

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE INGENIERÍA ELÉCTRICA



**DESARROLLO DE UN SISTEMA DE CONTROL Y
MONITOREO REMOTO DE UNA ESTACIÓN DE
BOMBEO EN ZONA RURAL DE EL SALVADOR**

PRESENTADO POR:

MARVIN ALEXANDER CEREN MONTES

CARLOS ALBERTO GONZÁLEZ AGUILAR

PARA OPTAR AL TÍTULO DE:

INGENIERO ELECTRICISTA

CIUDAD UNIVERSITARIA, FEBRERO DE 2017

UNIVERSIDAD DE EL SALVADOR

RECTOR INTERINO :

LIC. JOSÉ LUIS ARGUETA ANTILLÓN

SECRETARIA GENERAL :

DRA. ANA LETICIA ZAVALA DE AMAYA

FACULTAD DE INGENIERÍA Y ARQUITECTURA

DECANO :

ING. FRANCISCO ANTONIO ALARCÓN SANDOVAL

SECRETARIO :

ING. JULIO ALBERTO PORTILLO

ESCUELA DE INGENIERÍA ELÉCTRICA

DIRECTOR :

ING. ARMANDO MARTÍNEZ CALDERÓN

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE INGENIERÍA ELÉCTRICA

Trabajo de Graduación previo a la opción al Grado de:

INGENIERO ELECTRICISTA

Título :

**DESARROLLO DE UN SISTEMA DE CONTROL Y
MONITOREO REMOTO DE UNA ESTACIÓN DE
BOMBEO EN ZONA RURAL DE EL SALVADOR**

Presentado por :

**MARVIN ALEXANDER CEREN MONTES
CARLOS ALBERTO GONZÁLEZ AGUILAR**

Trabajo de Graduación Aprobado por:

Docente Asesor :

ING. WERNER DAVID MELÉNDEZ VALLE

San Salvador, Febrero de 2017

Trabajo de Graduación Aprobado por:

Docente Asesor :

ING.WERNER DAVID MELÉNDEZ VALLE

ACTA DE CONSTANCIA DE NOTA Y DEFENSA FINAL

En esta fecha, Jueves 9 de febrero de 2017, en la Sala de Lectura de la Escuela de Ingeniería Eléctrica, a las 4:30 p.m. horas, en presencia de las siguientes autoridades de la Escuela de Ingeniería Eléctrica de la Universidad de El Salvador:

1. Ing. Armando Martínez Calderón
Director

Firma: 




2. MSc. José Wilber Calderón Urrutia
Secretario

Firma: 

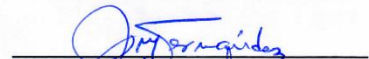
Y, con el Honorable Jurado de Evaluación integrado por las personas siguientes:

1- Ing. Werner David Meléndez Valle
(Docente-Asesor)

Firma:



2- Ing. José Miguel Hernández



3- Ing. Walter Leopoldo Zelaya Chicas



Se efectuó la defensa final reglamentaria del Trabajo de Graduación:

DESARROLLO DE UN SISTEMA DE CONTROL Y MONITOREO REMOTO DE UNA ESTACIÓN DE BOMBEO EN ZONA RURAL DE EL SALVADOR

A cargo de los Bachilleres:

- CEREN MONTES MARVIN ALEXANDER
- GONZÁLEZ AGUILAR CARLOS ALBERTO

Habiendo obtenido en el presente Trabajo una nota promedio de la defensa final: 8.0

(OCHO PUNTO CERO)

AGRADECIMIENTOS

Principalmente a Dios que me brinda la oportunidad de estar cada día con vida, él es quien me ha mostrado todo su apoyo en momentos difíciles y me ha dado fuerzas para continuar luchando por mis metas cuando ya estuve a punto de caer; por eso, deseo brindarle todos mis logros alcanzados desde lo más profundo de mi ser, infinitas gracias por ayudarme tanto.

A mi madrecita María Gricel Montes Marroquín de manera muy especial, ya que siempre ha estado brindándome su apoyo desde que comenzamos con esta lucha, gracias a ella lograba tener más valor de seguir luchando ante cualesquier adversidad además de apoyarme en lo económico y moral, fue capaz de generarme la motivación necesaria para terminar mis estudios y es de las más grandes bendiciones en esta tierra.

A mis dos hijas Gricel Alessandra y Tatiana Yamileth por gracias a ellas tengo una motivación extra para no decaer en mis estudios y lograr ser un ejemplo a seguir cada día.

A mi esposa Tatiana María que fue una de las que me dado el impulso final desde que llego a ser parte de mi vida se ha esforzado cada día dándome su apoyo en cada vez que estuve a punto de no continuar con este proyecto.

Deseo agradecer a todas las personas que en su respectivo momento me ha llevado en sus oraciones para fortalecer mi fe y con ella aumentar las ganas de luchar para alcanzar mis objetivos, metas y propósitos.

A los catedráticos de la escuela de Ingeniería Eléctrica que con paciencia me compartieron sus conocimientos.

A todos mis compañeros con los que estudiábamos juntos y compartíamos conocimientos como también a las personas que de manera directa e indirectamente apoyaron para llevar a cabo este trabajo.

Marvin Ceren.

AGRADECIMIENTOS

En primer lugar quiero agradecer a Dios padre todopoderoso por haberme permitido culminar esta etapa tan importante de mi vida que me ha dado el privilegio de convertirme en un profesional. A mi madre María Julia Aguilar Lucero que ha sido el ángel que Dios puso en mi vida y que por medio de muchos esfuerzos y sacrificios me sacó adelante, y supo guiarme de la forma correcta para formar a la persona que hoy soy. Que a pesar de mis caídas siempre me motivo a continuar y luchar, hoy con muchísimo gusto le ofrezco este triunfo que también es de ella.

A mi hermano José Alexander González Aguilar que a través de muchas cosas siempre me apoyo en las llegadas nocturnas, en las salidas a prisa, y en darme las palabras de aliento cuando el camino se tornaba difícil.

A la persona que por años ha formado parte de mi vida Aminta Beatriz. Ella ha sido la persona que siempre me motivo a sacar la mejor versión de mí, a tratar de superar mis límites y sin duda ha sido una excelente compañera a través de este viaje, de verdad gracias por estar ahí.

A mis tíos y tías que por las circunstancias de la vida muchos de ellos han sido para mí como hermanos, de verdad gracias por apoyarme cuando había que trasladarme por las madrugadas o ya entrada la noche, o incluso cuando los recursos económicos no eran suficientes. Sin duda sin su ayuda esto no hubiera sido posible. Mención especial a dos de ellos que hoy ya partieron de este mundo, mi tía Carmen que siempre tuvo palabras de aliento para mi persona y que siempre me preparaba deliciosos platillos aun si llegaba ya bastante entrada la noche. A Julio que fue como un hermano para mí, siempre me apoyo en las noches de desvelo, siempre atento si necesitaba un favor y que fue el que me escuchaba cuando las cosas personales se tornaban difíciles. A mis abuelos por apoyarme y creer en mí, a motivarme cuando a veces yo mismo dudaba de mis capacidades.

A mi grupo de amigos Luis, Cesar, Santiago, Patriz, y con mención especial a mi compañero de trabajo de graduación Marvin, sin duda compartir con ellos este camino ha sido una experiencia inolvidable, hemos reído, hemos llorado pero juntos hemos avanzado.

Finalmente, a cada una de las personas que en su momento contribuyeron en la consecución de este objetivo y que no he mencionado en estas líneas les estaré eternamente agradecido por aportar su granito de arena al proceso de formación que hoy me comprometo a dar lo mejor de mí como profesional y poner el conocimiento adquirido en función de los demás.

Sinceramente,

Carlos González.

PREFACIO

El uso de nuevas tecnologías ha venido en auge para los países en desarrollo, esto ha generado la necesidad de investigar y conocer métodos para mejorar el rendimiento de nuestras necesidades de trabajo en energía eléctrica.

Dentro de estas tecnologías, se encuentran los sistemas de automatización los cuales emplean muchos componentes electrónicos para llevar a cabo tareas establecidas. Uno de los recursos en mayor abundancia es el sistema de placas Arduino y sus derivados, que permiten utilizar diferentes métodos y tecnologías.

Los módulos de comunicación entre dichos componentes son muestra de una de las tecnologías que realiza la interacción entre cualquier sistema operativo y una placa Arduino, utilizando el principio de transferencia de bits. Ciertamente, es transformado en función de una comunicación de los módulos presentando sobre la pantalla las órdenes que se envían entre ellos, por lo que se convierte en una mayor estabilidad en transferencia de datos.

Para establecer una comunicación confiable entre ambos extremos del sistema, se hace uso del espacio radioeléctrico libre, mas específicamente de la frecuencia de los 915 MHz que es una frecuencia que puede ser utilizada para investigaciones médicas y científicas, según lo establecido por la unión internacional de telecomunicaciones (UIT) y avalado localmente por la superintendencia general de telecomunicaciones (SIGET).

El sistema consta de diferentes sensores para obtener control de las variables necesarias, para mantener el control en el centro de monitoreo de dicha planta y así puede ser posible a través de los módulos de comunicación que se pretenden implementar en el siguiente sistema de bombeo.

OBJETIVOS

GENERALES

- Diseño e implementación de un sistema de automatización, procesamiento y presentación de datos utilizando hardware y software libre.

ESPECIFICOS

- Evaluar el rendimiento del sistema de automatización y comprobación de datos, post instalación e implementación en planta de bombeo.
- Estudio del rendimiento entre los dos sistemas tanto manual como automatizado.
- Integración de Hardware y software para implementación de dicho sistema y comprobación de datos.

INDICE

Contenido

<i>CAPÍTULO I: PROPUESTA DE UN SISTEMA AUTOMATIZADO PARA MEJORAR EL FUNCIONAMIENTO DE ESTACION DE BOMBEO EN ZONA RURAL.....</i>	15
1.1 Introducción	15
1.2 Planteamiento del problema.....	17
1.3 ¿Que se pretende automatizar?	19
1.4 Diagrama en bloques de la solución propuesta	21
1.4.1 Placas de control para el arranque.....	21
1.4.2 Sensores	21
1.4.3 Relé, amplificadores de corriente y voltaje	22
1.4.4 Trasmisión RF.....	22
1.4.5 Control en zona remota.....	22
1.5 Determinación del proyecto y la manera de realizarlo	23
1.6 Características básicas de Arduino.....	24
1.6.1 Hardware	24
1.6.2 Arduino Uno	24
1.7 Software	26
1.8 Menú	26
1.9 Editor de texto.....	27
1.10 Dispositivos acoplables a Arduino	28
1.11 Sensores.....	28
1.12 Raspberry Pi [8]	29
1.13 Instalación del Sistema Operativo	32
1.14 módulos LL-RXR-27 [11]	33
1.14.1 Modos de operación.....	33
<i>2.0 CAPÍTULO II: CONSTRUCCION DEL PROTOTIPO PARA EL SISTEMA QUE SE DESEA AUTOMATIZAR TANTO EN LA ESTACIÓN DE MONITOREO COMO EN LA ESTACIÓN REMOTA.....</i>	35
2.1 HARDWARE [12]	35
2.1.1 Descripción de la interfaz del host	35
2.1.2 El flujo de mensajes.....	35
2.1.3 Iniciación por el maestro	36
2.1.4 Iniciación por el esclavo.....	36
2.1.5 Formato de los mensajes.....	37
2.1.6 Módulo de secuencia de inicio	40
2.1.7 La actualización del firmware del módulo.....	40

2.2 instalando Arduino Uno	43
Especificaciones	49
2.3 PRUEBAS DE LABORATORIO	50
2.4 Configuración e instalación de Raspberry pi3 [9].....	52
3.0 CAPÍTULO III: INTEGRACIÓN Y PRUEBA DE SUB- SISTEMAS.....	55
3.1 introducción	55
3.2 Hardware	56
3.2.1 Transformador de corriente de núcleo cortado [15].....	56
3.2.2 Fuente de voltaje 240 VAC/9 VAC	59
3.2.3 Tarjeta Rasperry PI 3	60
3.3 Software	61
3.5 Pruebas de campo	66
3.5.1 Actividades previas.....	67
3.5.2 Implementación del centro de monitoreo	68
3.5.3 implementación del sistema en sitio remoto	68
3.5.3 Aspectos importantes.....	68
Referencias generales.....	70
Recomendaciones generales.....	72
CONCLUSIONES GENERALES.....	73
Anexos	74
Anexo A: CODIGOS DE PROGRAMACION UTILIZADOS.....	74
A.1: Código LastVersion.INO.....	74
A.2: Código Fuente.py	75
A.3: Código Monitor_Center.py	76
Anexo B: Costo de materiales	78
Anexo C: Datasheet	80
Anexo c1: Datasheet de módulos RF	80
Anexo c2: Datasheet P2N2222A.....	82
Anexo c3: Datasheet relé de estado solido	84

LISTA DE FIGURAS

Figura 1.0: automatización de bomba [1]	15
Figura 1.0.1: Circuito de protección para el arranque de la bomba	18
Figura 1.0.2: Distancia de pozo de bombeo a tanque de distribución [4]	19
Figura 1.0.3: Estación de bombeo y tanque recolector	19
Figura 1.0.4: Esquema de bloques representativo para la comunicación entre las dos estaciones	21
Figura 1.1. Esquema de planta de bombeo	23
Figura 1.2: Frontal y reverso de la placa Arduino Uno [5]	24
Figura 1.3: Referencias de pines para Arduino Uno [5]	25
Figura 1.4 Configuración de Arduino en el programa [7]	26
Figura 1.5: Ventana del monitor serie	27
Figura 1.7: Logo Raspberry Pi [8]	29
Figura 1.9: Elementos Raspberry Pi. Modelo B [9]	30
Figura 1.10: Diagrama LAN9514 [10]	31
Figura 1.12: Captura programa Win32 Disk Imager	32
Figura 1.16: Modulo RF LL-RXR-27 [11]	33
Figura 1.17: Esquema de comunicación [11]	34
Figura 2.7: Descripción de componentes de la placa Arduino [13]	44
Figura 2.8: Conexión del cable USB a la placa Arduino	45
Figura 2.9: Asistente para Nuevo Hardware MS-Windows - Paso 1	45
Figura 2.10: Asistente para Nuevo Hardware MS-Windows - Paso 2	46
Figura 2.11: Asistente para Nuevo Hardware MS-Windows - Paso 3	46
Figura 2.12: Asistente para Nuevo Hardware MS-Windows - Paso 4	47
Figura 2.13: Relé de estado solido [14]	48
Figura 2.14: Transformador de corriente de núcleo partido [15]	49
Figura 2.15: Calibración de sensores y módulos	50
Figura 2.16: Comprobación de corriente RMS para relé de estado sólido	51
Figura 2.17: Comprobación del correcto funcionamiento del sistema	51
Figura 2.18: Raspberry PI3 [6]	52
Figura 2.19: Distribución de pines de Raspberry PI3 [6]	52
Figura 2.20: Sistema arrancado y funcionando	54
Figura 3.0: Circuito para la medición de corriente [7]	57
Figura 3.1: Circuito para la medición de voltaje [7]	58
Figura 3.2: Etapa de potencia del sistema [7].	59
Figura 3.2.1: fuente de voltaje 240V AC/ 9V AC	59
Figura 3.2.2: Esquema implementado para las mediciones de voltaje y corriente [7].	60
Figura 3.2.3: Líneas de código modificadas en el programa main.c	61
Figura 3.3: Líneas de código del archivo Monitor_center.py [16]	62
Figura 3.4: Líneas de código del programa Fuente.py [16]	63
Figura 3.5: Líneas de código programa LastVersion.INO	64
Figura 3.6: Página oficial de Link Labs de donde se obtiene el código para controlar los módulos [17]	64
Figura 3.7: Salida y opciones del ejecutable nomac_test [11]	65
Figura 3.9: Resultado obtenido de la simulación del enlace [19]	67
Figura 3.10: Centro de monitoreo	68
Figura 3.11: Comprobación de comunicación exitosa en sitio remoto	68
Figura 3.12: Antenas utilizadas para amplificar la señal	69
Figura 3.13: Lectura de voltaje línea a línea	69

LISTA DE TABLAS

Tabla 1.0: Características de Arduino Uno [6]	25
Tabla 1.1: Elementos Raspberry Pi 3 de modelo B+ [9]	30
Tabla 1.2: Especificaciones del hardware para módulo LL-RXR-27 [11]	33
Tabla 2.3: Características del transformador	50
Tabla 3.0: Opciones importantes del radio transmisor [18]	66
Tabla B0. Precios de Tarjeta, Módulos, sensores y Periféricos.	78
Tabla B1. Precios para amplificadores de señal.	78
Tabla B2. Precios para la instalación de equipo.	79

LISTA DE CÓDIGO

Código 1: La siguiente función calcula la suma de control utilizado por la interfaz de host para verificar la integridad del sistema y los paquetes de respuesta. _____ 39

CAPÍTULO I: PROPUESTA DE UN SISTEMA AUTOMATIZADO PARA MEJORAR EL FUNCIONAMIENTO DE ESTACION DE BOMBEO EN ZONA RURAL

1.1 Introducción

Aunque el ser humano todavía no está arraigado a las propiedades que ofrece la automatización es un hecho que, en un futuro estará instalada en cualquier lugar incluso ya lo tenemos instalado en la mayoría de nuestros hogares.

