes a obtener.
- **N** = Tamaño de la población total de estudiantes.
- **Z** = Nivel de confianza a utilizar del 95%
- **σ** = Variabilidad que puede tener el fenómeno, con un valor de 0.5
- **e** = Error muestral, 10%

3.3.2 Entrevista

Se refiere a la comunicación interpersonal que se establecerá con docentes que imparten materias en las carreras de Ingeniería, con el fin de obtener respuestas a las interrogantes planteadas sobre los problemas relacionados con la temática en estudio.

3.3.3 Investigación documental

Esta técnica centra su principal función en todos aquellos procedimientos que conllevan el uso óptimo y racional de los recursos documentales disponibles en las funciones de información. Se seleccionarán libros, tesis, reportes e informes que brinden información relevante al estudio.

3.4 Instrumentos de recolección de datos

Estos representan la forma en cómo se procederá a recabar la información necesaria para dar respuesta a su objeto de estudio. Los instrumentos de recopilación de datos para las fuentes primarias de información se describirán en los dos subapartados siguientes.

3.4.1 Cuestionario para encuesta

El cuestionario elaborado para la encuesta pretende alcanzar el objetivo descrito en ella, los datos serán obtenidos mediante un proceso de recopilación de información estandarizado, en el cual el estudiante individualmente completará los ítems propuestos para evitar opiniones sesgadas que pudieran influir en el resultado del estudio. Para esta investigación se utilizará un cuestionario de 23 preguntas que se encuentran en el Anexo No. 2. Una vez contestado este cuestionario se determinará si es factible proponer un entrenador para la realización de guías prácticas del departamento de Ingeniería y Arquitectura en la carrera de ingeniería de Sistemas Informáticos aplicadas a materias como Sistema Digitales, Microprogramación y Arquitectura de Computadoras.

3.4.2 Modelo de preguntas para entrevistas

Las preguntas de las entrevistas se enfocan en conocer la necesidad de proponer una nueva tecnología es decir un manual de guías prácticas para aplicarlas y desarrollarlas en la materia de Sistemas Digitales de la carrera de Ingeniería, y obtener la información útil siguiendo un esquema ordenado de interrogantes. Para esta investigación se utilizará instrumentos: el primero consta de un cuestionario de 9 preguntas ver anexo N°3 y el otro está formado por 6 preguntas para instituciones externas a la UESFMOcc. Ver anexo No. 4.

3.5 Determinación de la población y la muestra

La carrera de Ing. En Sistemas Informáticos del departamento de Ingeniería y Arquitectura cuenta con docentes y estudiantes, no todos serán objeto de estudio por diferentes razones.

Como se ha mencionado anteriormente en esta investigación utilizaremos dos tipos de fuentes de información: fuentes de información primarias y fuentes de investigación secundarias, entre las primarias tenemos a los estudiantes de la carrera de Ingeniería de Sistemas Informáticos de la Facultad Multidisciplinaria de Occidente.

Para determinar el tamaño de la muestra en el sector estudiantil se han tomado en cuenta las siguientes consideraciones:

- El nivel de confianza que se ha utilizado es el 95% para obtener una mayor representatividad en los datos.
- El porcentaje de error calculado es del 10%, con lo cual se obtiene un riesgo mínimo de sesgo en la información recolectada.
- Los valores de P y Q que representan la probabilidad de éxito y de fracaso en un experimento se han asignado en $P = 50\%$ y $Q = 50\%$.
- El sector estudiantil que se investigara está compuesto por alumnos de 3º, 4º y 5º año de las materias de sistemas digitales, arquitectura de computadoras, microprogramación, bases de datos sistemas de información gerencial de la carrera de Ingeniería de Sistemas Informáticos, se considera la población de 139 alumnos, la fórmula para determinar el tamaño de la muestra es la siguiente:

En donde:

- n = Tamaño de la muestra poblacional de estudiantes a obtener.
- N = Tamaño de la población total de estudiantes.
- Z = Nivel de confianza a utilizar del 95%
- σ = Variabilidad que puede tener el fenómeno, con un valor de 0.5
- e = Error muestral, 10%

$$n = \frac{139 (1.96)^2 (0.5)^2}{(139 - 1)(0.10)^2 + (1.96)^2 (0.5)^2}$$

$$n = 57$$

3.6 Diagnóstico de la situación actual

3.6.1 Aplicación de técnicas e instrumentos de recolección de datos

Las técnicas constituyen el conjunto de reglas y pautas que guían las actividades que realizan los investigadores en cada una de las etapas de la investigación científica. Las técnicas como herramientas procedimentales y estratégicas suponen un previo conocimiento en cuanto a su utilidad y aplicación de tal manera que seleccionarlas y elegir las resulte una tarea fácil para el investigador (Carrasco Díaz, 2005). Por tanto, en la presente investigación se utilizó las siguientes técnicas de recolección de datos:

La encuesta es un formato redactado en forma de interrogatorio para obtener información acerca de las variables que se investigan, puede ser aplicado personalmente o por correo y en forma individual o colectiva y debe reflejar y estar relacionado con las variables y sus indicadores.

Según Carrasco, consiste en la indagación, exploración y recolección de datos, mediante preguntas formuladas directa o indirectamente a los objetos que constituyen la unidad de

análisis del estudio investigativo. Las preguntas se formularon cara a cara entre el encuestador y el encuestado a manera de entrevista y cuando se realiza mediante instrumentos (forma indirecta) se denomina cuestionario, ejemplo para encuestar gran número de personas.

Entrevista

Normalmente se tienen varios entrevistadores, quienes deberán estar capacitados en el arte de entrevistar y conocer a fondo el cuestionario. Quienes no deberán sesgar o influir en las respuestas, por ejemplo, reservarse de expresar aprobación o desaprobación respecto de las respuestas del entrevistado, reaccionar de manera ecuánime cuando los participantes se perturben, contestar con gestos ambiguos cuando los sujetos busquen generar una reacción en ellos. Su propósito es lograr que se culmine exitosamente cada entrevista, evitando que decaiga la concentración e interés del participante, además de orientar a éste en el tránsito del instrumento Las explicaciones que proporcione deberán ser breves pero suficientes. Tiene que ser neutral, pero cordial y servicial. Asimismo, es muy importante que transmita a todos los participantes que no hay respuestas correctas o equivocadas. Por otra parte, su proceder debe ser lo más estándar posible (mismos señalamientos, presentación uniforme, entre otros.). Con respecto a las instrucciones del cuestionario, algunas son para el entrevistado y otras para el entrevistador. Este último debe recordar que al inicio se comenta: el propósito general del estudio, las motivaciones y el tiempo aproximado de respuesta, agradeciendo de antemano la colaboración. (Hernández Sampieri, Fernández Collado, & Baptista Lucio, 2010)

3.6.2 Descripción de la condición actual de la población estudiada

En la UES FMOcc se dan unos pequeños pasos con respecto a la tecnología de Arduino y NodeMCU, específicamente en la carrera de Ingeniería de Sistemas Informáticos. Se da a conocer parte de estas tecnologías en las materias de sistemas digitales, Arquitectura de computadoras y Microprogramación; pero estos son conocimientos agregados por el contexto evolutivo de la sociedad y debido a esto en las materias se han ido incluyendo.

Desde el año 2016, ninguna materia utilizaba la tecnología porque en el currículo no se contemplaba, si en los programas estuviese como un tema en específico por ley tendría que

implementarse, pero el currículo de ingeniería de sistemas informáticos data desde el año 1998 a la actualidad se está reformando.

En ese caso el programa de sistemas digitales se desarrolla tal y como lo dice la currículo del año 98, se propone a los estudiantes que realicen un proyecto y se da la libertad que lo desarrollen con pic, logos, Arduino o NodeMCU; eligiendo de esta manera la tecnología, consultando al docente acerca de la parte de hardware, de alguna forma de programación, sin tener guía específica de prácticas ni materiales necesarios para llevar a cabo un diseño, porque no se cuenta con ello en la facultad, se realizan proyectos que son buscados por los estudiantes de las materias de Sistemas Digitales, Arquitectura de Computadoras y Microprogramación.

3.7 Análisis e interpretación de los datos

Es necesario realizar un análisis de la información recopilada mediante la clasificación, procesamiento e interpretación de la información obtenida a través de los instrumentos de recopilación de datos. Es por ello, que se hará un análisis cuantitativo siendo para este caso un tratamiento estadístico de los datos recopilados y un análisis cualitativo para la información registrada que requiera un análisis más complejo.

3.8 Conclusiones y recomendaciones

En este apartado se formularán conclusiones específicas que marquen los hallazgos más importantes relacionados al estudio en cuestión. También, serán incluidas las recomendaciones según la interpretación de los datos y el criterio objetivo aplicado al estudio, de manera que éstas den la pauta para el cumplimiento de los objetivos y alcances previamente definidos.

Capítulo IV: Manual de Guías Prácticas utilizando Arduino y NodeMCU

4.1 Introducción

Esta guía está basada en componentes electrónicos para desarrollar prácticas, y enfocadas al área de estudiantes para que cada día estén inmersos en el mundo de la electrónica de una forma sencilla y efectiva al usar una placa de hardware libre (Arduino) que es muy fácil de manejar y de programar. Esta guía además también dispone de toda una serie de materiales interactivos que ayudarán a entender cómo hay que montar los proyectos electrónicos.

4.2 Objetivos.

- Aprender a utilizar el entrenador propuesto para Arduino.
- Conocer el entorno de programación utilizando el IDE de Arduino.
- Diseñar guías prácticas para la plataforma Arduino con software libre.
- Realizar programas y ejecutarlos sobre la plataforma Arduino.

4.3 Entrenador Arduino y NodeMCU

Introducción

El presente trabajo se ha realizado con el objeto de poner en práctica algunos de los conocimientos adquiridos en la carrera de Ingeniería de Sistema Informáticos.

La aparición de la plataforma de software libre Arduino en el mercado, ha despertado un gran interés en las personas por todos los proyectos que se pueden realizar incluyendo elementos electrónicos. Aunque la mayor parte de estos proyectos se realizan en diferentes ámbitos.

La idea de crear y proponer un entrenador de Arduino enfocado al desarrollo de guías prácticas, surgió por la necesidad crear proyectos y la innovación del ambiente tecnológico que cada día se considera más importante y útil para docentes y estudiantes de conocer e involucrarse en el desarrollo de nuevas aplicaciones.

Utilizar la placa Arduino en el desarrollo de proyectos es conveniente y confiable; además se puede utilizar NodeMCU en un mismo diseño y llegar a la fabricación de un entrenador para construir y presentar guías prácticas con las dos tecnologías.

Objetivos

- Diseñar un entrenador de Arduino y NodeMCU con elementos electrónicos.
- Presentar unas guías resueltas para demostrar la funcionalidad de cada elemento como: placa Arduino, ESP32, botones, leds, pantalla, sensores, motores, buzzer entre otros elementos.
- Proporcionar un prototipo donde poder desarrollar guías funcionales y de utilidad para la sociedad.

Arduino

Arduino y NodeMCU son una plataforma que aúna hardware y software libre, fácil de usar y flexible. Además, alrededor de Arduino y NodeMCU gira una comunidad de desarrolladores muy extensa en continuo movimiento, que desarrollan librerías y demás recursos que todo el mundo puede utilizar. Se pretende simplificar el uso de electrónica y programación de microcontroladores para hacerla accesible a un mayor número de personas.

Tanto de hardware como de software, son liberados con licencia de código abierto, lo que permite su uso, copia y modificación gratuita. El hardware en sus placas de desarrollo se compone de microcontroladores, pines para acceder fácilmente a los puertos de entrada y salida (tanto digitales como analógicos), los cuales se utilizan para conectarse con el exterior, e incluso se pueden conectar a placas de expansión (shields), que amplían las características de funcionamiento de la placa Arduino. Además, posee un puerto de conexión USB por donde se puede alimentar la placa y comunicarse con un ordenador para programar el microcontrolador.

Estas dos tecnologías Arduino y NodeMCU han crecido exponencialmente en los últimos años y actualmente ofrece una amplia gama de productos con diferentes características, desde placas de desarrollo de diferentes tamaños y funcionalidades hasta Shields de expansión pasando por impresoras 3D.

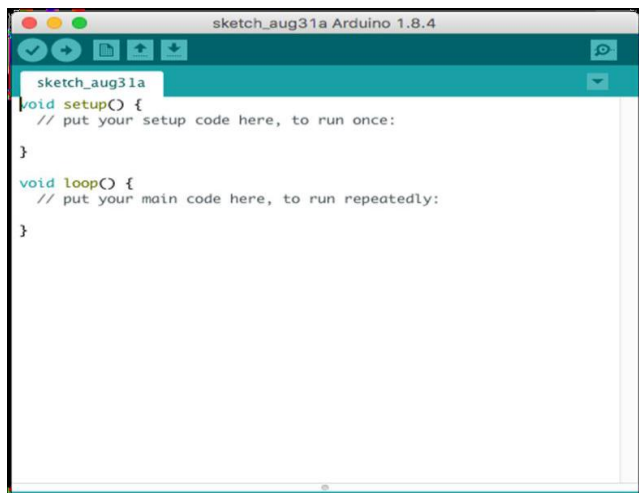
Entorno de desarrollo

Cuando se habla de entorno de desarrollo, se refiere a una aplicación o programa que facilita al desarrollador o programador el desarrollo de un software determinado. En el caso de Arduino, el entorno facilita la edición, compilación y carga a la placa determinada del código o programa que se suele denominar “sketch”, una de las grandes ventajas de la plataforma Arduino es que trabaja con licencias libres, por tanto, ofrece su propio entorno de desarrollo (IDE) de forma totalmente gratuita y facilita su descarga a través de la propia página web oficial.

El software está disponible para varios sistemas operativos. Recientemente se ha desarrollado además una herramienta de edición web, con la que no es necesario descargar ningún programa, ya que toda la edición se realiza online y es posible guardar los sketches en la “nube”.

El código del programa se escribe en el editor de texto. A la hora de escribir los sketches, estos siguen un esquema con tres partes claramente diferenciadas:

- En la parte superior del editor y fuera de cualquier función, se llama a las librerías y se definen las variables y constantes globales.
- `void setup()`: Es una función que se utiliza para configurar la placa, así como los posibles módulos. Esta parte del programa solo se ejecuta una vez cuando el microcontrolador arranca.
- `void loop()`: Es la función principal que ejecuta el código presente en ella de forma cíclica a modo de bucle. En ella se estructura la lógica del programa, y se llama a las demás funciones que deben de estar definidas fuera de esta.



```
sketch_aug31a Arduino 1.8.4
sketch_aug31a
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
```

Figura 10. Entorno de Arduino

Dentro de la barra de herramientas de acceso rápido, podemos encontrar seis botones que de izquierda a derecha desempeñan las siguientes funciones:

- **Verificar:** compila el código y muestra en el área de mensajes una confirmación si está todo correcto o un mensaje de error si por el contrario no se ha podido compilar correctamente.
- **Subir:** compila y si no hay errores, carga el programa a la placa.
- **Nuevo:** crea un nuevo sketch vacío.
- **Abrir:** despliega un submenú en el que se pueden abrir sketches de ejemplos de utilización de diferentes tecnologías, o un sketch previamente guardado.
- **Salvar:** guarda el sketch actual.
- **Monitor Serie:** abre el monitor serie, herramienta que permite la comunicación con la placa a través del puerto USB. Es posible tanto leer datos como enviar datos a la placa.

Manejo de librerías

Las librerías facilitan mucho las tareas de programación simplificando los códigos empleados. Además, la inmensa mayoría de las librerías son de código abierto para que todas las personas que las necesiten puedan hacer uso de ellas.

Existen dos tipos generales de librerías:

- **Librerías estándar:** cuando se descarga el IDE de Arduino, vienen incluidas una serie de librerías denominadas estándar. Con estas librerías se pueden ejecutar códigos de ejemplo que también se incluyen en la descarga. Estas librerías incluyen funciones para el manejo de comunicaciones básicas o algún tipo de hardware muy usado como son los servos motores o las pantallas LCD.

- **Librerías instaladas por el usuario:** Existen muchas más librerías con funciones útiles o drivers para dispositivos pertenecientes a todos los tipos de hardware. Estas librerías están disponibles para su descarga en sitios como Arduino Playground, GitHub o Google Code.

Gestor de librerías

Mediante esta opción se abre una ventana, en la que aparece una lista de librerías ya instaladas o que están listas para instalar. Se puede filtrar la búsqueda, o buscar por nombre. Para instalar una nueva librería basta con seleccionar la librería deseada, elegir la versión (si hay más de una disponible) y pinchar en instalar. El gestor de librerías está disponible desde la versión de IDE 1.6.2.

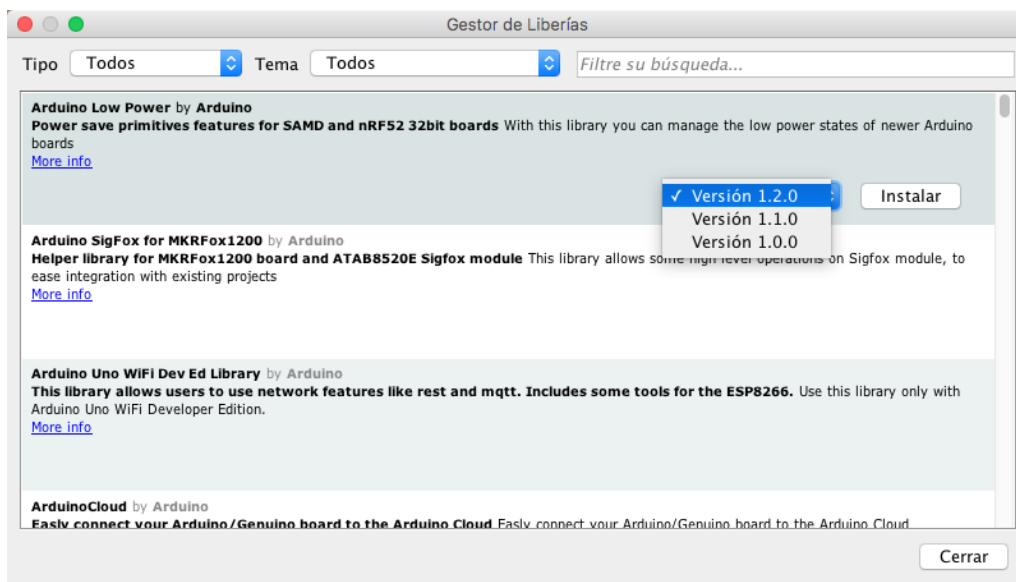


Figura 11. Gestor de librerías

Existe una librería llamada “SoftwareSerial” que permite usar cualquier par de pines de E/S para crear un puerto serie virtual, pero no siempre funciona correctamente y las velocidades de transmisión de datos son menores.

Justificación de la elección de la placa arduino.

La placa elegida para desarrollar este proyecto es la Arduino Mega, resulta conveniente enumerar algunas de las características fundamentales de esta placa.

- Microcontrolador: ATmega2560
- Tensión de operación: 5V
- Tensión de entrada (recomendada): 7-12V
- Tensión de entrada (límite): 6-20V
- Pines de E/S digitales: 54 (de los cuales 15 soportan salidas PWM)
- Entradas analógicas: 16
- Corriente soportada en los pines de E/S: 20mA
- Memoria flash: 256KB de los cuales 8KB son usados por el bootloader
- Memoria SRAM: 8KB
- Memoria EEPROM: 4KB
- Frecuencia del reloj: 16MHz
- Largo: 101.52mm
- Ancho: 53.3mm
- Peso: 37g

Entrenador es un sistema mediante el cual se aprende o se practica utilizar una tecnología, mediante los cuales se facilita el aprendizaje en la programación permitiendo simular y testear ciertas aplicaciones mediante el uso de periféricos como son indicadores Leds conectados a las salidas o interruptores o pulsadores en las entradas entre otros. Contaremos con una placa de desarrollo Arduino y los periféricos que se consideran oportunos aprender a controlar. A continuación, se detallarán imágenes de cómo está formado el entrenador y los circuitos que se obtuvieron del diseño.

Sensor PIR, Ultrasónico y DHT11

En este diagrama se tiene el sensor PIR que es un sensor infrarrojo positivo, es un sensor electrónico que mide la luz infrarroja radiada de los objetos situados en su campo de visión. También se muestra el sensor ultrasónico que miden la distancia mediante el uso de ondas ultrasónicas. El cabezal emite una onda ultrasónica y recibe la onda reflejada que retorna desde

el objeto, contando el tiempo entre la emisión y la recepción. El para medir la temperatura y humedad con Arduino.

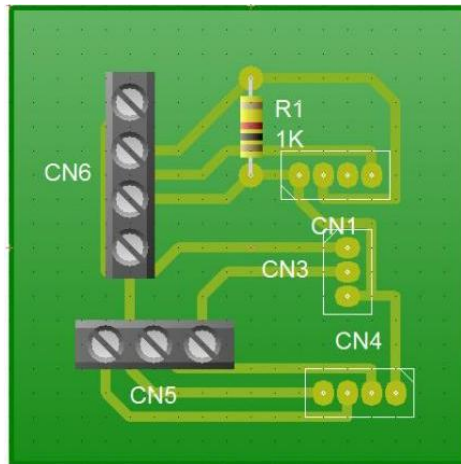


Figura 12. Diagrama de sensor PIR Ultrasónico

En este diagrama se muestra los RGB y se le llama así al tratamiento de la señal de video que trata por separado las señales de los tres colores rojo, verde y azul. Al usarlo independientemente, proporciona mayor calidad y reproducción más fiel del color. También las resistencias cuya función es la de oponerse al paso de la corriente eléctrica en el circuito.

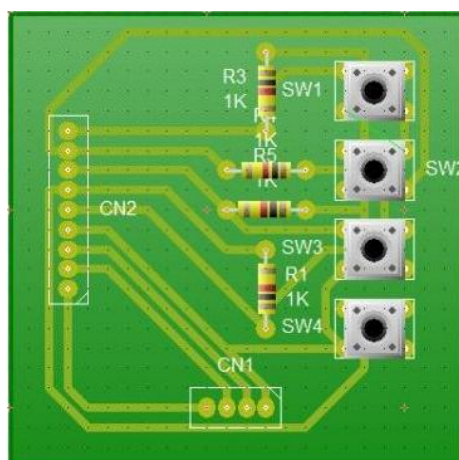


Figura 13. Diagrama con botones RGB y resistencias.

Esta placa de desarrollo permite hacer proyectos rápidamente con la plataforma ESP32. Tiene capacidades con tecnologías de WiFi y BlueTooth, contiene un procesador de doble núcleo. Es de bajo costo y consumo de energía. El ESP32 emplea un microprocesador Tensilica Xtensa LX6 en sus variantes de simple y doble núcleo e incluyendo interruptores de antena, balun de radiofrecuencia, amplificador de potencia, amplificador receptor de bajo ruido, filtros y módulos de administración de energía.

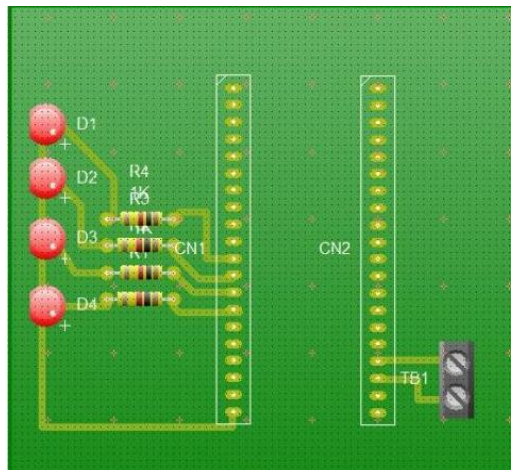


Figura 14. Diagrama ESP32

En este diagrama se demuestra el uso de los motores y es la parte sistemática de una máquina que es capaz de hacer funcionar el sistema, transformando algún tipo de energía (eléctrica) en energía mecánica capaz de realizar un trabajo.

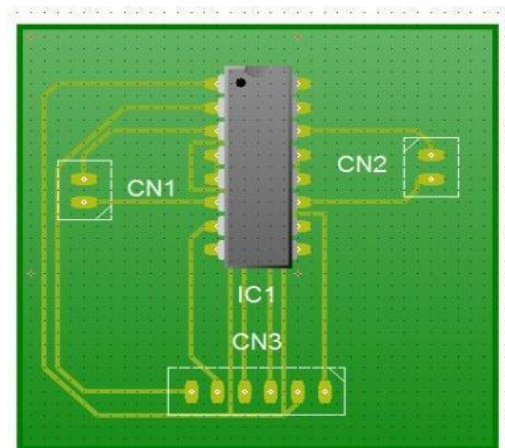


Figura 15. Diagrama de motores

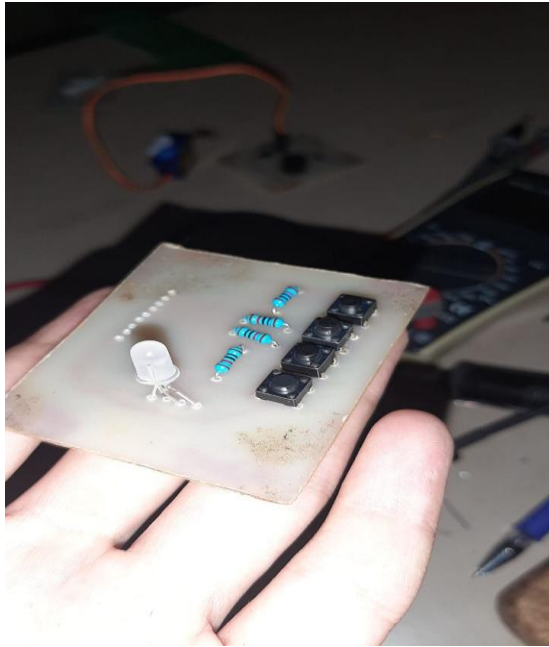


Figura 16. . La placa con sus resistencias en uso y botones RGB

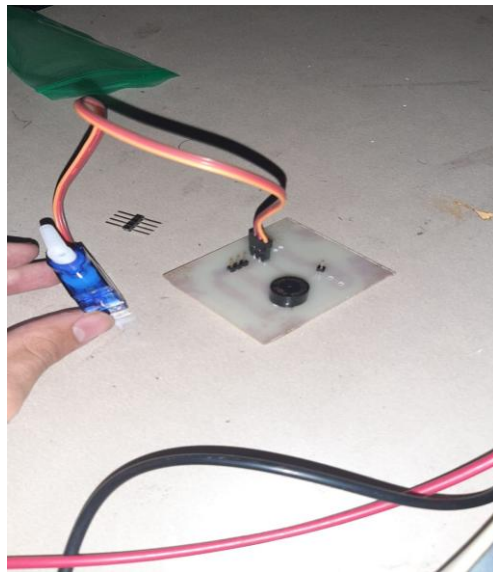


Figura 17. Servo y buzzer

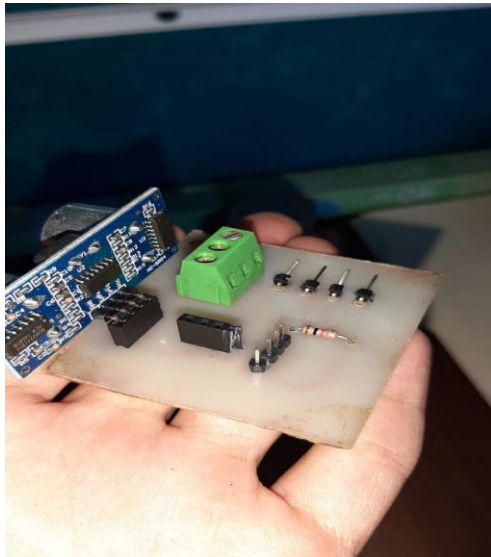


Figura 18. Placa Nano

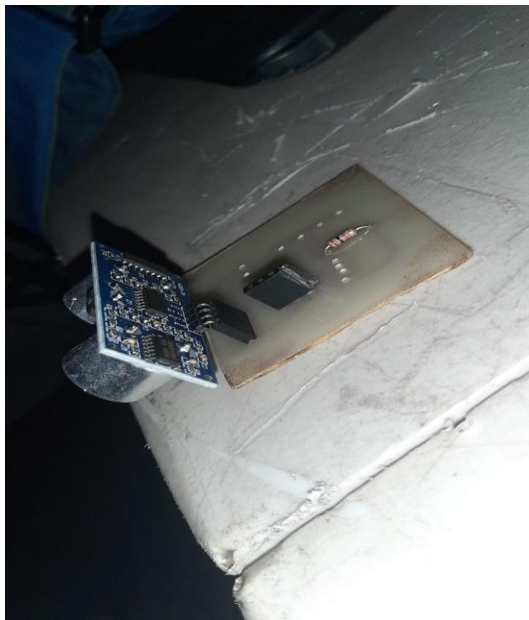


Figura 19. Sensor Ultrasónico



Figura 20. Plancha para diseño circuito impreso

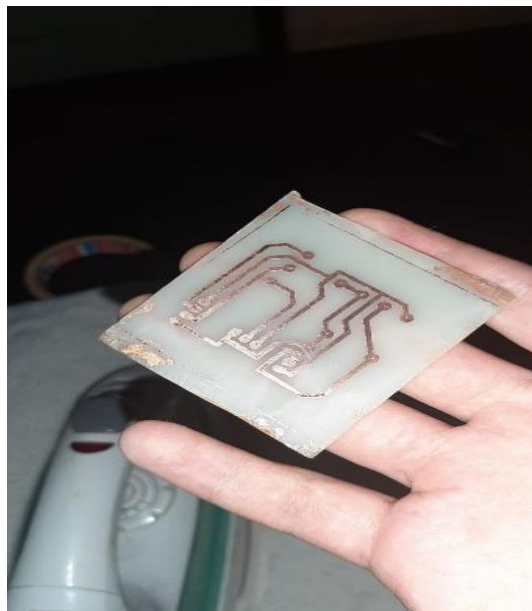


Figura 21. Circuito impreso

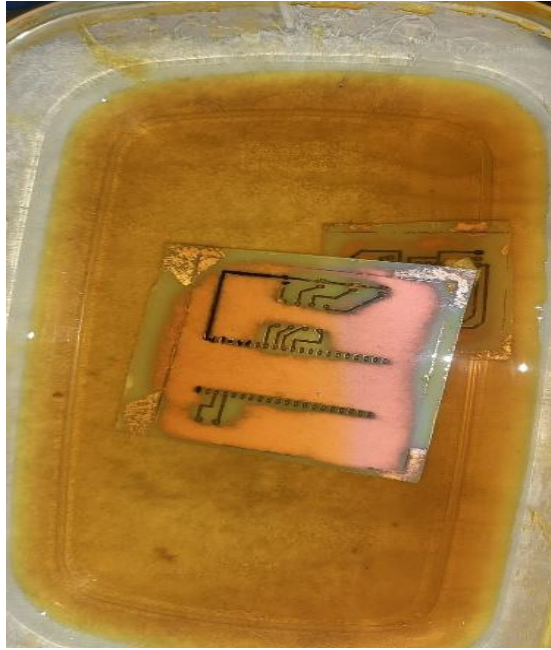


Figura 22. Placa o superficie

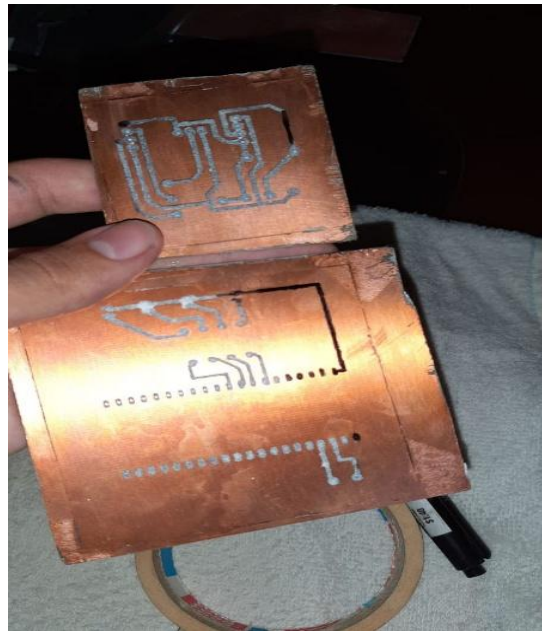


Figura 23. PCB diseñada



Figura 24. Sensor de movimiento

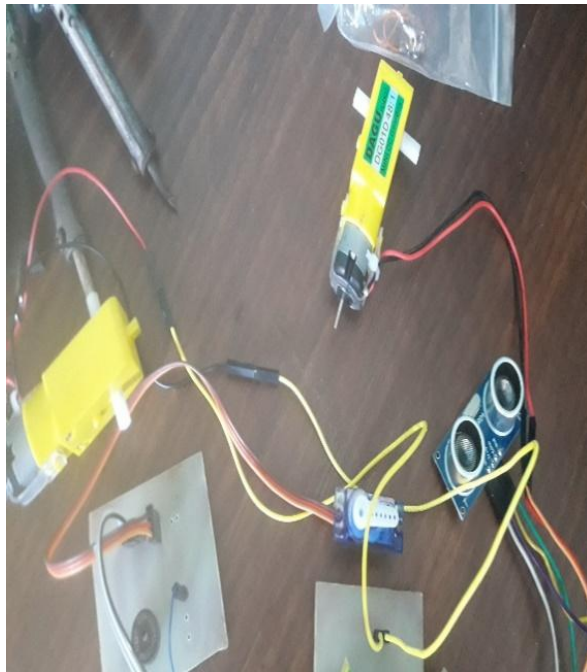


Figura 25. Motores



Figura 26. Pantalla LCD

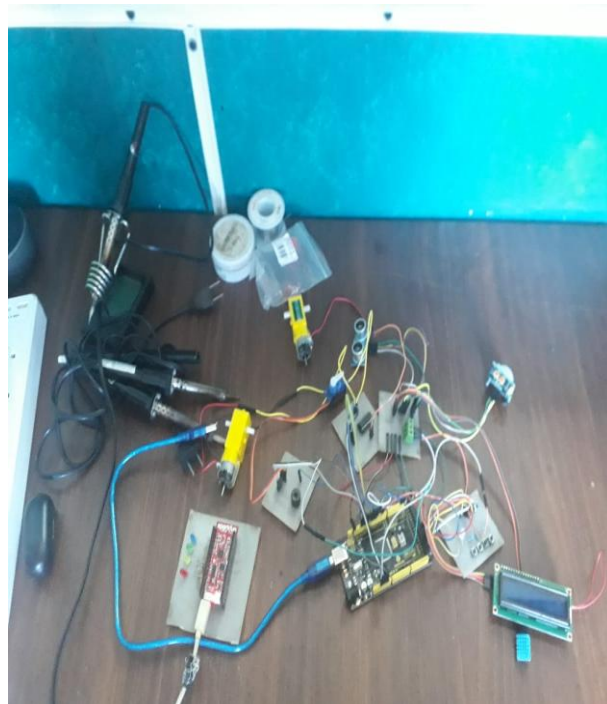
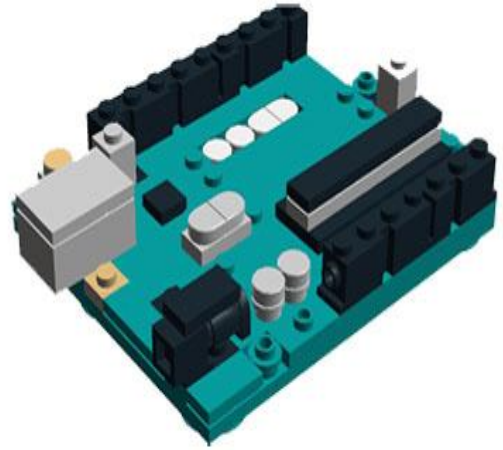


Figura 27. Elementos electrónicos



Manual de guías prácticas con tecnologías Arduino y NodeMCU



4.4 Prácticas con Arduino Mega

Ejercicio 1: programa que controle el parpadeo de un led sin interruptor.

Introducción: En esta práctica, se trabaja la parte electrónica que permite el encendido y apagado de un led con la programación del lenguaje IDE Arduino. Como paso previo se debe descargar e instalar el software IDE de Arduino. Referirse al capítulo 5 como guía para instalar el IDE de Arduino.

Objetivo: Crear un programa que controle el parpadeo de un led, que pueda variar el tiempo de estado alto y el tiempo de estado bajo del mismo.

Para esta práctica se utilizará la instrucción `delay(ms)`, esta instrucción permite apagar y encender un tiempo seleccionado en milisegundos.

Conceptos básicos

Placa Arduino: es una placa electrónica de hardware libre que incorpora un microcontrolador reprogramable y una serie de pines hembra.

Led: Puede traducirse como un diodo emisor de luz, un diodo a su vez, es una válvula de dos electrodos que permite el paso de la corriente electrónica en un único sentido.

Fuente poder: La fuente de alimentación o fuente de potencia es el positivo que convierte la corriente alterna, en una o varias corrientes continuas, que alimentan los distintos circuitos del aparato electrónico al que se conecta.

Cables: significa en su origen cuerda, o también cuerda fuerte, puede tratarse del cordón que presenta varios conductores aislados unos de otros y protegido por algún tipo de envoltura.

Protoboard: Es un tablero con orificios que se encuentran conectados eléctricamente entre sí de manera interna, habitualmente siguiendo patrones de líneas, en el cual se puede insertar componentes electrónicos y cables para el armado y prototipado de circuitos electrónicos y sistemas similares. Está hecho de dos materiales, un aislante, generalmente un plástico, y un conductor que conecta los diversos orificios entre sí.

Materiales:

1. placa Arduino
2. led
3. Fuente de poder
4. Cables
5. Protoboard

Desarrollo

Paso 1: Instalación del IDE de Arduino

Lo primero que se tiene que hacer es instalar el software de Arduino en la computadora, se puede descargar entrando a este link [Arduino Software](#) . Luego se le da clic en Windows installer (en este caso se usará Windows) también el instalador porque trae algunas ventajas extra.

Paso 2

Se puede asignar la función que encienda y apague a cualquier pin digital, exceptuando el 0 y 1, como se comentó anteriormente, por lo tanto, si se asigna el pin 13 como el responsable de esta función, el circuito a montar es muy sencillo:

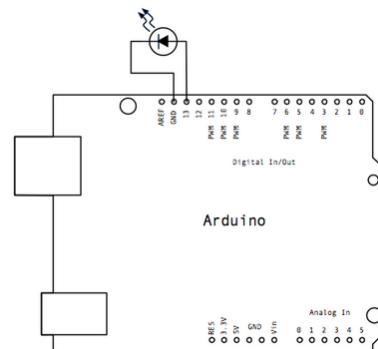
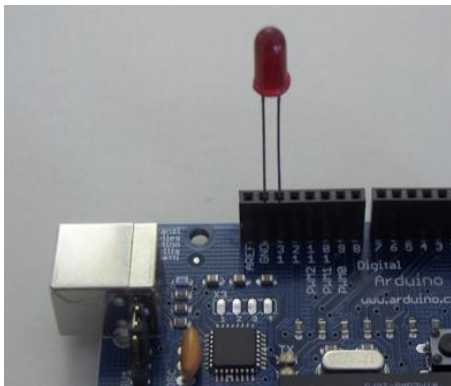


Figura 28. Esquema de circuito encendido y apagado led

Paso 3: Abrir el IDE de Arduino para comenzar a programar. A continuación, abrir el programa Arduino con el acceso directo creado en el escritorio.

Paso 4: Para poder empezar primero se debe entender como es la estructura básica de Arduino, que parámetros necesitamos escribir antes de escribir el código para nuestro proyecto.

Paso 5: Y el código de programación para el control del parpadeo del LED se muestra a continuación.

```
/*  
 * Intermitente  
 *  
 * Ejemplo básico con Arduino. Encendido y apagado de un led  
 * con una cadencia de 1 seg. Usando el PIN 13 como salida  
 * No es necesario usar una resistencia para el led  
 * La salida 13 de Arduino la lleva incorporada.  
 *  
 * http://www.arduino.cc/en/Tutorial/Blink  
 */  
int ledPin = 13; // Definición de la salida en el PIN 13  
void setup() //Configuración  
{  
  pinMode(ledPin, OUTPUT); // designa la salida digital al PIN 13  
}  
void loop() // bucle de funcionamiento
```

```
{  
digitalWrite(ledPin, HIGH); // activa el LED  
delay(1000); // espera 1 seg. (tiempo encendido)  
digitalWrite(ledPin, LOW); // desactiva el LED  
delay(1000); // espera 1 seg. (tiempo apagado)
```

Preguntas sobre la guía

- 1- ¿Qué línea de código determina el encendido y apagado de un led?
- 2- ¿Qué función desempeña el pin 13?
- 3- ¿Cuál es la función que activa un led?

Conclusión

Recomendaciones:

Ejercicio 2: programa que controle el parpadeo de un led con pulsador.

Introducción: Cuando se pulsa el pulsador (entrada 5 a “0”) se enciende y se apaga de forma intermitente la salida 13.

Objetivo: Conocer las partes de la placa Arduino Mega. Conocer la placa protoboard. Conocer el código de colores de una resistencia. Conocer cómo funciona un LED. Instalar Arduino Mega.

Conceptos básicos

Placa Arduino: es una placa electrónica de hardware libre que incorpora un microcontrolador reprogramable y una serie de pines hembra.

Led: Puede traducirse como un diodo emisor de luz, un diodo a su vez, es una válvula de dos electrodos que permite el paso de la corriente electrónica en un único sentido.

Fuente poder: La fuente de alimentación o fuente de potencia es el positivo que convierte la corriente alterna, en una o varias corrientes continuas, que alimentan los distintos circuitos del aparato electrónico al que se conecta.

Cables: significa en su origen cuerda, o también cuerda fuerte, puede tratarse del cordón que presenta varios conductores aislados unos de otros y protegido por algún tipo de envoltura.

Protoboard: Es un tablero con orificios que se encuentran conectados eléctricamente entre sí de manera interna, habitualmente siguiendo patrones de líneas, en el cual se puede insertar componentes electrónicos y cables para el armado y prototipado de circuitos electrónicos y sistemas similares. Está hecho de dos materiales, un aislante, generalmente un plástico, y un conductor que conecta los diversos orificios entre sí.

Pulsador: Es un operador eléctrico, que cuando se oprime, permite el paso de la corriente eléctrica y, cuando se deja de oprimir, la interrumpe.

Resistencia: Es toda oposición que encuentra la corriente a su paso por un circuito eléctrico cerrado, atenuando o frenando el libre flujo de circulación de las cargas eléctricas o electrones.

Materiales:

1. placa Arduino
2. led
3. Fuente de poder
4. Cables
5. Protoboard
6. Pulsador
7. Resistencia

Desarrollo:

Paso 1: Funcionamiento:

Cuando la E5 = 1 Entonces S13 = 0

Cuando la E5 = 0 Entonces S13 = 0-1 (Intermitente 200,200 ms)

Paso 2: Hacer que un led se encienda de forma intermitente de la siguiente manera: en cada ciclo, el encendido durara medio segundo y el apagado un segundo. Para ello, conectaremos el ánodo del led al pin 13. El cátodo del led ira a una resistencia de 220Ω para proteger el led. La resistencia ira a tierra. Hecho esto, ya sabemos que cuando el pin 13, que es digital, este en alto (5 V), el led se encenderá y cuando el pin 13 este en bajo (0 V), el led se apagará.

Paso 3: Se debe conseguir que el led se encienda mientras se esté accionando el pulsador. Para ello se utilizará Arduino simplemente como una pila.

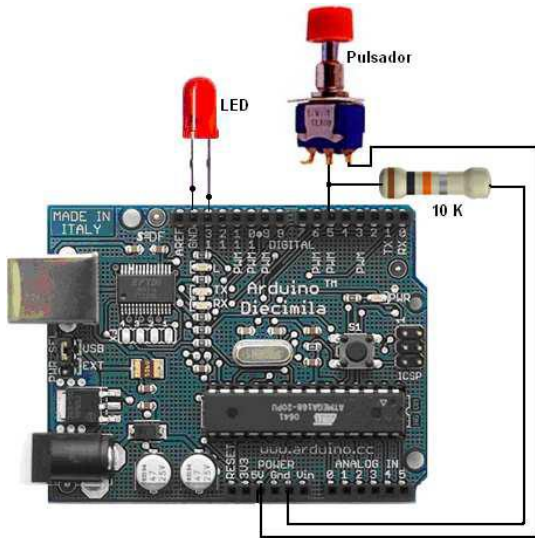


Figura 29. Diagrama de conexión de circuito de led con pulsador

Paso 4: A continuación, se muestra el código que representa toda la funcionalidad

Programa:

```
int ledPin= 13; // choose the pin for the LED

int inPin= 5; // choose the input pin (for a pushbutton)

int val= 0; // variable for reading the pin status

void setup() {

pinMode(ledPin, OUTPUT); // declare LED as output

pinMode(inPin, INPUT); // declare pushbutton as input

}

void loop(){

val= digitalRead(inPin); // lee valor de entrada

if(val== HIGH) { // chequea si el valor leído es "1" (botón presionado)

digitalWrite(ledPin, LOW); // pone el LED en OFF

} else{
```

```
digitalWrite(ledPin, LOW); // parpadea el LED
```

```
delay(200);
```

```
digitalWrite(ledPin, HIGH);
```

```
delay(200);
```

```
}
```

```
}
```

Preguntas sobre la guía

- 1- ¿Cómo haría que el led se encienda mientras al menos uno de los dos pulsadores este accionando?
- 2- ¿Cuál es la función de void loop()?
- 3- ¿Qué función desempeña la sentencia if?

Conclusión

Recomendaciones

Ejercicio 3: Secuencia Básica de encender y apagar tres leds.

Introducción: Se trata de encender y apagar tres leds colocados en las salidas 6, 7 y 8 (PIN6, PIN7 y PIN8) con una cadencia de 200 ms. Las variables asignadas a cada led son ledPin1, ledPin2 y ledPin3.

Objetivos: Conocer la funcionalidad de los pines (6, 7 y 8) y las variables asignadas a cada pin, así como su respectiva secuencia y de esta manera crear un programa que controle el encendido y apagado de tres leds.

Conceptos básicos:

Placa Arduino: es una placa electrónica de hardware libre que incorpora un microcontrolador reprogramable y una serie de pines hembra.

Led: Puede traducirse como un diodo emisor de luz, un diodo a su vez, es una válvula de dos electrodos que permite el paso de la corriente electrónica en un único sentido.

Fuente poder: La fuente de alimentación o fuente de potencia es el positivo que convierte la corriente alterna, en una o varias corrientes continuas, que alimentan los distintos circuitos del aparato electrónico al que se conecta.

Cables: significa en su origen cuerda, o también cuerda fuerte, puede tratarse del cordón que presenta varios conductores aislados unos de otros y protegido por algún tipo de envoltura.

Protoboard: Es un tablero con orificios que se encuentran conectados eléctricamente entre sí de manera interna, habitualmente siguiendo patrones de líneas, en el cual se puede insertar componentes electrónicos y cables para el armado y prototipado de circuitos electrónicos y sistemas similares. Está hecho de dos materiales, un aislante, generalmente un plástico, y un conductor que conecta los diversos orificios entre sí.

Un pulsador: es un operador eléctrico que, cuando se oprime, permite el paso de la corriente eléctrica y, cuando se deja de oprimir, lo interrumpe. - Un interruptor es un operador eléctrico, cuya función es abrir o cerrar un circuito de forma permanente.

Resistencia: Es toda oposición que encuentra la corriente a su paso por un circuito eléctrico cerrado, atenuando o frenando el libre flujo de circulación de las cargas eléctricas o electrones.

Materiales:

- 1- Plataforma Arduino
- 2- led
- 3- Fuente de poder
- 4- Cables
- 5- Protoboard
- 6- Pulsador
- 7- Resistencia

Desarrollo:

- 1- En el primer paso se debe instalar el software de Arduino en la computadora, se puede descargar entrando a este link [Arduino Software](#) . Luego se le da clic en Windows installer (en este caso se usará Windows) también el instalador porque trae algunas ventajas extra.
- 2- Seleccionar los leds y las resistencias necesarias para que cumplan con la función esperada.
- 3- Se muestra en el diagrama la forma que tendrán ya instalados los elementos electrónicos en el entrenador de Arduino.

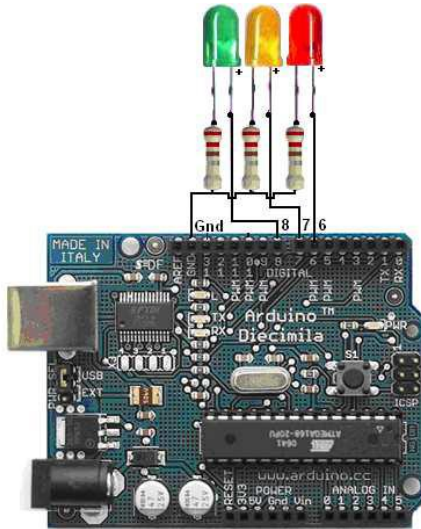


Figura 30. Diagrama de circuito de apagar y encender 3 leds

4- Secuencia de tiempo al encender y apagar un led se muestra a continuación.

Delay (200 ms)

Desactivar ledPin2

PIN7=LOW

Delay (200 ms)

Activa ledPin2

PIN7=HIGH

Delay (200 ms)

Desactivar ledPin3

PIN8=LOW

Delay (200 ms)

Activa ledPin3

PIN8=HIGH

PIN 7 220 ohm

ledPin2

PIN 8 220 ohm

5- Código para que los tres leds desempeñen la función de encender y apagar de forma secuencial.

Programa

```
// Encendido y apagado de 3 Leds
int ledPin1 = 6; // Define las salidas de los Leds
int ledPin2 = 7;
int ledPin3 = 8;

void setup() { // Configura las SALIDAS
pinMode(ledPin1, OUTPUT); // declarar Leds como SALIDAS
pinMode(ledPin2, OUTPUT);
pinMode(ledPin3, OUTPUT);
digitalWrite(ledPin1, LOW); // Apaga los Leds
digitalWrite(ledPin2, LOW);
digitalWrite(ledPin3, LOW);
}

void loop(){ //Bucle de Funcionamiento
digitalWrite(ledPin1, HIGH); // Apaga y enciende los leds cada 200 ms
delay(200);
digitalWrite(ledPin1, LOW);
digitalWrite(ledPin2, HIGH);
delay(200);
digitalWrite(ledPin2, LOW);
digitalWrite(ledPin3, HIGH);
delay(200);
digitalWrite(ledPin3, LOW);
}
```

Preguntas sobre la guía

- 1- ¿Qué función desempeña void setup?
- 2- ¿Para qué sirve la función Delay?
- 3- ¿En que se mide una resistencia?

Conclusión

Recomendaciones

Ejercicio 4: Funcionalidad de un pulsador utilizando la placa Arduino.

Introducción: El pulsador es un componente que conecta dos puntos de un circuito cuando es presionado. Para generar una señal de tensión con el pulsador, se necesita un divisor de tensión.

La resistencia R_{abajo} (pull-down) en el primer circuito fuerza a V_{out} , llegando a nivel CERO, hasta que se acciona el pulsador. Este circuito entrega una tensión alta, cuando se presiona el pulsador. Un valor para la resistencia de 10 k es recomendable.

Objetivo: Conocer las características y aplicaciones de un pulsador, trabajando en la plataforma arduino.

Conceptos básicos:

Placa Arduino: es una placa electrónica de hardware libre que incorpora un microcontrolador reprogramable y una serie de pines hembra.

Un pulsador: es un operador eléctrico que, cuando se oprime, permite el paso de la corriente eléctrica y, cuando se deja de oprimir, lo interrumpe. - Un interruptor es un operador eléctrico, cuya función es abrir o cerrar un circuito de forma permanente.

Led: Puede traducirse como un diodo emisor de luz, un diodo a su vez, es una válvula de dos electrodos que permite el paso de la corriente electrónica en un único sentido.

Fuente de poder: La fuente de alimentación o fuente de potencia es el positivo que convierte la corriente alterna, en una o varias corrientes continuas, que alimentan los distintos circuitos del aparato electrónico al que se conecta.

Cables: significa en su origen cuerda, o también cuerda fuerte, puede tratarse del cordón que presenta varios conductores aislados unos de otros y protegido por algún tipo de envoltura.

Protoboard: Es un tablero con orificios que se encuentran conectados eléctricamente entre sí de manera interna, habitualmente siguiendo patrones de líneas, en el cual se puede insertar componentes electrónicos y cables para el armado y prototipado de circuitos electrónicos y sistemas similares. Está hecho de dos materiales, un aislante, generalmente un plástico, y un conductor que conecta los diversos orificios entre sí.

Resistencia: Es toda oposición que encuentra la corriente a su paso por un circuito eléctrico cerrado, atenuando o frenando el libre flujo de circulación de las cargas eléctricas o electrones.

Materiales:

1. Placa arduino
2. Pulsador
3. Un diodo led
4. Fuente de poder
5. Cables para realizar las conexiones.
6. Protoboard
7. Una resistencia de 1K Ohmios.

Desarrollo:

Paso 1: En el segundo circuito, la resistencia Rarriba (pull-up) fuerza a nivel alto a Vout, mientras no se actúe sobre el pulsador. Al presionar el pulsador, se conecta Vout directamente con 0 V. Es decir, este circuito entrega un nivel bajo cuando se presiona el pulsador.

Paso 2: Se utiliza una resistencia pull-down, junto con un pulsador, para conectarla a un pin de entrada digital, y de esta forma, poder saber cuándo el pulsador es presionado. Si el pulsador está presionado, el valor del pin 10 será de 0 voltios (LOW) en caso contrario

Será de + 5 voltios (HIGH). En una placa protoboard debe haber una resistencia de 10K conectada entre el pin de entrada y tierra.

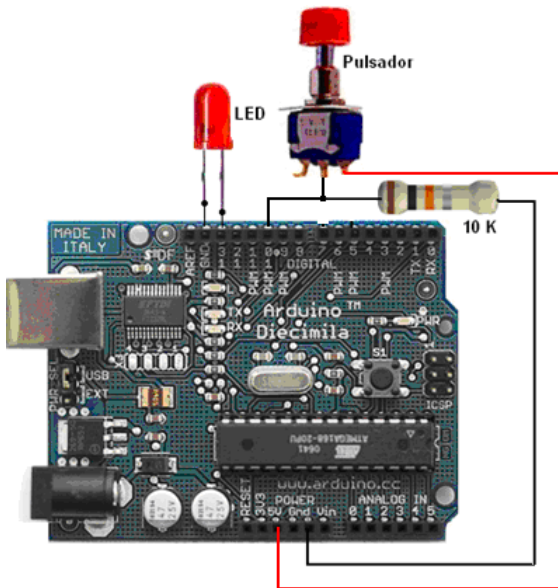


Figura 31. Diagrama de circuito de led y un pulsador

Paso 3: A continuación, un organigrama o esquema que muestra cómo estará formado el circuito.

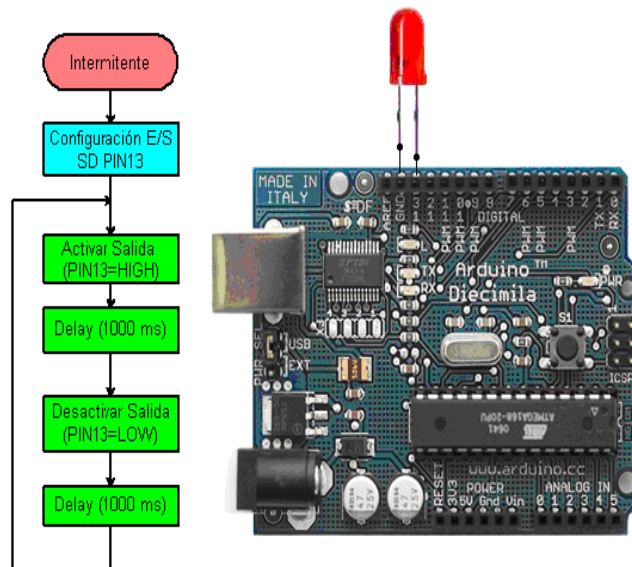


Figura 32. Esquema de circuito con led

Paso 4: De manera secuencial, se muestra a continuación el código fuente que corresponde a funcionalidad de un botón y un led.

Código fuente

Pulsador

```
* -----  
  
*  
  
* Detecta si el botón ha sido presionado o no  
  
* Y enciende el LED en consecuencia.  
  
*  
  
* Massimo Banzi  
  
*  
  
*/  
  
int ledPin = 13; // PIN del LED  
  
int inPin = 10; // PIN del pulsador  
  
int value = 0; // Valor del pulsador  
  
void setup() {  
  
  pinMode(ledPin, OUTPUT); // Inicializa el pin 13 como salida digital  
  
  pinMode(inPin, INPUT); // Inicializa el pin 10 como entrada digital  
  
}  
  
void loop() {  
  
  value = digitalRead(inPin); // Lee el valor de la entrada digital  
  
  digitalWrite(ledPin, value);  
  
}
```

Preguntas sobre la guía

- 1- ¿Cuál es la función del pulsador?
- 2- ¿Para qué sirve la fuente de poder?
- 3- ¿Describir brevemente que pasaría si no se utiliza digitalRead?

Conclusiones

Recomendaciones

Ejercicio 5: Sensor de fuerza

Introducción: Se trata de convertir un zumbador piezoeléctrico en un sensor de presión o fuerza utilizando este como sensor de entrada en uno de los pines de entrada analógica de Arduino (PIN 2).

El sensor de fuerza o presión mf01 auto adherible para Arduino es ideal para detectar una fuerza aplicada en la membrana. Básicamente, al detectar una flexión en la membrana el sensor cambia su resistencia interna. Este sensor está fabricado de 2 capas separadas por un espaciador, cuanto más se presiona, por ejemplo, más puntos de elemento activo tocan el semiconductor. Por lo tanto, esto hace que la resistencia disminuya. Cuando no es presionado, su resistencia es de aproximadamente 20 MΩ. Su rango de presión varía de 30 gramos a 1 Kg. Por lo que basta con ponerlo en un divisor de tensión, para poder medir la salida en voltaje con algún ADC.

Objetivos: Conocer la funcionalidad del sensor de fuerza o de presión a través de una práctica utilizando diferentes elementos electrónicos.

Conceptos básicos:

Sensor de fuerza: Los sensores de fuerza, o células de carga, son dispositivos que nos permiten obtener una señal eléctrica proporcional a la fuerza que se aplica sobre ellos. Estos transductores se presentan en múltiples formatos, ya que los requisitos mecánicos de los sistemas en los que se integran son también muy variados.

Placa Arduino: es una placa electrónica de hardware libre que incorpora un microcontrolador reprogramable y una serie de pines hembra.

Un pulsador: es un operador eléctrico que, cuando se oprime, permite el paso de la corriente eléctrica y, cuando se deja de oprimir, lo interrumpe. - Un interruptor es un operador eléctrico, cuya función es abrir o cerrar un circuito de forma permanente.

Led: Puede traducirse como un diodo emisor de luz, un diodo a su vez, es una válvula de dos electrodos que permite el paso de la corriente electrónica en un único sentido.

Fuente de poder: La fuente de alimentación o fuente de potencia es el positivo que convierte la corriente alterna, en una o varias corrientes continuas, que alimentan los distintos circuitos del aparato electrónico al que se conecta.

Cables: significa en su origen cuerda, o también cuerda fuerte, puede tratarse del cordón que presenta varios conductores aislados unos de otros y protegido por algún tipo de envoltura.

Protoboard: Es un tablero con orificios que se encuentran conectados eléctricamente entre sí de manera interna, habitualmente siguiendo patrones de líneas, en el cual se puede insertar componentes electrónicos y cables para el armado y prototipado de circuitos electrónicos y sistemas similares. Está hecho de dos materiales, un aislante, generalmente un plástico, y un conductor que conecta los diversos orificios entre sí.

Resistencia: Es toda oposición que encuentra la corriente a su paso por un circuito eléctrico cerrado, atenuando o frenando el libre flujo de circulación de las cargas eléctricas o electrones.

Materiales:

- 1- Sensor de fuerza
- 2- Placa arduino
- 3- Pulsador
- 4- Un diodo led
- 5- Fuente de poder
- 6- Cables para realizar las conexiones.
- 7- Protoboard
- 8- Una resistencia de 1K Ohmios.

Desarrollo:

Paso 1: Instalar el software libre arduino en la computadora con las respectivas librerías a utilizar en la codificación.

Paso 2: Recolectar la cantidad de materiales que sean necesarios para desarrollar dicha práctica, tomando en cuenta las características de cada material.

Paso 3: Esquema que muestra la ubicación de sensor de fuerza.

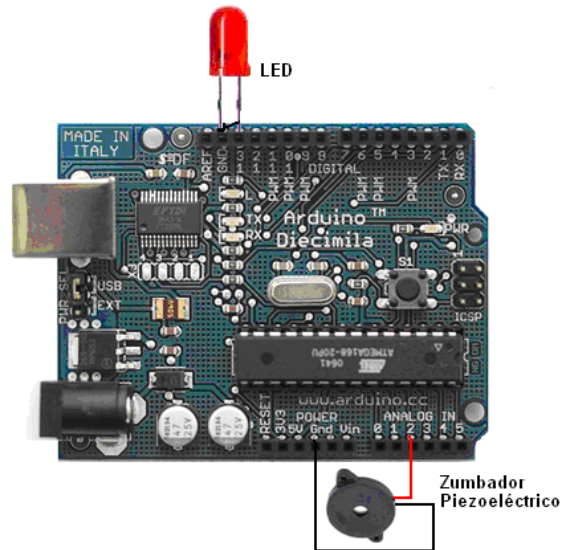


Figura 33. Diagrama del circuito con sensor piezoeléctrico

Paso 4: El código que corresponde al sensor de fuerza con las librerías necesarias.

```
/* Sensor piezoeléctrico
```

```
* -----
```

```
* Convertir un zumbador piezoeléctrico en un sensor de fuerza
```

```
*
```

```
* Created 24 October 2006
```

```
* copyleft 2006 Tod E. Kurt <tod@todbot.com
```

```
* http://todbot.com/
```

```
*/
```

```

int ledPin = 13;

int piezoPin = 2;

int THRESHOLD = 1; // Configura valor mínimo para que se encienda la salida
                    pin 13

int val = 0; // variable que almacena el valor leído por el sensor

int t = 0; // valor del intervalo de medida

void setup() {

pinMode(ledPin, OUTPUT);

Serial.begin(19200);

Serial.println("ready"); // indicador de espera

}

void loop() {

digitalWrite(ledPin,LOW); // indicador de reposo (esperando)

val = analogRead(piezoPin); // lectura de valor del piezoeléctrico

if( val > THRESHOLD ) { // si el valor leído es superior al mínimo establecido

digitalWrite(ledPin, HIGH); // activa salida 13

t = 0;

while(analogRead(piezoPin) > THRESHOLD) {

t++;

} // wait for it to go LOW (espera con una pequeña histéresis)

if(t>100) { // escribe en el puerto

```

```
Serial.print("knock! ");
```

```
//Serial.println(t);
```

```
}
```

```
}
```

```
}
```

Preguntas sobre la guía:

- 1- ¿Para qué sirve el sensor de fuerzas?
- 2- ¿Cuál es la funcionalidad de la sentencia if?
- 3- ¿Describir brevemente el siguiente código digital `Write(ledPin,LOW)` ?

Conclusión:

Recomendaciones:

Ejercicio 6. Control de un led RGB

Introducción: Se usará un led RGB, lo que permitirá obtener cualquier color en el led. Así, lo que se pretende es poner en el led RGB el color que queramos, controlado mediante tres potenciómetros.

Un led RGB, en realidad, son tres leds: uno rojo (R), uno verde (G) y uno azul (B), que son los tres colores primarios luz. Mezclando estos tres colores primarios, se puede obtener cualquier otro color. El modelo de led RGB con el que se ha hecho esta práctica tiene 4 patas en este orden: ánodo R (rojo), cátodo común, ánodo B (azul) y ánodo G (verde). Esto puede variar de unos modelos a otros.

Objetivo: Utilizar los colores básicos de tres leds (rojo, verde, azul), luego mezclar para obtener colores variados.

Conceptos Básicos:

Led RGB, RGB es una sigla formada por los términos de la lengua inglesa *red* (“rojo”), *green* (“verde”) y *blue* (“azul”). El concepto suele emplearse para referirse a un modelo cromático que consiste en representar distintos colores a partir de la mezcla de estos tres colores primarios. (Julian, 2014)

Potenciómetro: Es un resistor eléctrico variable de tres terminales y un contacto deslizante o giratorio que permite ajustar el voltaje o tensión eléctrica. También permite ajustar el amperaje o intensidad eléctrica.

Resistencia: Es toda oposición que encuentra la corriente a su paso por un circuito eléctrico cerrado, atenuando o frenando el libre flujo de circulación de las cargas eléctricas o electrones.

Materiales.

1. 1 led RGB,
2. potenciómetros de 10 k Ω ,
3. resistencias de 220 Ω (rojo-rojo marrón).

Desarrollo:

- 1- Instalar el software libre de Arduino y las librerías necesarias para poder codificar la funcionalidad que se desea mostrar.
- 2- Verificar que todo esté instalado correctamente y utilizar el circuito como guía para codificar paso a paso cada parte o elemento que se esté utilizando.
- 3- Se pretende controlar el color global del led. Por ejemplo, el rojo puro tiene un valor RGB: R=255, G=0; B=0.
- 4- El verde puro: R=0, G=255, B=0. El logo de Oikos Matematikon: R=103, G=22, B=116. Se debe controlar el led para conseguir los siguientes colores: rojo, verde, azul y logo Oikos Matematikon y logo Arduino.
- 5- Como pistas, en cuanto al led, se debe conectar cada ánodo a un pin digital tipo PWM~; por ejemplo, la pata R al pin 11, la pata G al 9 y la pata B al 10. En cuanto a los potenciómetros se debe conectar cada terminal medio a un pin analógico; por ejemplo, el potenciómetro que controle el color rojo al pin A2, el que controle el verde al A1 y el que controle el azul al A0.
- 6- El código diseñado a continuación es el que define la funcionalidad de los leds.

Sketch

```
const int ledR = 11; //el ánodo rojo del led al pin 11
const int ledG = 9; //el ánodo verde del led al pin 9
const int ledB = 10; //el ánodo azul del led al pin 10
const int potR = A2; //el terminal medio del poten rojo al pin A2
const int potG = A1; //el terminal medio del poten verde al pin A1
const int potB = A0; //el terminal medio del poten azul al pin A0
int brilloR; //nos dará la cantidad de rojo (entre 0 y 255)
int brilloG; //nos dará la cantidad de verde (entre 0 y 255)
int brilloB; //nos dará la cantidad de azul (entre 0 y 255)
void setup()
Serial.begin(9600); //esta sentencia es necesaria para usar la pantalla
pinMode(ledR, OUTPUT);
pinMode(ledG, OUTPUT);
pinMode(ledB, OUTPUT);
}
void loop() {
brilloR=analogRead(potR)/4;
brilloG=analogRead(potG)/4;
brilloB=analogRead(potB)/4;
analogWrite(ledR,brilloR);
analogWrite(ledG,brilloG);
analogWrite(ledB,brilloB);
Serial.print("R: ");
Serial.print(brilloR);
Serial.print(", G: ");
Serial.print(brilloG);
Serial.print(", B: ");
Serial.println(brilloB);
delay(100);
}
```

Preguntas sobre la guía

- 1- ¿Cuál es la función de esta sentencia `Serial.begin(9600)`?
- 2- ¿Cuál es la funcionalidad de `analogRead(potR)/4`?
- 3- ¿para qué sirve un potenciómetro?

Conclusiones:

Recomendaciones:

Ejercicio 7. Medidor de temperatura (dispositivo TMP36)

Introducción. Se pretende que según el calor que desprenda tu mano se encienda uno, dos o tres leds. Para ello se utilizará el sensor de temperatura TMP36. Cómo funciona el TMP36. Si se mira por su parte plana, el terminal izquierdo se conecta a 5 V, el terminal derecho se conecta a 0 V y el terminal central a un pin analógico; por ejemplo, al A0. El voltaje del terminal central estará comprendido entre 0 V y 5 V y será menor cuanto mayor sea la temperatura.

El TMP36 funciona entre $-50\text{ }^{\circ}\text{C}$ y $125\text{ }^{\circ}\text{C}$. No es muy preciso ya que tiene una incertidumbre de $\pm 1\text{ }^{\circ}\text{C}$, pero sirve. La función que relaciona el voltaje de salida del TMP36 con la temperatura se debe calcular a partir de estas pistas:

- El voltaje de salida del TMP36 vale 0 V cuando la temperatura es de $-50\text{ }^{\circ}\text{C}$.

Cada vez que la temperatura aumenta un grado el voltaje de salida aumenta 10 mV.

Con esas dos pistas se debe sacar la expresión de la temperatura (en grados centígrados) en función del voltaje (en voltios).

Objetivos: Medir la temperatura por medio de un elemento electrónico llamado medidor de temperatura.

Conceptos básicos:

Sensor de fuerza: Los sensores de fuerza, o células de carga, son dispositivos que nos permiten obtener una señal eléctrica proporcional a la fuerza que se aplica sobre ellos. Estos transductores se presentan en múltiples formatos, ya que los requisitos mecánicos de los sistemas en los que se integran son también muy variados.

Led: Puede traducirse como un diodo emisor de luz, un diodo a su vez, es una válvula de dos electrodos que permite el paso de la corriente electrónica en un único sentido.

Resistencia: Es toda oposición que encuentra la corriente a su paso por un circuito eléctrico cerrado, atenuando o frenando el libre flujo de circulación de las cargas eléctricas o electrones.

Materiales:

1. 1 sensor TMP36
2. leds rojos y
3. resistencias de $220\ \Omega$ (rojo-rojo-marrón).

Desarrollo:

Paso 1: Instalar la plataforma de software libre de Arduino. Descargándolo desde un sitio web.

Paso 2. Agregar las librerías que se van a necesitar para la codificación.

Paso 3: Se declara una variable llamada lectura que leerá el pin A0 (valor entre 0 y 1023); además, una variable en coma flotante llamada voltaje que dará el voltaje en voltios del pin A0 (valor entre 0 y 5). También una variable en coma flotante llamada temperatura que dará la temperatura en grados centígrados.

Paso 4: Ahora se explicará la parte de la práctica correspondiente a la iluminación de leds. Esta parte puede variar sus valores de temperatura de referencia en función de la temperatura que haga en la habitación. Cuando la temperatura este por encima de $35\ ^\circ\text{C}$ se encenderán los tres leds. Cuando la temperatura este entre $30\ ^\circ\text{C}$ y $34\ ^\circ\text{C}$ se encenderán dos leds. Cuando la temperatura este entre $25\ ^\circ\text{C}$ y $29\ ^\circ\text{C}$ se encenderá un led. Cuando la temperatura este por debajo de $25\ ^\circ\text{C}$ no se encenderá ningún led.

Paso 5: El código para medir la temperatura es el siguiente.

Sketch

```
const int led1 = 13;
```

```
const int led2 = 10;
```

```
const int led3 = 7;
```

```

const int temp1 = 35;
const int temp2 = 30;
const int temp3 = 25;
int lectura;
float voltaje;
float temperatura;
void setup() {
  Serial.begin(9600);
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(led3,OUTPUT);
}
void loop() {
  lectura=analogRead(A0);
  voltaje=lectura*5.0/1023.0;
  temperatura=100*voltaje-50;
  Serial.println(voltaje);
  if (temperatura>=temp1){
    digitalWrite(led1,HIGH);
    digitalWrite(led2,HIGH);
    digitalWrite(led3,HIGH);
  }
  else if (lectura>=temp2){
    digitalWrite(led1,HIGH);
    digitalWrite(led2,HIGH);
    digitalWrite(led3,LOW);
  }
  else if (lectura>=temp3){
    digitalWrite(led1,HIGH);
    digitalWrite(led2,LOW);
    digitalWrite(led3,LOW);
  }
}

```

```
}  
else {  
digitalWrite(led1,LOW);  
digitalWrite(led2,LOW);  
digitalWrite(led3,LOW);  
}  
delay(1000);  
}
```

Preguntas sobre sobre la guía

- 1- ¿Cuál es la función de Sensor?
- 2- ¿Para qué sirve delay?
- 3- ¿Qué significa la línea de código: digitalWrite(led1,LOW);?

Conclusiones:

Recomendaciones:

Ejercicio 8: Control del sensor de ultrasonidos

Introducción. En esta práctica vamos a aprender a medir la distancia desde un sensor a un objeto. Para ello, se usará el sensor de ultrasonidos HC-SR04. Se pide que por el serial salga la distancia, en cm, entre el sensor de ultrasonidos y el primer objeto que se interponga con el sensor.

El rango de medición del HC-SR04 está entre los dos 2 cm y los 500 cm; fuera de estos límites no mide con exactitud. La resolución es de 0,3 cm. La idea básica es que el sensor lanza un ultrasonido que chocara con el objeto regresando al sensor. Sabiendo el tiempo que dura este proceso y la velocidad del sonido (340 m/s), podremos calcular la distancia a la que se encuentra el objeto.

Objetivos: Calcular el valor de la distancia por medio de un elemento electrónico.

Conceptos Básicos:

Placa Arduino: es una placa electrónica de hardware libre que incorpora un microcontrolador reprogramable y una serie de pines hembra.

Sensor Ultrasónico:

Los sensores de ultra sonido o sensores ultrasónicos son detectores de proximidad que trabajan libre de partes mecánicas y que detectan objetos a distancia que van desde pocos centímetros hasta varios metros. Emite sonido y mide el tiempo que la señal tarda en regresar.

Materiales:

1. 1 sensor de ultrasonidos HC-SR04.
2. Placa Arduino

Desarrollo:

Paso 1: Instalar la plataforma de software libre de Arduino.

Paso 2: Agregar las librerías que se van a utilizar en la codificación.

Paso 3: El HC-SR04 tiene cuatro patas. La primera, la de la izquierda, hay que conectarla a 5 V. La cuarta pata, la de la derecha, a 0 V. La segunda pata se llama trigger y la conectaremos al pin 3, que declararemos de salida. Para que el sensor envíe el ultrasonido, debemos poner la pata trigger en alto durante exactamente 10 μ s, esto es, 0,01 ms. La tercera pata se llama echo y la conectaremos al pin 2, que declararemos de entrada.

Paso 4: Una vez que la onda ha regresado al sensor, la pata echo se pondrá en alto exactamente el tiempo que haya durado el proceso de envío y recepción del ultrasonido.

Paso 5: Se muestra la siguiente formula: $(\text{distancia_al_objeto}) = (\text{tiempo_ida_y_vuelta})/2 \cdot (\text{velocidad_sonido})$.

Paso 6: La instrucción `pulseIn(pecho, HIGH)` dará el tiempo en microsegundos que el pin que se le ha llamado pecho está en HIGH; esto es, el `(tiempo_ida_y_vuelta)` en microsegundos. Así, si a la variable `duración` le asignamos el valor `pulseIn(pecho, HIGH)` `duración = pulseIn(pecho, HIGH)`; se tendrá que `duración` es el `(tiempo_ida_y_vuelta)` en microsegundos. La velocidad del sonido es 340 m/s. Se podrá comprobar que es lo mismo que 0.034 cm/ μ s. Así, si a la variable `distancia` le asignamos el valor `duración/2*0.034` `distancia = duración/2*0.034`; Se tendrá que `distancia` es la distancia al objeto en centímetros. Si `distancia` es mayor o igual que 500 o menor o igual que 0, se dice que la medida no es fiable. Por tanto, en estos casos saldrá por el serial “-----“, en vez del valor de distancia.