Pero ¿qué es la automatización?

Según Julián Pérez Porto denomina automatización al acto y la consecuencia de automatizar. Este verbo, por su parte, alude a hacer que determinadas acciones se vuelvan automáticas (es decir, que se desarrollen por sí solas y sin la participación directa de un individuo).

El concepto suele utilizarse en el ámbito de la industria con referencia al sistema que permite que una máquina desarrolle ciertos procesos o realice tareas sin intervención del ser humano. La automatización permite ahorrar tiempo y, muchas veces, dinero.

Los orígenes de la automatización se encuentran en la prehistoria, con el desarrollo de las máquinas simples que minimizaban la fuerza que debían hacer las personas. La energía animal o humana.

Con el tiempo, comenzó a reemplazarse por energías renovables (como la energía eólica o la energía hidráulica). La siguiente etapa en el desarrollo de la automatización consistió en:

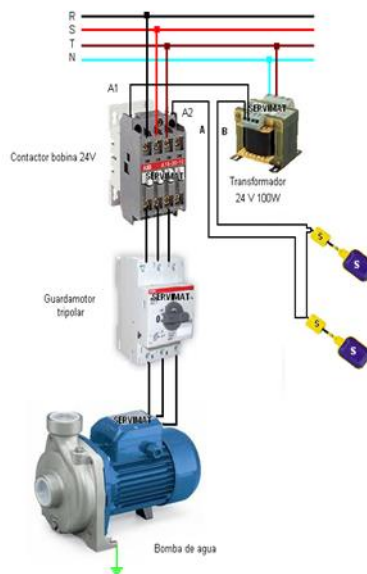


Figura 1.0: automatización de bomba [1]

El uso de la robótica y la informática, las cuales han permitido incrementar lo que se desea automatizar, en innumerables sectores industriales se utilizan máquinas que permiten estos procesos.

La automatización ofrece varias ventajas: además del ahorro de tiempo, suele favorecer la precisión en el desarrollo de tareas. Pero también se encuentran limitantes en aspecto negativo, al menos en el plano social, las máquinas que realizan estos trabajos pueden reemplazar a personas que de una u otra manera, pierden el empleo.

Es importante destacar, de todas maneras, que la automatización siempre requiere de algún tipo de control o supervisión por parte del ser humano. Además de la observación directa y del contacto físico con las máquinas, es habitual que se usen programas informáticos para esta labor [2].

Finalmente se puede definir como un conjunto de tecnologías aplicadas al control y al manejo inteligente de cualquier proceso que se esté automatizando, ya que permite una gestión eficiente del uso de la energía además de aportar seguridad, confort, y comunicación entre el usuario y el sistema. Para poder conseguir las propiedades comentadas anteriormente, es necesario que los sistemas recojan la información de su entorno con sensores y dispongan de la lógica para actuar en consecuencia utilizando actuadores.

Actualmente los sistemas automatizados tienen un precio muy alto de instalación con lo cual solo es posible verlo en empresas con grandes presupuestos. Estos suelen utilizar buses de transmisión de información que posibilitan una perspectiva robusta y diferente. Una alternativa económica consiste en la planteada en este capítulo y se basa en la utilización de placas Arduino.

En el presente proyecto utilizaremos esta tecnología que realiza las mismas funciones, pero a un precio más bajo. Esto se enfoca en una plataforma Arduino la cual se apoyará con otros dispositivos para poder construir un sistema automático simple. Arduino es una plataforma de hardware libre creada en 2005, basada en una placa con un micro controlador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

1.2 Planteamiento del problema

El interés en desarrollar este proyecto surge de las necesidades que presentan ciertas zonas rurales de nuestro país, para lograr tener el vital líquido como lo es el agua. De esta manera, se optó por colaborar en mejorar el proceso que utilizan para alcanzar abastecer sus hogares y ayudar con problema que se genera en el proceso tanto de captación como distribución de la obtención de agua en cada hogar.

De acuerdo a las investigaciones realizadas, el servicio de agua potable carece de una estabilidad porque en este lugar se encuentran bastantes limitantes, siendo estas las que provocan a la planta de bombeo un servicio ineficiente. Estas se generan tanto en la captación como en la distribución de ese vital líquido, para detallar el proceso se explicará a detalle de la siguiente manera: primeramente, se debe iniciar el encendido de la bomba para ello se debe adquirir un voltaje adecuado de la distribuidora siendo necesario una estabilidad de 240V porque la bomba no logra su arranque a menor tensión para empezar con el proceso de la captación de dicho pozo. Ya cuando se determinó esto, la bomba inicia su proceso el cual es llenar un tanque mediano ubicado a pocos metros del pozo, cabe mencionar que este suministro es de propiedad comunitaria.

Siguiendo con el proceso, luego que la bomba comienza la captación tiene que llenar el tanque mediano esto demora un tiempo aproximada de 2 horas, ya cuando este se llena completamente se tiene que apagar la bomba y abrir las válvulas que se dirigen a las tuberías del tanque de recolección, así se activa la bomba para impulsar esa agua. Es hasta entonces que el encargado de bombeo se dirige a donde está el tanque de almacenaje para verificar que se está recibiendo el agua correctamente la recolección, luego de haber hecho esto el bombero regresa a la estación de bombeo para realizar de nuevo el proceso de captación esto genera un tiempo aproximado de 1 hora en ida y vuelta esto ocurre durante dos procesos idénticos para llenar por completo el tanque recolector, la distancia de recorrido para el encargado es de aproximadamente un kilómetro y medio entre cada punto debido a que este camino es de difícil acceso, pero esto se logra en condiciones favorables como en el verano, lo cual es mucho más difícil en el invierno porque aumenta el tiempo para trasladarse de un punto a otro ya que se tiene que cruzar tres ríos que están dentro del recorrido.

El tiempo total de este proceso tiene un tiempo aproximado de cinco horas con las condiciones favorables, después con el tanque recolector lleno se distribuye el agua por gravedad esta es la manera en la que los usuarios de dicha zona reciben su abastecimiento de agua potable lo cual se logra si no hay ningún inconveniente, pero de haber alguno, el proceso es más lento.

Por ejemplo: cuando se quiebra o explota una tubería en el recorrido de la bomba al tanque de distribución se tiene que encontrar la fuga y luego llegar a detener la bomba generando más tiempo en dicho proceso por lo que implica una distribución del agua solo en unas pocas horas durante el día.

Viendo estas limitantes y con la demanda de la población se determinó que es necesario mejorar los tiempos de distribución de agua potable, de esta manera se lograría tener más horas de servicio. Para llegar a esta propuesta se tuvieron que hacer reuniones con la directiva de la comunidad y plantear nuestra alternativa que sería de mucho provecho para tener una mejor eficiencia, esto se puede llevar a cabo mediante la implementación de automatizar su sistema de captación ya que se estaría mejorando el tiempo enormemente aumentando la eficiencia y seguridad del equipo porque estarían controlados por sensores de voltaje y corriente trabajando para mejorar los arranque de la bomba, los cuales se pueden manejar desde los diferentes puntos ya sea desde el suministro como también en la distribución del agua potable. Además, se obtiene un funcionamiento eficaz de dicho proceso, dando origen a nuestra propuesta que consiste en instalar un centro de control en el lugar de captación del líquido y otro control remoto en la zona de distribución mejorando los tiempos de encendido y apagado de la bomba esto lo imprimiríamos con tecnología de bajo costo lo cual consiste en las siguientes partes:

Bomba Franklin Electric [3]

Modelo 2822019210 7.5 Hp

Voltaje 230V S: F MAX 42.1 AMP

Circuito de arranque



Figura 1.0.1: Circuito de protección para el arranque de la bomba

- ✓ Radiotransmisores a 915 MHz (Tx y Rx)
- ✓ Placa Arduino Uno Rev 3
- ✓ Circuitos de acondicionamiento de señal para corriente y voltaje
- ✓ Sensores de voltaje y corriente
- ✓ Tarjeta Raspberry pi 3
- ✓ Cable HDMI
- ✓ Monitor con entrada HDMI
- ✓ 2 breadboard
- ✓ Relé de estado sólido SDR-40
- ✓ Antenas Yagi 15 dBi
- ✓ Case para las diferentes tarjetas
- ✓ Accesorios para la instalación

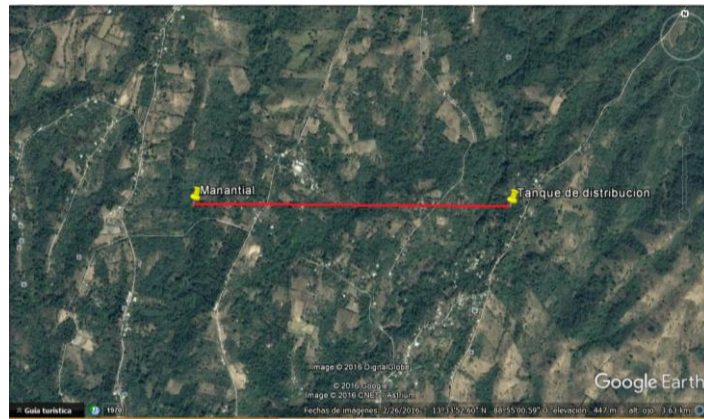


Figura 1.0.2: Distancia de pozo de bombeo a tanque de distribución [4]



Figura 1.0.3: Estación de bombeo y tanque recolector

1.3 ¿Que se pretende automatizar?

Se desea automatizar el arranque de la bomba desde un punto a otro, con el fin de mejorar el tiempo de servicio de ese suministro esto se llevara a acabo de la siguiente manera, primeramente se desea censar el nivel de voltaje y así evitar arranques bruscos del equipo estos se conectarían a

las líneas de alimentación de la bomba y al Arduino que se utiliza en el centro de control que se desee. Luego, se controlará la corriente de encendido con el relé, este se conectaría entre la línea de alimentación y el Arduino ubicados en el mismo lugar de la planta, en este proceso se necesita construir dos circuitos acondicionadores de señal uno para voltaje y el otro para corriente. Esto con el objetivo de no dañar la placa Arduino que se está utilizando. Esta placa conectada a la computadora son los que le dan la automatización del equipo y por medio de los módulos le dan la inteligencia a este circuito para poder controlarlos remotamente, dichos resultados se estarán monitoreando en los diferentes tipos de recolección de datos. A continuación, se mostrará algunas ventajas que se tendrían con el sistema ya instalado:

- Mejor control en el encendido de la bomba
- Mayor duración del equipo
- Protección en cada arranque
- Lectura de voltaje y corriente en tiempo real
- Control en el tiempo de llenado
- Aumentar el tiempo de servicio para suministrar el agua a los usuarios
- Mayor eficiencia al momento de un falla
- Mayor rendimiento en el tiempo para el encargado de bombeo
- Innovación en el servicio de agua
- Puede aumentar el número de usuarios ya que en menor tiempo podrían suministrar la demanda necesaria en las horas de mayor uso
- Sistema más ordenado dentro de las instalaciones sea en circuitos de protección como en sistema de alimentación de voltaje.

1.4 Diagrama en bloques de la solución propuesta.

ESTACION DE MONITOREO EN PLANTA DE BOMBEO ESTACION DE CONTROL REMOTAMENTE

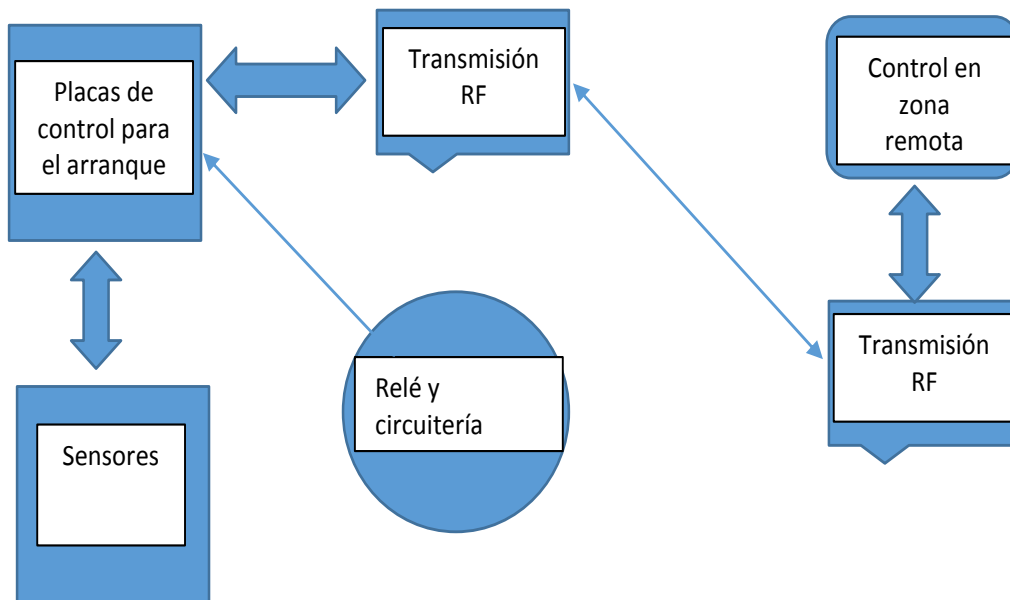


Figura 1.0.4: Esquema de bloques representativo para la comunicación entre las dos estaciones

1.4.1 Placas de control para el arranque

En esta caja de control se determina qué tipo de placas se utilizarán y se puedan acoplar al sistema. Para el caso se tiene una computadora que es la que se encarga en el control y acoplada a ella, se coloca el Arduino Uno que es el encargado en interrogar los demás componentes que luego se describirán, este bloque es el que da la inteligencia a nuestro sistema ya que por medio del software adecuado va siguiendo una serie de órdenes para lograr paso a paso la automatización que se desea.

1.4.2 Sensores

Este bloque corresponde a los sensores que se utilizan para tener un control adecuado de los datos que reflejan voltaje y corriente. Estos componentes son de vital importancia en este proceso. Para el caso, estos dos sensores necesitan ser calibrados adecuadamente ya que de la fiabilidad de los datos recolectados depende el correcto funcionamiento de todo el sistema. Estos sensores consisten en: un transformador de corriente de núcleo cortado y una fuente de alimentación AC/AC 240V/9V, los cuales van acoplados a sus respectivos circuitos de

acondicionamiento y de los circuitos a la placa Arduino que estará comunicándose con la computadora vía puerto serial para ir generando el proceso de automatización.

1.4.3 Relé, amplificadores de corriente y voltaje

Para poder tener un control exacto de cuando se activa la bomba, se necesita de un relé de excelente calidad, el cual es el encargado de dar el pase de energía necesaria para activar el control. En este sentido se realiza mediante la interrupción de una línea de voltaje y las otras salidas van directamente al circuito de encendido que a su vez se conecta al pin 13 de la placa Arduino, esto con el fin de evitar un daño irreparable en la placa ya que no soporta más de 5V y una corriente no mayor de 20 mA. Por esa razón, se utiliza un transistor BJT 2N2222A y unas resistencias. En conjunto, estos elementos ejercen la función de switch, ya que habilitan el flujo de corriente proveniente de una fuente de voltaje 120 VAC/9 VDC que suple la corriente de encendido del relé, que sometido a una tensión de 9 V tiene un valor de 80 mA, cuatro veces más la corriente máxima soportada por el Arduino.