Paso 7: La codificación para el control del sensor de ultrasonidos es la siguiente

Sketch

```
const int pecho = 2; //la pata echo del CI va al pin 2

const int ptrig = 3; //la pata trigger del CI va al pin 3

int duracion, distancia; //para calcular distancia

void setup() {

Serial.begin(9600); //in

pinMode(pecho, INPUT); //define el pin 2 como entrada (pecho)

pinMode(ptrig, OUTPUT); //define el pin 3 como salida (ptrig)

}

void loop() {

digitalWrite(ptrig, LOW); //al comenzar el loop el trigger debe estar en LOW

delay(10);

digitalWrite(ptrig, HIGH); //genera el pulso de trigger por 0,01 ms

delay(0.01);

digitalWrite(ptrig, LOW);

duracion = pulseIn(pecho, HIGH); //tiempo de ida y vuelta en microsegundos

distancia = (duracion/2)*0.034; //distancia

delay(10); //para darle tiempo

if ((distancia >= 500)||((distancia <= 0)){ //si la distancia es mayor de 5 m

Serial.println("-----

}

}
```

```
else{  
  
Serial.print(distancia);  
  
Serial.println(" cm");  
  
}  
  
delay(100); //para que los datos por el serial no aparezcan tan deprisa  
  
}
```

Preguntas sobre la guía:

- 1- ¿Cómo haría para que cuando la distancia fuese menor o igual a 10 cm también se encendiera un led?
- 2- ¿Qué determina esta porción de código `if ((distancia >= 500)||((distancia <= 0))`?
- 3- ¿Cuál es el elemento electrónico que determina el valor de la distancia?

Conclusión

Recomendaciones:

Ejercicio 9: Control de un servo I (librería Servo.h)

Introducción: Se pretende controlar un servo a través de un potenciómetro, la posición del servo dependerá de la posición del potenciómetro; el potenciómetro ira al pin analógico A0.

Un servo es una especie de motor que puede girar un ángulo de entre 0° y 180° con mucha precisión; de hecho, se le puede asignar exactamente en qué posición debe estar a través del ángulo. Los servos son fundamentales en robótica. Para controlar un servo con Arduino necesitamos cargar la librería Servo.h, esta librería contiene instrucciones relativas a los servos. De un servo salen tres terminales: el terminal rojo debe ir a 5 V, el terminal negro debe ir a 0 V y el terminal blanco debe ir a un pin digital de Arduino, por ejemplo el pin 9.

Objetivo: Controlar un servo a través de un potenciómetro.

Conceptos Básicos:

Un servomotor es un tipo especial de motor que permite controlar la posición del eje en un momento dado. Está diseñado para moverse determinada cantidad de grados y luego mantenerse fijo en una posición. (Antony, 2016)

Condensadores electrolíticos es un tipo de condensador que usa un líquido iónico conductor como una de sus placas. Típicamente con más capacidad por unidad de volumen que otros tipos de condensadores, son valiosos en circuitos eléctricos con relativa alta corriente y baja frecuencia.

Potenciómetro Un potenciómetro es un resistor eléctrico con un valor de resistencia variable y generalmente ajustable manualmente. Los potenciómetros utilizan tres terminales y se suelen utilizar en circuitos de poca corriente.

Materiales:

1 servo,

2 condensadores electrolíticos de $100 \mu\text{F}$,

1 potenciómetro de $10 \text{ k}\Omega$.

Desarrollo:

Paso 1: Instalar la plataforma de software libre Arduino.

Paso 2: Agregar las librerías que determinan el uso de variables necesarias en el código.

Paso 3: Utilizará condensadores en paralelo con el servo porque los motores y los servos consumen bastante corriente al arrancar. También se puede poner un servo en paralelo con el potenciómetro. La función de un condensador es que no se produzcan diferencias muy bruscas de corriente, estas pueden dañar los dispositivos. Mucho ojo porque los condensadores electrolíticos tienen polaridad, así que siempre conectaremos la pata larga al positivo y la corta al negativo.

Paso 4: Las librerías se cargan en las primeras líneas del sketch; así, para cargar la librería Servo.h se escribe en la primera línea del sketch:

```
#include <Servo.h>
```

Paso 5: Una vez se haya cargado la librería declarar un objeto tipo Servo al que se le llamará servito. Para ello, en la segunda línea del sketch, se escribe:

```
Servo servito;
```

Paso 6: Declarar la variable entera ángulo, que se utilizará para decirle al servo su posición y declara la variable entera lectura que leerá el valor del pin analógico A0, al que conectará la pata central del potenciómetro.

Paso 7: Declarar que se ha conectado su terminal blanco en el pin 9. Para ello, escribir en el void setup() la sentencia: servito.attach(9); Como se ha dicho, un servo puede tomar una posición de entre 0° y 180°. Si se quiere que tome la posición de 150°, escribiría: servito.write(150); //ordena a servito ponerse en la posición 150 grados delay(20); //para darle tiempo al servo a ponerse en la posición 150 grados.

Paso 8: Hacer girar al servo un ángulo en función de la lectura analógica A0, que depende de la posición del potenciómetro; por tanto, necesitará usar la función map para pasar de la lectura de A0 (entre 0 y 1023) al ángulo que debe girar (entre 0 y 179).

Paso 9: Se muestra el código con la respectiva funcionalidad.

Sketch

```
#include <Servo.h> //Carga la librería para trabajar con servos

Servo servito; //Declaramos un objeto tipo Servo llamado servito

int angulo; //Esta variable almacena el ángulo que debe girar servito

int lectura; //Esta variable a

void setup() {

servito.attach(9); //servito está conectado al pin 9

}

void loop() {

lectura=analogRead(A0);

angulo=map(lectura,0,1023,0,179); //calcula angulo a partir de lectura

servito.write(angulo); //ordena a se

delay(20); //para darle tiempo al servo a ponerse en la posición angulo

}
```

Preguntas sobre la guía

- 1- ¿Cuáles son librerías son indispensables en el desarrollo de esta guía?
- 2- ¿Cuáles son las variables definidas en el código de esta guía?
- 3- ¿Describir brevemente para qué sirve delay?

Conclusiones

Recomendaciones:

Ejercicio 10: Control de un motor de cc (Puente H L298N)

Introducción: Se pretende controlar un motor de cc mediante un pulsador y un potenciómetro. Según se presione o no el pulsador, el motor girara en un sentido o en otro. Con el potenciómetro se le dará más o menos voltaje al motor.

En las prácticas de los servos se han utilizado condensadores en paralelo para que la intensidad no varíe demasiado; así se evita que se pueda dañar el Arduino. Como los servos han funcionado en vacío (no han movido nada pesado), esta solución ha sido suficiente. Si los servos tuvieran que hacer más fuerza o fuesen servos más grandes la solución de los condensadores sería insuficiente. Lo mismo sucede con los motores de cc.

Los motores de cc no pueden ser alimentados directamente por Arduino, necesitan más intensidad de la que Arduino les puede dar. Si se conecta directamente un motor de cc a Arduino correrá el peligro de dañar Arduino e incluso el ordenador.

Para poder trabajar motores con Arduino se utilizará el módulo L298N. Este módulo es alimentado exteriormente por una batería de entre 6 V y 12 V y permite controlar hasta dos motores de cc, de manera que podremos hacerlos girar en ambos sentidos. Utilizando los pines PWM~ de Arduino, podrá incluso controlar la velocidad de los dos motores.

Objetivos: Controlar un motor de corriente continua por medio de un puente h.

Conceptos básicos:

Motor de corriente continua, denominado también motor de corriente directa, motor cc o motor dc. Convierte la energía eléctrica en mecánica, provocando un movimiento rotatorio.

Puente h

El puente H es un circuito electrónico que permite a un motor eléctrico DC girar en ambos sentidos, avanzar y retroceder. (Mecafenix, Ingeniería , 2017)

Potenciómetro

Un potenciómetro es un resistor eléctrico con un valor de resistencia variable y generalmente ajustable manualmente. Los potenciómetros utilizan tres terminales y se suelen utilizar en circuitos de poca corriente.

Pulsador

Interruptor como un dispositivo que cuando se encuentra “cerrado” deja pasar la corriente y cuando está “abierto” impide el paso de la misma, un pulsador no es más que un tipo de interruptor, el cual se mantendrá en posición de “cerrado” tanto tiempo como pulsado lo mantengamos, gracias a la lámina conductora que produce el contacto. (Isaxan, 2013).

Resistencia

Es toda oposición que encuentra la corriente a su paso por un circuito eléctrico cerrado, atenuando o frenando el libre flujo de circulación de las cargas eléctricas o electrones.

Batería: Acumulador eléctrico o simplemente pila, batería o acumulador, es un dispositivo que consiste en una o más celdas electroquímicas que pueden convertir la energía química almacenada en corriente eléctrica. Cada celda consta de un electrodo positivo, o ánodo, un electrodo negativo, o cátodo, y electrolitos que permiten que los iones se muevan entre los electrodos.

Materiales:

1 motor de cc de 6 o 9 V,

1 puente H L298N,

1 potenciómetro de 10 k Ω ,

1 pulsador,

1 resistencia de 10 k Ω (marrón-negro-naranja) y

1 batería de 6 o 9 V.

Desarrollo:

Paso 1: Instalar la plataforma de software libre Arduino con sus respectivas librerías.

Paso 2: Ver el L298N. En la entrada donde pone +12 conectar el positivo de la batería.

Paso 3: La entrada GND se pone a 0 V. Se puede conectar la entrada donde pone +5V a 5 V de Arduino, pero no es obligatorio. En las entradas OUT1 y OUT2 conectar los dos terminales del motor de cc. Las entradas IN1 e IN2 son las entradas de control e irán a dos pines PWM~; por ejemplo, IN1 al pin 10 e IN2 al pin 9. El modulo funciona así.

- Si IN1 está en HIGH, entonces OUT1 se pone al voltaje de la batería. Si IN1 está en LOW, entonces OUT1 se pone a 0 V.

- Si IN2 está en HIGH, entonces OUT2 se pone al voltaje de la batería. Si IN2 está en LOW, entonces OUT2 se pone a 0 V.

Paso 4: De esta manera, si IN1 está en HIGH e IN2 está en LOW, entonces el motor gira en un sentido. Si IN1 está en LOW e IN2 está en HIGH, entonces el motor gira en sentido contrario. Si ambos están en HIGH o ambos están en LOW, entonces el motor no gira. Como hemos conectado IN1 e IN2 a dos pines PWM~, podremos también controlar la velocidad del motor de cc.

Paso 5: Para el resto de la práctica, conectar la pata central del potenciómetro al pin analógico A0. Declararemos una variable entera a la que llamaremos vel que valdrá entre 0 y 255, pues estos son los valores que se pueden escribir en los pines 9 y 10 mediante analogWrite. La lectura del pin A0 va desde 0 hasta 1023; sin embargo, los valores que puede tomar vel van del 0 al 255, por lo que tendrá que escribir: $vel = \text{analogRead}(A0)/4$; El pulsador ira conectado al pin 2, que declarara de entrada, y a la resistencia, de forma que si pulso el pulsador, el pin 2 se pone a 5 V y si no lo pulso, el pin 2 se pone a 0 V.

Paso 6: Tendrá que escribir un condicional de manera que, si el pin 2 está en LOW, entonces en el pin 9 se escriba vel y en el pin 10 se escriba 0 para que el motor gire en un sentido. Por

el contrario, si el pin 2 está en HIGH, entonces en pin 9 se escribirá 0 y en el pin 10 se escribirá vel para que el motor gire en el sentido contrario.

Sketch

```
const int sentido = 2; //el pulsador va al pin de lectura 2

const int derA = 9; //pata 7 del CI al pin PWM~ 9

const int derB = 10; //pata 2 del CI al pin PWM~ 10

int vel; //a mayor valor mayor velocidad (entre 0 y 255)

void setup() {

pinMode(sentido, INPUT);

pinMode(derA, OUTPUT);

pinMode(derB, OUTPUT);

}

void loop() {

vel=analogRead(A0)/4; //vel vale entre 0 (0/4) y 255 (1023/4)

if (digitalRead(sentido)==LOW){

analogWrite(derA,vel);

analogWrite(derB,0);

}

else {

analogWrite(derA,0);

analogWrite(derB,vel);
```

}

}

Preguntas sobre la guía

- 1- Esta línea de código define analogWrite(derA,vel);:
- 2- La sentencia else sirve para:
- 3- La funcionalidad de void setup() { es:

Conclusión

Recomendaciones

Ejercicio 11. Casa domótica (módulo Bluetooth HC-06)

Introducción. En esta guía se pretende controlar dos leds y un servo mediante el móvil por medio de una App que se puede descargar del blog y al módulo Bluetooth HC-06.

Uno de los leds tendrá su ánodo en el pin 13, de manera que solo se controlará que se encienda o se apague.

Objetivo: Controlar dos leds y un servo mediante el móvil gracias a una App que se puede descargar del blog y al módulo Bluetooth HC-06.

Conceptos Básicos

Led es un diodo semiconductor que, cuando recibe tensión, genera luz. Un diodo, a su vez, es una válvula de dos electrodos que permite el paso de la corriente eléctrica en un único sentido. Puede decirse que un led es una fuente lumínica. Cuando se le aplica tensión a alguna de sus dos terminales, la recombinación de sus electrones provoca la liberación de energía en forma de fotones. (Julian, Definición de led, 2018)

Resistencia es toda oposición que encuentra la corriente a su paso por un circuito eléctrico cerrado, atenuando o frenando el libre flujo de circulación de las cargas eléctricas o electrones.

Servo motores un servomotor es un tipo especial de motor que permite controlar la posición del eje en un momento dado. Está diseñado para moverse determinada cantidad de grados y luego mantenerse fijo en una posición. (Antony, 2016)

Condensadores electrolíticos es un tipo de condensador que usa un líquido iónico conductor como una de sus placas. Típicamente con más capacidad por unidad de volumen que otros tipos de condensadores, son valiosos en circuitos eléctricos con relativa alta corriente y baja frecuencia.

Modulo Bluetooth HC-06 Sirve para acceder y comunicar de forma muy sencilla con tus proyectos electrónicos de forma inalámbrica.

Materiales:

2 leds rojos,

1 resistencia de 220 Ω (rojo-rojo-marrón),

1 servo motor,

1 condensador electrolítico de 100 μF y

1 modulo Bluetooth HC-06

Desarrollo:

Paso 1: Instalar la plataforma de software libre Arduino

Paso 2: Agregar las librerías necesarias para la funcionalidad de cada elemento

Paso 3: Se puede conectar el ánodo de un led al pin 13 y el cátodo directamente a GND, pues el pin 13 da menos de 5 V, por lo que no correrá el riesgo de perjudicar al led. Para encender este led se presionará instantáneamente el botón D de la App. Para apagarlo una vez que este encendido, volver a presionar instantáneamente el botón D de la App.

Paso 4: El segundo led tendrá su ánodo en el pin 11, que es PWM~, de manera que además de encenderlo o apagarlo poder variar su intensidad. La intensidad se controlará con los botones flecha arriba \uparrow y flecha abajo \downarrow .

Paso 5: El servomotor ira al pin 7. Presionar instantáneamente el botón flecha izquierda \leftarrow , el motor se pondrá en posición izquierda, esto es, 180° . Si presionamos instantáneamente el botón flecha derecha \rightarrow , el motor se pondrá en posición derecha, esto es, 0° . Presionar instantáneamente el botón logo OIKOS, el motor se pondrá en posición central, esto es, 90° .

Paso 6: Ver cómo funciona el módulo Bluetooth HC-06. Este módulo tiene cuatro patas. La pata RX debe ir al pin TX de Arduino (pin digital 1). La pata TX debe ir al pin RX de Arduino (pin digital 0). Así, estas patas van intercambiadas con el Arduino para poder establecer la conexión. La pata GND va a 0 V y la pata VCC va a 5V.

Paso 7: Es importante cargar el sketch en Arduino antes de conectar las patas RX y TX del módulo HC-06, cargar el sketch con estas patas conectadas dará error. Así, primero se carga el sketch en Arduino y luego conectaremos el módulo HC-06 a Arduino. Si tenemos que modificar el sketch y volverlo a cargar en Arduino, antes de cargarlo deberá desconectar el módulo de Arduino.

Paso 8: Ver cómo funciona la App. Es una App sencillísima. Cada vez que se presiona un botón de la App, esta mandara, vía Bluetooth, un carácter al serial. La equivalencia entre botones de la App y caracteres es la siguiente:

Botón A B C D E F ↑ ← logo → ↓ 0 1 2 3 4 5 6 7 8 9

carácter A B C D E F a b c d e 0 1 2 3 4 5 6 7 8 9

Paso 9: Una vez conectado el módulo HC-06 a Arduino, pero antes de que móvil y módulo estén conectados vía Bluetooth, el led interior que lleva el módulo estará parpadeando. Para conectar el móvil al módulo HC-06 mediante Bluetooth, dentro de la App, presionamos en Bluetooth. Cuando móvil y módulo estén conectados vía Bluetooth, el led del módulo lucirá sin parpadear.

Paso 10: El conjunto debe funcionar de la siguiente manera:

- Inicialmente, el led del pin 13 estará apagado. Si en la App se pulsa D y el led del pin 13 está apagado, entonces dicho led se encenderá. Si en la App se pulsa D y el led del pin 13 está encendido, entonces dicho led se apagará.

- Inicialmente, el led del pin 11 estará apagado. Cada vez que en la App se pulse el botón ↑, el led del pin 11 lucirá un poco más. Cada vez que en la App se pulse el botón ↓ el led del pin 11 lucirá un poco menos.

- Inicialmente, el servo estará en la posición de 90°. Si en la App se pulsa el botón ←, entonces el servo se pondrá en la posición 180°. Si en la App se pulsa el botón →, entonces el servo se pondrá en la posición 0°.

Paso 11: Dar las siguientes pistas para el sketch.

Declarar una variable tipo caracter o entera, que llamaremos lectura. En el void loop() esta variable recogerá el caracter que la App mande al serial mediante el grupo de sentencias:

```
if (Serial.available()>0){ //Si se ha mandado algo al serial...
```

```
lectura=Serial.read(); //lectura leerá lo que se ha mandado
```

```
}
```

```
else{ //Si no se ha mandado algo al serial...
```

```
lectura=' '; //lectura vale el carácter espacio en blanco
```

```
}
```

Paso 12: Declarar una variable de tipo entero, que se llamará brillo, y que inicialmente valdrá 0. En el void loop() hacer, mediante la función analogWrite, que el led de intensidad variable luzca con el valor de brillo. Si en la App se presiona el botón ↑ y el valor de brillo es menor o igual a 225, entonces incrementará el valor de brillo en 25. Si en la App se presiona el botón ↓ y el valor de brillo es mayor o igual a 25, entonces decrementará el valor de brillo en 25. Así los valores que tomará brillo estarán entre 0 y 250 y serán múltiplos de 25.

Paso 13: Declarar una variable de tipo entero, llamar encendido, y que inicialmente valdrá 0. Esta variable hará que si en la App pulse el botón D y encendido vale 0, entonces el led del pin 13 se encenderá y encendido pasará a valer 1. Si en la App pulsamos el botón D y encendido vale 1, entonces el led del pin 13 se apagará y encendido pasará a valer 0.

Paso 14: Declarar un objeto tipo servo, llamar servito. Recordar que antes habrá que cargar la librería Servo.h mediante: #include <Servo.h>. Este objeto ira al pin 7 de Arduino, por lo que en el setup() escribir: servito.attach(7); Recordar inicializar el serial en el setup() mediante: Serial.begin(9600); electrónico que necesita para crear una casa domotizada.

Paso 15: El código queda de la siguiente forma.

Sketch


```

#include <Servo.h> //Carga la librería para trabajar con servos

const int led1=13;

const int led2=11;

Servo servito; //Declaramos un objeto tipo Servo llamado servito

char lectura;

int encendido=0;//el carácter G no se puede conseguir con la App

int brillo=0;

void setup() {

Serial.begin(9600);

servito.attach(7); //servito está conectado al pin 7

pinMode(led1, OUTPUT);

pinMode(led2, OUTPUT);

digitalWrite(led1, LOW);

analogWrite(led2, 0);

servito.write(90);

}

void loop() {

analogWrite(led2,brillo);

if (Serial.available(>0){

lectura=Serial.read();

}

```

```
else{ //mientras no introduzcamos otro caracter, ...

lectura=' '; //lectura vale un espacio en blanco

}

if (lectura=='D'){

if (encendido==0){

digitalWrite(led1,HIGH);

}

if(encendido==1){

digitalWrite(led1,LOW);

}

encendido=1-encendido;

}

if (lectura=='b'){

servito.write(180);

}

if (lectura=='c'){

servito.write(90);

}

if (lectura=='d'){

servito.write(0);

}
```

```
if ((lectura=='a')&&(brillo<=225)){
```

```
brillo=brillo+25;
```

```
}
```

```
if ((lectura=='e')&&(brillo>=25)){
```

```
brillo=brillo-25;
```

```
}
```

```
delay(100);
```

```
}
```

Preguntas sobre la guía

- 1- ¿Cuál es la función de la siguiente línea de código `if ((lectura=='e')&&(brillo>=25))`?
- 2- ¿`servito.attach(7)`; la importancia de esta línea de código es?
- 3- ¿La casa domótica funciona con el módulo de?

Conclusiones

Recomendaciones

Ejercicio 12: Funcionalidad de cuatro pulsadores

Introducción: En esta práctica, se trabaja la parte electrónica que consta de cuatro pulsadores, permitiendo de esta manera controlar aplicaciones que se detallan a continuación: encendido y apagado de los Leds dependiendo del color que este encendido, ejecutara una acción diferente con la programación del lenguaje IDE Arduino. Como paso previo se debe descargar e instalar el software IDE de Arduino y algunas librerías que son necesarias para facilitar una serie de funciones relacionadas entre sí. Referirse al capítulo 5 como guía para instalar el IDE de Arduino.

Objetivo: Crear programas que controlen el encendido y apagado de varios leds mediante pulsadores; además hará la función de controlar motores, sensores de movimiento, sensores de temperatura, así como el buzzer que es un pequeño dispositivo electrónico que puede emitir sonidos para alertar de algún evento.

Conceptos básicos

Arduino mega es probablemente el microcontrolador más capaz de la familia Arduino. Posee 54 pines digitales que funcionan como entrada/salida; 16 entradas análogas, un cristal oscilador de 16 MHz, una conexión USB, un botón de reset y una entrada para la alimentación de la placa.

La comunicación entre la computadora y Arduino se produce a través del Puerto Serie. Posee un convertidor usb-serie, por lo que sólo se necesita conectar el dispositivo a la computadora utilizando un cable USB como el que utilizan las impresoras.(García Gonzalez, 2013)

Pulsador, interruptor como un dispositivo que cuando se encuentra “cerrado” deja pasar la corriente y cuando está “abierto” impide el paso de la misma, un pulsador no es más que un tipo de interruptor, el cual se mantendrá en posición de “cerrado” tanto tiempo como pulsado lo mantengamos, gracias a la lámina conductora que produce el contacto. (Isaxan, 2013)

Buzzer es un pequeño transductor capaz de convertir la energía eléctrica en sonido. Para hacerlos funcionar solo basta conectar el positivo con el + y la tierra o negativo con el – de una batería o cualquier fuente de corriente directa

Estos dispositivos se utilizan principalmente para alarmas y controles de sonido con un pequeño margen de frecuencia, como en los electrodomésticos. (Mecafenix, 2018)

Sensor ultrasónico, el funcionamiento es simple, envía una señal ultrasónica inaudible y nos entrega el tiempo que demora en ir y venir hasta el obstáculo más cercano que detecto.(Veloso, 2016)

PIR, los sensores infrarrojos pasivos son ideales para proyectos de detección de dichos movimientos. El sensor HC-SR501, sólo funciona cuando alguien se mueve en la franja que puede barrer su detector. Puede detectar movimiento de 3 hasta 7 metros de distancia. Este sensor de movimiento PIR tiene 3 pines, VCC, OUTPUT y GND, 2 potenciómetros para ajustar la sensibilidad y la demora. El retardo se puede configurar entre 5 y 300 segundos mientras que el potenciómetro de sensibilidad ajusta el rango de detección de aproximadamente 3 metros a 7 metros. (García, 2017)

Puente h es un circuito electrónico que permite a un motor eléctrico DC girar en ambos sentidos, avanzar y retroceder. (Mecafenix, Ingeniería , 2017)

Esp 32 es un SoC (System on Chip) diseñado por la compañía china Espressif y fabricado por TSMC. Integra en un único chip un procesador Tensilica Xtensa de doble núcleo de 32bits a 160Mhz (con posibilidad de hasta 240Mhz), conectividad WiFi y Bluetooth. (Llamas, 2018)

Pantalla LCD+I2C, el controlador de LCD I2C es un dispositivo que nos permite controlar una pantalla a través del bus I2C, usando únicamente dos cables. (Maker, 2018)

Dht 111 es un sensor digital de temperatura y humedad relativa del aire. Utiliza una comunicación digital con Arduino, de forma que no es necesaria la conexión a un pin analógico para realizar las lecturas.

Jumper o saltador es un elemento que permite cerrar el circuito eléctrico del que forma parte dos conexiones.

La función del cable macho-macho es con frecuencia usada en el tablero protoboard haciendo posible la conexión de dos elementos ingresados en dicho tablero.

Se conoce como macho-macho debido al fragmento que sobresale de los extremos del cable. (de anda, 2018)

Servo motor es un tipo especial de motor que permite controlar la posición del eje en un momento dado. Está diseñado para moverse determinada cantidad de grados y luego mantenerse fijo en una posición. (Antony, 2016)

Motor Dagu, se trata de un par de motores DC con caja reductora de DAGU. Estos motores ideales para ser usados en robots con tracción diferencial y ofrecen una instalación y uso muy fáciles.

Estos motores reductores requieren una tensión de 4,5 V con una corriente sin carga de 190mA, poseen una relación de transmisión de 48:1 y una velocidad de rueda de 140 RPM sin carga. (Carlos, 2020)

Led RGB, RGB es una sigla formada por los términos de la lengua inglesa *red* (“rojo”), *green* (“verde”) y *blue* (“azul”). El concepto suele emplearse para referirse a un modelo cromático que consiste en representar distintos colores a partir de la mezcla de estos tres colores primarios. (Julian, 2014)

Led es un diodo semiconductor que, cuando recibe tensión, genera luz. Un diodo, a su vez, es una válvula de dos electrodos que permite el paso de la corriente eléctrica en un único sentido.

Puede decirse que un led es una fuente lumínica. Cuando se le aplica tensión a alguna de sus dos terminales, la recombinación de sus electrones provoca la liberación de energía en forma de fotones. (Julian, Definicion de led, 2018)

Materiales:

Placa Arduino Mega

Pulsador

Buzzer

Sensor Ultrasónico

Sensor PIR

Puente H

ESP 32 Thing

Pantalla LCD+I2C

Pines ESP32

Sensor temperatura

Jumper

Servo motor

Motor Dago

Led RGB

Led Rojo

Led azul

Led amarillo

Led verde

Desarrollo

Este diseño de circuito nos muestra la conexión entre ellos, que ayudará a realizar el impreso de circuitos, los cuales contienen diferentes elementos electrónicos para desempeñar variedad de funciones.

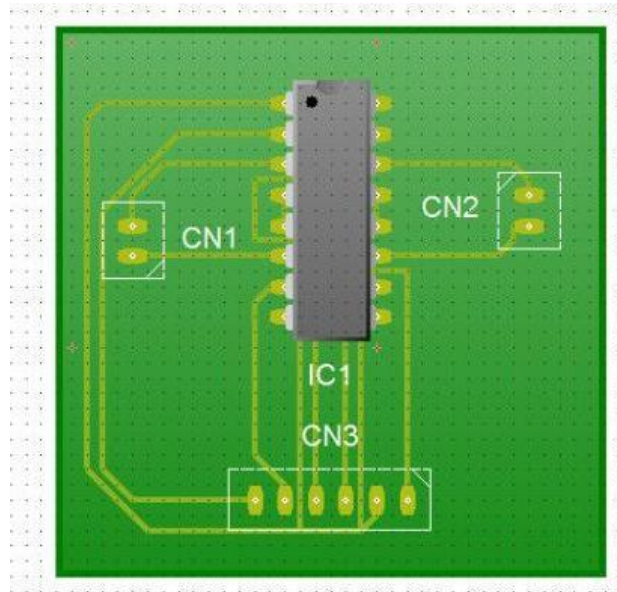


Figura 34. Motores

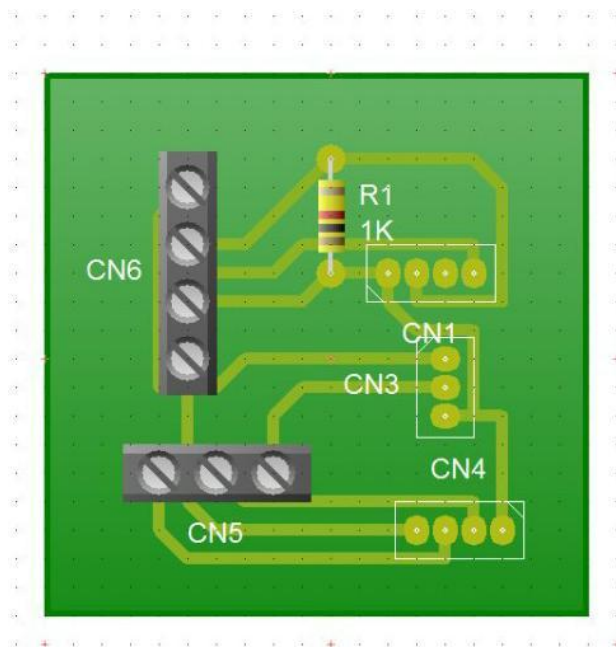


Figura 35. Ultrasónico PIR

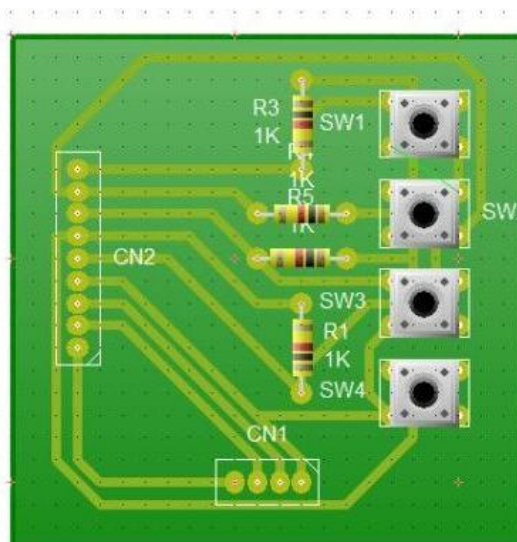


Figura 36. RGB

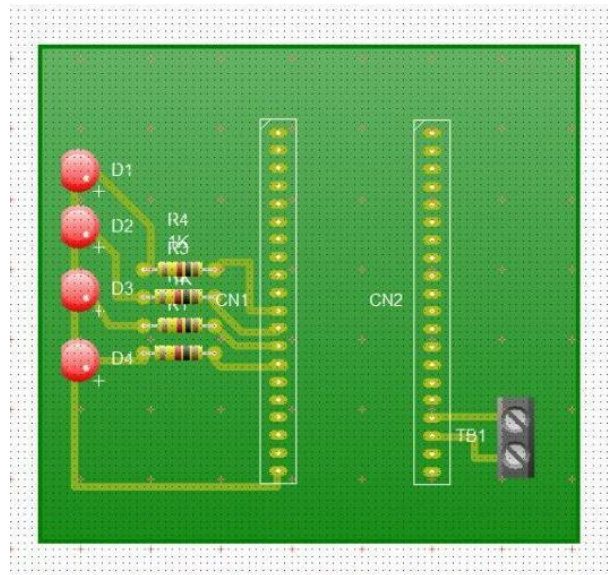


Figura 37. Esp 32

Paso 1:

Lo primero que se tiene que hacer es instalar el software de Arduino en la computadora, se puede descargar entrando a este link [Arduino Software](#) . Luego se le da clic en Windows

installer (en este caso se usará Windows) también el instalador porque trae algunas ventajas extra.

Paso 2:

En el IDE Arduino se tienen que incluir las siguientes librerías: LiquidCrystal, servo, dht

Paso 3: Digitar las siguientes líneas de código las cuales contienen sus respectivos comentarios con respecto a su funcionalidad.

```
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h>
```

```
// Incluimos la librería para poder controlar el servo
```

```
#include <Servo.h>
```

```
// Incluimos librería
```

```
#include <DHT.h>
```

```
// Definimos el pin digital donde se conecta el sensor
```

```
#define DHTPIN 9
```

```
// Dependiendo del tipo de sensor
```

```
#define DHTTYPE DHT11
```

```
// Inicializamos el sensor DHT11
```

```

DHT dht(DHTPIN, DHTTYPE);

// Declaramos la variable para controlar el servo

Servo servoMotor;

const int Trigger = 2; //Pin digital 2 para el Trigger del sensor

const int Echo = 3; //Pin digital 3 para el Echo del sensor

const int pinBuzzer = 11;

int GND = 12;

int B = 53;

int G = 51;

int R = 49;

int Boton1 = 47;

int Boton2 = 45;

int Boton3 = 43;

int Boton4 = 41;

int D1 = 7;

int I1 = 8;

int D2 = 5;

int I2 = 6;

int PIR = 4;

//Crear el objeto lcd dirección 0x3F y 16 columnas x 2 filas

LiquidCrystal_I2C lcd(0x27, 16, 2); //

```

```
void setup() {  
  
  pinMode(PIR, INPUT);  
  
  pinMode(I1, OUTPUT);  
  
  pinMode(D1, OUTPUT);  
  
  pinMode(I2, OUTPUT);  
  
  pinMode(D2, OUTPUT);  
  
  pinMode(GND, OUTPUT);  
  
  digitalWrite(GND, 0);  
  
  // Comenzamos el sensor DHT  
  
  dht.begin();  
  
  servoMotor.attach(10);  
  
  pinMode(Trigger, OUTPUT); //pin como salida  
  
  pinMode(Echo, INPUT); //pin como entrada  
  
  digitalWrite(Trigger, LOW); //Inicializamos el pin con 0  
  
  pinMode (R, OUTPUT);  
  
  pinMode (G, OUTPUT);  
  
  pinMode (B, OUTPUT);  
  
  // Inicializar el LCD  
  
  lcd.init();  
  
  //Encender la luz de fondo.  
  
  lcd.backlight();  
}
```

```

// Escribimos el Mensaje en el LCD.

lcd.setCursor(2, 0);

lcd.print("Cargando el");

lcd.setCursor(4, 1);

lcd.print("programa");

delay(5000);

lcd.clear();

Serial.begin(9600);

pinMode(Boton1, INPUT);

pinMode(Boton2, INPUT);

pinMode(Boton3, INPUT);

pinMode(Boton4, INPUT);

digitalWrite(R, HIGH);

digitalWrite(G, HIGH);

digitalWrite(B, HIGH);

}

void loop() {

lcd.setCursor(0, 0);

lcd.print("Esperando...");

int Lectura1 = digitalRead(Boton1);

int Lectura2 = digitalRead(Boton2);

```

```

int Lectura3 = digitalRead(Boton3);

int Lectura4 = digitalRead(Boton4);

int LecturaPIR = digitalRead(PIR);

if (LecturaPIR == HIGH)

{

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("Hay alguien?");

    delay(6000);

    lcd.clear();

}

if (Lectura4 == HIGH)

{

    digitalWrite(R, LOW);

    digitalWrite(G, HIGH);

    digitalWrite(B, LOW);

    delay(1000);

    long t; //tiempo que demora en llegar el eco

    long d; //distancia en centímetros

    digitalWrite(Triple, HIGH);

    delayMicroseconds(10); //Enviamos un pulso de 10us

```

```
digitalWrite(Triquer, LOW);

t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso

d = t / 59;      //escalamos el tiempo a una distancia en cm

Serial.print("Distancia: ");

Serial.print(d);    //Enviamos serialmente el valor de la distancia

Serial.print("cm");

Serial.println();

delay(100);        //Hacemos una pausa de 100ms

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Distancia:");

lcd.setCursor(0, 1);

lcd.print(d);

lcd.setCursor(3, 1);

lcd.print("cm");

delay(2000);

lcd.clear();

digitalWrite(I1, HIGH);

digitalWrite(D1, LOW);

delay(3000);

digitalWrite(I1, LOW);
```

```

    digitalWrite(D1, LOW);
}

if (Lectura2 == HIGH)
{
    digitalWrite(B, LOW);

    digitalWrite(G, LOW);

    digitalWrite(R, HIGH);

    delay(5000);

    // Leemos la humedad relativa

    float h = dht.readHumidity();

    // Leemos la temperatura en grados centígrados (por defecto)

    float t = dht.readTemperature();

    // Leemos la temperatura en grados Fahrenheit

    float f = dht.readTemperature(true);

    // Comprobamos si ha habido algún error en la lectura

    if (isnan(h) || isnan(t) || isnan(f)) {

        Serial.println("Error obteniendo los datos del sensor DHT11");

        return;

    }

    // Calcular el índice de calor en Fahrenheit

```



```
float hif = dht.computeHeatIndex(f, h);

// Calcular el índice de calor en grados centígrados

float hic = dht.computeHeatIndex(t, h, false);

Serial.print("Humedad: ");

Serial.print(h);

Serial.print(" %\t");

Serial.print("Temperatura: ");

Serial.print(t);

Serial.print(" *C ");

Serial.print(f);

Serial.print(" *F\t");

Serial.print("Índice de calor: ");

Serial.print(hic);

Serial.print(" *C ");

Serial.print(hif);

Serial.println(" *F");

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Humedad:");

lcd.setCursor(9, 0);
```

```
lcd.print(h);

lcd.setCursor(14, 0);

lcd.print("%");

lcd.setCursor(0, 1);

lcd.print("Temp C:");

lcd.setCursor(9, 1);

lcd.print(t);

delay(3000);

lcd.clear();

digitalWrite(I2, HIGH);

digitalWrite(D2, LOW);

delay(3000);

digitalWrite(I2, LOW);

digitalWrite(D2, LOW);

}

if (Lectura3 == HIGH)

{

digitalWrite(G, LOW);

digitalWrite(R, LOW);

digitalWrite(B, LOW);

delay(1000);
```

```

// Desplazamos a la posición 0°

servoMotor.write(0);

// Esperamos 1 segundo

delay(1000);

// Desplazamos a la posición 90°

servoMotor.write(90);

// Esperamos 1 segundo

delay(1000);

// Desplazamos a la posición 180°

servoMotor.write(180);

// Esperamos 1 segundo

delay(1000);

}

if (Lectura1 == HIGH)

{

digitalWrite(G, LOW);

digitalWrite(R, LOW);

digitalWrite(B, HIGH);

delay(1000);

//generar tono de 440Hz durante 1000 ms

tone(pinBuzzer, 440);

```

```
delay(1000);

//detener tono durante 500ms

noTone(pinBuzzer);

delay(500);

//generar tono de 523Hz durante 500ms, y detenerlo durante 500ms.

tone(pinBuzzer, 523, 300);

delay(500);

}

}
```

Preguntas sobre la guía

- 1- ¿Qué función desempeña la librería DHT?
- 2- ¿Qué pasaría si no se incluyen las librerías ya definidas en el código?
- 3- ¿Es necesario tomar en cuenta el circuito montado para desarrollar la programación?