1.4.4 Trasmisión RF

Este es el rol más significativo en la interconexión con las dos estaciones, ya que por medio de ellos se comunican los dos puntos, ahorrando al encargado de bombeo dos horas de recorrido. Este va acoplado a la computadora para la interrogación de cómo están trabajando los sensores y en base al resultado poder poner en marcha la estación de bombeo. Estos módulos tienen su propio software que se programa de acuerdo a las necesidades del usuario. Además, transmiten en una frecuencia de banda libre autorizada por la SIGET, los LL-RXR-27 operan en la banda libre de los 900 MHz y fácilmente se pueden configurar a la frecuencia específica de interés. Algo sumamente importante de mencionar, es el hecho que ambos módulos tienen que tener los mismos parámetros de operación. De esta forma, se garantiza que la comunicación se establecerá de forma exitosa.

1.4.5 Control en zona remota

Para propósitos de las pruebas se tiene que hacer uso de una computadora para el caso se utilizó una Raspberry PI3, aunque puede ser cualquier otra computadora, lo único que debe cumplir es que el sistema operativo pueda comunicarse con el software de los módulos, que para el caso están en un código ejecutable en Python. Adicionalmente, tienen que tener en buenas condiciones los puertos USB que es a través de ellos como se conecta el módulo y luego como usuario administrador o súper usuario para Linux se corre el ejecutable luego se comienza la transmisión ya con los dos módulos con los mismos parámetros y se comunican con la estación base para ello

se utilizan unas antenas yagi que aumenta la potencia de la señal tanto en la portadora como en la receptora, con todo esto ya puesto en marcha se inicia el proceso de captación en el pozo por medio de la bomba si en algún momento se decide detener el proceso se le dan las ordenes de módulo a módulo y así se empieza a utilizar la automatización en ambos sentidos.

1.5 Determinación del proyecto y la manera de realizarlo

Para crear dichos sistemas automatizados han de tenerse en cuenta varios aspectos, comenzando por conocer el capital del que se dispone para invertir en el sistema y seleccionar los dispositivos que más se ajusten tanto a las necesidades como al presupuesto. De poco serviría comprar un elemento con grandes prestaciones si luego no se va a aprovechar. Por ejemplo, en vez de utilizar como servidor del sistema una tarjeta FlexScada que tenga un precio alrededor de unos \$450.00 podríamos usar otras alternativas. Este año han aparecido alternativas muy baratas como la placa computadora Raspberry Pi3 que se puede obtener por unos \$70.00.

Luego está el factor estético que normalmente evitaría la instalación de cableado para comunicar las placas Arduino, es decir, aprovecharíamos dispositivos que trabajan inalámbricamente. Aunque el precio de un dispositivo inalámbrico es ligeramente superior, se podrá evitar la compra de cables, lo que poco a poco aumentaría el costo total haciéndolo incluso más caro. Pero también se debe saber que los elementos inalámbricos interfieren entre sí y eso por ejemplo en una zona densamente habitada en el que los vecinos también dispongan de este tipo aparatos puede reducir las prestaciones de la comunicación del sistema. Es por ello que cada caso hay que estudiarlo por separado y actuar en consecuencia.

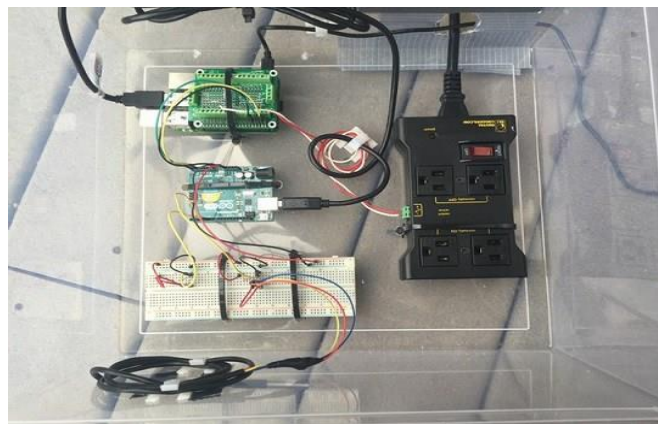


Figura 1.1. Esquema de planta de bombeo

1.6 Características básicas de Arduino.

En este apartado se describirán los principales elementos que componen una placa Arduino y el entorno de desarrollo en el que se programa el código, es decir la parte hardware y software que actúan sobre Arduino.

1.6.1 Hardware

Arduino una plataforma de hardware libre tanto su diseño como su distribución puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin haber adquirido ninguna licencia. Por eso existen varios tipos de placa oficiales, las creadas por la comunidad Arduino o las no oficiales creadas por terceros pero con características similares. En la placa Arduino es donde conectaremos los sensores, actuadores y otros elementos necesarios para comunicarnos con el sistema.

En el proyecto se han utilizado las placas Arduino Uno y Raspberry Pi3 que describiremos a continuación.

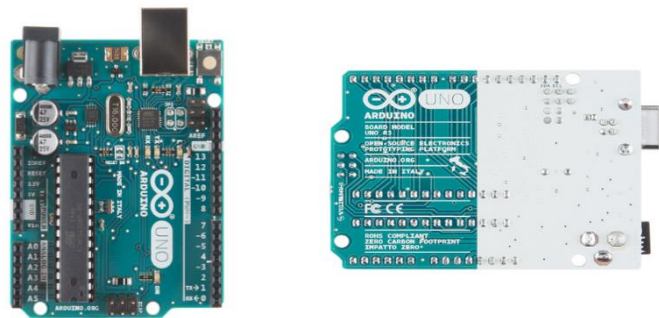


Figura 1.2: Frontal y reverso de la placa Arduino Uno [5]

1.6.2 Arduino Uno

Es un modelo diseñado y distribuido por la comunidad Arduino. La placa tiene un tamaño de 75x53mm. Su unidad de procesamiento consiste en un micro-controlador ATmega328. Puede ser alimentada mediante USB o alimentación externa y contiene pines tanto analógicos como digitales.

La tabla siguiente resume sus componentes y donde están ubicados los elementos más importantes que componen la placa Arduino Uno:

Micro-controlador	ATmega328
Voltaje operativo	5V
Voltaje de	7-12V

entrada(recomendado)	
Voltaje de entrada (limites)	6-20V
Pines digitales E/S	14 (de los cuales 6 proporcionan salida PWM)
Pines de entrada analógica	6
Corriente continua para pines E/S	40 mA
Corriente continua para pines de	
3.3V	50 mA
Memoria Flash	32 KB (ATmega328) de los cuales 0.5 KB son para el Bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Velocidad del reloj	16 MHz

Tabla 1.0: Características de Arduino Uno [6]

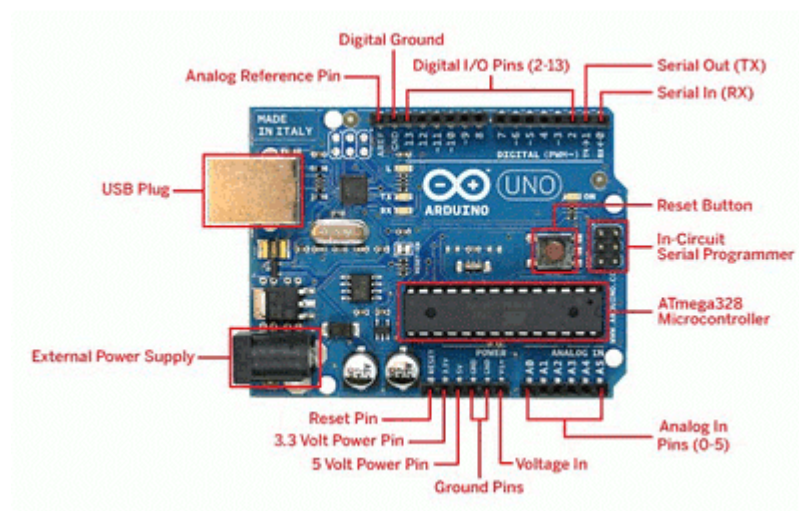


Figura 1.3: Referencias de pines para Arduino Uno [5]

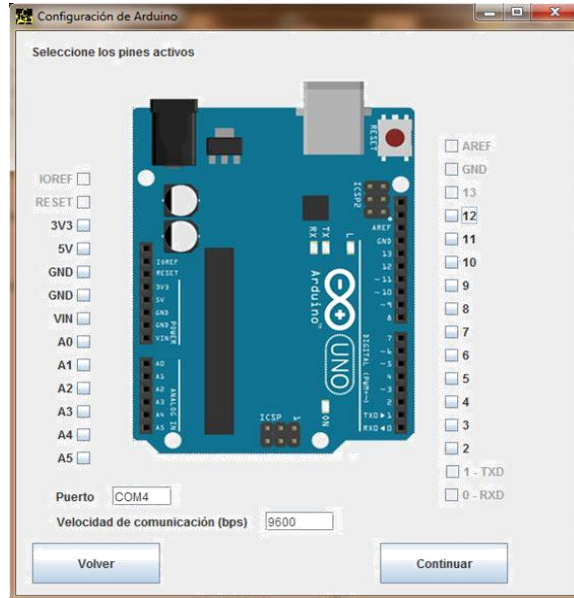


Figura 1.4 Configuración de Arduino en el programa [7]

1.7 Software

La plataforma Arduino tiene un lenguaje propio que está basado en C/C++ y por ello soporta las funciones del estándar C y algunas de C++. Sin embargo, es posible utilizar otros lenguajes de programación y aplicaciones populares en Arduino como Java, Processing, Python, Mathematica, Matlab, Perl, Visual Basic, etc. Esto es posible debido a que Arduino se comunica mediante la transmisión de datos en formato serie que es algo que la mayoría de los lenguajes anteriormente citados soportan. Es bastante interesante tener la posibilidad de interactuar con Arduino mediante esta gran variedad de sistemas y lenguajes ya que dependiendo de cuales sean las necesidades del problema que vamos a resolver podremos aprovecharnos de la gran compatibilidad de comunicación que ofrece.

La parte más importante se encuentra en Herramientas. Desde aquí podremos configurar el programa para que pueda comunicarse con la placa Arduino.

1.8 Menú

Pasando el ratón por Tarjeta aparecerá una lista con los tipos de placa Arduino que el programa comprende. Aquí seleccionaremos Arduino Uno (para la placa que deseamos) dependiendo de con cual estemos trabajando.

En el campo Puerto Serial seleccionaremos el que corresponda a nuestra placa que conectaremos mediante USB. Si utilizamos Windows el puerto tendrá un nombre del estilo COM pero en Linux será /dev/ttyUSBx donde x es un número. En caso de que aparezcan varios puertos serie y no sepamos cual es el de nuestra placa procederemos a desconectarla, anotamos los puertos que aparecen, reconectamos la placa y volvemos a mirar la lista de puertos. El nuevo puerto que haya aparecido será el de nuestra placa.



Figura 1.5: Ventana del monitor serie

1.9 Editor de texto

En esta área escribiremos la implementación (denominada por el programa sketch) para poder cargar en la placa Arduino los códigos que necesitamos. El programa tiene 3 partes.

La primera consiste en la inclusión de librerías y la declaración de constantes o variables globales que se podrán utilizar en cualquier función del programa.

La segunda es el método setup () que es el encargado de inicializar los dispositivos conectados a la placa y será ejecutado solo al iniciar el sistema.

La tercera parte consiste en el método loop () que ejecutará su código continuamente, es decir, en modo bucle. Aquí es donde se escribirá la lógica de la placa Arduino. Como el lenguaje es muy similar a C es posible crear otros métodos para separar bloques funcionales y dejar ordenado el programa.

1.10 Dispositivos acoplables a Arduino

Para conseguir las características de un sistema automático es necesario que además del órgano central que controle el sistema tengamos a disposición sensores que puedan recoger datos sobre la situación que deseemos trabajar en el control. Dependiendo de estos datos el sistema automático debe ser capaz de comunicarse con los actuadores para mejorar la situación de lo que se desea transmitir. También deben existir elementos con los que el usuario pueda comunicarse con el sistema y pueda hacer los cambios oportunos manualmente.

Los dispositivos estarán conectados mediante cables o directamente acoplados a la placa Arduino. Algunos de ellos disponen de librerías que deberemos adjuntar al programa para poder usar las utilidades que contengan. Para ello añadiremos la carpeta de la librería en la carpeta libraries del entorno de desarrollo de Arduino. Al principio del código del sketch incluiremos la librería con la línea:

```
#include <nombreLibreria.h>
```

Para utilizar los métodos de sensores y actuadores digitales debemos tener en cuenta que solo tenemos dos posibles valores, HIGH representa el nivel alto y LOW el nivel bajo.

En el caso de los analógicos su uso es levemente más complejo pero también más configurable ya que tiene que leerse o escribir un voltaje de 0 a 5 voltios que se representa en 10 bits (lectura) o en 8 bits (escritura), es decir la tensión puede tener 1024 (lectura) o 256 (escritura) valores distintos. Más adelante se hará un ejemplo con un sensor.

1.11 Sensores

Un sensor es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. Las variables de instrumentación pueden ser por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, movimiento, pH, etc.

Los sensores siempre que estén activados estarán tomando continuamente la situación actual de los parámetros cambiantes de la bomba y es el servidor o la placa Arduino quien leerá esta información y decidirá cómo actuar.

1.12 Raspberry Pi [8]

Raspberry Pi es un ordenador de placa reducida (SBC) de bajo coste, que se podría considerar como un ordenador de muy pequeño tamaño, comparable con el de una tarjeta de crédito, este fue desarrollado en Reino Unido por la fundación Raspberry Pi, con el objetivo principal de incitar tanto a niños en sus colegios como adultos a que aprendan sobre ordenadores y todo lo relacionado con ellos.

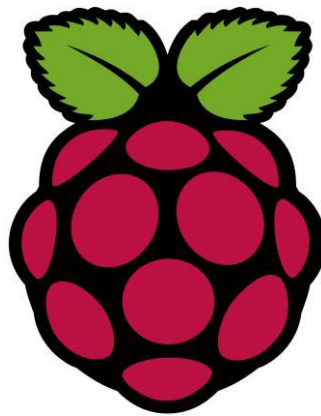


Figura 1.7: Logo Raspberry Pi [8]

La idea es tener una placa a la que poder conectar monitor, ratón y teclado y ayudar a personas de todas las edades a adentrarse en el mundo de la computación y la programación.

Esto surgió en 2006 cuando Eben Upton, Rob Mullins, Jack Lang and Alan Mycroft del laboratorio de informática de la Universidad de Cambridge empezaron a ver cómo había cambiado los conocimientos de los niños sobre la informática. En la década de 1990 la mayoría de los niños tenían mucha experiencia como programadores aficionados, en cambio en la década del 2000 solo eran capaces de realizar diseño web.

Para el proyecto de interés se utiliza una Raspberry PI3 con el objetivo de simular un ordenador muy completo además necesitamos acoplar los demás dispositivos que conforman el centro de control en la estación de bombeo.

Hoy en día la Raspberry Pi presenta tres modelos, los modelos A+ y B+, que están basados en sus predecesores A y B, y la nueva Raspberry Pi 3 modelo B. Las principales diferencias entre los

modelos antiguos A y B y sus versiones más recientes A+ y B+ respectivamente están en el almacenamiento, pasando de ser SD a microSD y en el caso del modelo A una reducción de tamaño y por tanto de peso. Otra diferencia del modelo B es el aumento de 2 a 4 puertos USB. En la siguiente tabla se puede observar la diferencia entre las tres versiones que se comercializan actualmente.

Dado que las demás características son similares y el modelo usado en este trabajo ha sido el modelo B, a continuación se va a pasar a detallar los componentes existentes en esta versión.