Conclusión

Recomendación

4.5 Ejercicios con NodeMCU

Ejercicio 1: Control de 4 leds mediante una aplicación móvil

Introducción: En esta práctica se relacionara IOT y electrónica para poder controlar 4 leds a través de comandos específicos con la programación del lenguaje IDE Arduino. Como paso previo se debe descargar la aplicación Telegram e instalar el software IDE de Arduino. Referirse al capítulo 5 como guía para instalar el IDE de Arduino.

Objetivos: Controlar 4 leds mediante comandos a través de una aplicación móvil y previamente programación JavaScript para la ejecución de dichos comandos.

Conceptos básicos:

Esp32, es creado por Espressif Systems, ESP32 es un sistema de bajo consumo y bajo costo en un chip SoC (System On Chip) con WiFi y modo dual con Bluetooth En el fondo, hay un microprocesador Tensilica Xtensa LX6 de doble núcleo o de un solo núcleo con un frecuencia de reloj de hasta 240MHz. ESP32 está altamente integrado con switch de antena , balun para RF, amplificador de potencia, amplificador de recepción con bajo nivel de ruido, filtros y módulos de administración de energía, totalmente integrados dentro del mismo chip!!.

Diseñado para dispositivos móviles; tanto en las aplicaciones de electrónica, y las de IoT (Internet de las cosas), ESP32 logran un consumo de energía ultra bajo a través de funciones de ahorro de energía

Led son componentes electrónicos que permiten el paso de la corriente en un solo sentido, en sentido contrario no dejan pasar la corriente. En el sentido en que su conexión permite pasar la corriente se comporta como un interruptor cerrado y en el sentido contrario de conexión, como un interruptor abierto.

Telegram es una plataforma de mensajería y VOIP desarrollada por los hermanos Nikolái y Pável Dúrov. La aplicación, anunciada oficialmente el 14 de agosto de 2013, está enfocada en la mensajería instantánea, el envío de varios archivos y la

comunicación en masa. El servicio lo administra una organización autofinanciada cuya sede principal opera en Dubái, Emiratos Árabes.

El servicio ofrece funcionalidades enfocadas en realizar charlas entre usuarios. Salvo excepciones, los mensajes son almacenados, o archivados en caso que desee ocultarlos, en la nube con opción de reenvío, borrado, acciones de búsqueda, realizar llamadas y adjuntos varios.

Materiales:

Placa ESP 32

4 leds (amarillo, rojo, verde, azul)

Telegram

Desarrollo:

Paso 1.

Se inicia sesión en el siguiente link shiftr.io con una cuenta de Gmail

Usuario: arduinoprogramable@gmail.com

Contraseña: soloesunaprueba

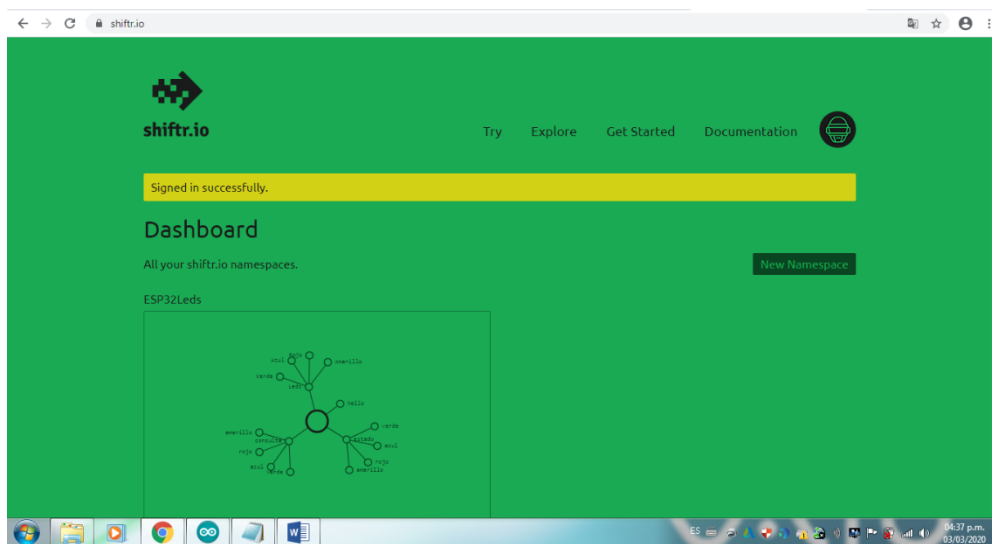


Figura 38. Vista de sitio web shiftr.io

Paso 2.

- Instalar node js en el siguiente link <https://nodejs.org/es/download/>
- Seleccionar sistema operativo de acuerdo al que esté instalado en su ordenador

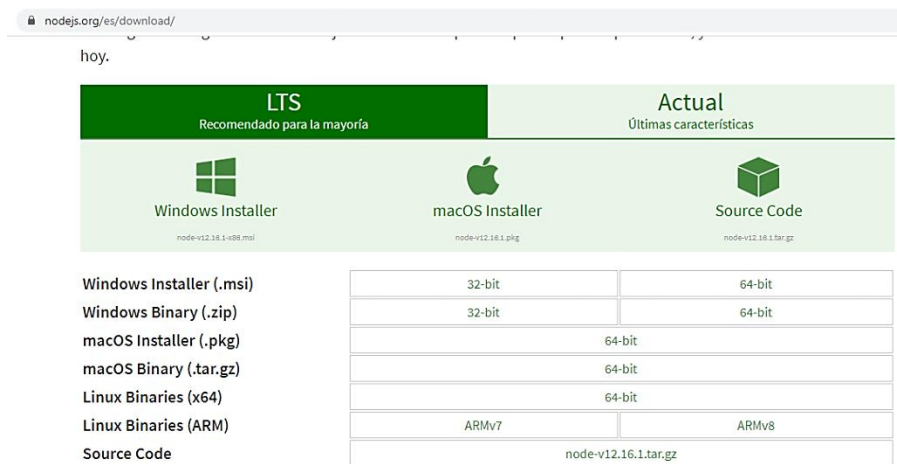
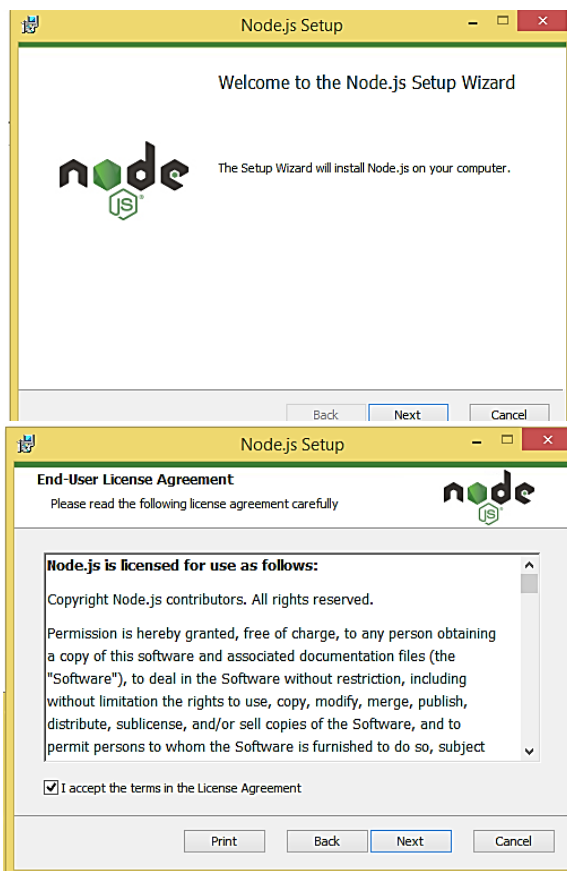


Figura 39. Vista de sitio web nodejs.org

- Dar clic en el instalador y seleccionar next



instalación de

Figura 40. Ventana de node

Figura 41. Ventana de términos de licencia

- Aceptar términos de licencia y dar clic en Next y se iniciará la instalación

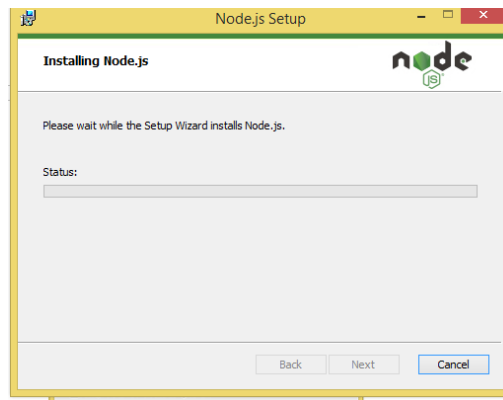


Figura 42. Ventana de progreso de instalación

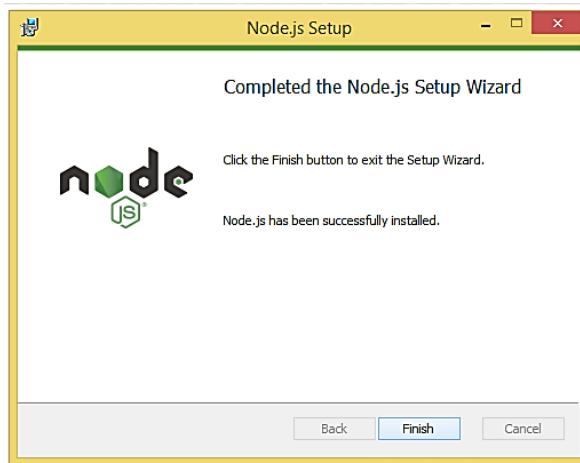



Figura 43. Ventana de finalización de instalación de node

- Dar clic en Finish

Paso 3.

- Abrir cmd (tecla Windows, luego digitar cmd y presionar tecla enter)
- Verificar en cmd si está instalado Node js de la siguiente manera:
- Clic en tecla Windows 
- Luego digitar cmd y presionar tecla enter

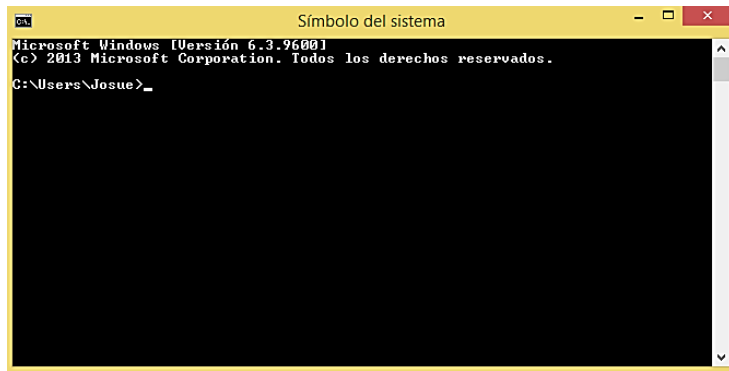


Figura 44. Ventana de cmd

Luego cmd en Windows digitar la siguiente dirección

- Node -v para verificar versión
- Luego dar clic en tecla Windows
- Y digitar Node

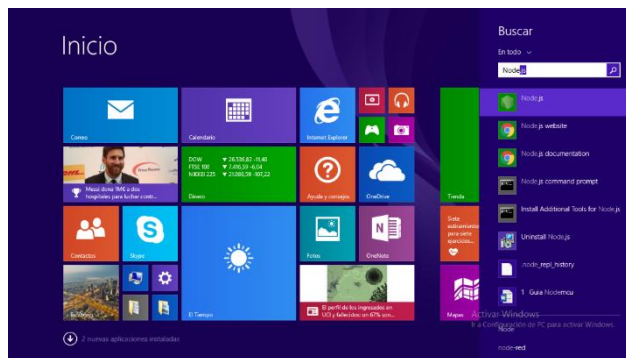


Figura 45. Pantalla de inicio

- Presionar tecla enter

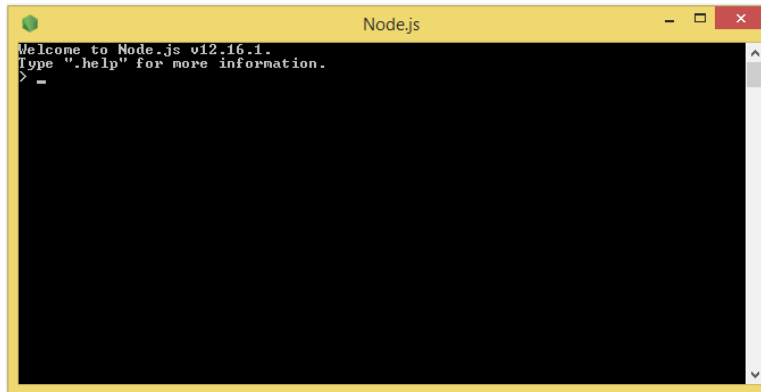


Figura 46. Ventana de node

- Luego en cmd digitar lo siguiente cd Downloads(que es ubicación de archivo llamado bot.txt.js) y presionar enter, al realizar esto se conectara a servidor para enviar órdenes a través de aplicación móvil Telegram

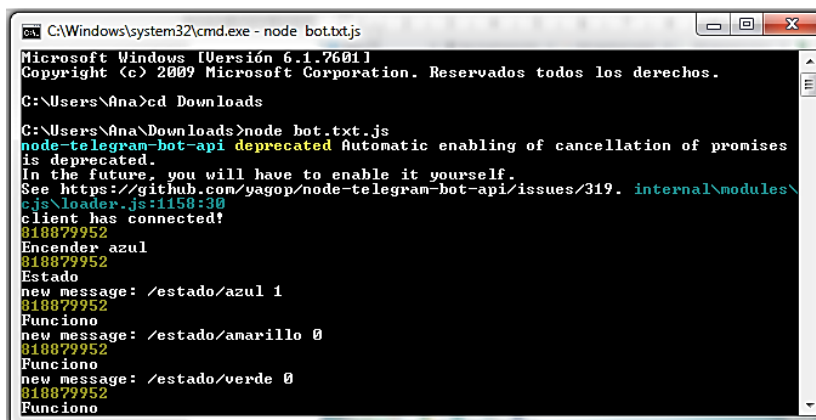


Figura 47. Ventana de cmd con visualización de órdenes enviadas al esp32

- Abrir Telegram en el celular luego enviar órdenes para encendido y apagado o estados de luces



Figura 48. Pantalla de aplicación Telegram

Paso 4. Abrir el IDE Arduino y digitar el siguiente código, es importante mencionar el importe de librería esp y mqtt de la siguiente manera:

- Instalar librería esp 32
- Herramientas >Placa>Gestor de tarjetas

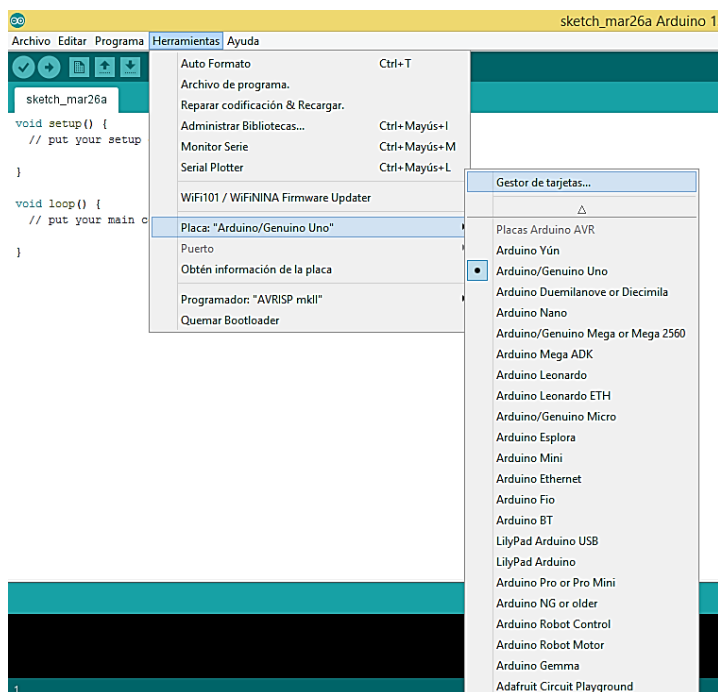
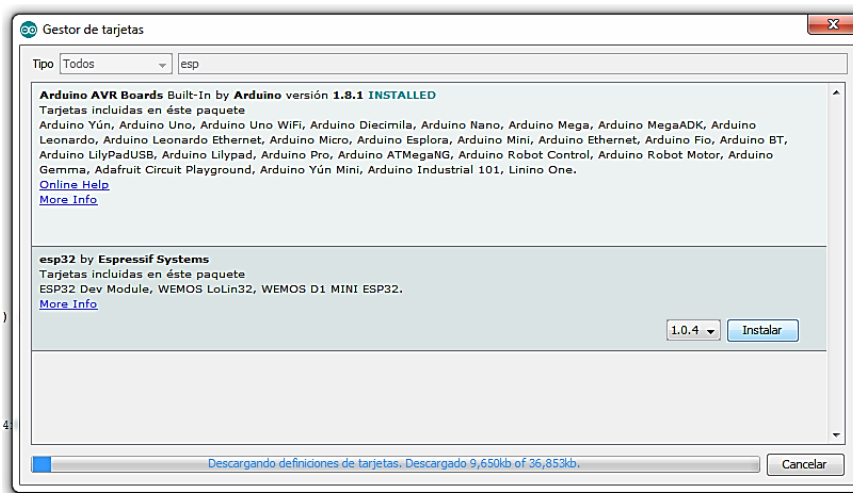


Figura 49. IDE arduino con opción para importar librería

Al seleccionar esa opción se desplegará una ventana así:

este



- Y buscamos esp en caso será 32 porque no viene incluido y

presionamos enter.

Figura 50. Ventana de Gestor de tarjetas

- Luego en la misma ventana digitar mqtt para importar librería

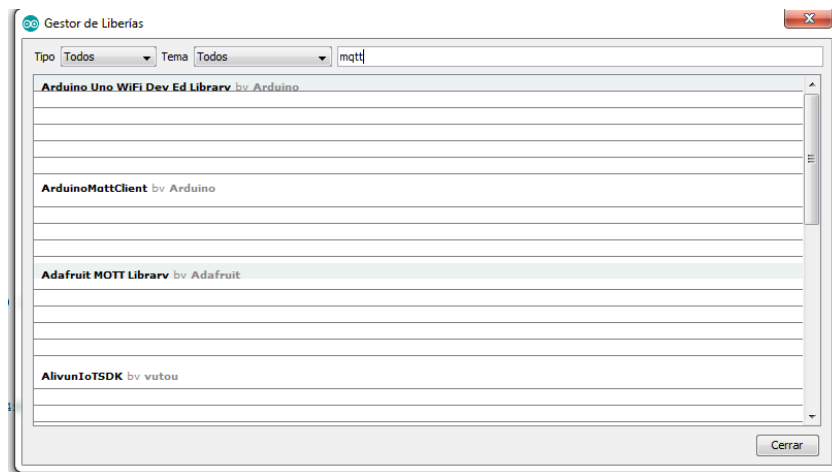


Figura 51. Búsqueda de librería mqtt

Paso

```
sketch_mar26a Arduino 1.8.10
Archivo Editar Programa Herramientas Ayuda
sketch_mar26a $
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
}
```

5. Digitar el siguiente código en el IDE Arduino

Figura 52. IDE Arduino

Código:

```
#include <WiFi.h>
#include <MQTT.h>
const char ssid[] = "ALSW";
const char pass[] = "25264897";
WiFiClient net;
MQTTClient client;
int A = 12;
int R = 13;
int V = 14;
int Az = 27;
int azul = 0;
int rojo = 0;
int amarillo = 0;
int verde = 0;
void connect() {
    Serial.print("checking wifi...");
```

```

while (WiFi.status() != WL_CONNECTED) {

  Serial.print(".");

  delay(1000);

}

Serial.print("\nconnecting...");

while (!client.connect("Esp32", "f064f4e3", "c4eeb2f5e4d50ed2")) {

  Serial.print(".");

  delay(1000);

}

Serial.println("\nconnected!");

client.subscribe("/Leds/Verde");

client.subscribe("/Leds/Amarillo");

client.subscribe("/Leds/Rojo");

client.subscribe("/Leds/Azul");

client.subscribe("/consulta/azul");

client.subscribe("/consulta/amarillo");

client.subscribe("/consulta/rojo");

client.subscribe("/consulta/verde");

// client.unsubscribe("/hello");

}

```

```
void messageReceived(String &topic, String &payload) {  
  
    Serial.println("incoming: " + topic + " - " + payload);  
  
    if (topic == "/Leds/Amarillo")  
  
    {  
  
        if (payload == "1") {  
  
            digitalWrite(A, HIGH);  
  
            amarillo = 1;  
  
        }  
  
        if (payload == "0") {  
  
            digitalWrite(A, LOW);  
  
            amarillo = 0;  
  
        }  
  
    }  
  
    if (topic == "/Leds/Rojo")  
  
    {  
  
        if (payload == "1") {  
  
            digitalWrite(R, HIGH);  
  
            rojo = 1;  
  
        }  
  
        if (payload == "0") {  
  
            digitalWrite(R, LOW);
```



```
    rojo = 0;
}
}
if (topic == "/Leds/Verde")
{
    if (payload == "1") {
        digitalWrite(V, HIGH);
        verde = 1;
    }
    if (payload == "0") {
        digitalWrite(V, LOW);
        verde = 0;
    }
}
if (topic == "/Leds/Azul")
{
    if (payload == "1") {
        digitalWrite(Az, HIGH);
        azul = 1;
    }
    if (payload == "0") {
```

```
digitalWrite(Az, LOW);

azul = 0;

}

}

if (topic == "/consulta/azul") {

    Serial.print("recibi el mensaje");

    if (azul == 0)

    {

        client.publish("/estado/azul", "0");

    }

    else {

        client.publish("/estado/azul", "1");

    }

}

if (topic == "/consulta/rojo") {

    Serial.print("recibí el mensaje");

    if (rojo == 0)

    {

        client.publish("/estado/rojo", "0");

    }

    else {
```

```
    client.publish("/estado/rojo", "1");  
  
    }  
  
    }  
  
    if (topic == "/consulta/amarillo") {  
  
        Serial.print("recibi el mensaje");  
  
        if (amarillo == 0)  
  
        {  
  
            client.publish("/estado/amarillo", "0");  
  
        }  
  
        else {  
  
            client.publish("/estado/amarillo", "1");  
  
        }  
  
    }  
  
    if (topic == "/consulta/verde") {  
  
        Serial.print("recibi el mensaje");  
  
        if (verde == 0)  
  
        {  
  
            client.publish("/estado/verde", "0");  
  
        }  
  
        else {  
  
            client.publish("/estado/verde", "1");  
  
        }  
  
    }  
  
}
```

```

    }

}

}

void setup() {

  Serial.begin(115200);

  WiFi.begin(ssid, pass);

  // Note: Local domain names (e.g. "Computer.local" on OSX) are not supported by Arduino.

  // You need to set the IP address directly.

  client.begin("broker.shiftr.io", net);

  client.onMessage(messageReceived);

  connect();

  pinMode(A, OUTPUT);

  pinMode(R, OUTPUT);

  pinMode(V, OUTPUT);

  pinMode(Az, OUTPUT);

}

void loop() {

  client.loop();

  delay(10); // <- fixes some issues with WiFi stability

  if (!client.connected()) {

    connect();
  }
}

```

}

}

Preguntas.

1. ¿Por qué es necesario la instalación del software Nodejs?
2. ¿Es importante la importación de la librería mqtt?
3. Que función desempeña esta línea `Serial.begin`

Conclusiones:

Recomendaciones:

Ejercicio 2. Parpadeo de un led/BLINK

Introducción: Hacer parpadear un LED con la programación del lenguaje IDE Arduino. Como paso previo se debe descargar e instalar el software IDE de Arduino. Referirse al capítulo 5 como guía para instalar el IDE de Arduino.

Objetivo: Crear un programa que controle el parpadeo de un led que pueda variar el tiempo de estado alto y el tiempo de estado bajo del mismo.

Para esta práctica se utilizará la instrucción `delay(ms)`, esta instrucción permite apagar y encender un tiempo seleccionado en milisegundos.

Conceptos básicos

Antes de iniciar la construcción del circuito y realizar su programación es importante recordar unos pequeños conceptos que nos permitirán evitar errores en un futuro:

Cables Dupont: Un jumper o saltador es un elemento que permite cerrar el circuito eléctrico del que forma parte dos conexiones.

La función del cable macho-macho es con frecuencia usada en el tablero protoboard haciendo posible la conexión de dos elementos ingresados en dicho tablero.

Se conoce como macho-macho debido al fragmento que sobresale de los extremos del cable. (De anda, 2018)

Una Protoboard es un instrumento que permite probar el diseño de un circuito sin la necesidad de soldar o desoldar componentes. Las conexiones en una Protoboard se hacen con solo insertar los componentes lo que permite armar y modificar circuitos con mayor velocidad.

Es importante saber que los procesadores ESP trabajan a 3,3 voltios, tanto las entradas como las salidas digitales. Siempre debemos comprobar que al conectar un componente a este voltaje funcionará correctamente (se deberá verificar que el voltaje es suficiente para que el componente funcione o que el voltaje no excesivo para quemarlo).

Los diodos LED (light emitting diode), como cualquier diodo, tienen polaridad. Esto significa que la corriente eléctrica circula desde el ánodo (+) hacia el cátodo (-) y no en sentido contrario, por lo que debemos estar atentos a la hora de conectarlos.

El LED tiene un voltaje de funcionamiento aproximado entre 1,8 y 2,1 voltios. La intensidad de funcionamiento que debe circular por él está comprendida entre 10 y 20 miliamperios (mA). Estos valores son genéricos, ya que varían en función del modelo, fabricante, color...

Si conectáramos directamente el LED en el circuito, sin una resistencia que disipe el exceso de energía que suministra la placa, se quemaría y dejaría de funcionar en poco tiempo.

Los pines que sirven para suministrar corriente para el encendido del LED, son todos los pines GPIO (General Purpose Input/Output). Esto será independiente del modelo de placa que utilicemos (NodeMCU, Mini-NodeMCU, Wemos D1, Wemos D1 R2...).

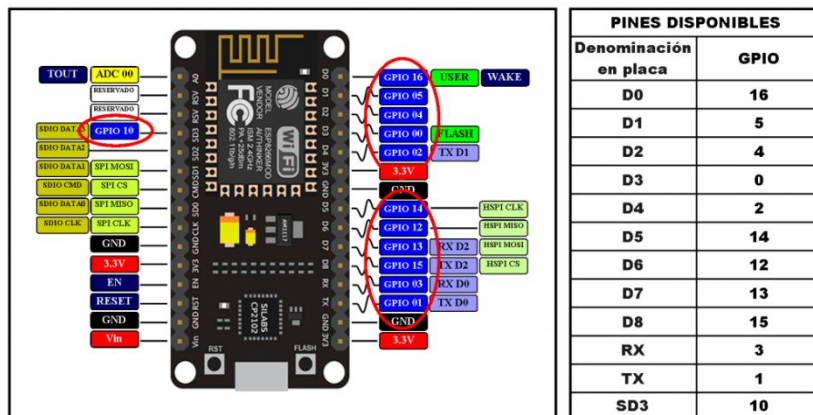


Figura 53. Placa NodeMCU

Materiales:

1. ESP32
2. Breadboard
3. Cables dupont(macho macho 10 cm)
4. Led
5. Resistencia de 100 ohmios

Desarrollo

Paso 1: Instalación del IDE de Arduino

Lo primero que se tiene que hacer es instalar el software de Arduino en la computadora, se puede descargar entrando a este link [Arduino Software](#) . Luego se le da clic en Windows installer (en este caso se usará Windows) también el instalador porque trae algunas ventajas extra.

Paso 2. El circuito que se debe conectar en una placa modelo NodeMCU es el siguiente (en otros modelos de placas será igual, únicamente se tendrá que elegir la salida adecuada):

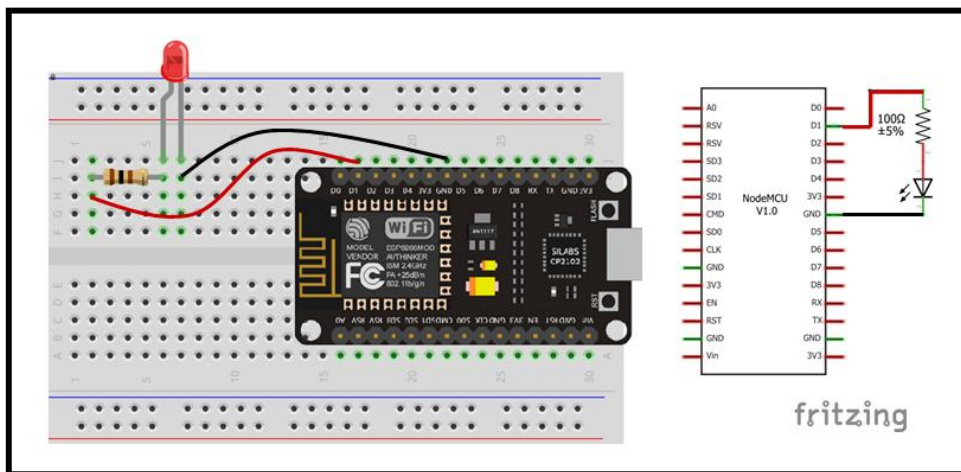


Figura 54. Diagrama del circuito con conexiones

Paso 3. Abrir el IDE de Arduino para comenzar a programar. A continuación, abrir el programa Arduino con el acceso directo creado en el escritorio.

Para poder empezar primero se debe entender como es la estructura básica de Arduino, que parámetros necesitamos escribir; antes de escribir el código para nuestro proyecto.

Y el código de programación para el control del parpadeo del LED sería:

```
int LED_aparpadear = 5;
```



```

// Definimos la variable y el número del GPIO
// a conectar el LED a parpadear.

// Lo hacemos en el encabezado del programa para que sea una
//variable global, accesible a todo el programa

void setup() { // La función “setup” inicializa las configuraciones de
// los diferentes elementos del programa.

// Únicamente se ejecuta una vez.

pinMode(LED_aparpadear, OUTPUT);

// Inicializa el LED_a parpadear como una salida.

}

void loop() {

// La función “loop” se ejecuta una y otra vez, indefinidamente.

digitalWrite(LED_aparpadear, HIGH); // Enciende el LED.

delay(1000); // Espera un segundo.

digitalWrite(LED_aparpadear, LOW); // Apaga el LED.

delay(2000); // Espera dos segundos.

}

```

Paso 4. Conectamos la placa NodeMCU con el ordenador a través del cable USB.

Pulsamos SUBIR, en el IDE de Arduino y este nos preguntará el nombre que queremos asignarle al fichero. En este caso se ha nombrado Sketch001_Parpadeo_LED, pero podría ser

otro cualquiera. Una vez asignado el nombre, el IDE compilará el programa. A continuación lo subirá a la placa y el LED que hemos conectado en la protoboard comenzará a parpadear.

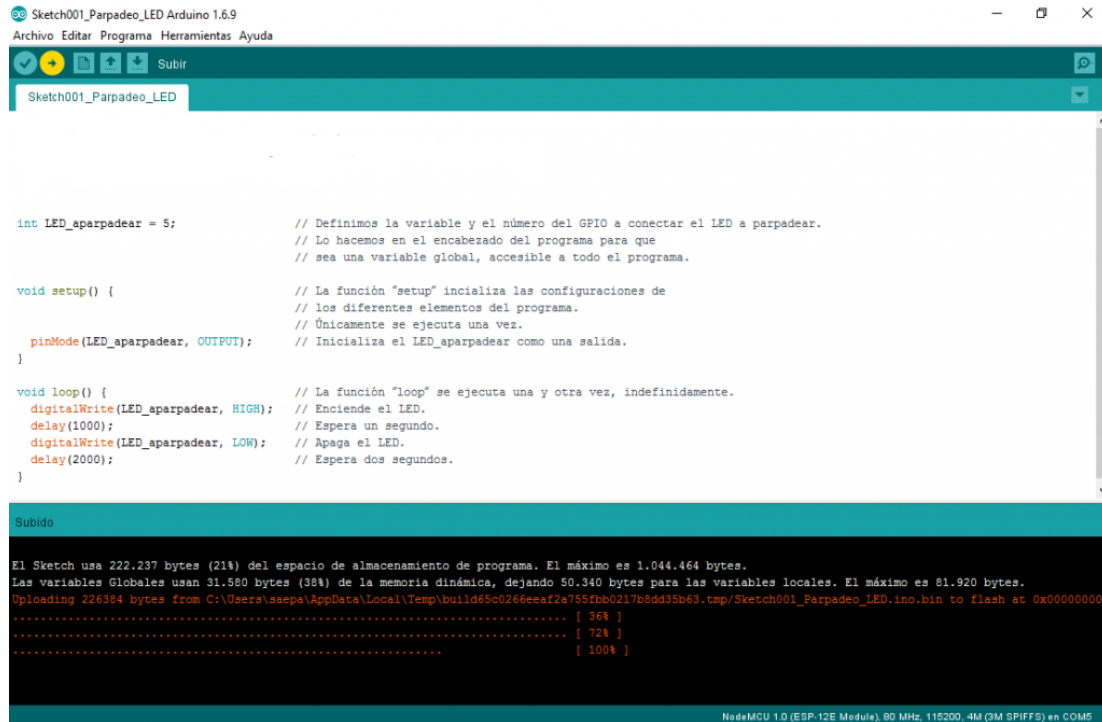


Figura 55. Ejecución del programa de parpadeo de led

Pregunta

S:

- ¿Qué ocurre si se modifica la función delay?
- ¿Qué función desempeña el pin 5?
- ¿Qué ocurriría si conectáramos directamente el LED en el circuito, sin una resistencia?

Conclusión:

Recomendaciones:

Ejercicio 3.- Servidor web modo access point con procesador ESP- Encendido y apagado de un led

Introducción: En esta práctica se trabajará la parte electrónica y web de manera que se utilizará un servidor para controlar el encendido y apagado de un LED a través los dispositivos clientes que se conecten, como PC's, Smart Phones, Tablets, etc.

Objetivo: Crear un servidor WiFi de área local (Wireless Local Area Network – WLAN), configurándolo como un punto de acceso por software (Software Enabled Access Point – SoftAP o simplemente AP). Esta es una de las tres formas de conexión WiFi de que dispone el procesador ESP8266 instalado en el NodeMCU

Conceptos básicos

En el modo Access Point el NodeMCU difunde un SSID (Service Set Identifier), es decir, el “nombre de red” que se visualiza desde los dispositivos WiFi clientes (salvo que lo ocultemos, por supuesto). La conexión se realiza cuando el NodeMCU autoriza las peticiones de conexión de los clientes.

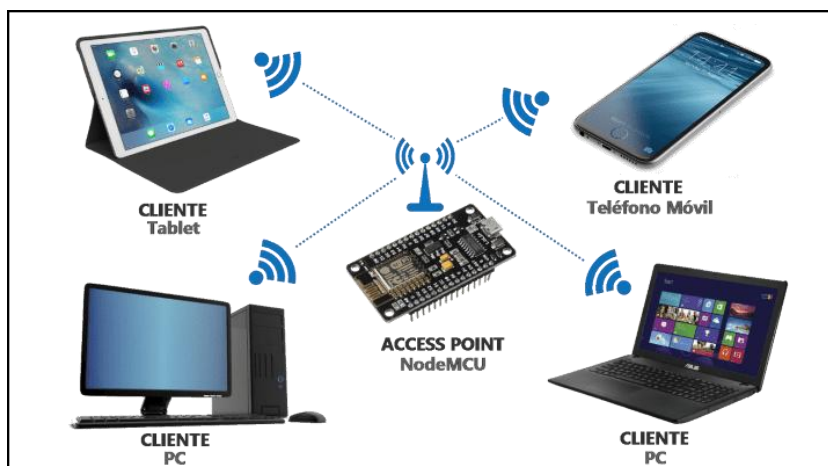


Figura 56. Esquema de funcionalidad de placa NodeMCU

Por defecto la dirección IP del NodeMCU, que nos sirve para conectar con él desde un navegador Web, es la 192.168.4.1 y asigna a los clientes las siguientes direcciones IP disponibles, 192.168.4.2, 192.168.4.3, etc.