Figura 1.9: Elementos Raspberry Pi. Modelo B [9]

Nº	Identificador
1	Salida puertos USB 2.0
2	Pines GPIO
3	Conector DSI Display Module
4	Conector CSI Camera Module
5	Ranura para la tarjeta microSD
6	Puerto Ethernet (RJ45)
7	Alimentación micro USB
8	Leds indicadores
9	Salida de audio
10	Salida HDMI

Tabla 1.1: Elementos Raspberry Pi 3 de modelo B+ [9]

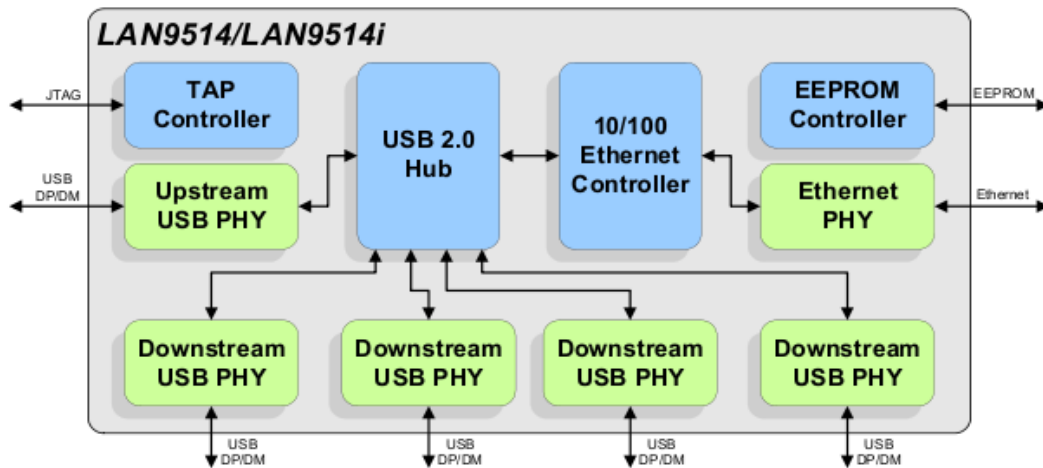


Figura 1.10: Diagrama LAN9514 [10]

La Raspberry Pi está diseñada para ejecutar el sistema operativo GNU/Linux de código abierto. Raspbian OS es la distribución por excelencia para la Raspberry Pi. Es la más completa y optimizada de las existentes, optimizando el código de ésta para la Raspberry Pi.

La distribución permite moverse ágilmente en el hardware de la Raspberry Pi, con un entorno de escritorio LXDE y Midori como navegador web predeterminado. Además incluye herramientas de desarrollo muy interesantes, como IDLE para Python, Scratch para programar video juegos.

RISC OS, Arch Linux es otro de los grandes nombres en cuanto a distribuciones Linux. Se caracteriza por su simplicidad, elegancia, coherencia del código y minimalismo. Pero la simplicidad no quiere decir facilidad de uso, ya que Arch Linux es bastante conocida por ser poco amigable y recomendable solo para gente con conocimientos más elevados. Ahora Arch Linux soporta ARM y por tanto también puede instalarse en la Raspberry Pi.

Pidora es básicamente una distribución Linux Fedora especialmente optimizada para funcionar en ARM. Por el resto de características es similar a Fedora, la hermana pequeña de Red Hat, y mantenida por los mismos desarrolladores de esta comunidad libre.

OpenELEC es otro de los sistemas operativos oficiales de la Raspberry Pi y por tanto se incluye en NOOBS. Se trata de una distribución Linux especialmente pensada para crear un centro multimedia barato con la Raspberry Pi. Con él se puede disponer de todo el contenido multimedia y acceso a Internet para transformar una TV en una smartTV.

Para ello, OpenELEC incluye paquetes de códec de audio y vídeo, drivers, y se basa en el famoso Kodi (anteriormente conocido como XBMC, siglas de Xbox Media Center). Kodi es un centro multimedia que fue creado en un inicio para la videoconsola Xbox, pero el desarrollo hizo que se portara a otras plataformas. Se completa con reproductores de audio, vídeo, presentación de diapositivas, visores de imágenes, reportes de clima, y otras funciones implementadas mediante plug-ins.

1.13 Instalación del Sistema Operativo

Dado que se va a trabajar con Python el sistema operativo que se va a instalar en este caso es Raspbian.

Para poder instalar la distribución elegida es necesario usar un ordenador con lector de tarjetas SD. Usando un ordenador con Windows, el primer paso que hay que realizar es obtener el sistema operativo del sitio web autorizado

Posteriormente, se necesita descargar el software gratuito "Win32 Disk Imager". En Image File se escoge el fichero que contenga la imagen del Sistema Operativo, en Device hay que asegurarse que se elige la unidad correcta asociada a la tarjeta SD y mediante el botón Write se pasa el sistema a la tarjeta.

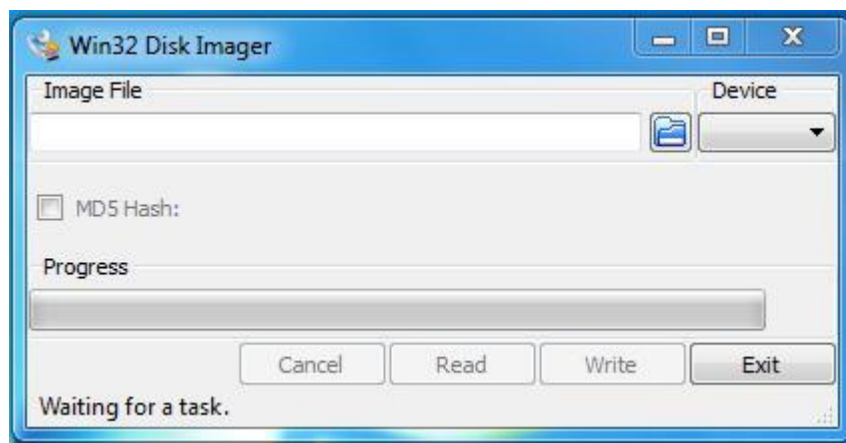


Figura 1.12: Captura programa Win32 Disk Imager

Una vez finalizado este proceso, ya se puede insertar la tarjeta en Raspberry Pi y comenzar a trabajar.

1.14 módulos LL-RXR-27 [11]

Para la conexión entre módulo LL-RXR-27 y las tarjetas Arduino se utilizan divisores de tensión para la protección del módulo esto se realiza con los dos voltajes de alimentación que permite el módulo LL-RXR-27 de 3.3v y 5V respectivamente.



Figura 1.16: Modulo RF LL-RXR-27 [11]

Especificaciones de hardware LL-RXR-27

Bandas de frecuencia	902 a 928 MHz ISM (US); 868 MHz ETSI (UE)
Tasa de datos en el aire	183 bps a 37,5 kbps
Potencia de salida	Programable: 26 dBm a 12 dBm en pasos de 0.6 dB
Tensión de alimentación	3,5 V a 5,5 V
VCC	3.3 V
Regulador a bordo	V IN = 3,5 - 5,5 V V OUT = 3.3 V Corrientes máx. = 150 mA I q = 1 μ A PSRR = 50 dB

Tabla 1.2: Especificaciones del hardware para módulo LL-RXR-27 [11]

1.14.1 Modos de operación

El módulo LL-RXR-27 está diseñado para ser manejado por un micro-controlador externo. Ya que ha desarrollado una interfaz de host para operar el módulo en el modo de enlace sinfonía o el modo NOMAC. Estos modos se describen a continuación. En este capítulo se pretende mostrar los sensores y periféricos utilizados, la forma de funcionamiento sus características y cálculos matemáticos previos para la implementación del software que cada uno de los dispositivos

requiere para el funcionamiento y en algunos casos presentar datos obtenidos experimentalmente para compáralos con datos teóricos.

La combinación del módulo LL-RXR-27 con el modulo sensor de corriente 120Amp, no se hubiese realizado sin el modulo LL-RXR-27 que es el encargado en realizar el trabajo de enviar vía RF los datos de censar voltaje y corriente DC de una sola vez, para esto estaba previsto este módulo en el planteamiento inicial del proyecto. Esta combinación surge por los resultados obtenidos de corriente mediante el módulo RF, no son estables y ya que el parámetro de corriente de cortocircuito es de mucha importancia para el análisis de los dos sistemas automatizados se requiere un sensor que retorne datos de corriente estables y lo más exacto posible como el utilizado

Modo Symphony Link



Figura 1.17: Esquema de comunicación [11]

Este es un modo de transmisión propietario de Link Labs. Se implementa una red en estrella sobre la base de un sistema Alojamiento ranurado. Un enlace utilizando el modo Symphony se representa en la figura, arriba.

En el modo Symphony, el módulo de LL-RXR-27 participa como criterio de valoración con las redes Sinfónica de enlace compatible. El módulo se conecta automáticamente a una pasarela Sinfónica de Enlace en rango. La puerta de enlace transmite mensajes de enlace ascendente desde el módulo al conductor, la plataforma de servicios de datos basado en la nube Enlace Labs. Del mismo modo, los mensajes de los relés de puerta de enlace descendente enviados desde conductor a dirigirse individualmente módulos.

En el modo de sinfonía Enlace, el módulo se controla mediante un pequeño conjunto de funciones de alto nivel. Los detalles del funcionamiento de la red - timing, selección de frecuencia, velocidad de datos, la potencia de transmisión, encriptación, etc. - son controlados automáticamente por el apretón de manos entre el módulo y la puerta de entrada para optimizar la capacidad de la red, el alcance y el consumo de energía.

Modo NOMAC

NOMAC modo desactiva el firmware de control de acceso al medio (MAC) y permite que el sistema host para emitir comandos de bajo nivel. Modo NOMAC da el control directo del sistema de acogida de los parámetros de radio del módulo, sin ningún tipo de control automático de la red. NOMAC es útil para la investigación y el desarrollo de la red mientras que el diseño o experimentar con nuevos protocolos MAC.

2.0 CAPÍTULO II: CONSTRUCCION DEL PROTOTIPO PARA EL SISTEMA QUE SE DESEA AUTOMATIZAR TANTO EN LA ESTACIÓN DE MONITOREO COMO EN LA ESTACIÓN REMOTA

A continuación, se detalla de forma específica la manera en la cual serán configurados e instalados cada uno de los componentes necesarios para lograr una automatización de la planta de bombeo, por esa razón se brindaran todos los detalles de cada proceso.

2.1 HARDWARE [12]

- ✓ Módulos radiotransmisores LL-RXR-27

2.1.1 Descripción de la interfaz del host

El módulo de LL-RXR-27 está diseñado para ser controlado por un micro-controlador externo (MCU) de la elección del usuario. Este MCU se conoce como el host externo y se comunica con el módulo a través de un punto de punto asíncrono universal de recepción y transmisión de conexión (UART).

La biblioteca de los comandos disponibles para pasar de la conexión UART (una biblioteca de C), así como la propia conexión física, se hace referencia a que la interfaz de host

2.1.2 El flujo de mensajes

La interfaz de host implementa un protocolo maestro o esclavo donde el anfitrión externo es "maestro" y el módulo es "esclavo". El protocolo permite que tanto el maestro y el esclavo para iniciar el intercambio de datos, aunque el mecanismo es diferente para cada uno.

La interfaz de host es compatible con dos tipos de mensajes: Comando y paquetes de respuesta. El maestro siempre envía paquetes de comando, mientras que el esclavo siempre envía paquetes de respuesta.

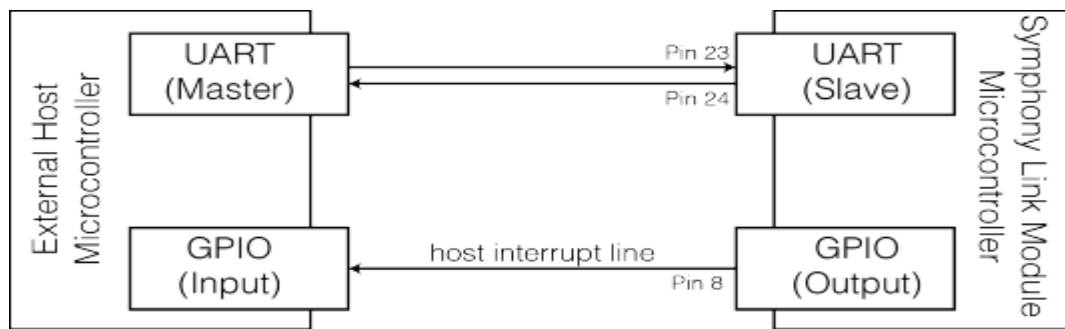


Figura 2.1: Pines encargados de la comunicación entre los módulos [11]

2.1.3 Iniciación por el maestro

El anfitrión externo actúa como el maestro de protocolo y envía únicamente el UART paquete de órdenes a la vez. El maestro debe esperar a que un paquete de respuesta que se devuelve desde el esclavo antes de enviar otro paquete de órdenes. El esclavo responde dentro de los 300 ms a un paquete de órdenes con una sola respuesta del paquete.

2.1.4 Iniciación por el esclavo

El módulo no puede iniciar directamente una transacción UART. Cualquier flujo no deseado de los datos de esclavo a maestro se da en la forma de una solicitud de interrupción encuestados. El esclavo inicia una solicitud de interrupción encuestada afirmando la línea de anfitrión de interrupción de uso general de entrada y salida (GPIO). La línea de interrupción anfitrión el Pin 8 del módulo. Si es evidente cuando el módulo afirma la línea de interrupción de host, el host externo utiliza apropiadas UART Comando de paquetes para consultar el motivo de la interrupción, a través de un campo de datos denominado el Host interfaz de interrupción Banderas (Banderas IRQ). Dependiendo de lo que se establecen las banderas, los maestros de los ingresos sobre una base de bandera específica, posiblemente emitir órdenes adicionales para continuar el intercambio de datos.

La línea de interrupción de acogida se afirma cada vez que se establece una bandera de interrupción. Se afirmó después de la orden maestra y así borra todas las banderas de interrupción pendientes.

Configuración UART:

Nominal Velocidad de transmisión	115200 bps
Datos	8 bits
Paridad	Ninguna
Bits de parada	1 bit
Control de flujo	Ninguna

Tabla 2.0: Parámetros para lograr una comunicación entre LL-RXR-27 [11]

2.1.5 Formato de los mensajes

Los paquetes de comando enviado al módulo y de los paquetes de respuesta que devuelve se adhieren a un formato específico. Las siguientes tablas describen el formato de los paquetes de instrucción y respuesta, respectivamente. Cada tipo de paquete concluye con dos bytes de suma de comprobación, que se utiliza para verificar la integridad del paquete, el compute checksum es la función, utilizada para calcular la suma de comprobación, como se describe a continuación.

El comando bytes de ellos mismos ese enumeran "códigos" OP

Formato de los comandos de paquetes		
Número de bytes	Descripción	Comentarios
0	Wakeup preámbulo 0	Siempre igual a 0xFF
1	Wakeup preámbulo 1	Siempre igual a 0xFF
2	Wakeup preámbulo 2	Siempre igual a 0xFF
3	Wakeup preámbulo 3	Siempre igual a 0xFF
4	Inicio de trama	Siempre igual a 0xC4
5	comando Byte	Especifica qué comando principal está contenida en el mensaje
6	número de aviso	Número de 8 bits incrementado por el anfitrión externo. Rollo esperada a lo largo: 253, 254, 255, 0, 1, 2,...

7	longitud de carga útil (MSB)	La mayoría byte significativa de la longitud de carga útil. Nota: longitud de carga útil máxima es de 256 bytes.
8	longitud de carga útil (LSB)	Byte menos significativo de la longitud de la carga útil.
9	Carga útil byte 0	
10	Payload byte 1	
11	Carga útil byte 2	
longitud de carga útil + 9	byte de suma de comprobación (MSB)	La mayoría byte significativo de CRC de 16 bits. Vea la sección de suma de comprobación para más detalles.
longitud de carga útil + 10	byte de suma de comprobación (LSB)	Byte menos significativo de CRC de 16 bits.