Vamos a hacer un repaso de las funciones más importantes de la librería ESP8266WiFi.h, utilizadas por el ESP8266 para trabajar en modo Access Point (AP):

WiFi.softAP(ssid, password, channel, hidden)

El software que permite a un dispositivo que no ha sido específicamente construido como un router, convertirse en un punto de acceso WiFi (router virtual) se conoce como Software Enabled Access Point, o abreviadamente SoftAP.

Para crear el Access Point con el NodeMCU recurrimos a este software a través de la instrucción WiFi.softAP(ssid, password, channel, hidden). Los cuatro parámetros que la componen se definen a continuación:

ssid: es un parámetro obligatorio, el nombre de red, SSID(Service Set Identifier). Debe tener entre 8 y 63 caracteres.

password: es un parámetro opcional, la clave. Para una conexión WPA2-PSK debe tener al menos 8 caracteres. Si no se especifica, el Access Point estará abierto a los clientes.

channel: es un parámetro opcional, para establecer el canal WiFi de comunicación. Los canales válidos son del 1 al 13, por defecto se utiliza el canal 1.

hidden: es un parámetro opcional, para permitir ver u ocultar el SSID. 1 permite ver la red y 0 la oculta.

Wifi.softAPdisconnect()

La instrucción para finalizar la conexión de los dispositivos conectados al Access Point es Wifi.softAPdisconnect().

WiFi.softAPConfig(local_ip, gateway, subnet)

Como se ha comentado, la red creada por softAP tiene la dirección IP 192.168.4.1. Esta dirección se puede cambiar utilizando la instrucción WiFi.softAPConfig(local_ip, gateway, subnet). Los tres parámetros que la componen se definen a continuación:

local_ip: dirección IP del Access Point.

gateway: es la dirección IP de la pasarela o puerta de enlace (puede coincidir con la local_ip).

subnet: dirección IP de la subred.

WiFi.softAPIP() Devuelve la dirección IP con la que se ha creado la conexión (AccessPoint). Por defecto la dirección IP será: 192.168.4.1.

WiFi.softAPmacAddress(macAddr) Para conocer la dirección MAC del AccessPoint utilizamos la función Wifi.softAPmacAddress(macAddr). El parámetro macAddr es voluntario, y es un array de seis elementos que permite almacenar cada uno de los elementos de la dirección MAC.

WiFi.softAPgetStationNum() Para ver el número de dispositivos conectados (clientes/estaciones), utilizamos la instrucción WiFi.softAPgetStationNum(). Como máximo se pueden conectar cinco dispositivos.

Materiales:

ESP32

Breadboard

Cables dupont (macho 10 cm)

Led

Resistencia de 100 ohmios

Paso 1. Circuito

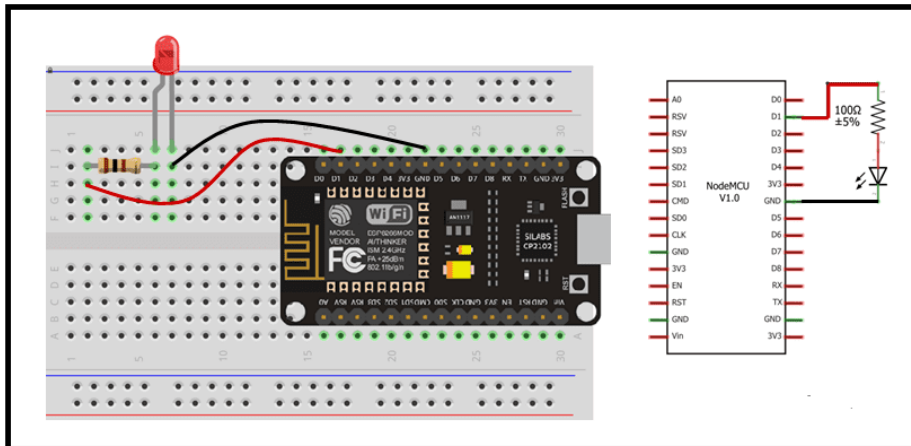


Figura 57. Diagrama de circuito control de led con servidor Access point

El led estará conectado en el puerto GPIO02 (D4).

Este led, como comprobaremos, está relacionado en lógica inversa con el led que tiene el ESP8266 del NodeMCU junto a la antena. Esto significa que cuando un led se enciende, el otro se apaga.

Paso 2. Digitar en un bloc de notas el siguiente código html

Código html

```
<!DOCTYPE HTML>
```

```
<meta charset='UTF-8'>
```

```
<html>
```

```
<!--Se envía el estado del LED en función del estado del puerto GPIO-->
```

```
<h1>El LED está ahora: APAGADO/ENCENDIDO</h1>
```

```
<!--Se crean enlaces para modificar el estado del LED-->
```

```
<p>Presiona <a href='/LED=ON'>AQUÍ</a> para encender el LED</p>
```

```
<p>Presiona <a href='/LED=OFF'>AQUÍ</a> para apagar el LED</p>
```

```
<br>
```

```
<!--Se crean cajas de comprobación (checkbox) para modificar el estado del LED-->
```

```
<input type='checkbox' onClick=location.href='/LED=ON'> ENCENDER </input>
```

```
<br>
```

```
<input type='checkbox' onClick=location.href='/LED=OFF'> APAGAR </input>
```

```
<br><br>
```

```
<!--Se crean botones para modificar el estado del LED-->
```

```
<button type='button' onClick=location.href='/LED=ON'> ENCENDER </button>
```

```
<button type='button' onClick=location.href='/LED=OFF'> APAGAR </button>
```

```
<br><br>
```

```
<!--Se crean botones con estilos para modificar el estado del LED-->
```

```
<button type='button' onClick=location.href='/LED=ON' style='margin:auto; background-color:green; color:#A9F5A9; padding:5px; border:2px solid black; width:200;'><h2>ENCENDER</h2> </button>
```

```
<button type='button' onClick=location.href='/LED=OFF' style='margin:auto; background-color:red; color:#F6D8CE; padding:5px; border:2px solid black; width:200;'><h2>APAGAR</h2> </button>
```

```
<br><br>
```

```
<!--Se crea un único botón para modificar el estado del LED-->
```

```
<!--Se hace la pregunta en función del estado del puerto GPIO-->
```

```
<button type='button' onClick=location.href='/LED=ON/OFF'> ENCENDER/APAGAR </button>
```


Un
página

navegador Web
visualizará la
HTML así:

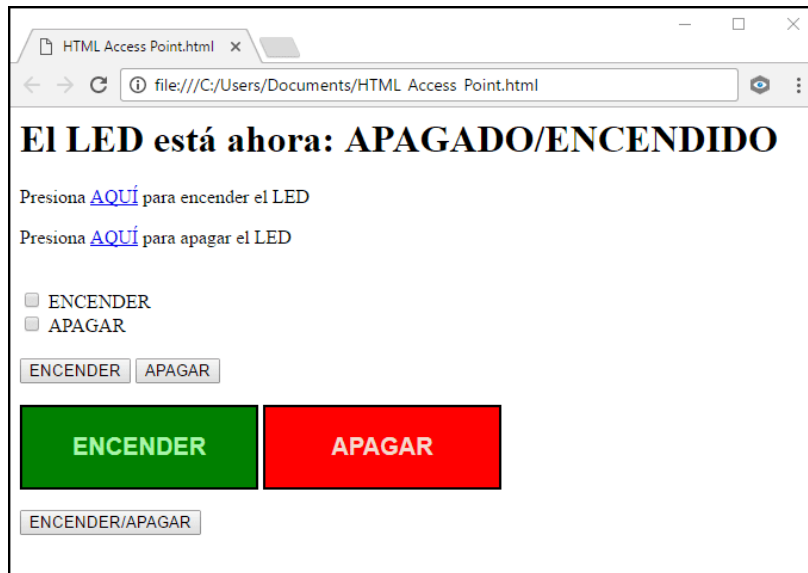


Figura 58. Visualización de página html con funciones

Como se puede observar, se han introducido varios controles/botones para encender o apagar el LED, utilizando diferentes elementos HTML como `<a>`, `<checkbox>` o `<button>`.

Lo que tienen en común todos ellos, en su funcionamiento, es la forma en que se realiza la petición desde el cliente, a través del navegador al servidor (NodeMCU). En la petición, para lograr la comunicación se une la IP (obtenida con la propiedad `ref`) con la cadena `"/LED=ON"` o `"/LED=OFF"` en función de lo que hayamos pulsado. Quedaría así:

`http://192.168.4.1/LED=ON` o `http://192.168.4.1/LED=ON`

De esta manera el NodeMCU, recibe la petición `"/LED=ON"` o `"/LED=OFF"` y la analiza utilizando la función `peticion.indexOf()`, para determinar las acciones a realizar.

Paso 4. Digitar en el IDE Arduino el siguiente código para nodemcu

```
#include <ESP8266WiFi.h> //Incluye la librería ESP8266WiFi
```

```

const char ssid[] = "NodeMCU-ESP8266"; //Definimos la SSID de nuestro servidor WiFi -
nombre de red-

const char password[] = "12345678";

//Definimos la contraseña de nuestro servidor

WiFiServer server(80); //Definimos el puerto de comunicaciones

int PinLED = 2; //Definimos el pin de salida - GPIO2 / D4

int estado = LOW; //Definimos la variable que va a recoger el estado del LED

void setup() {

Serial.begin(115200);

pinMode(PinLED, OUTPUT); //Inicializamos el GPIO2 como salida

digitalWrite(PinLED, LOW); //Dejamos inicialmente el GPIO2 apagado

server.begin(); //inicializamos el servidor

WiFi.mode(WIFI_AP);

WiFi.softAP(ssid, password); //Red con clave, en el canal 1 y visible

//WiFi.softAP(ssid, password,3,1); //Red con clave, en el canal 3 y visible

//WiFi.softAP(ssid); //Red abierta

Serial.println();

Serial.print("Direccion IP Access Point - por defecto: ");

//Imprime la dirección IP

Serial.println(WiFi.softAPIP());

Serial.print("Direccion MAC Access Point: "); //Imprime la dirección MAC

```

```

Serial.println(WiFi.softAPmacAddress());

//IPAddress local_ip(192, 168, 1, 1); //Modifica la dirección IP

//IPAddress gateway(192, 168, 1, 1);

//IPAddress subnet(255, 255, 255, 0);

//WiFi.softAPConfig(local_ip, gateway, subnet);

//Serial.println();

//Serial.print("Access Point - Nueva direccion IP: ");

//Serial.println(WiFi.softAPIP());

}

void loop()

{

// Comprueba si el cliente ha conectado

WiFiClient client = server.available();

if (!client) {

return;

}

// Espera hasta que el cliente envía alguna petición

Serial.println("nuevo cliente");

while(!client.available()){

delay(1);

}

```

```

// Imprime el número de clientes conectados

Serial.printf("Clientes conectados al Access Point: %dn", WiFi.softAPgetStationNum());

// Lee la petición

String peticion = client.readStringUntil('r');

Serial.println(peticion);

client.flush();

// Comprueba la petición

if (peticion.indexOf('/LED=ON') != -1) {

estado = HIGH;

}

if (peticion.indexOf('/LED=OFF') != -1){

estado = LOW;

}

//Enciende o apaga el LED en función de la petición

digitalWrite(PinLED, estado);

// Envía la página HTML de respuesta al cliente

client.println("HTTP/1.1 200 OK");

client.println(""); //No olvidar esta línea de separación

client.println("<!DOCTYPE HTML>");

client.println("<meta charset='UTF-8'>");

```

```

client.println("<html>");

//Imprime el estado del led

client.print("<h1>El LED está ahora: ");

if(estado == HIGH) {

client.print("ENCENDIDO</h1>");

} else {

client.print("APAGADO</h1>");

}

//Se crean enlaces para modificar el estado del LED

client.println("Presiona <a href='/LED=ON'>AQUÍ</a> para encender el LED<br>");

client.println("Presiona <a href='/LED=OFF'>AQUÍ</a> para apagar el LED<br><br>");

//Se crean cajas de comprobación (checkbox) para modificar el estado del LED

client.println("<input type='checkbox' onClick=location.href='/LED=ON'> ENCENDER
</input><br>");

client.println("<input type='checkbox' onClick=location.href='/LED=OFF'> APAGAR
</input><br><br>");

//Se crean botones para modificar el estado del LED

client.println("<button type='button' onClick=location.href='/LED=ON'> ENCENDER
</button>");

client.println("<button type='button' onClick=location.href='/LED=OFF'> APAGAR
</button><br><br>");

```

```
//Se crean botones con estilos para modificar el estado del LED
```

```
client.println("<button type='button' onClick=location.href='/LED=ON' style='margin:auto; background-color:green; color:#A9F5A9; padding:5px; border:2px solid black; width:200;'><h2> ENCENDER</h2> </button>");
```

```
client.println("<button type='button' onClick=location.href='/LED=OFF' style='margin:auto; background-color:red; color:#F6D8CE; padding:5px; border:2px solid black; width:200;'><h2> APAGAR</h2> </button><br><br>");
```

```
//Se crea un único botón para modificar el estado del LED
```

```
if(estado == HIGH) {
```

```
client.print("<button type='button' onClick=location.href='/LED=OFF'> APAGAR </button><br><br>");
```

```
} else {
```

```
client.print("<button type='button' onClick=location.href='/LED=ON'> ENCENDER </button><br><br>");
```

```
}
```

```
client.println("</html>");
```

```
delay(1);
```

```
Serial.println("Petición finalizada"); // Se finaliza la petición al cliente. Se inicializa la espera de una nueva petición.
```

```
//Desconexión de los clientes
```

```
//WiFi.softAPdisconnect();  
  
}
```

Paso 5. Conexión con el access point

Para poder conectar con la red WiFi realizaremos los siguientes pasos:

- Desde el dispositivo cliente -con el que queremos conectar- (PC, Smart Phone, Tablet...) buscaremos las redes WiFi visibles.
- Conectaremos con la red NodeMCU-ESP8266 con la clave 12345678.
- Abriremos nuestro navegador Web e iremos a la dirección <http://192.168.4.1>
- En el navegador aparecerá el texto para verificar el estado del LED y los botones para controlar su encendido y apagado.

Preguntas:

¿Por qué se ha configurado con 1 el delay?

¿Cuál es la Ip que utiliza NodeMCU por defecto?

Conclusión:

Recomendaciones:

Ejercicio 5. Modos de ahorro de energía -deep sleep

Introducción: El consumo de energía de los procesadores ESP8266 es muy elevado, los podemos resumir rápidamente como “devoradores de energía”. El motivo fundamental es la energía necesaria para realizar las conexiones WiFi, pero en multitud de proyectos la facilidad de conexión WiFi es el motivo por el que resultan útiles estos procesadores.

Objetivo: conocer los modos de ahorro de energía de que disponen para poder construir proyectos que en situaciones ideales duren años, sin tener que cambiar las baterías o depender del uso de enchufes.

Conceptos básicos:

Resistencia: Es toda oposición que encuentra la corriente a su paso por un circuito eléctrico cerrado, atenuando o frenando el libre flujo de circulación de las cargas eléctricas o electrones.

Los diodos LED (light emitting diode), como cualquier diodo, tienen polaridad. Esto significa que la corriente eléctrica circula desde el ánodo (+) hacia el cátodo (-) y no en sentido contrario, por lo que debemos estar atentos a la hora de conectarlos.

El LED tiene un voltaje de funcionamiento aproximado entre 1,8 y 2,1 voltios. La intensidad de funcionamiento que debe circular por él está comprendida entre 10 y 20 miliamperios (mA). Estos valores son genéricos, ya que varían en función del modelo, fabricante, color...

La protoboard es una placa de pruebas para electrónica que contiene numerosos orificios en los que es posible insertar cables y otros elementos electrónicos para montar circuitos provisionales.

La ventaja de este dispositivo es que no requiere soldar sus componentes para tener un circuito operativo.

El Cable dupont macho macho es bastante práctico y sencillo de usar, permitiendo conexiones entre boards de manera ordenada y funcional.

Deep-Sleep: como podrá ver a continuación, es el modo que genera mayor ahorro, pero a costa de dejar la placa en suspenso. La única parte de la placa que funciona durante este modo es reloj en tiempo real, para poder reiniciarla cuando haya finalizado el tiempo de reposo.

Modo de ahorro de energía deep-sleep, el modo de ahorro de energía deep-sleep es uno de los tres modos que disponen los microprocesadores ESP (modem-sleep, light-sleep y deep-sleep).

Es el procedimiento que genera mayor ahorro, pero a costa de desactivar la WiFi, el reloj del sistema –System Clock– y la unidad central de proceso –Central Process Unit o CPU-, por lo que la placa queda en suspenso, finalizando todos los procesos activos.

Únicamente permanece en funcionamiento el reloj en tiempo real –Real Time Clock o RTC-, que será quien envíe la señal para que se pueda reiniciar el microprocesador, cuando finalice la cuenta atrás de tiempo de suspenso que hayamos establecido.

Es por este motivo por el que se debe conectar con un cable el pin GPIO16 (D0) con el pin de Reset (RST). Si no se hiciera la placa no se reiniciaría.

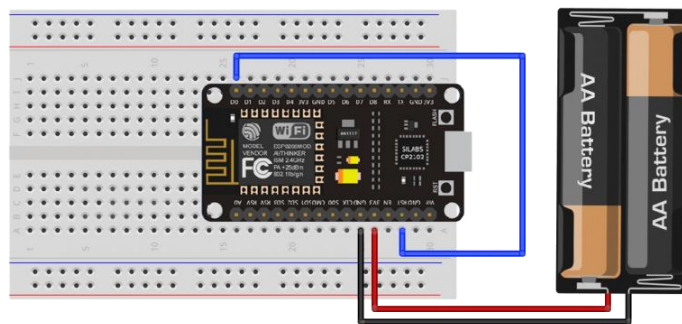


Figura 59. Diagrama de circuito con baterías

El consumo del microprocesador en modo deep-sleep baja de un consumo medio de 80/100 miliamperios a ~20 microamperios conforme indica la guía técnica de Espressif de Low Power Solutions.

Es posible alimentar al microprocesador ESP8266 desde el pin de 3,3V en una placa NodeMCU. El voltaje a utilizar debe estar dentro de los límites establecidos para el microprocesador, es decir, entre 3 y 3,6 V. Haciéndolo de esta forma evitamos el consumo de energía que se produce en el regulador de tensión.

Función deep-sleep: ESP.deepSleep() tiene dos variables: ESP.deepSleep
(tiempo_en_us, modo_reinicio)

tiempo_en_us (uint32_t): tiempo en microsegundos que debe estar el microprocesador en suspenso.

El tiempo máximo que se puede mantener el microprocesador en suspenso es de aproximadamente 71 minutos y 30 segundos. Esto se debe a que los microprocesadores ESP8266 trabajan a 32 bits, por lo tanto el número más alto que admiten es $2^{32} = 4.294.967.295$. Dado que el tiempo de suspenso se da en microsegundos (μs) tenemos que:

$$4.294.967.295 \mu\text{s} = 4.294,967295 \text{ s} \approx 71' 30''$$

modo_reinicio: indica cómo se debe proceder cuando se reinicia el microprocesador con la calibración de la señal de radio para la WiFi. Existen cuatro posibilidades:

WAKE_RF_DEFAULT: cuando se reinicia el microprocesador, únicamente se calibra la señal de radio si el chequeo da error (init data byte 108>0).

SLEEP_TIME, WAKE_RFCAL: cuando se reinicia el microprocesador siempre se calibra la señal de radio. Esto incrementa el consumo.

SLEEP_TIME, WAKE_NO_RFCAL: cuando se reinicia el microprocesador no se calibra la señal de radio. Esto reduce el consumo.

WAKE_RF_DISABLED: cuando se reinicia el microprocesador se deshabilita la señal de radio (como en el modo modem sleep). Este es el modo con menor consumo, pero no permite ni enviar ni recibir datos vía WiFi.

Para hacer uso de la función deep-sleep se propone un ejemplo muy sencillo, el control de encendido y apagado de un led (blink), pero el tiempo que el LED esté apagado, en vez de estar el microprocesador ESP8266 funcionando, estará en modo deep-sleep.

Así mismo haremos que cada vez que se reinicie el microprocesador se deshabilite la señal de radio (WAKE_RF_DISABLED) ya que no será necesario ni el envío ni la recepción de datos vía WiFi.

El tiempo que se ha establecido que debe permanecer el LED encendido es de 1 segundo y apagado 5 segundos.

Materiales:

1. Placa NodeMCU.
2. Led.
3. Resistencia 100Ω / 0,25W.
4. Breadboard.
5. Cables protoboard (Dupont).

Paso 1: Abrir el IDE de Arduino para comenzar a programar. A continuación, abrir el programa Arduino con el acceso directo creado en el escritorio. Luego crear el circuito presentado a continuación:

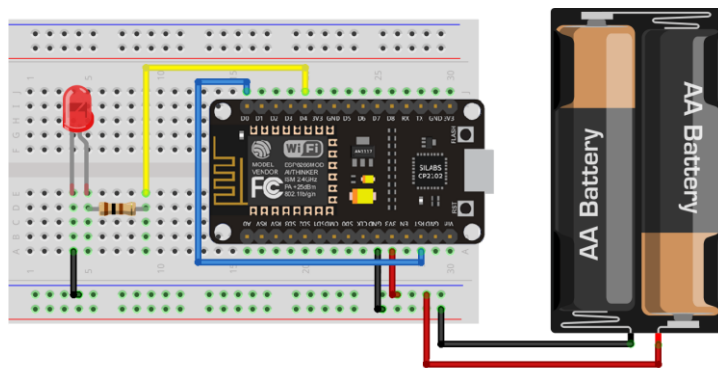


Figura 60. Diagrama de circuito Deep sleep

Importante: Es necesario desconectar el cable GPIO16 (D0)-RST para subir el sketch a la placa y reconectarlo para que funcione su ejecución.

```

int LED = 2;           // Pin LED GPIO02 (D4)

uint32_t TIEMPO_DeepSleep = 5e6; // Tiempo en modo deep-sleep en microsegundos

uint8_t TIEMPO_Referencia;    // Tiempo de referencia para encender 1 segundo el LED

void setup() {

    pinMode(LED, OUTPUT);      // Inicializa el LED_aparpadear como una salida

    digitalWrite(LED, HIGH);   // Enciende el LED

    TIEMPO_Referencia = millis();

}

void loop() {

    if (millis()-TIEMPO_Referencia > 1000){

        //ESP.deepSleep(TIEMPO_DeepSleep, WAKE_RF_DEFAULT); // Calibración de señal de
radio si es necesario

        //ESP.deepSleep(TIEMPO_DeepSleep, WAKE_RFCAL);      // Calibración de señal de
radio siempre

        //ESP.deepSleep(TIEMPO_DeepSleep, WAKE_NO_RFCAL);   // Sin calibración de la
señal de radio

        ESP.deepSleep(TIEMPO_DeepSleep, WAKE_RF_DISABLED); // Deshabilita la señal de
radio después del reencendido

    }

}

```

Es necesario reconectar el cable que va del pin GPIO16 (D0) al Reset (RST) para que el microprocesador se pueda reiniciar y el sketch funcione correctamente.

Preguntas:

¿En qué consiste la función DeepSleep ?

¿Qué pasaría si el tiempo que se ha establecido en el cual debe permanecer el LED encendido es de 2 segundos y apagado 6 segundos?

Conclusión:

Recomendaciones

Capítulo V: Manual de Usuario Arduino y NodeMCU

5.1 Introducción

Arduino puede tomar información del entorno a través de sus pines de entrada, de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores. El Microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en wiring) y el entorno de desarrollo Arduino (basado en processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software.

Las placas pueden ser hechas a mano o compradas montadas de fábrica; el software puede ser descargado de forma gratuita. El presente manual de prácticas basado en la placa Arduino, se elaboró con la finalidad de apoyar a los docentes y estudiantes. Universitarios respectivamente, en la construcción gradual de sistemas basados en Microprocesador, especialmente pensados para su uso en un amplio rango de aplicaciones industriales y de gran consumo, como dispositivos embebidos. Las prácticas descritas en este material se derivan de fuentes bibliográficas, así como de documentos libres de internet.

5.2 Manual de usuario para Arduino

Instalación

Como instalar Arduino en Windows

Los siguientes son los pasos que tienes que seguir para empezar a trabajar con la placa Arduino:

1.- Comprar una placa Arduino y el cable USB

La placa Arduino es una tarjeta I/O simple la cual se basa en el procesador ATmega8 de Atmel. La tarjeta impreso (PCB) y está compuesta por un circuito componentes electrónicos.

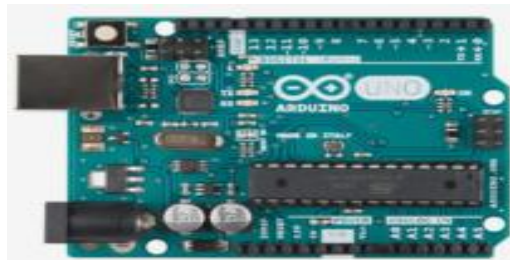
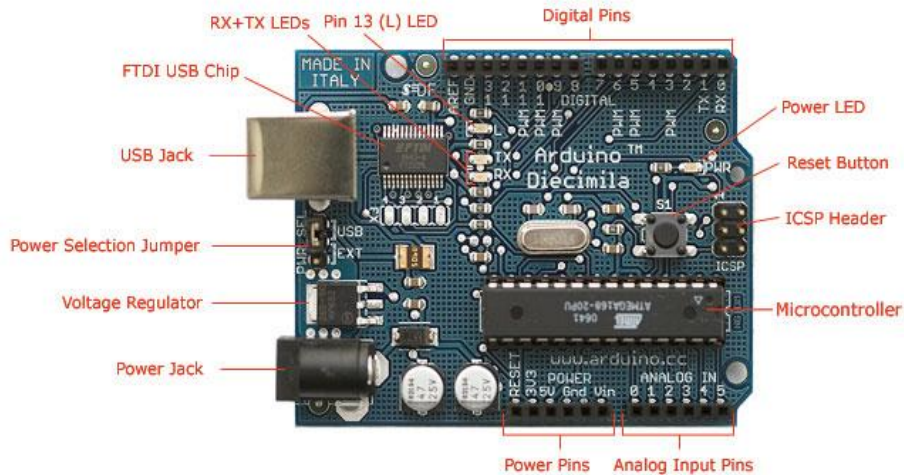


Figura 61. Placa Arduino Uno



Photograph by SparkFun Electronics. Used under the Creative Commons Attribution Share-Alike 3.0 license.

Figura 62. Arduino Duecimila

Hay varias formas de obtener una placa Arduino:

Comprar una placa: La forma más sencilla es comprando una ya lista.

Construir tu propia placa: si quieres construir tu propio PCB descarga los archivos CAD desde la página de Hardware. Extrae el archivo .brd y envíalo a un fabricante de PCB. Ten en cuenta que un fabricante de PCB puede cobrarte bastante caro por hacer una sola placa. Es mejor juntar a un grupo de 20 a 30 personas. Como los archivos CAD están a libre disposición también puedes adaptar la placa a tus propios requerimientos.

Comprar las partes: Compra las partes tu tienda de electrónica preferida. La versión serial ha sido diseñada con componentes básicos, para que puedas encontrar los componentes

fácilmente. La versión USB requiere de habilidades más avanzadas en cuanto a lo que es soldadura ya que utiliza componentes SMD como el chip FTDI.

Programar el bootloader: para que el ambiente de desarrollo pueda programar el chip, este tiene que ser programado con un código especial llamado bootloader.

2.- Descargar el entorno de desarrollo Arduino

Para programar la tarjeta necesitas el entorno de desarrollo Arduino. Descargar en el siguiente enlace <https://www.arduino.cc/en/Main/Software>

Seleccionar el tipo de sistema operativo que corresponda

Download the Arduino IDE



Figura 63. Ventana de descarga de software arduino

3.- Instalar el entorno de desarrollo de Arduino (IDE)

Ejecuta el instalador del programa y sigue los pasos de instalación. Ejecuta el instalador del programa y sigue los pasos de instalación.



Aceptar los términos y condiciones de la licencia.

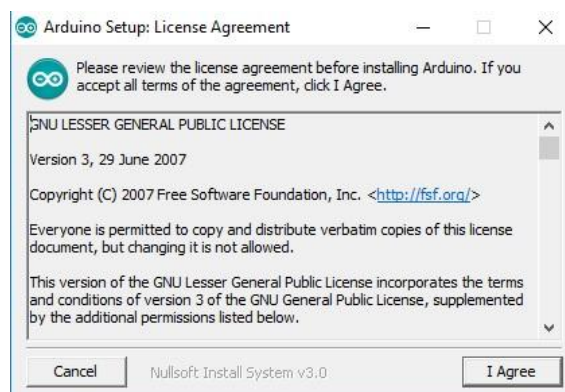


Figura 64. Ventana de términos de licencia

Seleccionar todas las opciones para que instale todos los complementos y drivers necesarios.

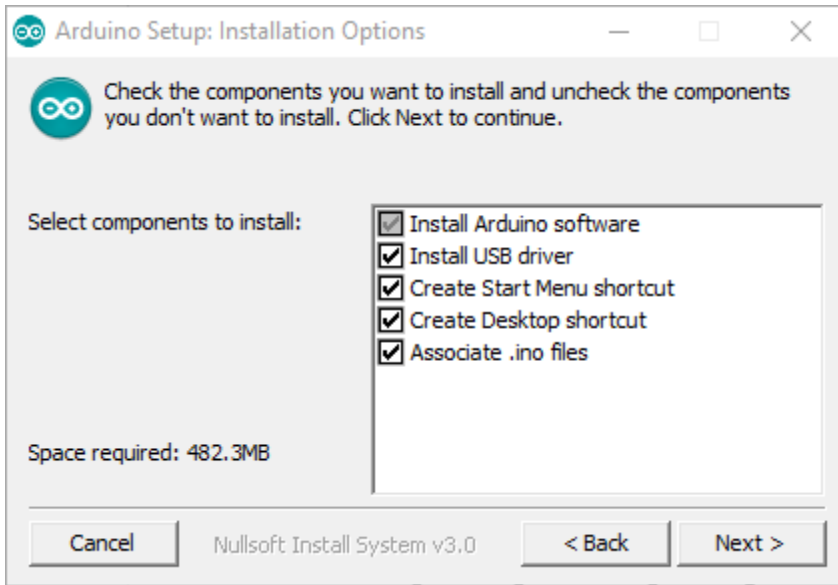


Figura 65. Ventana de Opciones de instalación

Seleccionar la ruta de instalación y presiona «install».

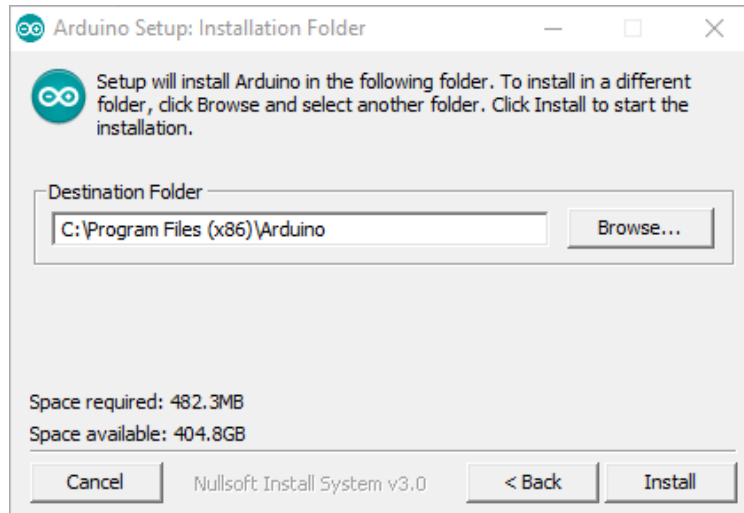


Figura 66. Ventana de ubicación de destino

Esperar un par de minutos que termine el proceso de instalación.

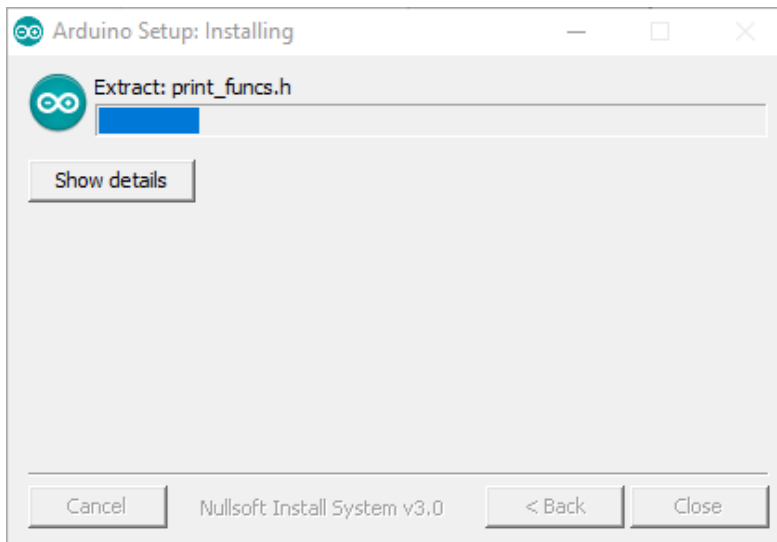


Figura 67. Ventana de progreso de instalación

Si estás utilizando la placa Arduino, necesitarás instalar los drivers del chip FTDI. Sigue estos pasos para completar la instalación.

Si está usando una placa Arduino Uno original, los drivers se instalan automáticamente con el IDE.

4.- Conectar la tarjeta Arduino

La alimentación puede ser provista por el puerto USB o por una fuente externa (6-12V). Cualquiera sea el modo de alimentación conecta el cable USB al pc.

El LED de encendido debería iluminarse.

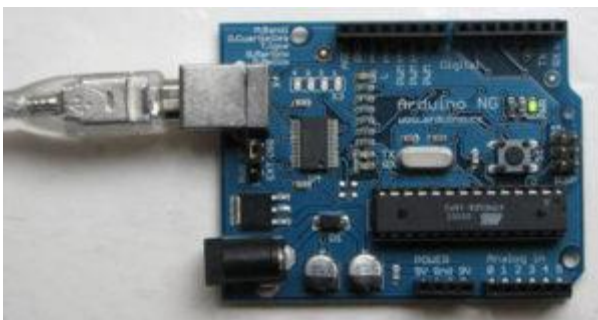
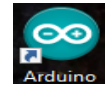


Figura 68. Placa arduino con conexión usb

5.- Ejecutar el ambiente de desarrollo Arduino.

Ve a escritorio y haz doble clic en la aplicación Arduino.



7.- Cargar el programa en la tarjeta

Abre el

sketch de ejemplo: Archivo

> Ejemplos

> 01.Basics > Blink

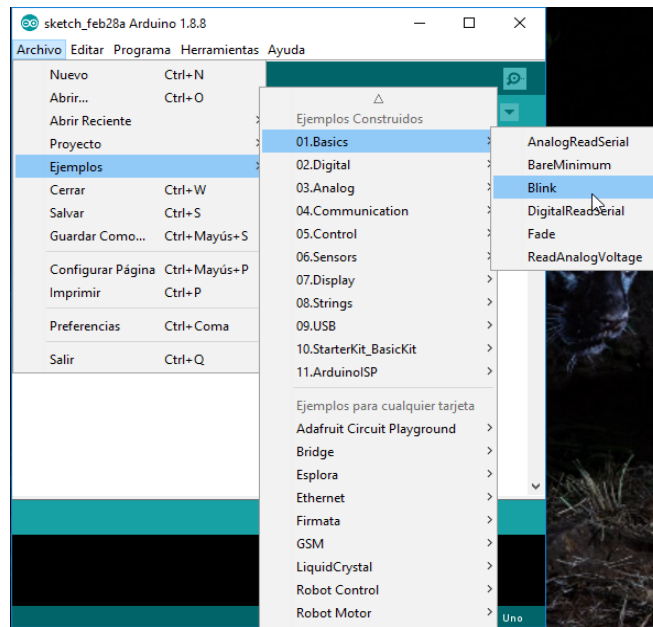


Figura 69. Menú para ejemplo de arduino

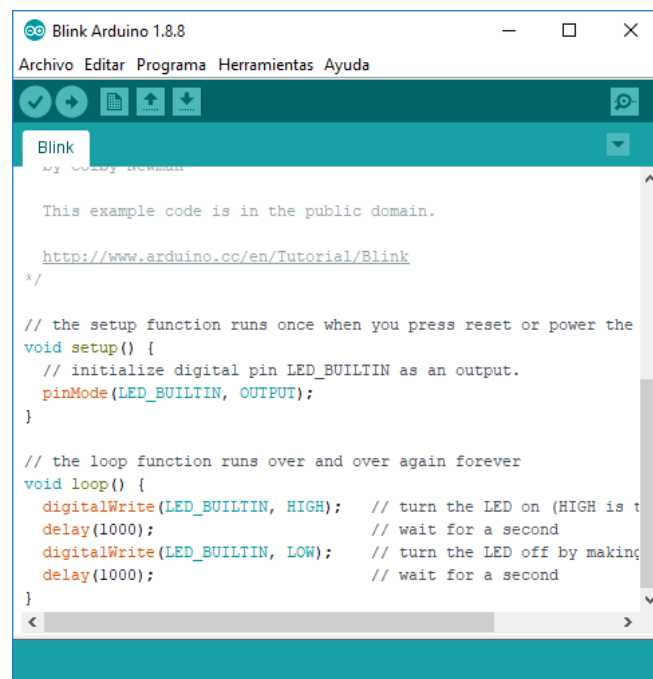


Figura 70. Ejemplo en el IDE arduino

Selecciona el puerto COM en el que tienes conectada la tarjeta Arduino en el menú Herramientas >

Puerto.