Tabla 2.1: Comprobación de byte para el envío y recibo de datos [11]

Formato de respuesta de paquetes		
Número de bytes	Descripción	Comentarios
0	Inicio de trama	Siempre igual 0xC4
1	byte de comando	Especifica el comando principal a la que el módulo está respondiendo.
2	número de aviso	Especifica el número de mensajes a la que el módulo está respondiendo.
3	byte ACK	00 = ACK: Comando Reconocido 01 = NACK: comando no es admitida 02 = NACK: Suma de comprobación incorrecta 03 = NACK: Tamaño de carga útil fuera de rango 04 = NACK: Valor de carga útil fuera de rango
4	longitud de carga útil (MSB)	Byte de mayor valor que representa el número total de bytes de carga útil
5	longitud de carga útil (LSB)	Byte menos significativo que representa el número total de bytes de carga útil
6	Carga útil byte 0	
7	Payload byte 1	

8	Carga útil byte 2	
...	...	
longitud de carga útil + 6	byte de suma de comprobación (MSB)	La mayoría byte significativo de la suma de comprobación de 16 bits. Vea la sección de la suma de comprobación para más detalles.
longitud de carga útil + 7	byte de suma de comprobación (LSB)	Byte menos significativo de la suma de comprobación de 16 bits.

Tabla 2.2: Finalización de la comprobación de los bytes [11]

compute_checksum

```

/**
 * @brief
 * compute_checksum
 *
 * @param[in] buf
 * byte array to compute checksum on
 *
 * @param[in] len
 * size of the byte array in bytes
 *
 * @return
 * The 16-bit checksum
 */
static uint16_t compute_checksum(uint8_t* buf, uint16_t len)
{
    uint16_t i;
    uint16_t crc = 0;
    for(i = 0; i < len; i++)
    {
        crc =(crc>>8)|(crc<<8); crc ^= buf[i];
        crc ^= (crc & 0xff) >> 4;
        crc ^= crc << 12;
        crc ^= (crc & 0xff) << 5;
    }
    return crc;
}

```

Código 1: La siguiente función calcula la suma de control utilizado por la interfaz de host para verificar la integridad del sistema y los paquetes de respuesta.

2.1.6 Módulo de secuencia de inicio

El módulo se iniciará automáticamente cuando se conecta la alimentación y cada vez que la línea de Reset se afirma y luego de este proceso el módulo notifica al host externo que la secuencia de inicio se encuentra completa de manera permanente y así el "Reset" bandera IRQ para afirmar que la línea fue recibida con éxito. El host externo debe esperar a que el indicador de IRQ de reposición antes de interactuar con el módulo.

2.1.7 La actualización del firmware del módulo

En este apartado se dará una forma más sencilla de actualizar el firmware como se describen varios procedimientos firmware a un módulo LL-RXR-27.

Si usted está tratando de actualizar o modificar el firmware de un módulo montado en una placa de evaluación, es más fácil de utilizar la capacidad de actualización de firmware integrado.

En el momento de la fabricación, el micro-controlador Renesas (Renesas R5F5116ADNE) en la LL-RXR-27 está programado con la versión de firmware más reciente. Durante este proceso, cada módulo se le asigna una dirección de cifrado y claves únicas MAC. Un gestor de arranque también se carga en cada MCU, para facilitar futuras actualizaciones de firmware.

Método 1: Uso de Python con `bootloader_upload_firmware.py`

Descargar el **`bootloader_upload_firmware.py`** script en Python, proporcionada por Link Labs. Este script utiliza el estándar **XMODEM** y **PySerial** bibliotecas, así que por favor asegúrese de que se agreguen a la instalación de Python.

El script también utiliza **`ll_ifc.py`**, un envoltorio de Python para controlar un módulo de un ordenador.

Coloque el archivo de firmware cifrado deseado, por ejemplo **`rlp_multi_mac.bin.encrypted`**, en la ruta del archivo correspondiente.

Conectar el módulo a su ordenador a través de USB. Tenga en cuenta qué puerto COM del puente de USB a UART enumera. Por ejemplo, en el siguiente cuadro, el puente de USB a UART de una placa de evaluación de Link Labs ha enumerado a COM21.



Figura 2.2: comprobación de puerto COM a utilizar en la instalación

Desde la terminal de comandos, ejecute el siguiente comando usando el puerto COM adecuado y la ruta del archivo binario codificado.

- `python bootloader_upload_firmware.py --device COMxx ~/path/to/encrypted/binary`

Por ejemplo, el comando podría ser algo como esto:

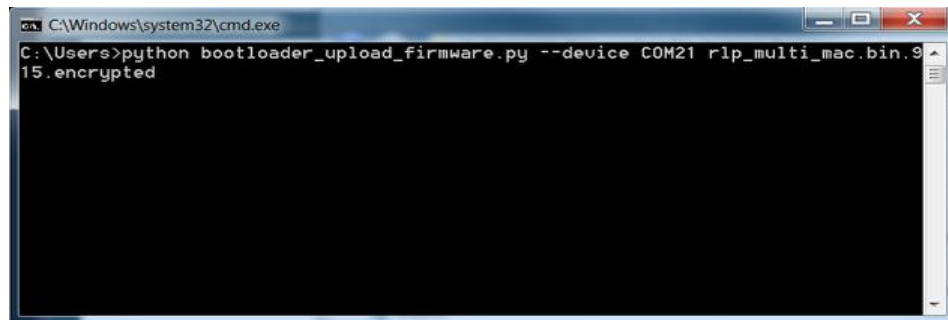


Figura 2.3: Cargado el archivo de firmware

Verificar que se alcanza el siguiente resultado exitoso:



Figura 2.4: Firmware ya instalado en modulo LL-RXR-27

Método 2: Uso de Tera Term

El gestor de arranque en la LL- RXR-27 tiene una interfaz legible. Puede ver esta interfaz mediante la conexión de un terminal para UART del módulo.

Conectar el módulo a su ordenador a través de USB y abrir un terminal. En este ejemplo, estamos usando Tera Term.

Configuración del menú

Seleccione **Puerto Serie**. Compruebe el puerto serie es el puerto COM correcto (en este ejemplo, COM21), con los siguientes valores:

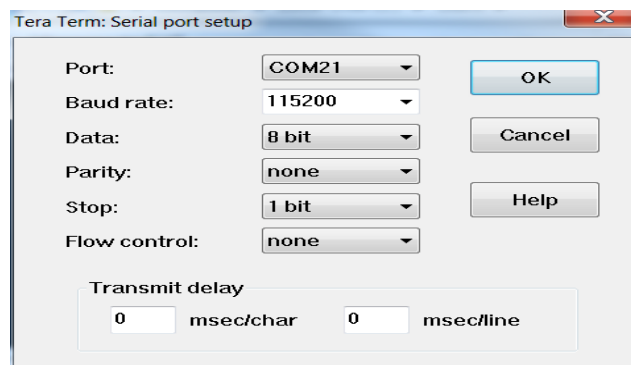


Figura 2.5: Elección de puerto COM para instalación de firmware utilizando método 2

Arrancar el modulo. Esto se puede lograr mediante comandos de interfaz de ordenador, o mediante la celebración de la **BOOT** baja pasador al reiniciar el módulo. La ventana de Tera Term se actualizará con lo siguiente:

```
Module Bootloader (RX111)
```

```
Release v0.10
```

```
Firmware is...OK
```

```
Enter a command:
```

```
h - Help
```

```
u - Upload encrypted firmware
```

```
v - Verify current firmware
```

```
r - Reset
```

Enviar el carácter U de esta manera 'U', mientras que la ventana de Tera Term está activo. Esto comenzará el proceso bootloading.

A continuación, realice inmediatamente los siguientes pasos.

Seleccione **Archivo > Transferencia > XMODEM> Enviar**.

Aparecerá un árbol de archivos; vaya al archivo binario encriptado apropiado y haga clic en **Enviar**. Tera Term enviará el nuevo firmware para el módulo.

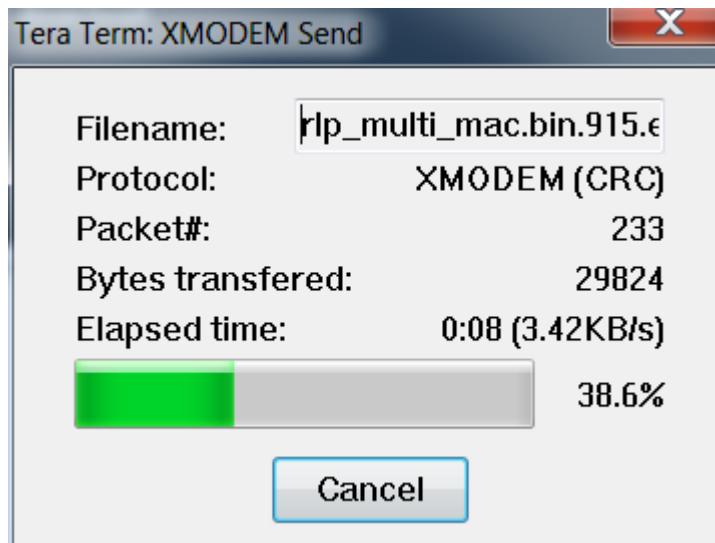


Figura 2.6: Cargando el firmware en el módulo con el método 2 [11]

Después de la finalización con éxito, la ventana de Tera Term se actualizará con:

Verifying...OK

Activating...DONE

Cualquier otra respuesta indica un error en la verificación de la imagen del firmware. Una vez que los módulos envían esta respuesta, el firmware ha sido validado y cargado, y se debe enviar el carácter **r** para reiniciar el módulo en el nuevo firmware [12].

- ✓ Placas Arduino Uno y Due

2.2 instalando Arduino Uno

Esta sección explica como instalar el software Arduino en un ordenador que ejecute cualquiera de los siguientes Sistemas Operativos: Windows, Mac OS X, GNU/Linux.

Como conectar tu placa Arduino al ordenador y cargar tu primera rutina.

Estos son los pasos que se seguirán:

Obtener una placa Arduino y un cable.

Descargar el entorno Arduino.

Instalar los drivers USB.

Conectar la placa.

Conectar un LED.

Ejecutar el entorno Arduino.

Subir un programa.

Buscar el Leds que parpadea

También necesitas un cable USB estándar (parecido con el que conectarías a una impresora USB, por ejemplo)

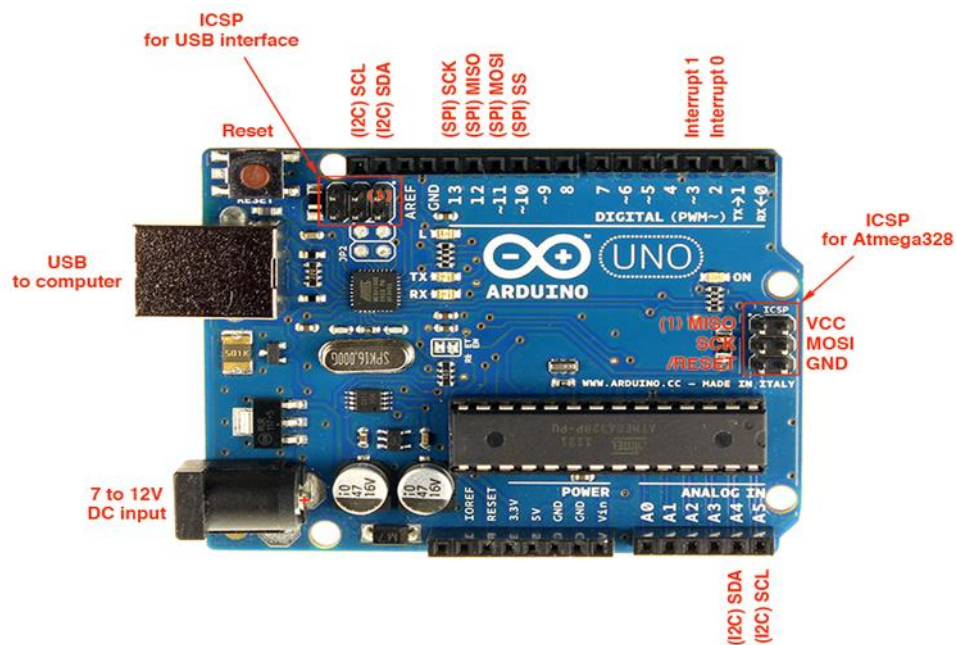


Figura 2.7: Descripción de componentes de la placa Arduino [13]

La fuente de alimentación se selecciona mediante el jumper entre los conectores del USB y alimentación. Para alimentar la placa desde el puerto USB (bueno para controlar dispositivos de baja potencia como Leds), coloca el jumper en los dos pines más cercanos al conector USB. Para alimentar la placa desde una fuente externa (6-12 V), coloca el jumper en los dos pines más

cercanos al conector de alimentación. En cualquier caso, conecta la placa a un puerto USB de tu ordenador.

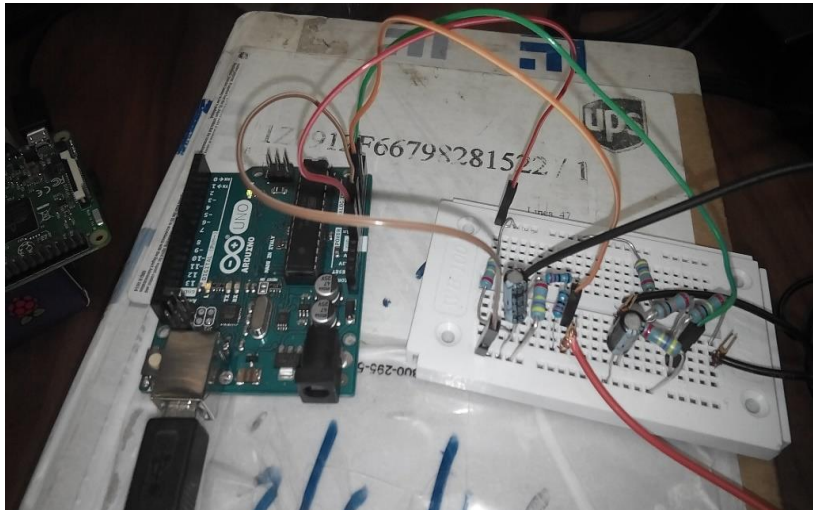


Figura 2.8: Conexión del cable USB a la placa Arduino

El LED de alimentación debería encenderse.

El asistente para Añadir Nuevo Hardware debería abrirse. Indícale que no conecte a Windows Update y haz clic en siguiente.



Figura 2.9: Asistente para Nuevo Hardware MS-Windows - Paso 1

Selecciona instalar desde una lista o ubicación especificada (Avanzado) y haz clic en siguiente.



Figura 2.10: Asistente para Nuevo Hardware MS-Windows - Paso 2

Asegúrate de buscar el mejor driver en estas ubicaciones que está marcado; desmarca buscar dispositivos extraíbles y marca incluir esta ubicación en la búsqueda y navega a la ubicación donde descomprimiste los drivers USB en el paso anterior. Haz clic en siguiente.

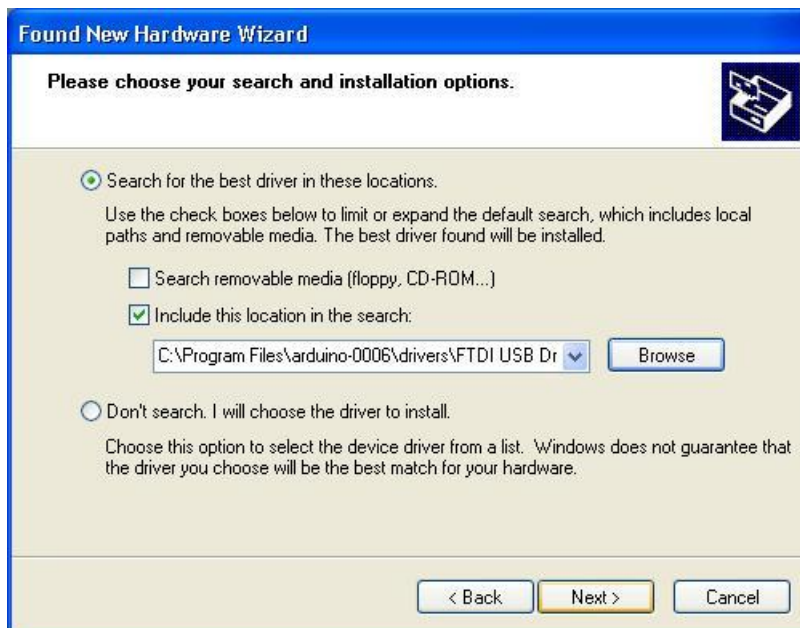


Figura 2.11: Asistente para Nuevo Hardware MS-Windows - Paso 3

El asistente buscará el driver y dirá que se encontró USB Serial Converter. Haz clic en finalizar



Figura 2.12: Asistente para Nuevo Hardware MS-Windows - Paso 4

El asistente para Añadir Nuevo Hardware aparecerá de nuevo. Realiza los mismos pasos. Esta vez, se encontrará un USB Serial Port.