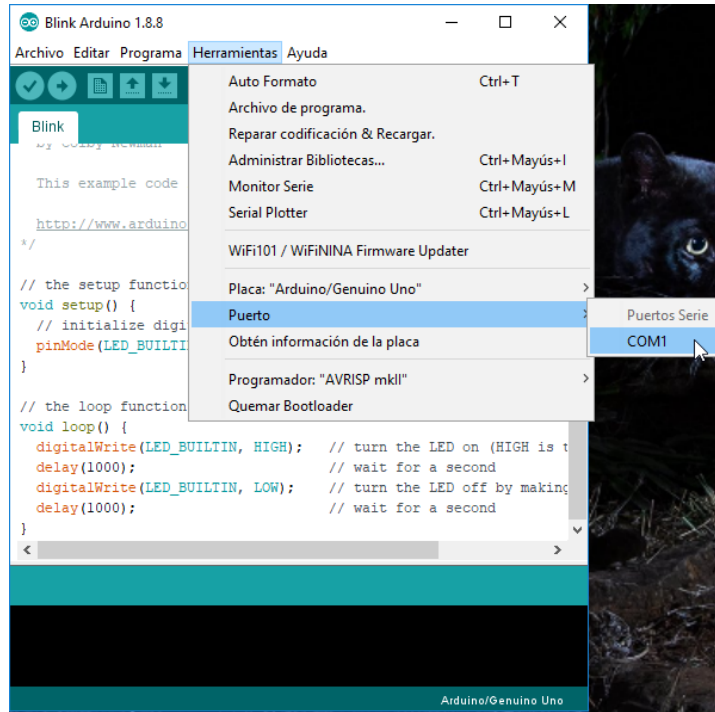
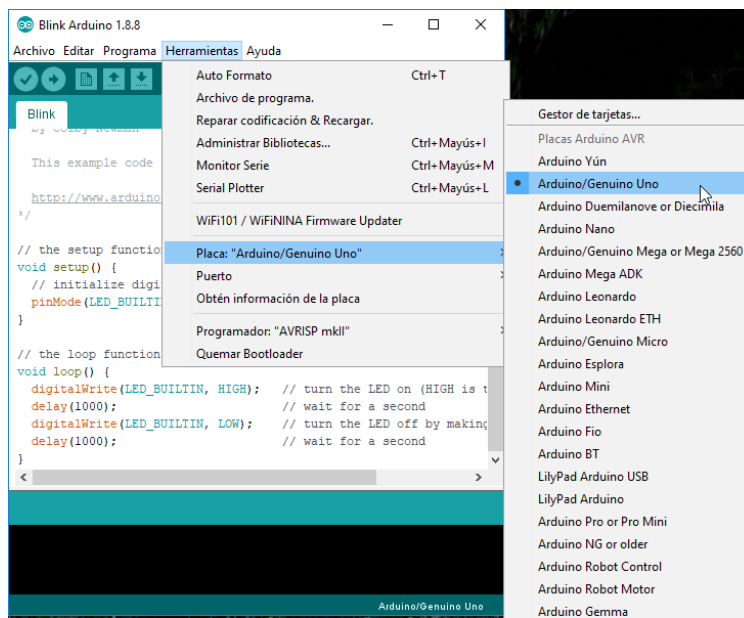


Figura 71. Ventana de selección de puerto

Se
se está
chip
tarjeta y



necesitará especificar el microcontrolador que utilizando. Observa el que está instalado en tu seleccionar en Herramientas >Placa:

Figura 72. Ventana de selección de placa

Si tienes la nueva Arduino Diecimila, simplemente haz clic en el botón “Upload”. Si estás utilizando una Arduino Mini o una Arduino NG u otra tarjeta, necesitarás presionar el botón de reset presente en la placa inmediatamente antes de apretar el botón “Upload”. Espera unos segundos y deberías ver los LED’s RX y Tx de la tarjeta parpadeando. Si el programa se cargó satisfactoriamente un mensaje aparecerá en la barra de status “Done uploading”.

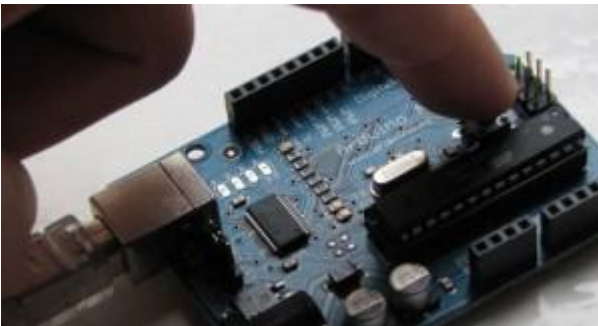


Figura 73. Visualización de placa arduino con led parpadeando

8.- Mirar como pestañea el LED

Unos segundos después de que se haya cargado el programa el LED empezará a parpadear. ¡Si lo hace felicitaciones! Ya ha realizado tu primer paso con las placas Arduino.

5.3 Introducción al manual NodeMCU

La plataforma ESP32 ofrece a bajo costo capacidades wifi, BlueTooth y BLE. Es la evolución del chip ESP8266 mejorando sus capacidades de comunicación y procesamiento computacional. Tiene un CPU de dos núcleos de hasta 240Mhz que se pueden controlar independientemente. Además integra internamente una gran cantidad de periféricos incluyendo: sensores táctiles capacitivos, sensor de efecto Hall, amplificadores de bajo ruido, interfaz para tarjeta SD, Ethernet, SPI de alta velocidad, UART, I2S e I2C.

El punto fuerte de esta plataforma es su gran comunidad en internet que le brinda soporte y desarrolla constantemente nuevas herramientas para su uso.

Para su desarrollo cuenta con gran variedad de software, lenguajes de programación, frameworks, librerías, código/ejemplos y otros recursos. Los más comunes a elegir son: Espressif (Espressif IoT Development Framework) desarrollado por el fabricante del chip, Arduino (en lenguaje C++), Simba Embedded Programming Platform (en lenguaje Python), RTOS's (como Zephyr Project, Mongoose OS, NuttX RTOS), MicroPython, LUA, Javascript (Espruino, Duktape, Mongoose JS), Basic.

La tarjeta NodeMCU-32 está diseñada especialmente para trabajar en protoboard. Posee un regulador de voltaje en placa que le permite alimentarse directamente del puerto USB. Los pines de entradas/salidas trabajan a 3.3V. El chip CP2102 se encarga de la comunicación USB-Serial. Aplicado en Mini Servidores Web, Procesamiento digital, Webcams, Domótica y más.

5.4 Objetivos del manual

El presente manual se ha creado con la finalidad de orientar al usuario (docentes como universitarios) la forma como debe instalar el programa requerido para poder utilizarlo en múltiples proyectos multidisciplinarios que podrán aportar beneficios para la sociedad.

5.5 Manual de usuario para NodeMCU

Instalación

Para empezar necesitamos tener instalado el Python 2.7

Python es una solución rápida y cómoda para programar cositas en nuestros micros y por eso llevaba tiempo buscando alguna manera de programar Arduino con Python, y en concreto con una variante de reciente aparición llamado MicroPython, que está pensada para correrla en micro controladores de este estilo.

MicroPython es un desarrollo de un intérprete de Python 3, pensado para correr en micro controladores con pocos recursos comparados con un PC, y por eso han hecho algunas concesiones como no incluir muchas de las librerías estándar que el Python 3 normal incluye, pero que pueden ser cargadas con import cuando se requieran y que además incluye librerías que se pueden importar para acceder al hardware de forma cómoda.

MicroPython no viene porque sea una versión reducida, sino de que es una versión para microcontroladores.

Para poder flashear nuestro NodeMCU vamos a necesitar crear un entorno de trabajo adecuado, con unos cuantos programas necesarios (En este caso desde Windows 10), y conviene que lo hagamos en orden:

Instalar Python 2.7: Si. Aunque MicroPython es Python 3.0, algunas de las utilidades precisas requieren la versión 2.7 (O eso dicen, que no lo he probado)

- Instalar pip o comprobar que está disponible.
- Instalar esptool.
- Descargar la imagen que vamos a flashear

Vamos a la [página oficial de Python](#) y descargamos la versión 2.7: Cuando termina la descarga basta con que ejecutéis el fichero.

Os hará varis preguntas tontas que podéis responder con intro y cuando finalice deberías tener instalado todo en C:\Python27

Comprobar que el path apunta a Python. Para ello, escribe “entorno” en la ventana de búsqueda:

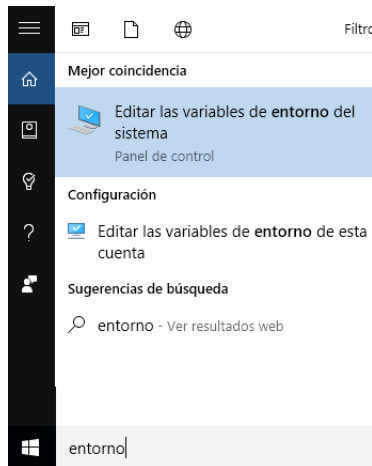


Figura 74. Ventana de búsqueda para variables de entorno

Elegir la primera opción

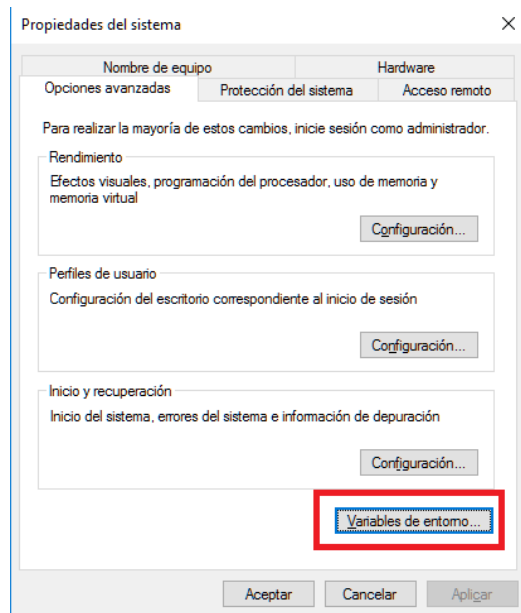


Figura 75. Ventana propiedades del sistema

Seleccionar variables de entorno y busca Path:

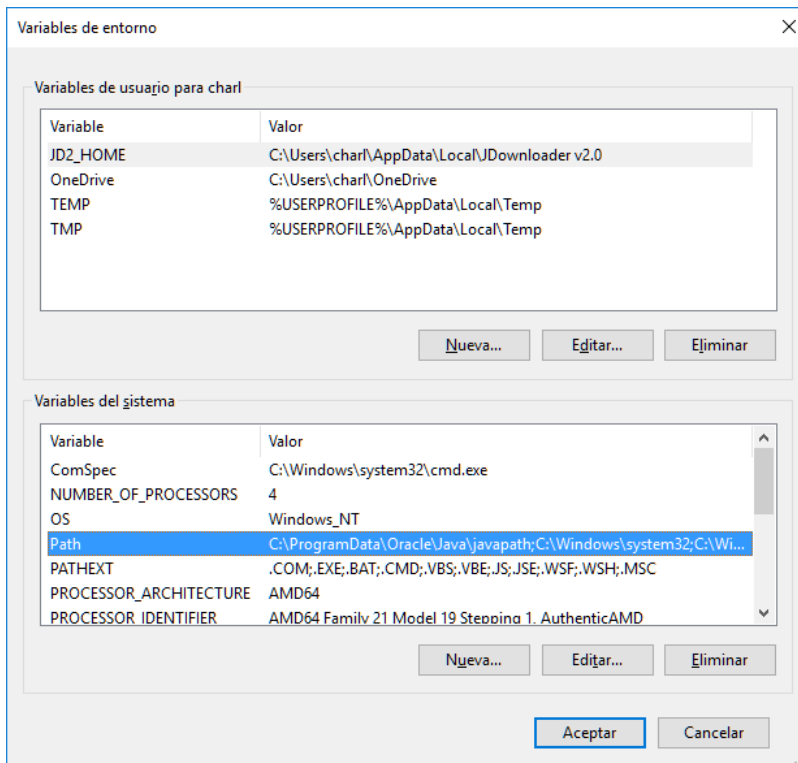


Figura 76. Ventana de variables de entorno

Clic en editar para ver los valores de path:

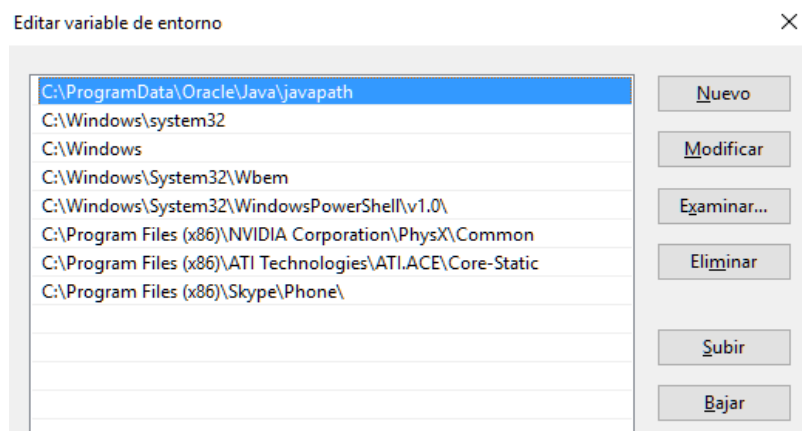


Figura 77. Edición de variables de entorno

Como puedes ver, no ha incluido el path a los directorios de Python con lo que vamos a tener problemas cuando queramos ejecutar algo. Tenemos que añadir un par de líneas al path:

C:\Python27

C:\Python27\Scripts

Clic en el botón Nuevo y añadir las dos líneas hasta que quede así:

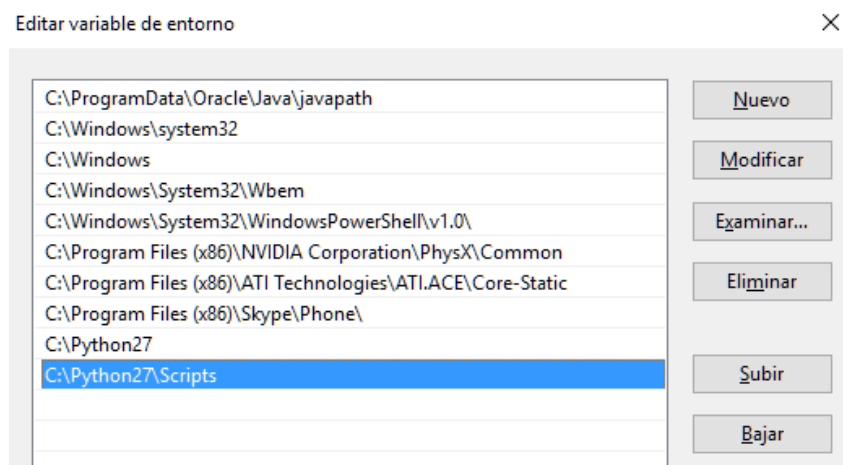


Figura 78. Edición de variables seleccionando botón nuevo

Clic en aceptar dos veces

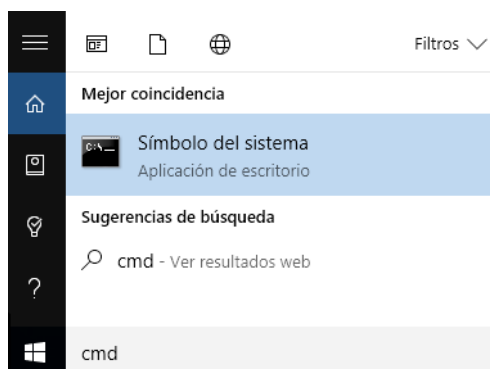
Colocar icono con Python 2.7 en el escritorio de la siguiente manera: ir a botón de inicio\todos los programas\Python 2.7 y seleccionar IDLE que es el entorno gráfico.

Podemos lanzar el entorno gráfico de Python 2.7 desde el menú de aplicaciones, pero también nos interesa tener instalado pip que es algo así como un instalador estándar de paquetes en Python.

Abrir una consola de comandos en Windows, escribiendo cmd en la barra de búsqueda y presionar enter

Escribe:

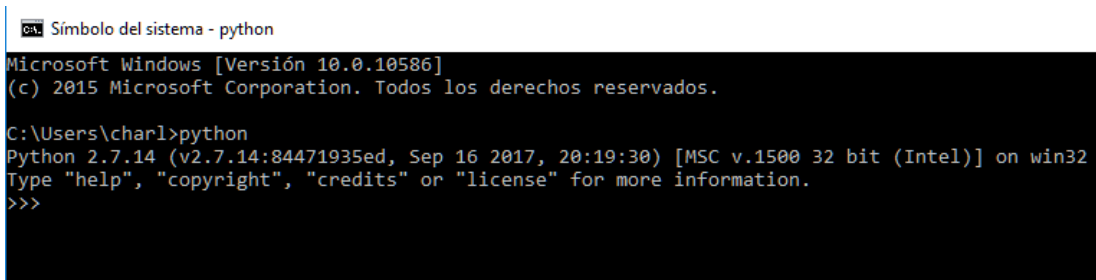
[intro]



Print("Hello world")

Figura 79. Ventana de inicio Windows

Vamos a comprobar que tenemos instalado Python y en el path. Escribe Python + [Intro]:



```
Símbolo del sistema - python
Microsoft Windows [Versión 10.0.10586]
(c) 2015 Microsoft Corporation. Todos los derechos reservados.
C:\Users\charl>python
Python 2.7.14 (v2.7.14:84471935ed, Sep 16 2017, 20:19:30) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Figura 80. Ventana símbolo del sistema

Se visualizará el prompt de Python (>>>) y si se observa de esa manera indica que fue satisfactoria la instalación. Para salir presionar [CTRL] + Z e [Enter]. Escribe pip [Intro], para asegurarnos de que está instalado, verificar variedad de opciones sin error.

La utilidad esptool sirve para volcar el firmware de MicroPython a nuestro ESP8266 / NodeMCU y aunque hay algunas utilidades gráficas que son más cómodas, Escribe en la consola:

Pip install esptool

☐ Símbolo del sistema

```
C:\Users\charl>pip install esptool
Collecting esptool
  Using cached esptool-2.1.tar.gz
Collecting pyserial>=2.5 (from esptool)
  Using cached pyserial-3.4-py2.py3-none-any.whl
Collecting pyaes (from esptool)
  Using cached pyaes-1.6.1.tar.gz
Collecting ecdsa (from esptool)
  Using cached ecdsa-0.13-py2.py3-none-any.whl
Installing collected packages: pyserial, pyaes, ecdsa, esptool
  Running setup.py install for pyaes ... done
  Running setup.py install for esptool ... done
Successfully installed ecdsa-0.13 esptool-2.1 pyaes-1.6.1 pyserial-3.4
C:\Users\charl>
```

Figura 81. Verificación de instalación de pip en ventana símbolo del sistema

Tenemos instalado un entorno Python adecuado para flashear nuestro NodeMCU o esp8266, y ahora se descargará la imagen que vamos a volcar. Para ello vamos a la página de descargas de MicroPython

Se observará que hay muchos micro controladores soportados además el que utilizaremos, buscar la sección: “Firmware for ESP8266 boards” y descargar la última versión estable disponible. En este caso:

esp8266-20170823-v1.9.2.bin (elf, map) (latest)

Es conveniente tener en cuenta que dependiendo del esp8266 que tengamos puede tener 512K o 1024 K de memoria RAM.

Si ud tiene NodeMCU o cualquier modelo con un ESP12, ESP12E no tendrá problemas y se puede descargar el firm de 1024, pero muchos otros modelos de esp son de 512K y se tendrá que buscar la imagen adecuada.

Descargar la imagen que corresponda

Ahora es buen momento para conectar nuestro NodeMCU al USB y determinar el puerto serie que levanta. Para ello, pulsar la tecla [Windows] +X y se abrirá un menú especial. Elegir administrador de dispositivos:

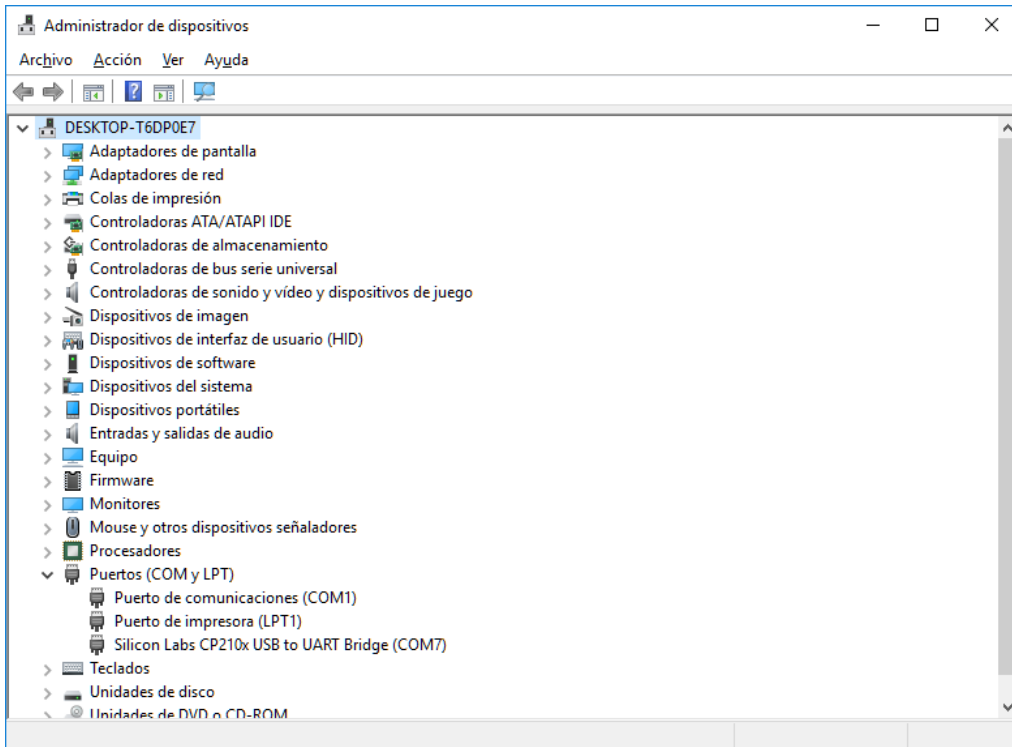


Figura 82. Ventana administrador de dispositivos

Revisar puertos Com y LPT y se visualizará que puerto com1 se ha creado, en este caso el COM7 para el USB del NodeMCU.

El fabricante recomienda borrar la memoria flash actual de nuestro micro. Abrir una consola de comandos y escribir:

```

C:\Users\charl>esptool.py --port COM7 erase_flash
esptool.py v2.1
Connecting...
Detecting chip type... ESP8266
Chip is ESP8266
Uploading stub...
Running stub...
Stub running...
Erasing flash (this may take a while)...
Chip erase completed successfully in 5.1s
Hard resetting...

C:\Users\charl>
C:\Users\charl>_

```

esp
tool
.py
--
por
t
CO
M7

erase_flash

Figura 83. Instalación satisfactoria

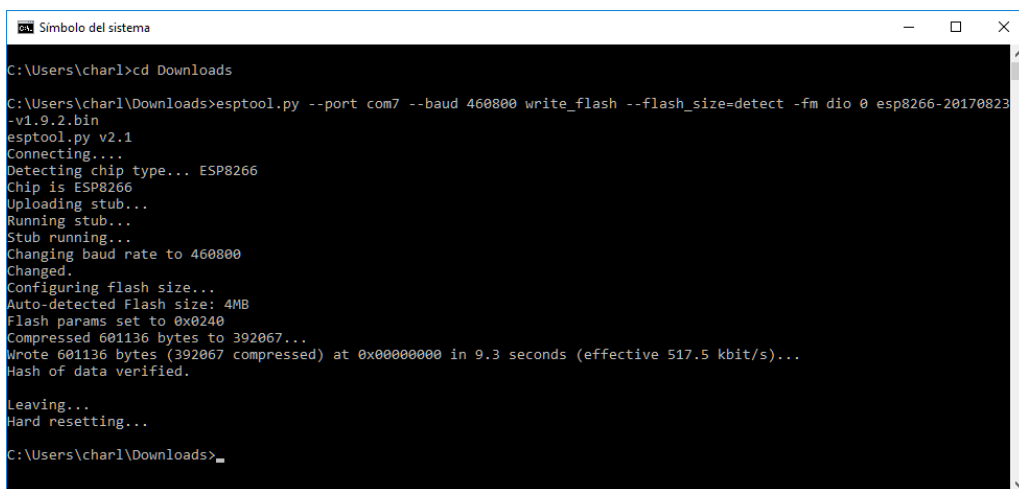
Cambiar com7 por tu puerta USB.

Observar el mensaje que termina con “chip erased completed successfully” y podemos utilizar la imagen que hemos descargado a nuestro micro. Ir al directorio donde se ha descargado (Normalmente el de descargas).

Abrir cmd desde la barra de tareas deberías estar en tu directorio de usuario y por tanto con cd downloads tu ubicación sería descargas y digitar:

```
esptool.py --port com7 --baud 460800 write_flash --flash_size=detect -fm dio 0 esp8266-20170823-v1.9.2.bin
```

Donde com7 es el puerto USB y esp8266-20170823-v1.9.2.bin es el fichero que se ha descargado, asegurarse de escribir los nombres correctos donde corresponda.



```
Símbolo del sistema
C:\Users\charl>cd Downloads
C:\Users\charl\Downloads>esptool.py --port com7 --baud 460800 write_flash --flash_size=detect -fm dio 0 esp8266-20170823-v1.9.2.bin
esptool.py v2.1
Connecting...
Detecting chip type... ESP8266
Chip is ESP8266
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 460800
Changed.
Configuring flash size...
Auto-detected Flash size: 4MB
Flash params set to 0x0240
Compressed 601136 bytes to 392067...
Wrote 601136 bytes (392067 compressed) at 0x00000000 in 9.3 seconds (effective 517.5 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting...
C:\Users\charl\Downloads>
```

Figura 84. Visualización del firm cambiado

Visualizar como el firm se ha cambiado mientras la luz de comunicación del esp8266 va parpadeando y si todo ha ido bien ya tenemos instalado MicroPython en NodeMCU.

La instrucción de arriba cambia el firm a 460800 baudios, es una velocidad más que respetable, si observa que con algunos cables largos da problemas, se recomienda disminuir la velocidad hasta 115200, por ejemplo.

Lo primero es conectar a nuestro NodeMCU mediante la puerta serie y un programa de terminal como putty o TeraTerm, comprobar la comunicación, colocar estos parámetros en \\setup\\serialport:

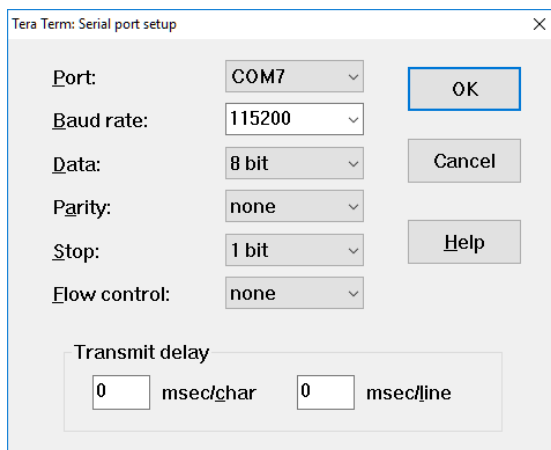


Figura 85. Programa serial teraterm

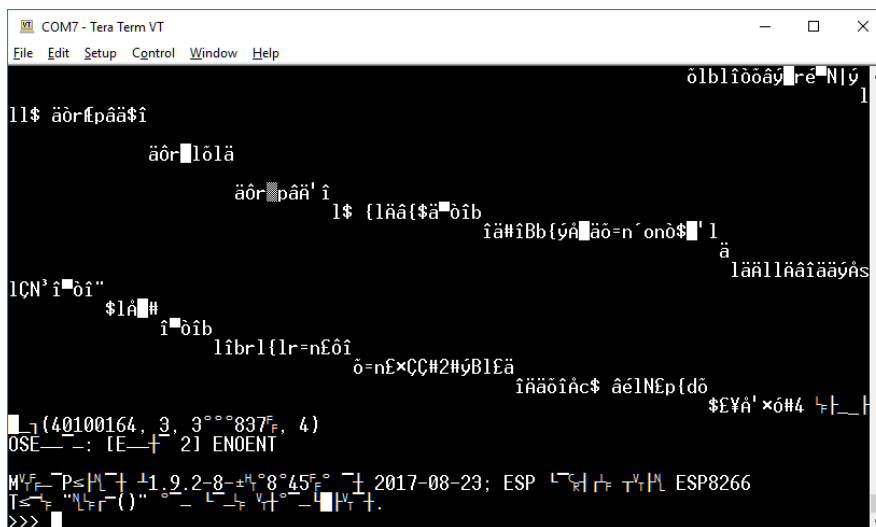


Figura 86. Ventana de terminal

Las primeras veces que se ejecute la terminal se observará tipografía china, típicos de velocidad incorrecta, se corregirán cuando se establezca a 115200 baud, pero al principio se observara así:

Escribir:

[intro] Print("Hello world")

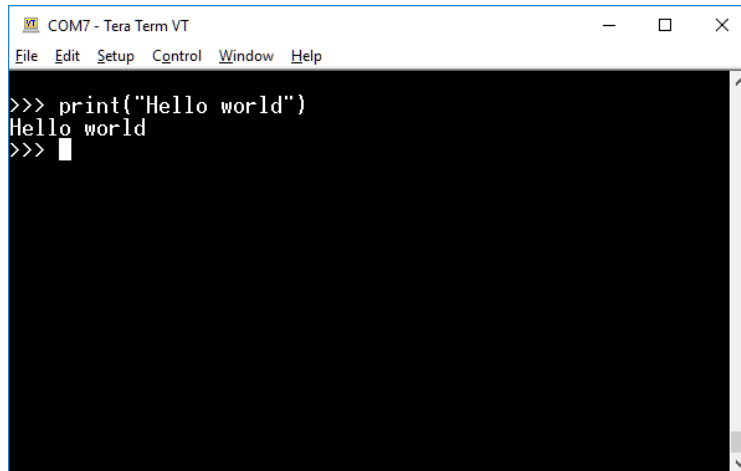


Figura 87. Ventana de terminal

O alguna operación como 23 elevado a 43: 23^{43}

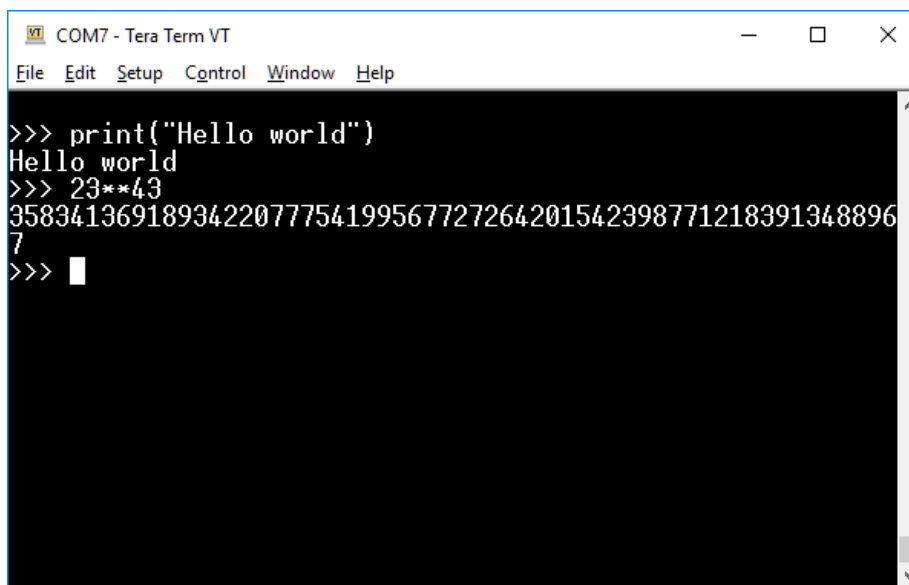


Figura 88. Ventana teraterm con operación matemática

Lo que indica que la instalación está lista y MicroPython dispuesto a trabajar.

Además se debe descargar las últimas versiones del código del emulador Arduino para el ESP32 y para ello tenemos que descargar e instalar el entorno gráfico de GIT.

GIT es un repositorio organizado de código en desarrollo que contiene copias de casi cualquier cosa que se te ocurra y de prácticamente cualquier cosa que tenga un mínimo de interés y sea de interés general, donde los programadores voluntarios van creando y depurando sus cosas.

El interface Arduino para ESP8266 y ESP32 son un par de magníficos ejemplos de las cosas que podemos encontrar

Vamos a descargar el gestor de GIT desde [aquí](#). Ejecuta el programa una vez descargado y se visualizará de esta forma:

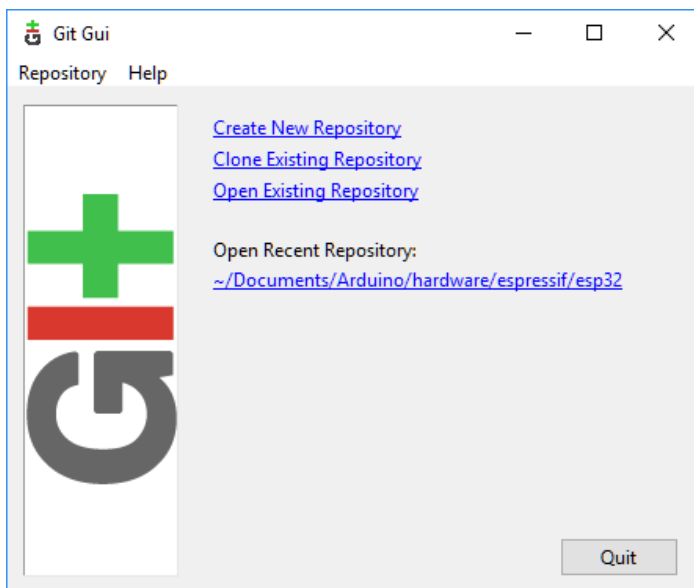


Figura 89. Ventana Git Gui

Presionar la segunda opción: Clone existing repository.

Se debe descargar el código del esp32 a tu pc, y para ello debes indicarle que el repositorio de origen es:

<https://github.com/espressif/arduino-esp32.git>

Y el destino será la dirección de tu directorio Arduino, que se encuentra en mis documentos\Arduino mas la dirección donde se colocará /Arduino/hardware/expressif/esp32.

En este caso quedará así:

C:\Users\charl\Documents\Arduino\hardware\espressif\esp32

Pero en su caso debes asegurarse de colocarle el path correcto del directorio Arduino:

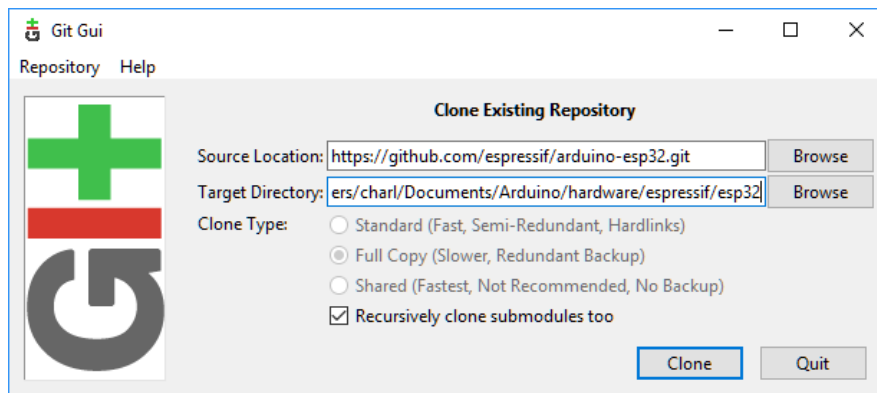
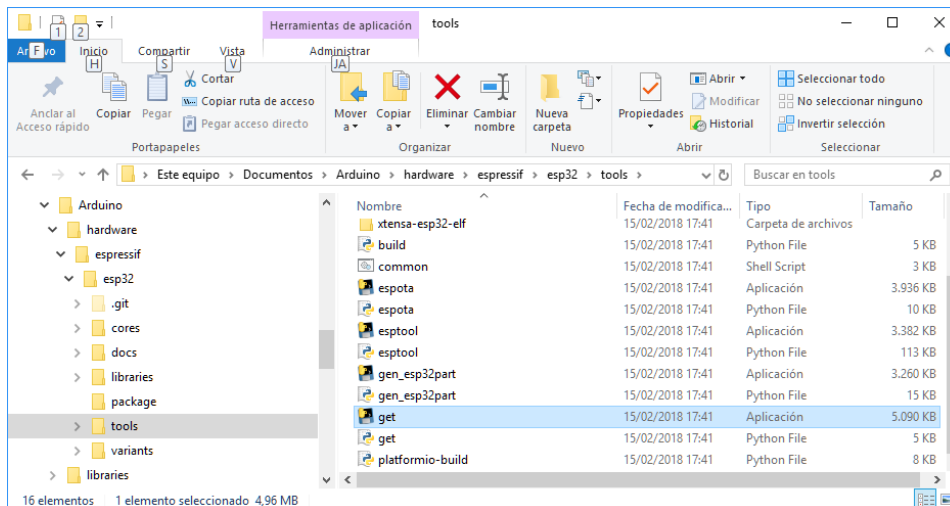


Figura 90. Ventana clone repository

Clic en el botón Clone y empezara a descargar el código a la PC.

Cuando finalice el clonado, ejecutar un programa llamado get que se encuentra en el directorio tools:

Muestra la estructura de directorios del repositorio ESP32 en Github



Hacer un solo clic y esperar mientras se realiza la descarga

Figura 91. Ventana herramientas de aplicación

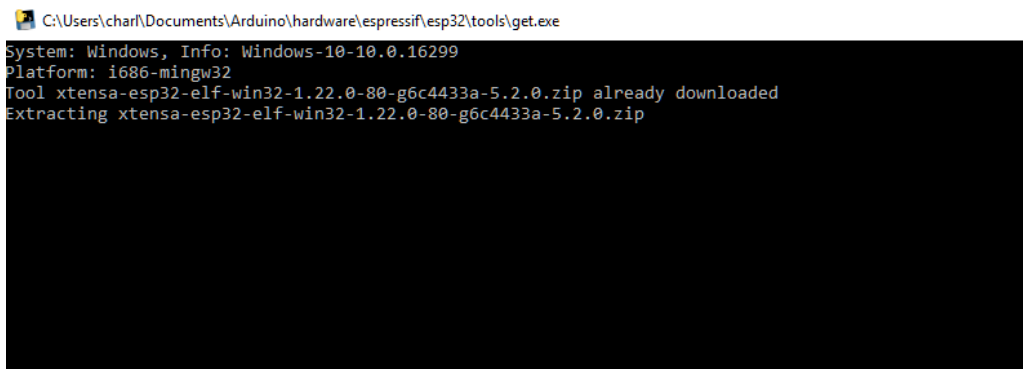
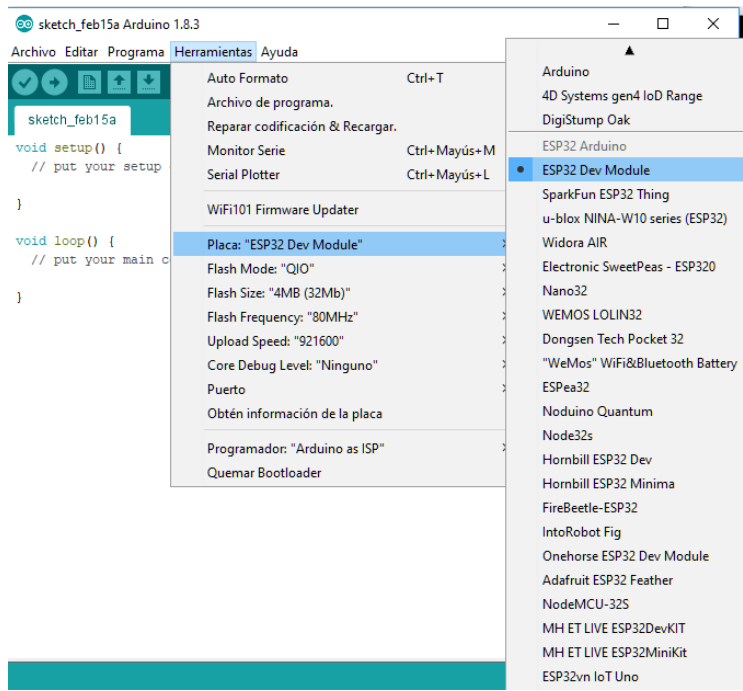


Figura 92. Verificación de sistema descargado

Seleccionar nuestro módulo ESP32. Arrancar IDE Arduino y empezar por conectar tu Node32. Elegir el procesador.

Se

en



observará que hay una nueva sección de placas con múltiples modelos basados el chip ESP32:

Figura 93. Ventana de menú herramientas

Seleccionar el puerto serie y ya estamos listos para probar nuestro primer programa.

Ejemplo

Vete a ejemplos y encontraras un buen montón de programas de prueba para tu esp32. Carga:

```
\\Ejemplos\ESP32\ChipID\GetChipID
```

Compíllalo, pero pon a 115200 baudios la consola de Arduino. EL programa te devolverá un identificador interno del chip que solo sirve para comprobar que todo funciona.

Ejemplo:

Hacer un Blinking Led en nuestro modulo, el LED está en el GPIO 1:

```
Int ledPin = 1;
```

```
void setup()
```

```
{  pinMode (ledPin, OUTPUT);
```

```
  Serial.begin(115200);
```

```
  Serial.println(LED_BUILTIN);
```

```
}  
  
void loop()  
  
{ digitalWrite (ledPin, HIGH);  
  
  delay(1000);  
  
  digitalWrite (ledPin, LOW);  
  
  delay(1000);  
  
}
```

Conclusión

Se ha podido instalar correctamente el software que nos permitirá interactuar con el hardware de arduino y NodeMCU.

5.6 Simulador Fritzing

Fritzing es un software para la automatización del diseño electrónico (Electronic Design Automation) para diseñadores, artistas y cualquier persona que tenga interés en la electrónica y el desarrollo de prototipos. El objetivo de Fritzing es proveer las herramientas que faciliten la documentación y el intercambio de proyectos. (Oliveira, 2011)Fritzing posee una interfaz de usuario amigable que facilita y agiliza el trabajo. Incluye las siguientes secciones de trabajo:

- Vista del Proyecto – es donde el circuito electrónico virtual es construido y editado en vista de protoboard (placa de prototipos), esquema o PCB.
- Ventana de Paleta – incluye la librería de piezas, el inspector de piezas, historial de acciones y navegador de proyectos.
- Creador de Piezas – es una herramienta para modificar o crear piezas para Fritzing.

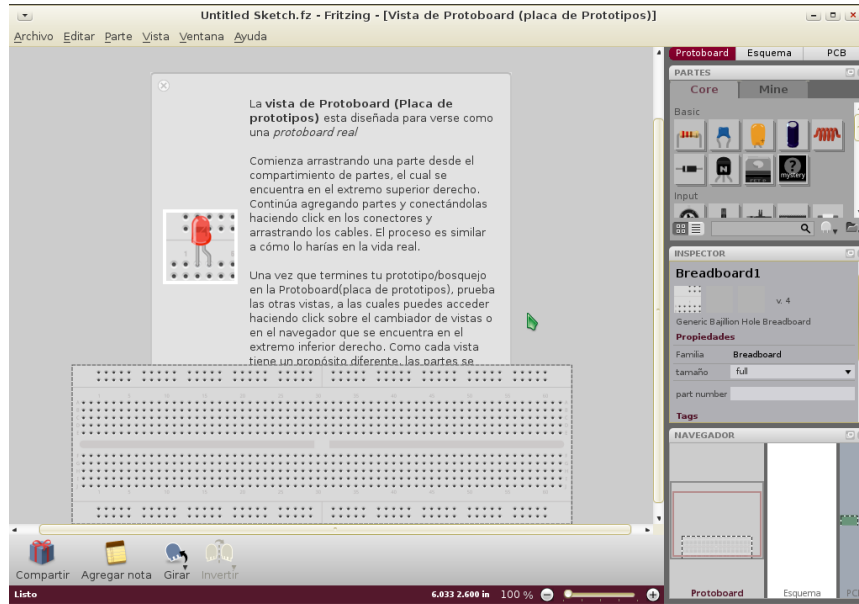


Figura 94. Vista Inicial de Fritzing al iniciar lo por primera vez

El ambiente de Fritzing puede ser reordenado de acuerdo a las preferencias y necesidades del usuario. Las diferentes secciones y ventanas pueden ser movidas, cambiar el tamaño, combinadas, ocultadas o mostradas como objetos flotantes. (Oliveira, 2011)

Vista del Proyecto.

Es donde el circuito electrónico virtual es construido y editado en la vista de protoboard (placa de prototipos), de esquema o PCB. Para cambiar entre vistas, simplemente haga clic en Navegador de vistas o uso el intercambiador de vistas (el cual puede ser mostrado o escondido a través de la sección Ventana de la barra de menú). Mientras se realizan cambios en alguna de las vistas, los cambios tomarán efecto en todas las demás vistas, inmediatamente.

- Vista de protoboard (placa de prototipos), es donde usualmente comenzará un nuevo proyecto. Aquí podrá construir fácilmente su circuito virtual exactamente de la misma forma en que se verá su circuito real, de esta forma se evitan errores que pueden ocurrir en la transición de la capa física al esquema de circuito.

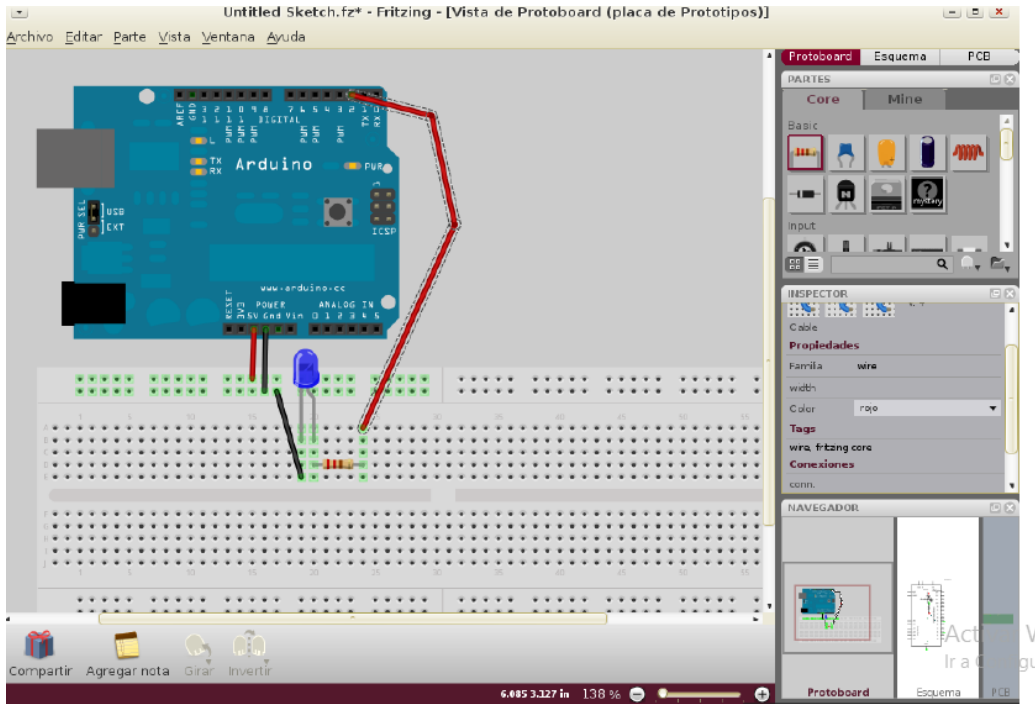


Figura 95. Vista de Protoboard (Placa de Prototipos)

Vista de esquema, muestra el circuito construido en la vista de protoboard como un diagrama de circuito, y es práctico para aquellos que están acostumbrados o que desean aprender los símbolos de circuitos estándar. (Oliveira, 2011)

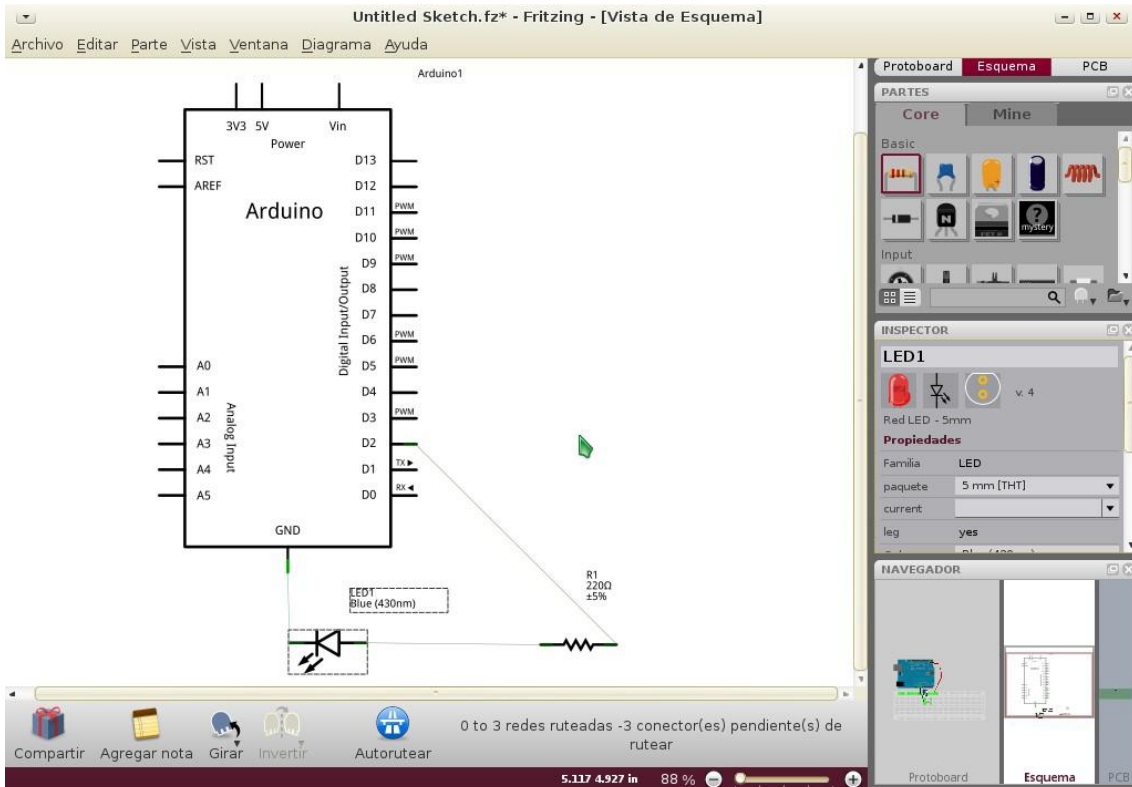


Figura 96. Vista de Esquema a Vista de PCB, permite

e diseñar y exportar la documentación necesaria para la producción de Tablas de Circuito Impreso (Printed Circuit Board). (Oliveira, 2011)

Ventanas de Paleta.

Las ventanas de paleta proveen partes, herramientas e información. Estas pueden ser mostradas o escondidas a través de la sección Ventana de la barra de menú. Estas incluyen:

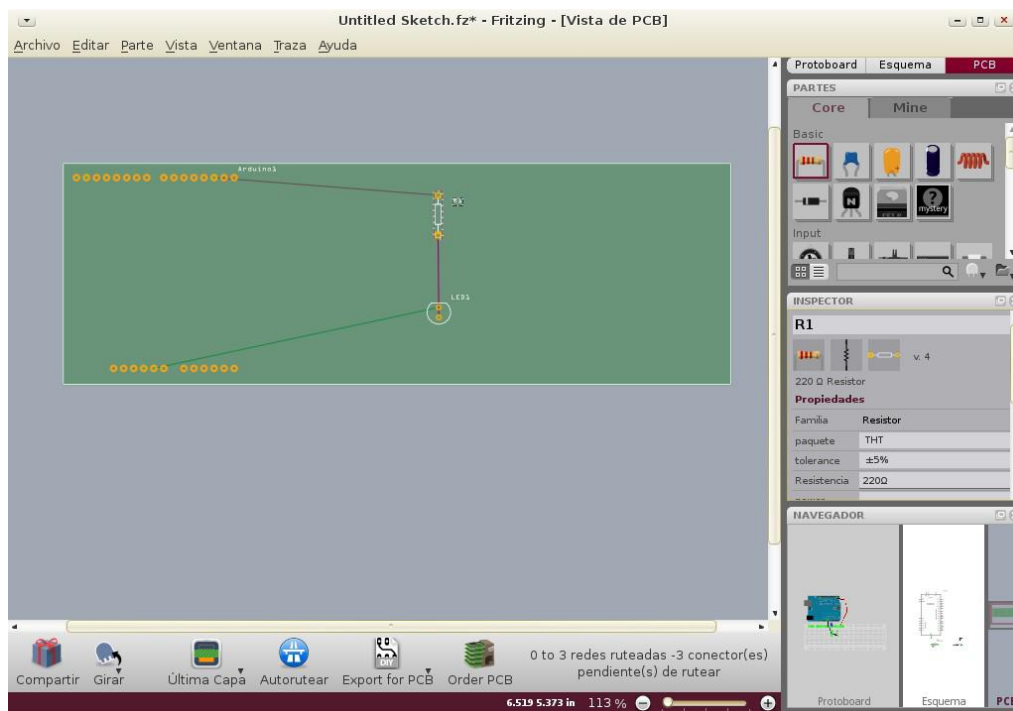


Figura 97. Vista de PCB

Intercambiador de vistas.

Permite cambiar entre las diferentes vistas del proyecto.

Intercambiador de Vistas

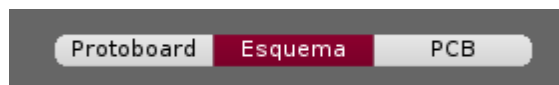


Figura 98. Vista intercambiador de vistas

Librería de piezas.

Tiene una selección de partes electrónicas que se pueden arrastrar y soltar directamente en la vista del proyecto. Las piezas están dispuestas en contenedores. Fritzing viene con un contenedor base (o núcleo), que muestra una buena colección de piezas, pero también permite crear contenedores propios (“Mi contenedor”), por tanto se puede organizar los contenedores base y personalizados de la manera que prefiera. Por ejemplo, puede crear un contenedor mientras trabaja, que solo contendrá las piezas usadas en ese proyecto, o puede crear un contenedor para partes de un tipo específico. Esta característica es útil para mantener el orden y el acceso rápido a las piezas requeridas a través de los grupos. (Oliveira, 2011)

En la parte inferior de cada contenedor encontrará opciones para cambiar de vista, así como para la administración de las piezas y los contenedores:

Mostrar como iconos/lista, proporciona dos maneras de ver las piezas en la librería.

- Pieza, tiene las siguientes opciones:

Nuevo: Abre el Creador de Piezas.

Importar: Permite importar piezas a Fritzing.

Editar: Abre el Creador de Piezas y permite la edición de la pieza seleccionada en el contenedor.

Exportar: Permite exportar la pieza seleccionada en el contenedor.

Remove: Borra la pieza seleccionada del contenedor.

- Contenedor, tiene las siguientes opciones:

Nuevo: Abre un contenedor vacío.

Abrir: Permite abrir contenedores existentes en el sistema.

Abrir núcleo: Abre el contenedor núcleo.

Cerrar: Cierra en contenedor actual.

Guardar: Guarda el contenedor actual.

Guardar como: Guarda el contenedor actual como uno nuevo.

Exportar: Permite exportar el contenedor actual.

Renombrar: Cambia el nombre al contenedor actual. (Oliveira, 2011)



Figura 99. Librería de Piezas

Note las “piezas misteriosas” en la librería (iconos con un signo de pregunta - ?), los cuales son de gran ayuda cuando no se puede encontrar una pieza en el contenedor, y no se quiere pasar por todo el proceso de creación de piezas. Las “piezas misteriosas” permiten definir rápidamente una pieza nueva y sus conectores.

- Inspector de Piezas.

Muestra información de la pieza seleccionada (nombre, icono, propiedades y etiquetas).

Algunos de estos datos, como el nombre de la pieza o sus propiedades, pueden ser modificadas directamente a través de esta ventana. (Oliveira, 2011)

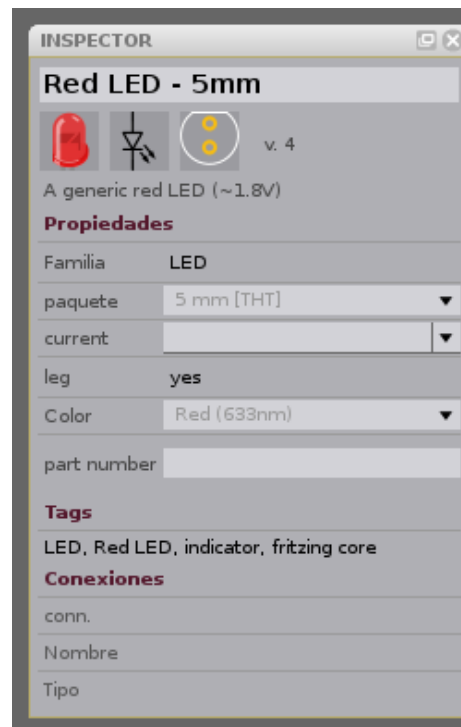


Figura 100. Inspector de Piezas

- Historial de acciones.

Muestra un listado de las últimas acciones ejecutadas y un acceso rápido a la función Deshacer.



Figura 101. Historias de Acciones

El Navegador.

Permite cambiar entre las diferentes vistas del proyecto. También muestra una vista macro del circuito y ayuda a la navegación dentro de la vista del proyecto (usando el macro de la vista). (Oliveira, 2011)

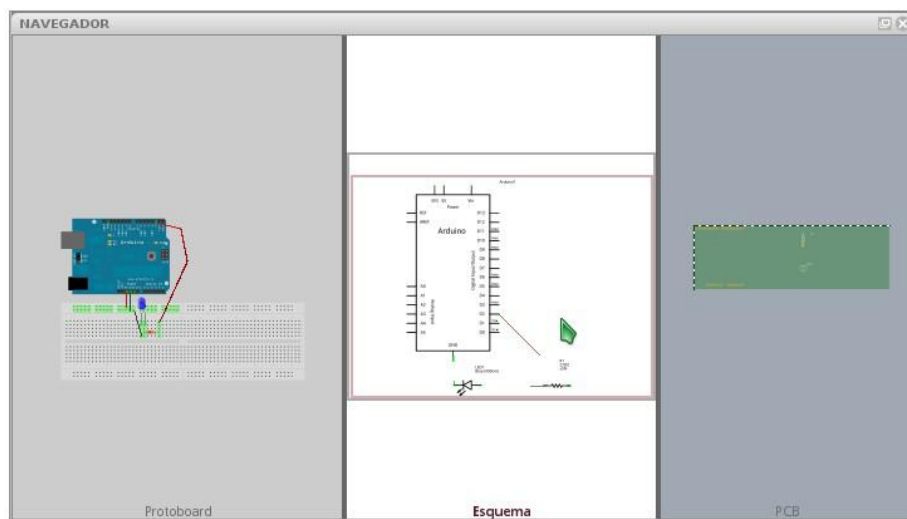


Figura 102. Navegador

Creador de Piezas.

El creador de piezas, es una herramienta para modificar y crear nuevas piezas para Fritzing, y tiene su ventana independiente. Las propiedades y los gráficos de las piezas pueden ser editadas usando esta herramienta. Para abrir el creador de piezas, seleccione desde la barra del menú:

“Pieza > Nuevo” o “Pieza > Editar”.

A continuación se muestran una referencia de todas las opciones del Creador de Piezas:

- Nombre, el nombre de la pieza.
- Icono, el icono de la pieza mostrado en la Librería de Piezas (es cargado presionando “Cargar icono...” debajo del icono).
- Gráficos, “Cargar una imagen...” carga una imagen para la vista correspondiente (Placa, Esquema o PBC).
- Zoom, acerca o aleja para la edición de conectores o puntos de anclaje.
- Especificaciones,
Etiqueta, proporciona un etiquetado adicional, por ejemplo LED#1.
Descripción, breve descripción referente a la pieza.
Etiqueta de control, etiqueta de ID para encontrar una pieza (un campo de búsqueda será integrado en el futuro).
- Autor, nombre del autor/creador de la pieza.
- Conectores,
Lista de conectores, lista todos los conectores de la pieza.
Nombre, el nombre del conector (por ejemplo: “pin0: ground”).

Icono Macho/Hembra, alterna entre los símbolos macho y hembra del conector.

Descripción, breve descripción del conector.

Agregar un conector, agrega un nuevo conector a la lista y gráficos.

- Guardar como pieza nueva, guarda la información nueva o editada como una nueva pieza. (Oliveira, 2011)

Un flujo de trabajo en Fritzing consiste, básicamente, en los siguientes pasos:

1. Construcción de un circuito real – es muy importante que primero construya el circuito en el mundo real y lo pruebe, antes de crearlo virtualmente en Fritzing.

2. Reconstruir el circuito en Fritzing – esto es realizado arrastrando piezas de la Librería de Piezas a la Vista del Proyecto, y hacer las conexiones correspondientes. Si no puede encontrar alguna pieza en la librería, Fritzing permite crear piezas personalizadas a través del Creador de Piezas. Alternativamente puede usar la Pieza Misteriosa (indicada por un icono en forma de signo de interrogación - ?), que le permitirá definir rápidamente una nueva pieza y sus conectores. Las piezas son conectadas usando la placa y cables, los cables son creados automáticamente al hacer clic y arrastrar un conector de cualquier pieza.

3. Editar un circuito – arrastrar y soltar, copiar y pegar, selección múltiple, rotación, historial de acciones, etc., son algunas de las funciones que se pueden encontrar en la barra de menú y secciones de Fritzing. Puede curvar un cable, creando puntos de curvatura. También puede crear Notas o Etiquetas a su bosquejo.

4. Cambiar las propiedades de la pieza – las propiedades de las piezas pueden ser encontradas y modificadas a través del Inspector de Piezas en la sección de la Ventana de

Paleta. Cuando una pieza es seleccionada en la Vista de Proyecto, información relevante es mostrada en el Inspector de Piezas.

5. Cambiar entre las Vistas de Proyecto – usando la paleta del Navegador o el Cambiador de Vistas, puede cambiar entre la vista de placa, esquemática y PCB. Cualquiera de estas vistas

puede ser usada como en entorno principal de trabajo, y puede ser seleccionada en cualquier momento.

6. Diseñar un PBC – cuando el circuito esta completado en la vista de placa, ya puede diseñar el esquema de PBC en la vista PBC.

7. Documentar un Proyecto – un proyecto puede ser documentado y compartido usando la función de Exportar desde la barra de menú. “Guardar como recurso compartido...” guarda todos los archivos necesarios y piezas personalizadas, relacionadas al proyecto como una carpeta compartida. (Oliveira, 2011)

Una Placa de Circuito Impreso (Printed Circuit Board – PCB) puede ser hecha en casa o en un pequeño laboratorio usando proceso de grabado DIY. También puede ser realizado por un servicio de manufactura de PCB profesional para una fabricación más precisa. Fritzing permite diseñar un esquema PCB para ambos casos. Usando la función de Ruta Automática, puede generar automáticamente pistas de cobre y lograr diseños profesionales, los cuales pueden ser exportados en PDF y archivos Gerber. (Oliveira, 2011)

Conclusiones

Considerando que el propósito principal de este trabajo de graduación es elaborar un manual de prácticas para la plataforma Arduino y NodeMCU, las cuales se han desarrollado en un entrenador, para mejorar la enseñanza - aprendizaje en la carrera de Ingeniería de Sistemas Informáticos: caso práctico en la materia de Sistemas Digitales, a fin de dar soporte a los conocimientos teóricos y convertirlos en un elemento útil en el proceso de enseñanza-aprendizaje de docentes y estudiantes.

La población de estudiantes y docentes en quienes se centró el objeto de estudio en la facultad, han manifestado su agrado frente a este tipo de propuesta al determinar la aplicabilidad de la plataforma Arduino y NodeMCU en la materia de Sistemas Digitales de la carrera de Ingeniería de Sistemas Informáticos y poder aplicar de manera fácil y didáctica sus conocimientos teóricos.

Se ha valorado el diagnóstico realizado por medio de encuestas y entrevistas para aportar por un manual como propuesta para las prácticas de la plataforma Arduino y NodeMCU y motivar la participación constante del estudiante, usando ayudas tecnológicas que permitan la fácil asimilación de los temas de la asignatura y utilizando un entrenador, Software y bibliografía para facilitar el proceso de aprendizaje.

Con la elaboración de un manual de usuario para la implementación de las guías prácticas, manejo de la tecnología y recursos electrónicos, los estudiantes no solo cuentan con la información teórica sino que también pueden desarrollar las aplicaciones de manera real y funcional por medio de un entrenador de desarrollo para hardware y software con Arduino y NodeMCU, logran transitar desde el diseño y la simulación, hasta la comprobación de una práctica, se les brinda a los alumnos de tercero y quinto año de la carrera de Ingeniería de Sistemas Informáticos.

Este trabajo de grado deja creadas las bases para la implementación de nuevas guías y otros proyectos con la plataforma Arduino y NodeMCU.

Recomendaciones

- A partir del desarrollo de guías prácticas se podrán crear otros proyectos que se pueden considerar como recomendaciones y sugerencias para trabajos en la plataforma de Arduino y NodeMCU.
- Mediante la resolución de las prácticas propuestas, motiven a los estudiantes a diseñar nuevos proyectos y realizar las siguientes actividades:
- Desarrollar guías prácticas permite verificar las limitaciones, conocimientos sobre elementos electrónicos y alcances que se pueden obtener al hacer uso de nuevas tecnologías.
- Elaboración de proyectos, a efectos de integrar los conocimientos sobre las herramientas electrónicas y tecnologías introducidas en un entrenador, los alumnos deberán realizar trabajos basados en la realidad tecnológica.
- Para diseños funcionales los estudiantes propondrán y mezclarán las metodologías de programación adecuadas para la resolución óptima de problemas, dado que en desarrollo de software siempre hay muchos caminos para obtener un resultado.
- Es de suma importancia seguir los pasos correlativos del proceso de diseño y desarrollo de una aplicación basada en Arduino y NodeMCU, para comprobar que el diseño es algo muy simple a nivel teórico, pero a la hora de implementarlo en la práctica es más complicado de lo que parece.
- Utilizar la tecnología entre Arduino y NodeMCU que mejor parezca de acuerdo a la experiencia adquirida, ya que en ambas se pueden crear proyectos funcionales y de calidad.

Referencias Bibliográficas

admin. (15 de junio de 2019). *6 proyectos con ESP8266 orientados a la tecnología de IoT*. Obtenido de electrogeek: electrogeekshop.com/6-proyectos-con-esp8266-orientados-a-la-tecnologia-del-iot/

Alemán Suárez, J. D., & Mata Mendoza, M. A. (2006). Guía de Elaboración de un manual de prácticas de laboratorio, taller o campo: asignaturas teórico prácticas. *Tesis de subdirección de planes y programas de estudio*. Universidad Autónoma Chapingo, México.

Antony, G. G. (2 de diciembre de 2016). *¿Qué es y cómo funciona el servomotor?* Obtenido de <http://panamahitek.com/que-es-y-como-funciona-un-servomotor/>

arduino, P. (2020). *Sensores de Polvo*. Obtenido de Proyectos con Arduino: <http://proyectosconarduino.com/sensores/polvo-particulas/>

arduinoodhtics. (2015). *Historia*. Obtenido de arduinoodhtics.weebly.com: <https://arduinoodhtics.weebly.com/historia.html>

Carlos, G. D. (13 de febrero de 2020). Obtenido de <https://alsw.net/>

Carrasco Díaz, S. (2005). *Metodología de investigación científica*. Peru: San Marcos.

De anda, N. (23 de Octubre de 2018). *Factor evolución*. Obtenido de Jumper macho-macho: <https://www.factor.mx/portal/base-de-conocimiento/cables-jumper-macho-macho/>

esp8266. (1 de junio de 2018). *NodeMCU, la popular placa de desarrollo conb ESP8266*. Obtenido de luisllamas.es: <https://www.luisllamas.es/esp8266-nodemcu/>

Grokhotkov, I. (2017). *Librería ESP8266Wifi*. Obtenido de Esp8266 Arduino Core: <https://esp8266-arduino-spanish.readthedocs.io/es/latest/esp8266wifi/readme.html>

Guirado Roda, J. F. (2014). *NODEMCU una plataforma para IOT de código abierto*. Obtenido de [descubrearduino](http://descubrearduino.com): <https://descubrearduino.com/nodemcu/>

Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, M. d. (2010). *Metodología de la Investigación*. Mexico: Mc-Graw-Hill.

Herranz, B. (2019). *Desarrollo de aplicaciones para IoT con el módulo ESP32*. Obtenido de ebuah.uah.es: https://ebuah.uah.es/dspace/bitstream/handle/10017/35420/TFG_Benito_Herranz_2019.pdf?sequence=1&isAllowed=y

Isaxan. (19 de noviembre de 2013). *edca*. Obtenido de <http://elcajondeardu.blogspot.com/2013/11/>

Julian, P. P. (2014). *Definición de RGB*. Obtenido de (<https://definicion.de/rgb/>)

Julian, P. P. (2018). *Definición de led*. Obtenido de (<https://definicion.de/led/>)

Lema11, L. (3 de abril de 2016). *Historia de arduino resumido*. Obtenido de es.slideshare.net: <https://es.slideshare.net/luislema11/historia-del-arduino-resumido>

Macho, J. C. (2020). *GUÍA A LA PROGRAMACIÓN DE ARDUINO III – LIBRERÍAS*. Obtenido de Prometec: www.prometec.net

Martín, G. (2019). *6 proyectos con ESP8266 orientados a la tecnología del IoT*. Obtenido de programarfacil.com: <https://programarfacil.com/esp8266/proyectos-con-esp8266-iot/>

Massimo Banzi, M. S. (2009). *Introducción a Arduino* (2016 ed.). Estados Unidos: O'Reilly. Obtenido de makezine.com

No, D. (30 de agosto de 2016). *ESPloradores*. Obtenido de comparacion de placas nodemcu: https://www.esploradores.com/comparacion-de-placas-nodemcu_/

Oliveira, G. (2011). *Fritzing Primeros Pasos*. Venezuela.

Proyecto. (2020). *Detector de Gases*. Obtenido de Proyectos con Arduino: <http://proyectosconarduino.com/proyectos/detector-de-gases/>

proyecto. (2020). *sensor de lluvia*. Obtenido de proyectosconarduino: <http://proyectosconarduino.com/sensores/sensor-de-lluvia/>

robótica, T. d. (2010). *Guía básica de Arduino*. Bogotá.

Torrente, O. (2013). *Arduino. Curso práctico de formación* (Primera ed.). México: Alfaomega Grupo Editor, S.A de C.V.

Torrento Peromingo, L. (07 de julio de 2014). *Tipos de placa Arduino*. Obtenido de nosinmiarduino: <http://nosinmiarduino.blogspot.com/2014/07/tipos-de-placas-arduino.html>

Ubach Granados, C. (2017). *Programando directamente un ESP8266. Tesis de Ingeniería (Industrial)*. Universidad Técnica Superior de Ingeniería Industrial, Barcelona.

Yuan, M. (12 de junio de 2017). *Conozca NodeMCU y su placa DEVKIT*. Obtenido de ibm: www.ibm.com

Anexos

Anexo 1. Glosario

Fehaciente: Que prueba o da fe de algo de forma indudable

Tecnología: Es la ciencia aplicada a la resolución de problemas concretos. Constituye un conjunto de conocimientos científicamente ordenados, que permiten diseñar o crear bienes o servicios que facilitan la adaptación al medio ambiente y de las necesidades esenciales y los deseos de la humanidad.

Anteproyecto: Es la primera versión que se desarrolla de un programa o de una norma, antes de que sea presentado en búsqueda de la aprobación o de la puesta en marcha.

Instrumentos de recolección de datos: Se refiere a cualquier tipo de recurso que utiliza el investigador; para allegarse de información y datos relacionados con el tema de estudio. Por medio de estos instrumentos, el investigador obtiene información sintetizada que podrá utilizar e interpretar en armonía con el Marco Teórico.

Guía: Una guía es algo que rige u orienta. Una guía puede ser el documento que incluye los principios o procedimientos para encausar una cosa o el listado con información que se refiere a un asunto específico.