La primera rutina que subirás a la placa Arduino hace parpadear un LED. El Arduino Uno tiene una resistencia incorporada y un LED en el pin 13.

Ejecutar el entorno Arduino.

Abrir la carpeta de Arduino y hacer doble clic en la aplicación Arduino.

Subir un programa.

Abrir la rutina de ejemplo de parpadeo del LED: File > Sketchbook > Examples > Digital > Blink.

Seleccionar el dispositivo serie de la placa Arduino desde el menú Herramientas > Puerto Serie. En Windows, este debería ser COM1 o COM2 para la placa serie Arduino, COM3, COM4 o COM5 para la placa USB. Para descubrirlo, abrir el Administrador de Dispositivos de Windows

(En la pestaña Hardware o en el Panel de Control de Sistema). Buscar un USB Serial Port en la sección Puertos; esa es la placa Arduino.

- ✓ Relé de corriente alterna



Figura 2.13: Relé de estado solido [14]

Se puede utilizar el relé de estado sólido SSR-40 DA en equipos industriales, transformadores, reguladores de temperatura y sistemas de protección. Soporta hasta 40A, voltaje entre 24 y 380 VAC. No hay contactos mecánicos, con este relé se pueden encender y apagar varias veces el equipo en un corto período de tiempo.

Relé de estado sólido (SSR siglas en inglés) se refiere a un dispositivo semiconductor que puede realizar las mismas funciones que un relé electromecánico común, con la diferencia de no tener partes mecánicas o contactos.

Esto aumenta la vida útil del relé, reduce la producción de ruido durante el funcionamiento y aumenta la seguridad de conducción.

Especificaciones:

- Modelo: relé de estado sólido SSR-40 DA
- Tensión de entrada / Control: 3 ~ 32 V CC
- Tensión de funcionamiento: 24 ~ 380 VAC
- Tensión de servicio: 250V
- apoyado Corriente máxima: 40A

- Tiempo conmutación: <10 ms
- tipo de contacto: Normalmente abierto
- temperatura de funcionamiento: -20 ° C a +80 ° C
- material: ABS
- Dimensiones: 60x45x28mm
- Peso: 105

✓ Sensor de corriente



Figura 2.14: Transformador de corriente de núcleo partido [15]

Especificaciones

Sensor de corriente núcleo dividido/transformador SCT-016 con conector terminal

Modelo: SCT-016 entrada nominal actual: 0.01a/120a gire relación: 1:3000

Características:

Apertura tamaño: 16mm * 16mm,

No linearity ±3 % (10%-120% de corriente de entrada)

1 m cable que va, estándar Φ3. 5 tres núcleos enchufe de salida.

Tipo de salida de corriente y tensión de salida (salida de tensión Resistencia de muestreo Tipo incorporado)

Usos: se utiliza para la medición de corriente, monitor y protección del motor de ca, Equipos de iluminación, compresor de aire et

Material del núcleo: Ferrite

Resistencia mecánica: el número de conmutación es no menos de 1000 veces (prueba a 25 ° c)

Safety índice: rigidez dieléctrica (entre el depósito y salida) 1000 V a/1 Min

Propiedad de resistencia al fuego: de acuerdo con UL94-VO

Temperatura de trabajo: -25 ° C ~ 70 ° C

Modelo	SCT-016
Apertura Diámetro (Mm)	16
Relación de transformación	1:3000
Frecuencia de trabajo	50 HZ-1 kHz
Tamaño del esquema (mm) a/b/c	29.5x30.8x46
Corriente nominal de entrada	0-120A
Linealidad	± 1%
Tipo de salida	Salida de corriente
Salida de corriente/tensión	40mA
Rigidez dieléctrica (ca/1 min)	1000 V
Liderando el modo de salida de señal	50 cm
Modo de montaje	Tipo de suspensión
Material de núcleo de hierro	Ferrita
Peso	70g

Tabla 2.3: Características del transformador

2.3 PRUEBAS DE LABORATORIO

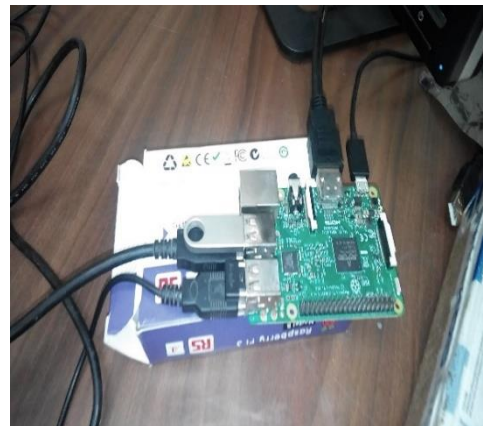


Figura 2.15: Calibración de sensores y módulos



Figura 2.16: Comprobación de corriente RMS para relé de estado sólido



Figura 2.17: Comprobación del correcto funcionamiento del sistema

2.4 Configuración e instalación de Raspberry pi3 [9]



Figura 2.18: Raspberry PI3 [6]

Si se posee un case o carcasa para la Raspberry Pi, lo primero que se debe hacer es insertar la placa dentro, haciendo coincidir las aperturas para los puertos adecuadamente y luego apretar los 4 tornillos que se adjuntan. Ahora ya está lista y protegida para poderla manipular de una mejor forma.

Después, todos los cables, como los USB del teclado y el ratón que servirán para la configuración del sistema, conectar el HDMI a tu pantalla o TV y la fuente de alimentación, pero no la conectar a la red eléctrica, ya que eso provoca que la Raspberry se encienda y aún no se ha insertado la tarjeta SD con el sistema operativo. Entonces esto será lo siguiente.

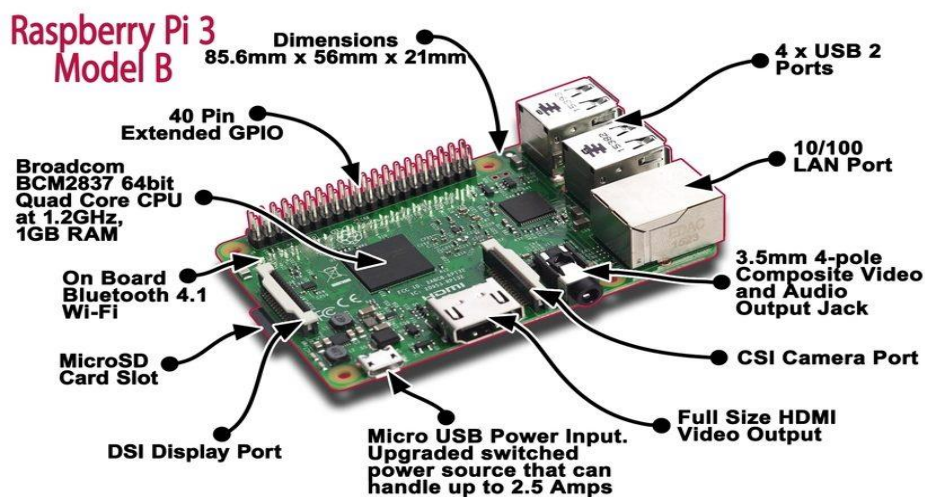


Figura 2.19: Distribución de pines de Raspberry PI3 [6]

Preparar la tarjeta con el sistema operativo elegido es sencillo. Como digo, en nuestro caso es NOOBS, pero si has elegido otro el procedimiento es igual, sustituyendo la imagen de NOOBS por la del otro sistema, aunque puede que algunos sistemas operativos puedan diferir un poco en el procedimiento:

Descomprime el ZIP de NOOBS que hayas descargado. El contenido deberás pasarlo a nuestra SD previamente preparada como te indicamos a continuación.

Preparar la SD desde Windows

Una vez hayas descargado el ZIP de NOOBS desde la web oficial, debes proceder de la siguiente manera:

Formatear la SD utilizando la aplicación SD Association Formating Tool o directamente con las opciones de formateo de Windows. Para ello, vas a Equipo (Mi PC) con la SD insertada en la ranura de tu PC y le haces clic con el botón derecho en la unidad correspondiente a la tarjeta. Pulsas sobre Formatear y dejas los valores por defecto (debe de ser FAT como formato), si lo deseas puedes escribir un nombre de etiqueta del volumen.

Ahora el contenido descomprimido del ZIP debes copiarlo y pegarlo o arrastrando al interior de la SD. Si utilizas un software de compresión/descompresión, puedes elegir la opción extraer cuando abras el archivo ZIP y seleccionar la SD como destino directamente.

Ya tienes instalado NOOBS, ahora se puede retirar la SD de la ranura del PC con seguridad e insertarla en la Raspberry Pi.

No conectar la fuente de alimentación a la red eléctrica porque se iniciaba la Pi automáticamente (recuerda que la Pi no tiene interruptores ON/OFF), ya que has insertado la SD con NOOBS en su ranura y el resto de cables están conectados. Los Leds comienzan a encender y en la pantalla verás unos colores primero y lo siguiente será el menú principal de NOOBS cuando arranquemos nuestra Raspberry Pi. La pantalla del Recovery Tools nos mostrará los sistemas operativos disponibles en la versión de NOOBS que te hayas descargado. Igualmente puedes seleccionar el idioma en el menú Lenguaje y cambiar a español.

Para comenzar a usar uno de los sistemas operativos disponibles (recuerda que en las nuevas versiones de NOOBS solo está disponible Raspbian OS para usarlo offline, el resto debes descargarlos para instalarlos), selecciona el que deseas y pulsa ENTER o en el botón Install que aparece. Para cambiar de SO solo tienes que pulsar la tecla Shift al iniciar la Raspberry. Por cierto, como puedes comprobar, aparece un icono a la derecha de cada entrada de sistema operativo en el menú de NOOBS y verás cómo los sistemas que están en la SD tienen un símbolo de una tarjeta SD y los que tienes que instalar desde la red tienen un símbolo de una conexión Ethernet.

Vamos a suponer que vas a utilizar Raspbian OS, uno de los principales y más recomendables para uso genérico y que además es el único offline que se integra en la NOOBS. Cuando es la primera vez que inicias Raspbian ya no es necesario hacer login como en las antiguas versiones y luego introducir el comando startx para iniciar el sistema gráfico, ahora ya lo hace automáticamente.

Una vez finalizado, si todo ha ido correctamente, ya puedes comenzar a utilizar tu mini PC y a instalar las aplicaciones que necesites.

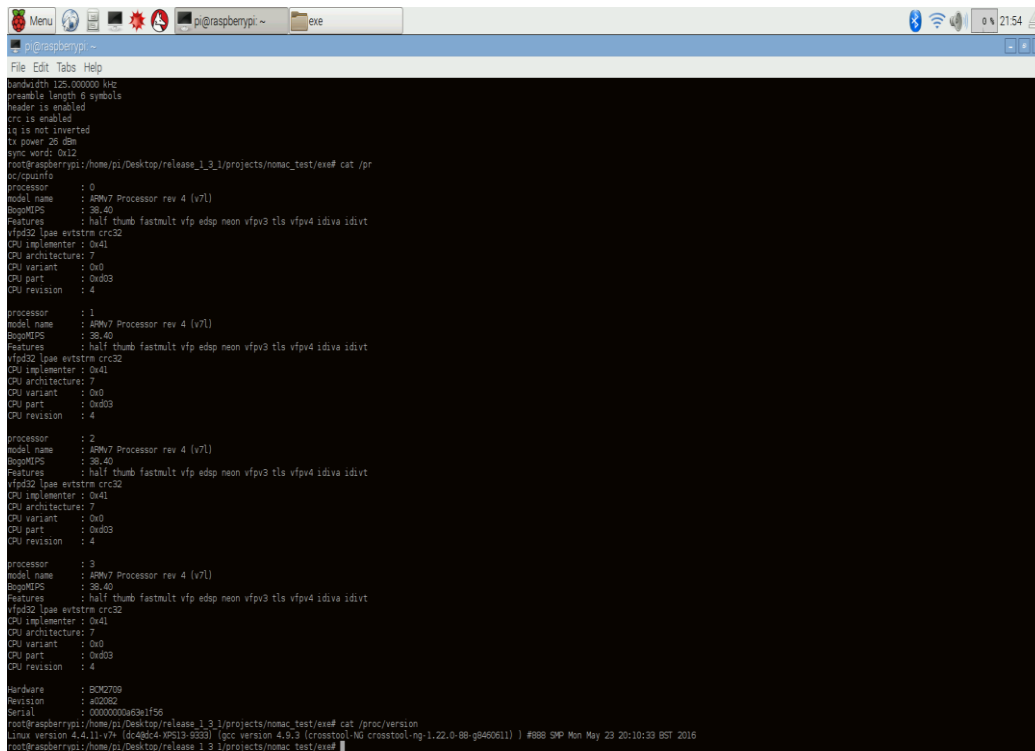


Figura 2.20: Sistema arrancado y funcionando

3.0 CAPÍTULO III: INTEGRACIÓN Y PRUEBA DE SUB- SISTEMAS.

3.1 introducción

En el presente capitulo, se detalla cómo se ha ido integrando cada componente al sistema deseado. Se sabe que dentro de todo sistema, es necesario ajustar cada detalle para un correcto funcionamiento de cada una de las piezas o partes que se utilizan y así asegurar en gran medida el funcionamiento en conjunto para determinar el éxito de lo que se está planteando. En esta oportunidad esa regla se debe cumplir en la brevedad posible, por ello que se tiene que implementar un orden lógico de cada una de sus partes para el acoplamiento del sistema. Aclarando lo anterior se continúa: Como primer pasó, realizar un estudio detallado de los módulos RF utilizados, para ser más precisos será el LL-RXR-27. Se sabe que este módulo como ya se ha mencionado en los capítulos anteriores posee diferentes modos de operación como se describe a continuación: Symphony Link y NoMac.

En el estudio realizado para la familiarización con dichos módulos, se encontró que los primer modo de operación obliga al usuario a utilizar un tercer dispositivo como puerta de enlace o Gateway, lo que influye en un aumento para el costo total de todo el sistema, para una mejor comprobación de dicha afirmación basta con visitar el sitio de Link Labs el cual confirma, que la adquisición de un kit para utilizar el modulo en el modo Symphony, tendría un costo adicional de \$1000 aproximadamente y si para esto se necesita soporte de fábrica, el precio tiene un aumento del 320 % costando alrededor de \$3,200. Teniendo claro este punto, es útil saber que esos valores no consideran aun, coste de envío ni impuestos.

Todo lo anteriormente mencionado deja como única opción el modo NoMac. Aunque, este modo se debe ejecutar exclusivamente utilizando la consola para introducir los comandos del sistema operativo que satisface las necesidades del usuario, generando la posibilidad de interactuar con los parámetros de configuración de los módulos, lo cual es sumamente importante si se desea poder controlar los módulos como radio transmisores, es oportuno mencionar que los módulos han sido diseñados bajo el concepto del IoT (internet de las cosas).

3.2 Hardware

- Transformador de corriente de núcleo cortado
- Fuente de voltaje 240 VAC/9 VAC
- Una tarjeta Raspberry PI 3
- Una tarjeta Arduino Uno Rev. 3
- Un relé de estado sólido 40 A máx
- Resistencias y capacitancias de diferentes valores para adecuar las señales a censar.
- Una computadora personal para el sitio remoto
- Dos antenas Yagi 15 dBi 915 MHz
- Dos radiotransmisores para transmitir a 915MHz
- Dos cables de 7 metros con conectores tipo N y SMA
- Dos tubos de aluminio de 6 metros
- Un monitor con entradas HDMI
- Un teclado y mouse USB

3.2.1 Transformador de corriente de núcleo cortado [15]

Es el encargado en estar monitoreando los parámetros de corriente RMS que demanda el sistema ya en funcionamiento. El cual, está generando los valores que luego se envían al Arduino para comprobación de datos de dicho sistema, siendo uno de los pilares fundamentales en este ámbito ya que es constante en la interrogación del sistema para verificar un correcto funcionamiento en la corriente demandada por la bomba.