Práctica: Es la acción que se desarrolla con la aplicación de ciertos conocimientos.

Manual: Se denomina manual o toda guía de instrucciones que sirve para el uso de un dispositivo. La corrección de problemas o el establecimiento de procedimientos de trabajo.

ESP8266: Es un chip de bajo costo Wifi con una pila tcp/ip completa y un microcontrolador, fabricado por Espressif.

Placa Arduino: Es una plataforma de desarrollo basado en una placa electrónica de hardware libre que incorpora un microcontrolador reprogramable y una serie de pines hembra, los que

permiten establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores de una manera muy sencilla.

Placa NodeMCU: Es la placa de desarrollo basado en el ESP8266 totalmente abierta, a nivel de software y de hardware.

Anexo2. Modelo de entrevista para Institución



UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA

Institución: _____

Entrevistador: _____

Entrevistado _____

Fecha: _____ Hora: _____

¿Su institución implementa proyectos utilizando la tecnología Arduino o NodeMCU en alguna área?

Si su respuesta es sí pase la a la siguiente pregunta, en caso que no, gracias por su tiempo.

¿Si su institución implementa proyectos, con qué equipo cuenta su laboratorio para realizan las prácticas con estas tecnologías?

¿Las prácticas son realizadas por medio de guías o los alumnos implementan un proyecto con la plataforma Arduino y NodeMCU?

guías _____ proyecto _____ ambos _____

¿Qué tipo de prácticas realizan, es decir que tipos de dispositivos involucran en el proceso?

¿Qué tipos de proyectos se han desarrollado con Arduino y NodeMCU?

Anexo 3. Formato de cuestionario para entrevista sector docente



UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA

Entrevistador: _____

Entrevistado _____

Fecha: _____ Hora: _____

ENTREVISTA

¿Ha implementado ud en su materia nuevas tecnologías como Arduino y NodeMCU?

SI_____ NO_____

¿Si su respuesta es si: de qué forma lo ha implementado?

¿Cuál es su proyección a futuro de acuerdo con la tecnología Arduino y NodeMCU?

¿Estaría de acuerdo en ser parte de un mundo innovador en el área de software libre, que se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software de acuerdo a las necesidades que se le presenten?

SI_____ NO_____

¿Estaría en la disponibilidad de conocer la funcionalidad e importancia de la plataforma Arduino y NodeMCU, sabiendo que sirve para facilitar el uso de la electrónica en proyectos multidisciplinares?

SI_____ NO_____

¿Cuáles son las herramientas que utiliza como docente específicamente para la enseñanza de la programación de los Arduino y NodeMCU?

Computadora_____ Arduino IDE _____ Placa Arduino _____

Elementos electrónicos _____ Herramientas para electrónica _____

¿Cuenta con recursos para desarrollar prácticas con los estudiantes, como componentes electrónicos: resistencias, motores, Leds, cooler y diodos?

SI_____ NO_____

¿Considera necesario un manual de usuario y un manual de guías prácticas de Arduino y NodeMCU para el manejo de la tecnología y recursos electrónicos?

SI_____ NO_____

¿Considera necesario un entrenador digital para desarrollar guías prácticas con Arduino y NodeMCU?

SI_____ NO_____

Anexo 4. Formato de cuestionario para estudiantes



UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA

Encuestador: _____

Fecha: _____ Hora: _____

Estimado estudiante:

La siguiente encuesta tiene como propósito principal recolectar valiosa información acerca del conocimiento que los estudiantes de la carrera de Ingeniería de Sistemas Informáticos poseen de la plataforma Arduino y NodeMCU.

Indicaciones: Marque con una “X” su respuesta

1- ¿Qué ciclo está estudiando? (Coloque el número)

2- ¿Tiene conocimiento sobre Arduino?

Si _____ NO _____

3- En el caso que sí, mencione ¿qué es lo que conoce?

App motores

Sensores

Pantallas lcd

- Pulsadores
- Fotorresistencias LDR
- Condensadores
- Buzzer

4- Si no conoce nada, pase a la pregunta #10

5- ¿Puede programar los Arduino en el lenguaje C/C++?

SI _____ NO _____

6- ¿Ha realizado algún trabajo o proyecto tecnológico con Arduino?

SI _____ NO _____

7- Si su respuesta es sí, seleccione una o más opciones de estos proyectos

- Alarma de eventos con google calendar
- Bascula conectada
- Jardinera avanzada
- Controlador midi
- Lector de huellas para puerta de garaje
- Persianas controladas por la voz
- Coche controlado por Joystick
- Lector de huellas dactilares
- Comprobador de carga de pilas y baterías
- Otro

8- ¿En alguna materia de la carrera le enseñaron a programar la placa Arduino?

SI ___ NO _____

9- En cual/es

• Microprogramación

• Arquitectura de Computadoras

• Sistemas Operativos

• Otra

10- ¿Le gustaría que se desarrollasen prácticas de la plataforma Arduino en diferentes materias de su carrera?

SI_ _____ NO _____

11- ¿Considera que es importante incluir los Arduino en los programas de estudio de Ingeniería de Sistemas Informáticos?

Si ___ NO _____

12- ¿Qué proyecto propondría para que fuese implementado con arduino?

13- ¿Conoce sobre la tecnología NodeMCU?

SI_____ NO_____

14- Si su respuesta es no, pase a la pregunta # 22

15- ¿Posee conocimiento en la programación de NodeMCU con el lenguaje LUA?

SI _____ NO _____

16- ¿Ha realizado algún trabajo o proyecto tecnológico con NodeMCU?

Si _____ NO_____

17- Si su respuesta es sí, seleccione una o más opciones de estos proyectos

- Alarma Wifi con Blynk
 - Conexión de procesador ESP8266 como cliente de un servidor web
 - Modos de ahorro de energía DEEP-SLEEP
 - Control de dispositivos : encendido, apagado y/o regulación
 - Alarma por correo electrónico
 - Ecuador gráfico
 - Otro
-

18- ¿En alguna materia de la carrera de han construido proyectos con tecnología

NodeMCU?

SI___ NO___

19- Si se han construido proyectos con tecnología NodeMCU, ¿En qué materias?

- Microprogramación
- Arquitectura de Computadoras
- Sistemas Operativos
- Otra

20- ¿Tiene conocimiento en el tema de Internet de las Cosas (IoT)?

SI_____ NO_____

21- ¿Ha escuchado sobre el tema Domótica?

SI_____ NO_____

22- ¿Le gustaría que se desarrollasen prácticas de la plataforma NodeMCU?

Introducción a NodeMCU--- breve concepto

SI_ _____ NO_____

23- ¿Qué practicas propondría con la plataforma NodeMCU

Anexo 5. Análisis y tabulación de datos

Objetivo: Conocer en que ciclo se encuentra estudiando el alumno

1- ¿Qué ciclo está estudiando?

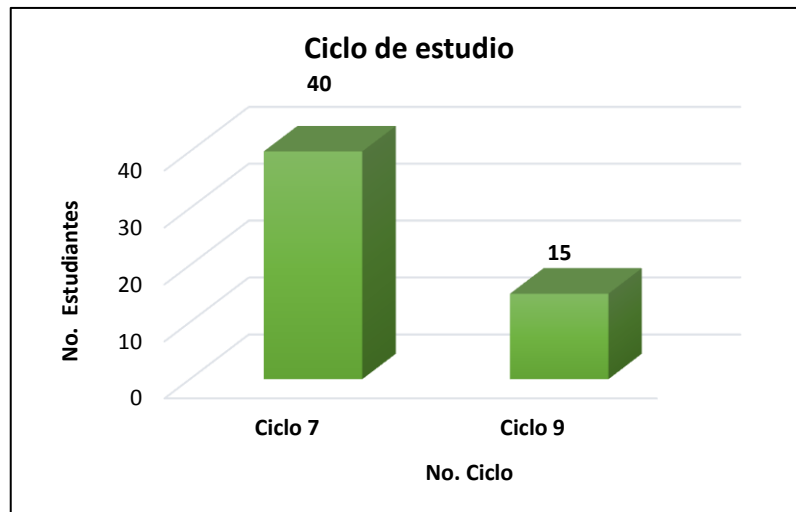


Figura A5.1 Grafico de ciclo de estudio

Se ha tomado como base para las encuestas la población de la carrera de Ingeniería de Sistema Informáticos de tercer y quinto año, porque se analizó que cada estudiante ha adquirido conocimientos de acuerdo al ciclo académico que está cursando y el plan de estudio de cada materia.

Objetivo: Percibir el nivel de conocimiento que los estudiantes de la carrera de Ingeniería de Sistemas Informáticos de la UES FMOcc poseen sobre la tecnología arduino.

2- ¿Tiene conocimiento sobre Arduino?

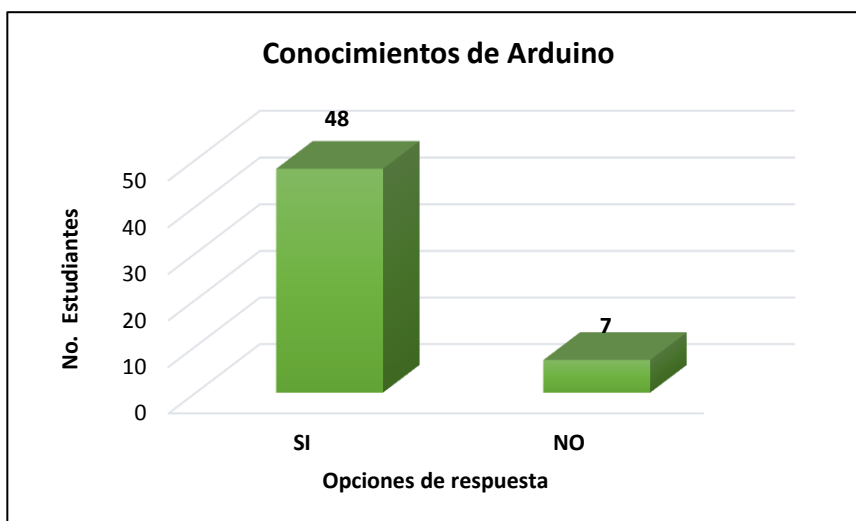


Figura A5.2 Gráfico de conocimientos de Arduino

La mayoría de estudiantes encuestados tienen conocimientos sobre la plataforma libre de arduino, pero no todos han desarrollado proyectos innovadores, por escasez de recursos en el medio en que se han enfrentado durante el periodo de estudio.

Objetivo: Averiguar que elementos de arduino conocen los estudiantes

3- En el caso que sí, mencione ¿qué es lo que conoce?

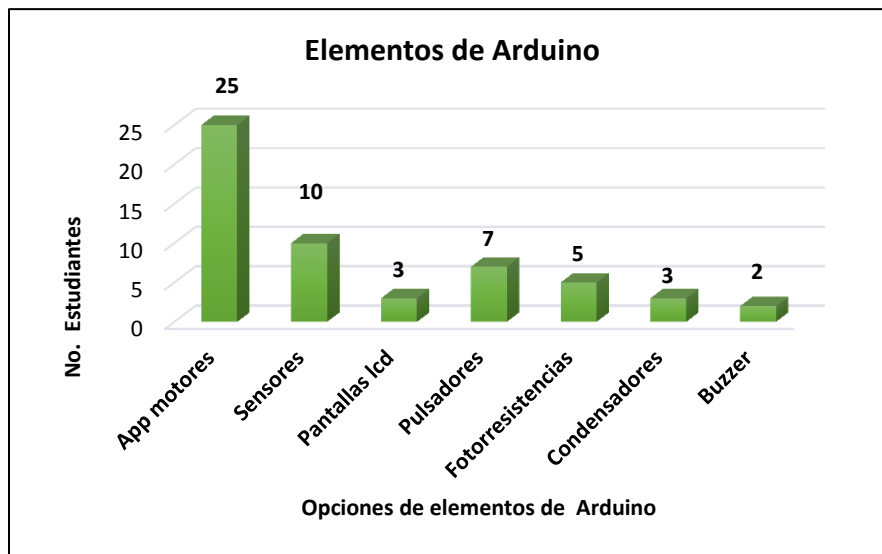


Figura A5.3 Grafico de elementos de arduino

Se ha concluido que el sector estudiantil evaluado, un porcentaje conoce de algunos elementos electrónicos y su utilidad, pero otra parte de la población no ha descubierto la funcionalidad total que estos tienen ni sus respectivas características. Y muchos estudiantes si conocen de esta nueva tecnología de Arduino que es una plataforma de software libre; pero los

conocimientos adquiridos en algunos casos han sido por sus propios medios, porque les interesó, y otros que si han sido instruidos en materias antes cursadas en la carrera de estudio.

Objetivo: Medir el nivel de conocimientos que los estudiantes poseen en la programación del lenguaje C/C++.

5- ¿Puede programar los Arduino en el lenguaje C/C++?

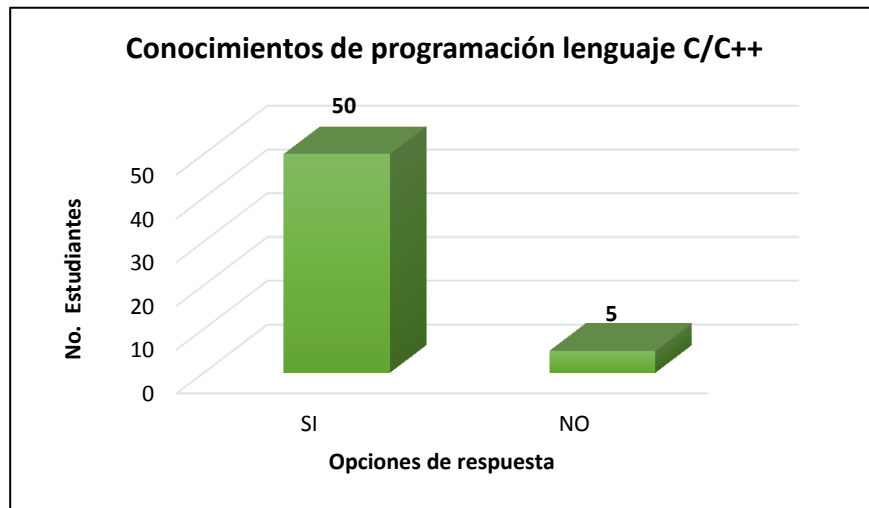


Figura A5.4 Grafico de conocimientos de programación lenguaje c/c++

Los mayoría de estudiantes que han diseñado proyectos han utilizado lenguaje c/c++ y mezclando también otros tipos de lenguaje para una mejor funcionalidad y que cumplan con las expectativas por la cual se eligió crear el proyecto.

Objetivo: Averiguar si los estudiantes han realizado algún proyecto con arduino.

6- ¿Ha realizado algún trabajo o proyecto tecnológico con Arduino?

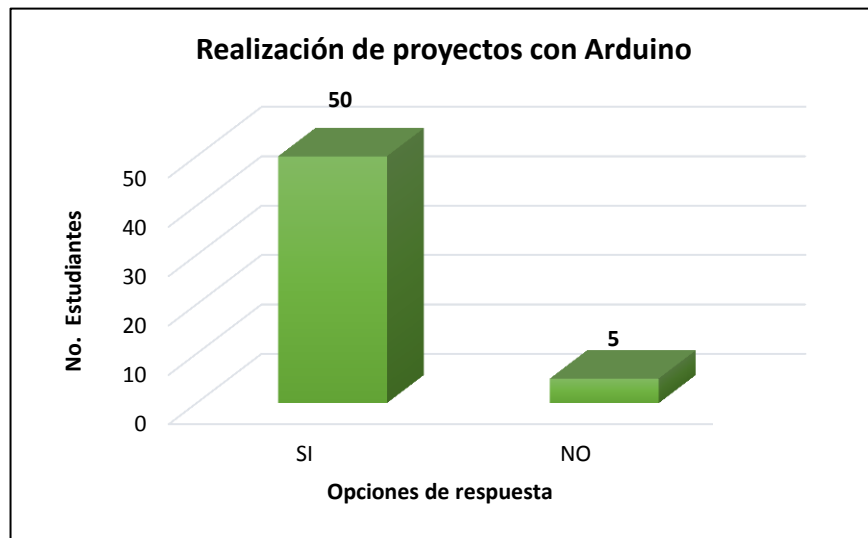


Figura A5.5 Grafico de realización de proyectos con arduino

Se determinó que la mayoría de estudiantes ya han desarrollado alguno de los trabajos tecnológicos etiquetados y, además; mencionaron otros que no se encuentran en la lista como pulseras para ciegos entre otros. Considerando también a algunos de ellos están en el campo de aprendizaje.

Objetivo: Conocer que tipos de proyectos con arduino han realizado los estudiantes con la tecnología arduino.

7- Si su respuesta es sí, seleccione una o más opciones de estos proyectos

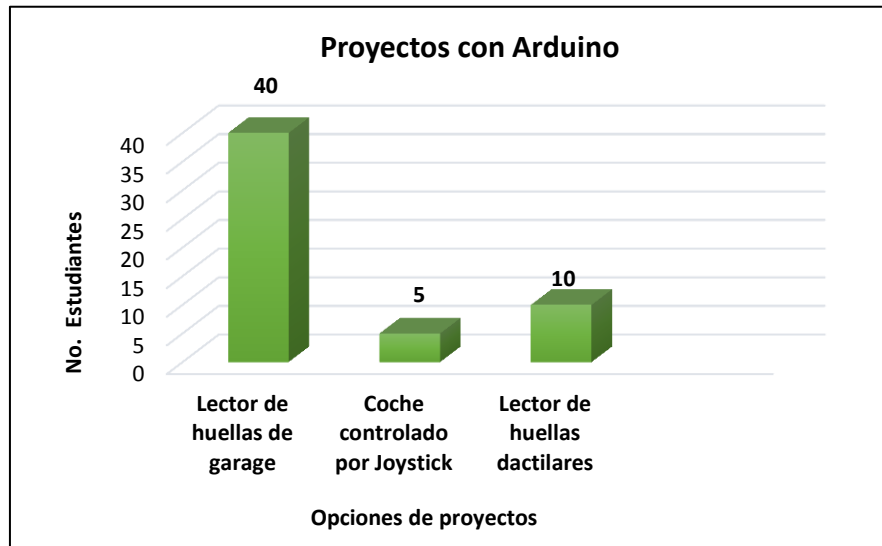


Figura A5.6 Grafico de proyectos con arduino

Se determinó que la mayoría de estudiantes ya han desarrollado alguno de los trabajos tecnológicos etiquetados y, además; mencionaron otros que no se encuentran en la lista como pulseras para ciegos entre otros. Considerando también a algunos de ellos están en el campo de aprendizaje.

Objetivo: Investigar que materias han empleado la enseñanza con la tecnología arduino.

8- ¿En alguna materia de la carrera le enseñaron a programar la placa Arduino?

En cual/es



Figura A5.7 Grafico de materias con programación Arduino

En algunas materias de tercer y quinto año de la carrera de ingeniería de sistemas informáticos se adquieren conocimientos básicos sobre como programar la placa arduino y cuál es la funcionalidad que esta tiene. En las tres materias mencionadas se imparten conocimientos básicos sobre Arduino, desde la arquitectura hasta como integrar elementos para desarrollar proyectos.

Objetivo: Medir la importancia del desarrollo de prácticas con arduino en la carrera de Ingeniería de Sistemas Informáticos de la UES FMOcc.

10- ¿Le gustaría que se desarrollasen prácticas de la plataforma Arduino en diferentes materias de su carrera?

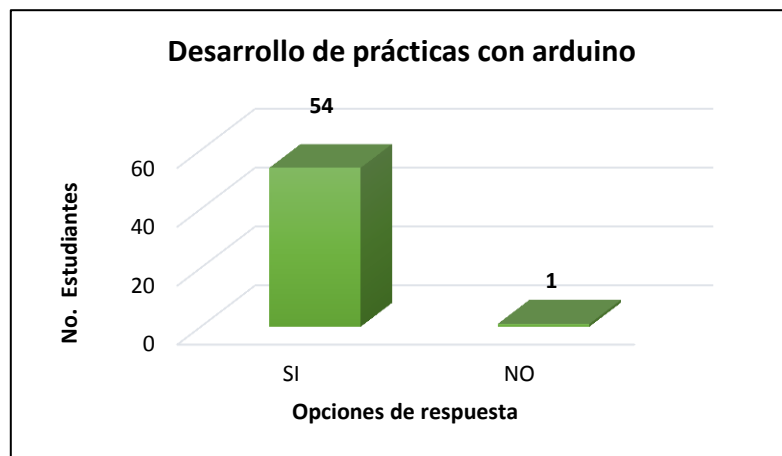


Figura A5.8 Grafico de desarrollo de prácticas con Arduino

Cada estudiante respondió de acuerdo a los proyectos que les gustaría diseñar y mostraron interés en el aprendizaje y desarrollo utilizando la placa Arduino en diferentes áreas para que se pudiesen aplicar en varias materias de la carrera de Ingeniería de Sistemas Informáticos.

Objetivo: Determinar la importancia de la inclusión de arduino en los programas de la carrera de Ingeniería de Sistemas Informáticos.

11- ¿Considera que es importante incluir los Arduino en los programas de estudio de Ingeniería de Sistemas Informáticos?

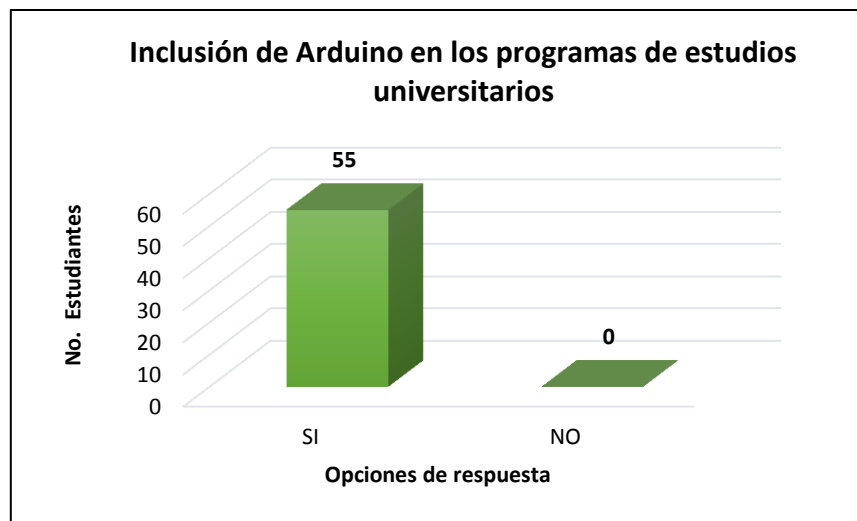


Figura A5.9 Grafico de inclusión de Arduino en estudios universitarios

La repuesta que predominó en el campo de investigación fue positiva, ya que algunos estudiantes tienen conocimientos básicos para comenzar a diseñar proyectos, por lo tanto, les parece de suma importancia que oficialmente ya este respaldado en los programas de estudio.

Objetivo: Conocer el nivel de conocimiento que los estudiantes poseen sobre la tecnología NodeMCU

13- ¿Conoce sobre la tecnología NodeMCU?

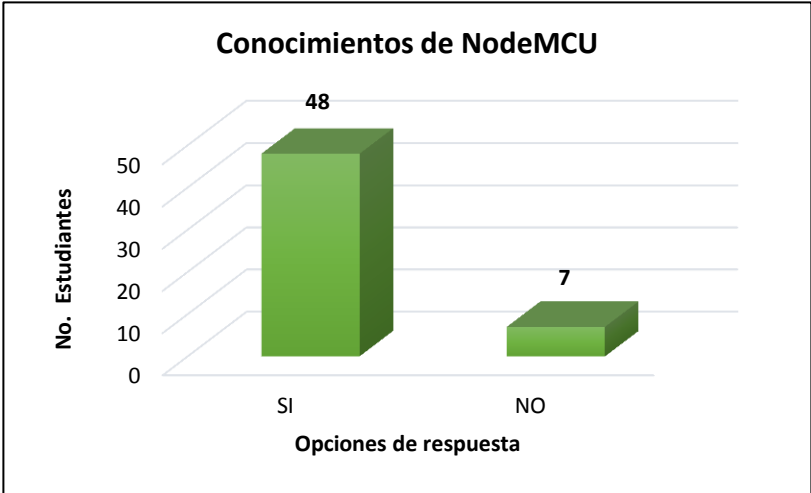


Figura A5.10 Grafico de conocimientos de NodeMCU

Definitivamente en esta pregunta las respuestas han sido equitativas, existe población que ha trabajado con esta tecnología nueva NodeMCU y han creado proyectos funcionales, pero la otra parte de encuestados solo lo han escuchado como un término cualquiera.

Objetivo: Medir el nivel de conocimiento de los estudiantes en programación con lenguaje LUA

15- Posee conocimiento en la programación de NodeMCU con el lenguaje LUA?

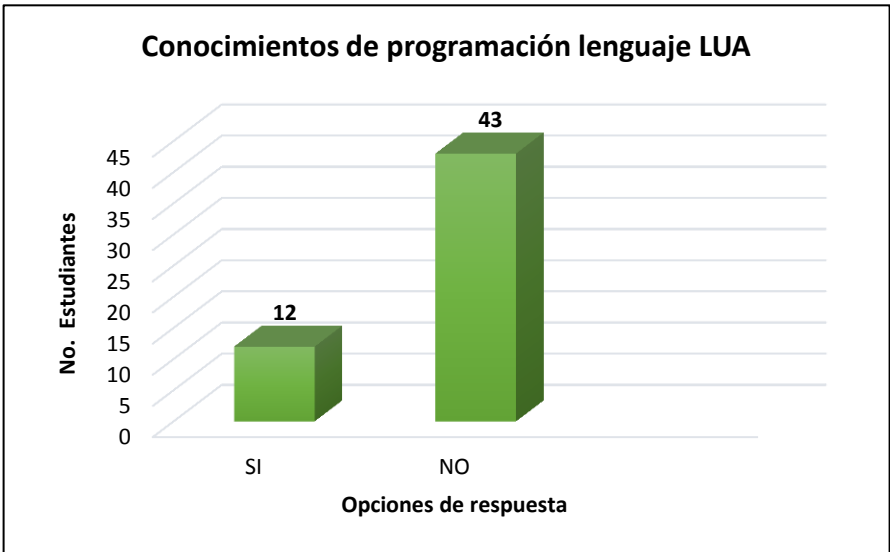


Figura A5.11 Grafico de conocimientos en programación LUA

Hay personas que han desarrollado aplicaciones para comunidades con el lenguaje LUA y que les ha parecido muy fácil de comprenderlo, mientras para la otra parte de la población estudiada siguen siendo poco conocidos.

Objetivo: Conocer la realización de trabajo o proyectos con la tecnología NodeMCU por los estudiantes.

16- ¿Ha realizado algún trabajo o proyecto tecnológico con NodeMCU?

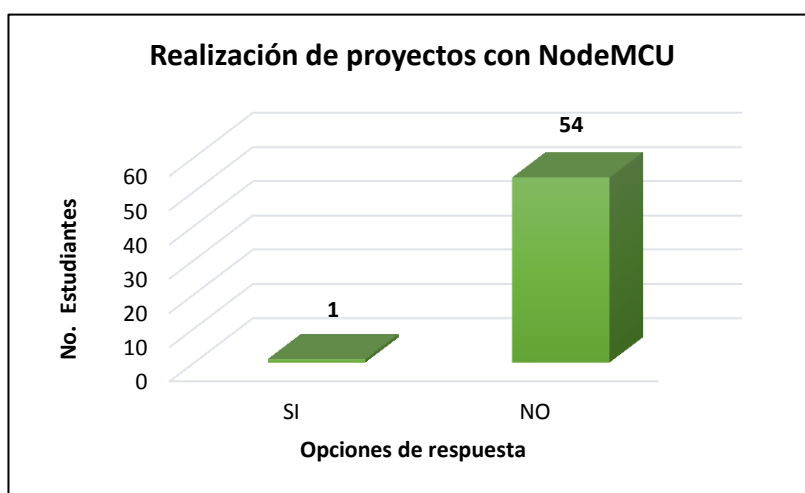


Figura A5.12 Grafico de Realización de proyectos con NodeMCU

Se determinó que la minoría de estudiantes ha realizado proyectos con la tecnología NodeMCU, por lo tanto se determinó con esta investigación que algunos estudiantes todavía están en proceso de aprendizaje.

Objetivo: Conocer que tipos de proyectos han realizado los estudiantes de Ingeniería de Sistemas Informáticos de la UES FMOcc.

17- Si su respuesta es sí, seleccione una o más opciones de estos proyectos

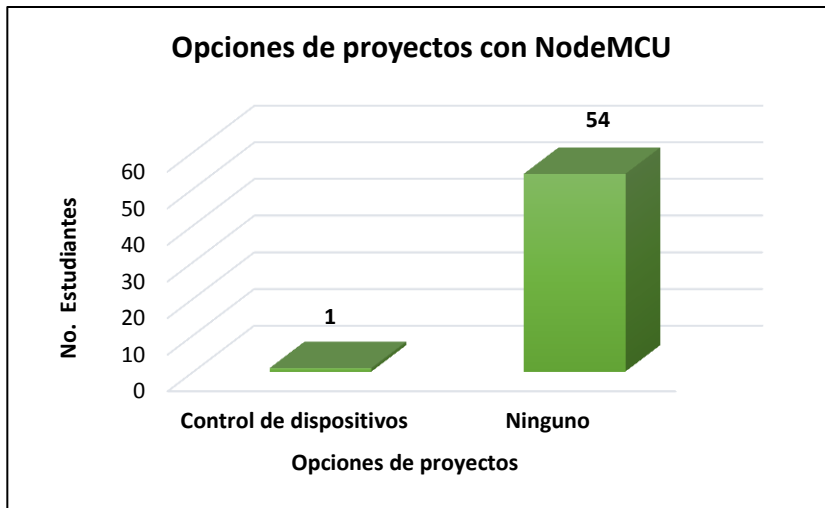


Figura A5.13 Gráfico de opciones de proyectos con NodeMCU

La mayoría de estudiantes no han realizado proyectos con NodeMCU por lo cual se diagnosticó que se requiere de esta tecnología para que los estudiantes opten por enriquecer conocimientos y de esta manera puedan comprender la importancia y utilidad de proyectos para contribuir de forma positiva a la sociedad.

Objetivo: Investigar las materias que han utilizado la tecnología NodeMCU en proyectos.

18- Si se han construido proyectos con tecnología NodeMCU, ¿En qué materias?

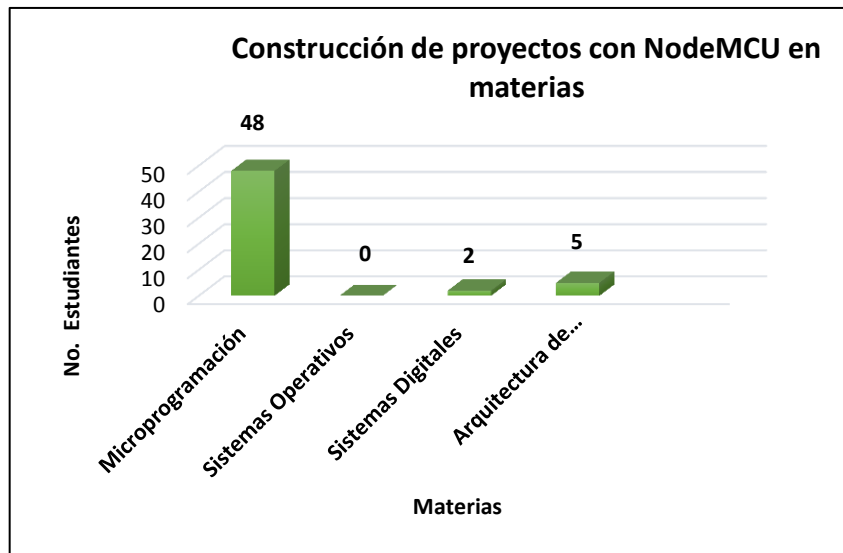


Figura A5.14 Gráfico de construcción de proyectos con NodeMCU

Se comprobó por medio de la investigación que, en algunas materias de tercer y quinto año de la carrera de Ingeniería de Sistemas Informáticos, les dejan tareas de proyectos grupales durante el ciclo y que deben ser desarrolladas en la tecnología NodeMCU.

En algunas de las tres materias antes mencionadas utilizan NodeMCU como propuesta para trabajar y diseñar proyectos utilizando los recursos disponibles en cada estudiante.

Objetivo: Indagar el nivel de conocimiento que los estudiantes tienen en IOT.

20- ¿Tiene conocimiento en el tema de Internet de las Cosas (IoT)?

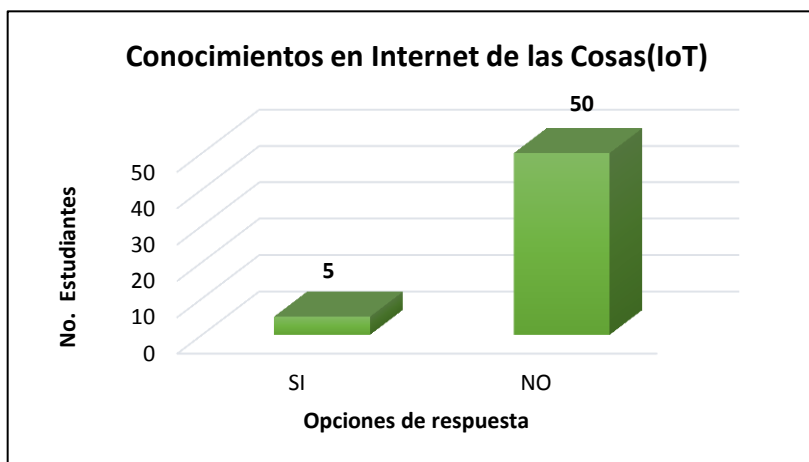


Figura A5.15 Grafico de conocimientos de IoT

Pocos estudiantes tienen conocimiento sobre este término y a que se refiere, la gran mayoría asegura que han hecho uso de ello sin darse a conocer su significado.

Objetivo: Medir el nivel de conocimiento de los estudiantes en el tema de domótica.

21- ¿Ha escuchado sobre el tema Domótica?

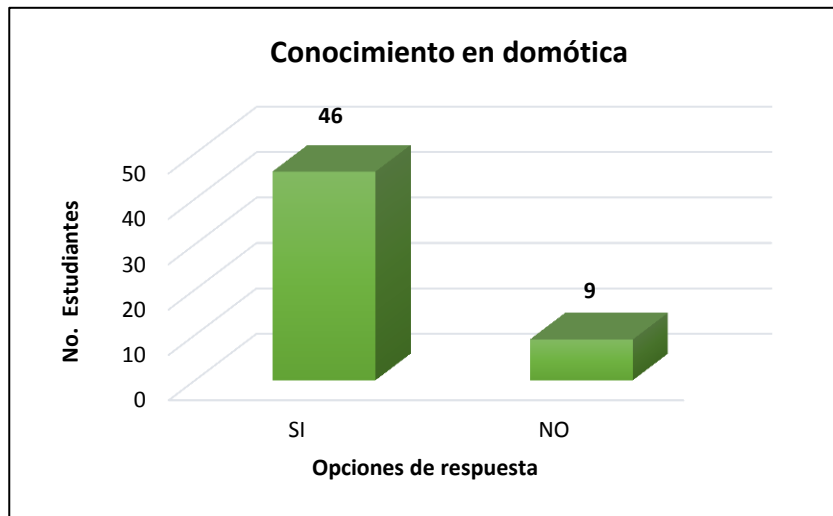


Figura A5.16 Grafico de conocimiento en domótica

Se determinó que la mayoría de estudiantes conocen sobre domótica, porque de manera práctica y de primera instancia han investigado, probado y creado algo domotizado.

Objetivo: Determinar el grado de importancia del desarrollo de prácticas con NodeMCU.

22- ¿Le

gustaría que se

desarrollasen

la plataforma

NodeMCU?

prácticas de

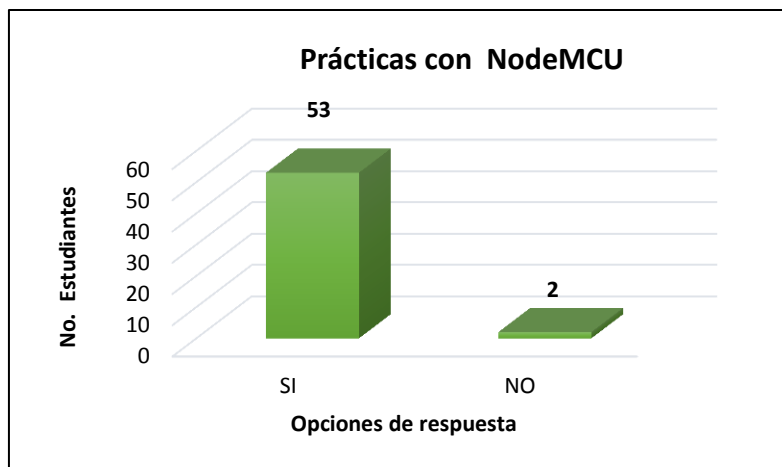


Figura A5.17 Gráfico de prácticas con NodeMCU

La mayoría de estudiantes se muestran interesados en descubrir más sobre esta tecnología por lo cual proponen que se desarrollen practicas con la tecnología NodeMCU.

Objetivo: Conocer las propuestas de proyectos que los estudiantes tienen con respecto a tecnologías como Arduino y NodeMCU.

23- Propuestas de proyectos con las tecnologías mencionadas

Proyectos propuestos por estudiantes de UES FMOcc	
Arduino	NodeMCU
UES domotizada	Seguridad de UES domotizada
Puertas con sensor de proximidad	Comandos por voz
Alarmas con sensores	Control de luces automatizada
Lector de control de estudiantes	Casa domótica

Figura A5.18 Proyectos propuestos por estudiantes de UES FMOcc

Además, se determinó que algunos estudiantes también desean diseñar algo nuevo poniendo en práctica la creatividad mediante la tecnología arduino.

Se diagnosticó que existen varias inquietudes y propuestas por los encuestados con conocimientos sobre NodeMCU como la casa domotizada, carros electrónicos para detectar productos, control de clima, entre otros

Anexo 6. Presupuesto

PRESUPUESTO PARA LA REALIZACIÓN DEL ESTUDIO				
RECURSOS	DESCRIPCIÓN DEL RECURSO	COSTO UNITARIO	FRECUENCIA	TOTAL
Viáticos	Transporte de los desarrolladores del tema en estudio al punto de reuniones.	\$10.00	Semanal (34semanas)	\$340.00
	Alimentación en caso de reuniones extensas.	\$9.00	Semanal (10semanas)	\$90.00
Datos móviles y saldo telefónico	Necesario para la comunicación entre los investigadores de tema y personas relacionadas al estudio, y búsqueda de información inmediata.	\$3.00	Semanal (20semanas)	\$60.00
Servicio de internet	Para búsqueda de información, envíos de esta y ediciones en línea.	\$3.00	Mensual (7 meses)	\$21.00
Energía eléctrica	Consumo equivalente en relación con el estudio, como en computadoras, impresoras, entre otros.	\$6.00	Mensual (7 meses)	\$42.00
TRABAJO DE GRADO				
Papelería	Incluye papel, encuadernado de trabajo de grado, lápices, lapiceros, entre otros.	-	Todo el estudio	\$30.00
Impresión	Impresión de documentos.	-	Todo el estudio	\$30.00
MATERIALES Y EQUIPO				
Materiales	Materiales y recursos para desarrollar prácticas			\$290.00
Imprevistos (10%)				\$30.00
COSTO TOTAL				\$933.00

		CRONOGRAMA DE ACTIVIDADES																																																
O	ACTIVIDADES:	PERIODO DE AGOSTO 2019-MAYO 2020																																																
		FECHA APROX		AGOSTO			SEPTIEMBRE			OCTUBRE			NOVIEMBRE			DICIEMBRE			ENERO			FEBRERO			MARZO			ABRIL			MAYO																			
		Inicio	Fin	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42					
1	Elaboración de perfil	12-08-19	17-08-19	1																																														
2	Inscripción de tesis	19-08-19	23-08-19		1																																													
3	Elaboración de anteproyecto	25-08-19	02-09-19			1	2																																											
4	Corrección de anteproyecto	09-09-19	20-09-19				1	2	3	4	5	6																																						
5	Presentación de anteproyecto	04/11/2019	08/11/2019										1																																					
6	Desarrollo del marco teórico	11/11/2019	25-11-2019											1	2																																			
7	Desarrollo de la metodología de la investigación	01/01/2020	10/01/2020																			1	2																											
8	Recopilación de información	13/01/2020	14/02/2020																					1	2	3	4	5																						
9	Procesamiento de datos	17/02/2020	24/03/2020																																															
10	Análisis de datos	31/03/2020	14/04/2020																																															
11	Elaboración de análisis	16/04/2020	30/04/2020																																															
12	Revisión	05/05/2020	10/05/2020																																															
13	Presentación preliminar de tesis	12/05/2020	16/05/2020																																															

Anexo 7. Cronograma