Es importante destacar que este sensor debe ser calibrado de acuerdo al motor al que se le monitorean las diferentes variables eléctricas, para ello se debe acceder al código Arduino y desde allí introducir los parámetros que se necesitan para la medición mediante el número de bits que envía, ya que además de la corriente genera un factor de potencia ya con todo definido se compila el código Arduino y luego se carga a la placa Arduino Uno que se está empleando en el sistema. Luego, se obtienen los valores de la medición y para comprobación de que es un valor fiable se verifica con un amperímetro de tenaza el cual indica el mismo valor que se obtiene en pantalla.

El detalle del circuito para la medición de corriente se muestra a continuación:

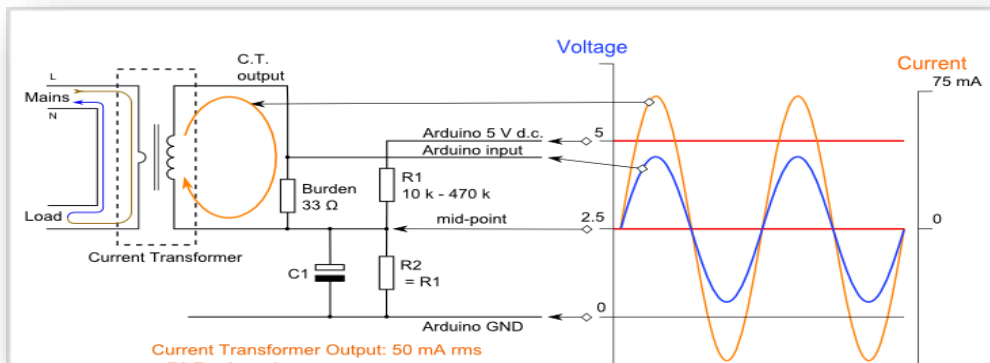


Figura 3.0: Circuito para la medición de corriente [7]

Como punto inicial fue necesaria la transformación de la corriente inducida por la bomba en el devanado secundario del transformador, a una señal de voltaje equivalente, esto se logró colocando una resistencia de carga a la salida del transformador, el valor de la misma depende de la corriente de salida, ya que la amplitud de la señal que entra al Arduino no debe ser mayor a 5 V, de lo contrario la entrada del Arduino sufriría un daño irreversible. Para el caso se calculó de la siguiente manera:

$$I_{rms} = 40 \text{ mA} \qquad I_{max} = \sqrt{2} * I_{rms} = 56.57 \text{ mA} \qquad I_{pp} = I_{max} * 2 = 113.13 \text{ mA}$$

$$R = 5 \text{ V} / 113.13 \text{ mA} = 44.19 \Omega \approx 44 \Omega$$

Otro aspecto importante a tener en cuenta con este circuito es el hecho que el rango de entrada del Arduino va de 0 V a 5 V, es decir no acepta valores negativos, es por ello que fue necesaria la aplicación de un Offset a la señal de interés, esto se logró utilizando dos resistencias similares de 10 KΩ alimentadas desde el Arduino con 5 V, con esto se logra que el punto medio de la señal de entrada se situó en 2.5 V.

Para el caso del voltaje se conecta la fuente AC-AC en paralelo con la carga para poder medir el voltaje de la misma, el circuito utilizado es el siguiente:

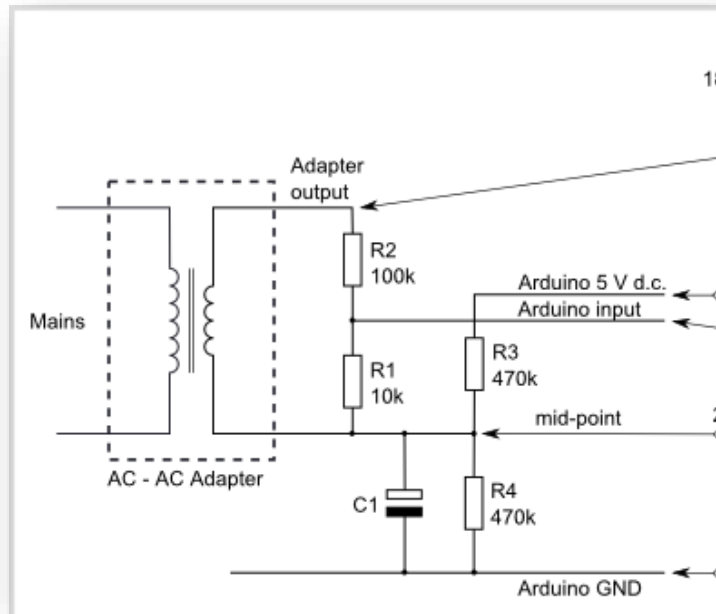


Figura 3.1: Circuito para la medición de voltaje [7]

Como se recuerda la fuente provee 9 V en la salida pero al operar sin carga estas fuentes debido a la poca regulación de voltaje que poseen pueden llegar hasta los 12 V, es decir un $V_{pico} = 16.97$ V, Es por ello que se debe hacer uso de un divisor de voltaje para escalar la señal original, para el caso:

$$\frac{10K}{100K + 10K} * 16.97 = 3.08 V_{pp}$$

Con ese valor de voltaje no se sobrepasan los 5 V que soporta el Arduino en su entrada, y al igual que en el caso de la corriente se aplicó un offset para que el punto medio de la señal de entrada sea 2.5 V.

USO DE RELE DE ESTADO SOLIDO PARA ACTIVACION Y DESACTIVACION DE LA CARGA

La activación o desactivación del sistema de bombeo se realiza a través de una de las salidas digitales del Arduino, configurando el estado de la misma en alto o en bajo según se necesite. Para pasar del mundo de la electrónica donde se manejan corrientes de unos cuantos miliamperios a manejar una bomba sumergible de 10 HP que cuando opera a 240 V demanda una corriente de 31.25 Amperios es necesario el uso de un elemento adicional como lo es el relé de estado sólido. Dentro del marco de la implementación del sistema se detectó que a pesar de que el voltaje de activación del relé iba de 3 a 30 VDC la corriente que demandaba era superior a los 20 mA que es

la corriente máxima que el Arduino puede manejar en sus salidas. Por ejemplo a 5 V el relé demanda una corriente de 48 mA

Para solventar esta problemática se decidió hacer uso de un transistor BJT 2N2222A, ya que este tipo de transistor soporta según su datasheet una corriente de hasta 800 mA, como también una fuente 120 VAC / 9 VDC para poder suministrar la corriente necesaria. El esquema implementado se muestra en la figura 3.10 donde M represente el relé a utilizar.

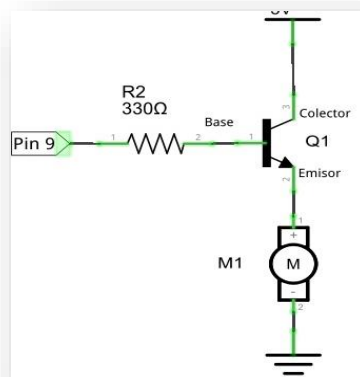


Figura 3.2: Etapa de potencia del sistema [7].

3.2.2 Fuente de voltaje 240 VAC/9 VAC



Figura 3.2.1: fuente de voltaje 240V AC/ 9V AC

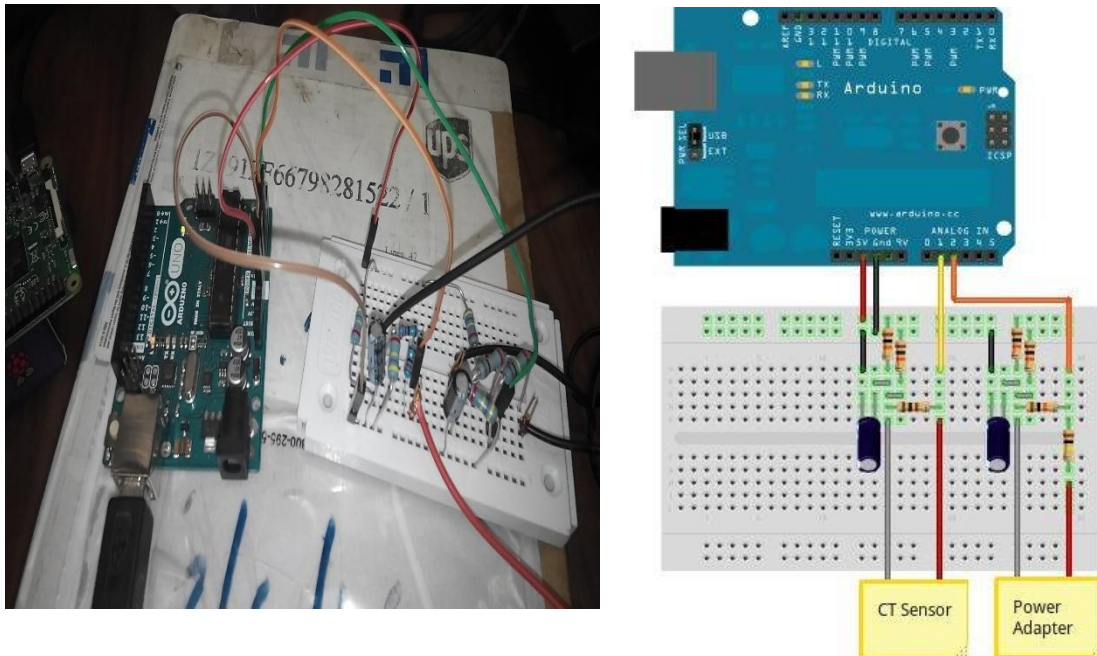


Figura 3.2.2: Esquema implementado para las mediciones de voltaje y corriente [7].

Esta es la encargada en el control de voltaje el cual se conecta al Arduino mediante un arreglo de esta manera se puede estar censando los dos parámetros de interés del sistema creado. Adicionalmente, dan la oportunidad al usuario final de obtener una medición en tiempo real de voltaje y corriente RMS.

3.2.3 Tarjeta Raspberry PI 3

Esta tarjeta se encuentra configurada para el centro de monitoreo del control remoto, esto significa que estará en el tanque de distribución de los usuarios interrogando vía RF por medio de los módulos, todas las peticiones necesarias para llevar a cabo el proceso primeramente, esto se detalla de la siguiente manera:

- 0 Este código significa apagar el sistema
- 1 Código utilizado para una petición de encendido
- 2 Se transmite este código para anunciar que el voltaje del lado remoto no es el adecuado.
- 7 Este código es utilizado cuando se solicita una actualización de los valores de voltaje y corriente.

Si no se recibe ninguno de estos códigos se entiende que la comunicación entre ambos sitios no es viable en ese momento.

3.3 Software

main.c:

Es el programa a través del cual se puede interactuar con los módulos RF en modo NoMac, cuando se compila este programa se genera un archivo ejecutable llamado nomac_test, es importante mencionar que este programa por estar escrito en “C” goza de una valiosa portabilidad de un sistema operativo a otro. Dentro del marco de integración y pruebas fue necesaria la modificación de dos aspectos importantes dentro de este programa que no favorecían al proceso de automatización, la primera con respecto al tiempo de ejecución del modo recepción. Esto se logró a través de una modificación en el valor de la variable Receive_time, para el caso del centro de monitoreo se fijó un valor de 8000 ms y en el caso del sitio remoto se ajustó a 3000 ms. Con respecto a la segunda modificación realizada, fue necesario el cambio en la forma de presentar los datos por parte del programa, dado que este regresaba la información recopilada con un formato que utilizaba múltiples líneas, lo cual no era conveniente para el diseño del sistema.

```
if (( rv == 0 ) && (rx_len > 0))
{
    //printf("Received %d byte packet  RSSI= %d SnR= %0.2f:\n\t",
    // rx_len - 3, *((int16_t *)&rx_buf[0]), (int8_t)rx_buf[2] / 4.0); //se comento para estandarizar la transmision en
    la cadena de datos
    for (c = 3; c < rx_len; c++)
    {
```

Figura 3.2.3: Líneas de código modificadas en el programa main.c

Monitor_Center.py

Este programa es el que se ejecuta en la Raspberry utilizada en el centro de control y el más extenso de todos los programas utilizados. Actualmente, se ejecuta desde la consola de comandos de Linux, invocando al intérprete de Python 2.7. Lo anterior no significa que el usuario final no pueda interactuar con el programa, ya que ha sido estructurado de tal manera que el usuario puede controlar cada una de las acciones del sistema, introduciendo la opción de interés desde el teclado de la computadora. La función principal del programa es establecer una interfaz entre la etapa de comunicación y el usuario final, ya que a través de los valores ingresados desde el teclado, dicho programa utiliza el archivo nomac_test para poder transmitir un código hacia el sitio remoto, automáticamente se coloca en modo de recepción y espera por la respuesta enviada por el modulo ubicado en el manantial.

```
Monitor_center.py x
#!/usr/bin/python

from subprocess import call
from subprocess import check_output
from sys import exit

call(["./nomac_test", "-C", "1"])           #seteamos coding rate a 4/5
call(["./nomac_test", "-S", "8"])         #Seteamos spreading factor
call(["./nomac_test", "--mac_set", "0"])   #Seteamos en modo NoMac
call(["./nomac_test", "-f", "915000000"])  #Seteamos la frecuencia de operacion
call(["./nomac_test", "-p", "26"])        #Seteamos la potencia de operacion
call(["./nomac_test", "-w", "1"])         #ancho de banda de 125 KHz
call(["./nomac_test", "--antenna_set", "1"]) #Para utilizar antena externa
call(["./nomac_test", "-a"])             #Se guarda la configuracion
call(["./nomac_test", "-g"])             #Comprobacion de configuracion correcta

opcion=0
while opcion!=1 and opcion!=2:
    opcion=input("Bienvenido al centro de control, preste 1 para encender el sistema, o 2 para salir \n")
    if opcion==1:
        call(["./nomac_test", "-t", "1"])   #mandamos comando de encendido
        a=check_output(["./nomac_test", "-r"]) #esperamos respuesta del lado remoto por diez segundos
        respuesta=a[163:164]
        print respuesta
        val=len(a)
        while (val<=163):
            accion=input("Imposible conectar con el sistema remoto: preste 0 para salir o 1 para reintentar")
            if accion==0:
                print("La sesion ha terminado, gracias")
                exit()
            if accion==1:
                call(["./nomac_test", "-t", "1"]) #mandamos comando de encendido
                a=check_output(["./nomac_test", "-r"]) #esperamos respuesta del lado remoto
                respuesta=a[163:164]
                val=len(a)
```

Figura 3.3: Líneas de código del archivo Monitor_center.py [16]

Fuente.py:

Este programa se ejecuta de forma permanente en la consola del lado remoto del sistema, básicamente su función es esperar por una petición del centro de monitoreo. Cuando se recibe un código este lo traslada al Arduino y espera por la respuesta del mismo, la cual incluye un código de retorno y en la mayoría de los casos, los valores de voltaje y corriente leídos.

El rol de este programa es de vital importancia ya que permite la automatización de la comunicación entre el centro de control y el lado remoto, convirtiéndose así en una interfaz entre el control principal y los diferentes sensores y actuadores que se encuentran del lado de la fuente.

```

Fuentes.py x
print ("Esperando petición del centro de monitoreo")
#a=check_output(["./nomac_test","-r"]) #Obligamos al programa a escuchar por lo menos una vez
a='inicio'
lon=len(a)
var=1
while var==1:
    while(lon<=163):
        a=check_output(["./nomac_test","-r"])
        lon=len(a)
        #print lon
        #print a
    codigo=a[163:164]
    if codigo=='0':
        info=serial.Serial('/dev/ttyACM0',9600,timeout=3)
        info.write('0')
        retorno=info.readline()
        if retorno[0:1]=='0':
            call(["./nomac_test","-t","0"])
    if codigo=='1':
        info=serial.Serial('/dev/ttyACM0',9600,timeout=3)
        info.write('1')
        retorno=info.readline()
        if retorno[0:1]=='1':
            call(["./nomac_test","-t",retorno])
        if retorno[0:1]=='2':
            call(["./nomac_test","-t",retorno])
    if codigo=='7':
        info=serial.Serial('/dev/ttyACM0',9600,timeout=3)
        info.write('7')
        retorno=info.readline()
        call(["./nomac_test","-t",retorno])
a='inicio'
lon=len(a)

```

Figura 3.4: Líneas de código del programa Fuente.py [16]

LastVersion.INO

Este es el programa que controla el funcionamiento de la placa Arduino, que a su vez maneja los sensores y el relé de activación, este código ejecutará una acción siempre y cuando se reciba una petición de parte de la computadora a la cual está conectado vía serial-USB, y devolverá a la misma el resultado de la ejecución. Esto incluye código de respuesta y los valores censados.

```
LastVersion
#include "EmonLib.h"           // Include Emon Library
EnergyMonitor sensores;      // Create an instance

//char dataString[50]={0};
char dato;
int led=13;
void setup(){
  Serial.begin(9600);
  pinMode(led,OUTPUT);
  sensores.voltage(2,112.5,1.7);
  sensores.current(1,52.5);
}

void loop(){
  //sprintf(dataString,"%02X",a);
  //Serial.println(dataString);
  //a++;
  //delay(2000);
  //Serial.begin(9600);
  if(Serial.available()>0){
    dato=Serial.read();
    if (dato=='0'){
      digitalWrite(led,LOW);
      Serial.println("0");
    }
  }
}
```

Figura 3.5: Líneas de código programa LastVersion.INO

3.4: Control de los módulos

Para configurar los módulos en modo NoMac es necesario el programa creado en C que se ejecuta en la computadora del usuario. Dicho programa permite comunicarse con los módulos. Para ello es necesario descargar la carpeta release_1_3_2.tar.gz del sitio oficial de Link Labs. En la figura siguiente se muestra lo anteriormente mencionado:



Figura 3.6: Página oficial de Link Labs de donde se obtiene el código para controlar los módulos [17]

Al descomprimir el archivo descargado se encuentran diversos códigos para los diferentes modos de operación, utilizando únicamente el NoMac. Todas las configuraciones y pruebas se hicieron en ambos controles tanto en la estación de monitoreo como el tanque de distribución, esto se llevó a cabo en el sistema operativo Ubuntu 14.04 LTS corriendo sobre una Dell Inspiron 6400 y del lado del centro de monitoreo se utilizó el sistema Raspbian Jessie corriendo sobre una Raspberry PI 3. Después de ejecutar el comando “make” en la consola de Linux, es creada una carpeta llamada exe, en donde se encuentra un ejecutable llamado nomac_test que permite la comunicación con el módulo, como se observa en la imagen siguiente donde ejecutando la opción “-h” proporciona la siguiente salida:

```

C:\Windows\system32\cmd.exe
C:\Users\Carlos>nomac_test.exe -h
Usage:
--baudrate [-b] configure baudrate of the tty device, default: 115200
--coding_rate [-C] configure coding rate [-4] : 4/5, 4/6, 4/7, 4/8
--tone [-c] Transmit a CW tone
--delete_settings [-d] delete saved settings from flash
--device [-D] choose tty device, default: \\.\COM3
--echo_mode [-e] enter echo mode
--echo_test [-E] Peer-to-peer test send/receive with matching echo mode end-no
de
--freq [-f] configure rx/tx frequency Hz
--get_radio_params [-g] Get the radio parameters of the module
--help [-h] print this help message
--tx_power_set [-p] configure tx output power
      LLRLP20 [-2 to +20 dBm] LLRRR26 [-11 to +26 dBm]
--rx [-r] Place module in receive continuous mode
--rxh [-R] Place module in receive continuous mode, print out raw hex
--restore_defaults [-R] Restore default radio settings
--save_settings [-s] store radio settings to flash
--sleep [-S] Place module in sleep mode
--spreading_factor [-S] spreading factor [-6-12]
--sync_word LoRa sync word [LoRaWAN: 0x34, default: 0x12]
--tx [-t] <tx message string> Place the module in transmit mode
--txh [-x] <tx hexadecimal data> Place the module in transmit mode
--bandwidth [-w] configure bandwidth [0-3] : 62.5k, 125k, 250k, 500k, default:
3
--reset_ncu [-u] Reset the module (takes a few seconds)
--bootloader [-U] Reset to Bootloader mode (takes a few seconds)
--key [-K] Set the encryption key
--mac_set MAC mode set
--mac_get MAC mode get
--eui64 get the IEEE EUI-64 unique identifier
--uid get unique identifier
--irq_flags_get get state of IRQ Flags Register
--irq_flags_clear clear bits in IRQ Flags Register
--preamble_len Set the preamble length in symbols
--sleep_block Block sleep
--sleep_unblock Unblock sleep
--antenna_set Set the antenna configuration
--antenna_get Get the antenna configuration
--receive_time Number of milliseconds to poll when in receive mode, 0 indicate
s infinite loop, default: 0
--iq_inversion_set Set polarity: 0 for normal, 1 for inverted

C:\Users\Carlos>_
  
```

Figura 3.7: Salida y opciones del ejecutable nomac_test [11]

OPCION	DESCRIPCION
-C	Establece la tasa de codificación extra de los paquetes
-f	Establece la frecuencia de operación (902-928 MHz)
-g	Obtiene los parámetros de la configuración actual
-p	Establece la potencia de operación del módulo, para el modelo utilizado el rango permitido es de +11 dBm a +26 dBm
-r	Pone en funcionamiento el modo de recepción
-R	Permite en caso de ser necesaria la restauración de los diferentes parámetros a los

	valores de fábrica.
-a	Guarda la configuración actual en la memoria del modulo
-t	Se utiliza para transmitir los mensajes
-w	Configura el ancho de banda utilizado, las opciones: 62.5 K, 125 K, 250 K y 500 K
--mac_set	Establece el modo de operación del modulo
--mac_get	Obtiene el modo de operación actual del modulo
--antena_set	Configura cual antena utilizar la construida internamente sobre la placa o la del conector SMA
--antena_get	Determina cual antena está siendo utilizada por el modulo

Tabla 3.0: Opciones importantes del radio transmisor [18]

```

266 lines (219 sloc) | 10.7 KB
Raw Blame History
1 /*
2  Emon.cpp - Library for openenergymonitor
3  Created by Trystan Lea, April 27 2010
4  GNU GPL
5  modified to use up to 12 bits ADC resolution (ex. Arduino Due)
6  by boredman@boredomprojects.net 26.12.2013
7  Low Pass filter for offset removal replaces HP filter 1/1/2015 - RM
8  */
9

```

Figura 3.8: Autorización para uso público del monitor de energía basado en Arduino.

3.5 Pruebas de campo

Para verificar el funcionamiento del sistema diseñado, fue necesario desplazarse hasta la zona de estudio en el cantón San Antonio Arriba, jurisdicción de Santiago Nonualco, La Paz. Y en dicha visita se implementaron y configuraron todos los elementos que conforman el esquema propuesto para lograr automatizar el proceso requerido.

3.5.1 Actividades previas

1. Se realizaron las pruebas de laboratorio respectivas para calibrar previamente todo el sistema
2. Se corrobora a nivel de laboratorio que el radioenlace es funcional y que su implementación era viable en la zona anteriormente descrita.
3. Uso de software Radio Mobile para simular y conocer el comportamiento del enlace.



Figura 3.9: Resultado obtenido de la simulación del enlace [19]

Algo importante, que se debe mencionar es que el nivel de la recepción obtenido por medio de la simulación cumple con los estándares requeridos para entablar la comunicación entre los puntos de interés. Lo anterior se basa en el hecho que el modulo utilizado trabaja hasta con -137 dBm, y el obtenido se situó en -88.0 dBm.

3.5.2 Implementación del centro de monitoreo



Figura 3.10: Centro de monitoreo

La posición del centro de monitoreo se estableció en el tanque de distribución tomando en cuenta que este lugar posee una mayor elevación con respecto al sitio donde se ubica el manantial, lo que a su vez favorece la comunicación entre ambos sitios.

3.5.3 implementación del sistema en sitio remoto



Figura 3.11: Comprobación de comunicación exitosa en sitio remoto

3.5.3 Aspectos importantes

- La comunicación entre ambos sitios se establece haciendo uso de antenas con mayor ganancia que la originalmente poseen los módulos, específicamente tipo Yagi con una ganancia de 15 dBi en la banda de 915 MHz.



Figura 3.12: Antenas utilizadas para amplificar la señal

- Debido al voltaje elevado que la distribuidora proporciona a la estación de bombeo no se pudo realizar la prueba con el equipo de bombeo, por lo cual se optó por una carga alternativa, siendo esta una luminaria incandescente de 100 Watts



Figura 3.13: Lectura de voltaje línea a línea

Referencias generales

- [1] coparoman. [En línea]. Available: <https://www.google.com/sv/search?q=bombas+de+agua+automatizadas&>.
- [2] J. P. P. P. 2016., «Definicion.de,» 2016. [En línea]. Available: <http://definicion.de/automatizacion/>. [Último acceso: 20 Junio 2016].
- [3] F. Electric, «Franklin Electric,» 15 Noviembre 1944. [En línea]. Available: <http://franklinagua.com/>.
- [4] g. maps, «google maps,» [En línea]. Available: <https://www.google.com/sv/maps/place/Santiago+Nonualco/@13.5312365,-88.942672,1032m/>.
- [5] usinainfo, insainfo, 14 Octubre 2008. [En línea]. Available: <http://www.usinainfo.com.br>. [Último acceso: 10 Julio 2016].
- [6] comohacer.eu. [En línea]. Available: <http://comohacer.eu/configurar-raspberry-pi-desde-cero/>.
- [7] O. e. m. Arduino, «open energy monitor,» [En línea]. Available: <https://learn.openenergymonitor.org/electricity-monitoring/ctac/how-to-build-an-arduino-energy-monitor>.
- [8] R. Sofware, «RASPBerry PI FOUNDATION,» [En línea]. Available: <https://www.raspberrypi.org/downloads/>.
- [9] R. p. B+, Raspberry PI, 6 Septiembre 2014. [En línea]. Available: <https://www.raspberrypi.org/products/model-b-plus/>.
- [10] R. Lan9514. [En línea]. Available: <https://www.google.com/sv/search?q=Diagrama+LAN9514&>.
- [11] L. Labs, «Documentos de modulos,» [En línea]. Available: <http://docs.link-labs.com/m/52162/1/477731-hardware-specifications>.
- [12] s. p. d. l. Labs. [En línea]. Available: <https://www.link-labs.com/>.
- [13] E. SCHEMATIC. [En línea]. Available: <http://www.electroschematics.com/7958/arduino-uno-pinout/>.
- [14] FOTEK, «FOTEK RELÉ DE ESTADO SOLIDO,» [En línea]. Available: <http://www.fotek.com.hk/page1e.htm>.
- [15] t. d. corriente. [En línea]. Available: <https://www.google.com/sv/>

- [16] Python, «python documentos,» [En línea]. Available: <https://www.python.org/doc/>.

- [17] LL-RXR-27-USB, «link labs,» [En línea]. Available: <https://www.google.com/sv/search?q=ll-rxr-27&espv=2&biw=988&bih=447&>.
- [18] L. Labs y L. L. Documentos, «Link Labs,» Link Labs, [En línea]. Available: <http://docs.link-labs.com/m/52162/1/478033-modes-of-operation>. [Último acceso: 20 01 2016].

- [19] R. Mobile, Radio Mobile, 9 Julio 2000. [En línea]. Available: <http://www.cplus.org/rmw/rme.html>. [Último acceso: 23 Marzo 2016].
- [20] WIKIPEDIA. [En línea]. Available: <https://es.wikipedia.org/wiki/Wikipedia>.
- [21] AliExpress, «distribuidora de equipo electrónico,» Distribuidora de equipo Eléctrico, 15 06 2008. [En línea]. Available: <https://www.google.com/sv/search?q=rele+de+corriente+alterna&>. [Último acceso: 25 2 2016].
- [22] A. primium, «Amason.es,» [En línea]. Available: <https://www.amazon.es/YHDC-Transformador-corriente-120A-40mA/dp/B01EFVKSZG>.
- [23] b. d. i. p. modulos. [En línea]. Available: http://www.link-labs.com/doxygen/group__link__labs__interface__library.html.
- [24] E. Electronic. [En línea]. Available: <http://www.electronicaembajadores.com/datos/pdf1/ss/ssmg/ssmgc30.pdf>.
- [25] R. p. GPIO, «pinterest,» [En línea]. Available: <https://www.google.com/sv/search?q=pinnes+gpio+raspberry+pi+3&>.

Recomendaciones generales

- Para los lugares donde no existe una línea vista entre las antenas es posible establecer la comunicación entre los módulos de comunicación utilizando repetidores activos o pasivos.
- Para una mayor comodidad y transparencia de todo el proceso ante el usuario final, el uso de una interfaz gráfica sería de gran utilidad.
- Se sugiere la instalación de un regulador de voltaje para mantener estable la señal de alimentación del sistema de bombeo
- Ante la posibilidad de fallas en la red eléctrica de la distribuidora que brinda el servicio de energía en la zona, se considera que tanto el sitio remoto como el de control deberían estar conectados a una unidad de respaldo de energía (UPS), para poder finalizar procesos y para protección de los diferentes elementos del sistema.
- En caso de cambio de Hardware, se debe verificar que el software necesario este instalado correctamente en el nuevo dispositivo.
- En la actualidad, existen opciones similares a los radiotransmisores utilizados en el presente trabajo con un menor costo de adquisición, por lo cual se debe estudiar su uso en función de la compatibilidad con los demás componentes del sistema y la portabilidad que puedan llegar a ofrecer.

CONCLUSIONES GENERALES

- ☐ Se determinó que el sistema desarrollado para el control automático sobre el sistema de bombeo es operativo y viable para ser implementado en la zona.
- ☐ Se comprueba que la implementación de un sistema de este tipo conlleva una mayor eficiencia en la distribución del agua potable ya que reduce los tiempos de operación, logrando que el servicio sea brindado con más regularidad en la comunidad.
- ☐ Un de las ventajas más significativas del sistema implementado es que puede aplicarse a diferentes ambientes y que tiene la capacidad de manejar diferentes tipos de cargas eléctricas monofásicas.
- ☐ La explotación del espacio de banda libre entre ellas los 915 MHz permitiría a los beneficiarios del sistema no depender de un sistema de comunicación privado generando un ahorro a los encargados de administrar dicho proyecto.
- ☐ Se verificó que los datos proporcionados por los sensores utilizados poseen un alto grado de confiabilidad ya que de la precisión de los mismos depende el correcto funcionamiento del sistema de bombeo.
- ☐ A nivel de costos el sistema diseñado e implementado tiene un valor de \$ 760.64. En comparación a una automatización industrial este se convierte en una opción más económica.
- ☐ Para corroborar los fines y alcances que se proyectaron en el perfil de trabajo, se puede decir que los objetivos han sido alcanzados en su totalidad. Además, los resultados han sido precisos en función del objetivo general del proyecto.

Anexos

Anexo A: CODIGOS DE PROGRAMACION UTILIZADOS

A.1: Código LastVersion.INO

```
#include "EmonLib.h" // Se utiliza la librería EmonLib
EnergyMonitor sensores; // Se crea una instancia de la clase

char dato;
int led=13;
void setup(){
  Serial.begin(9600); //Velocidad utilizada
  pinMode(led,OUTPUT); //Se configura el pin 13 como salida
  sensores.voltage(2,112.5,1.7); // (pin de monitoreo, calibración)
  sensores.current(1,52.5); // (pin de monitoreo, calibración)
}

void loop(){
  if(Serial.available(>0){ //Se espera hasta que haya una petición
    dato=Serial.read(); //se captura el dato recibido
    if (dato=='0'){ //cero=apagar el sistema
      digitalWrite(led,LOW);
      Serial.println("0"); //apagado correctamente
    }
    if(dato=='1'){ //uno= encender el sistema
      sensores.calcVI(20,2000); //(ciclos analizados, tiempo máx)
      float voltaje=sensores.Vrms; //se obtiene el valor de V leído
      if(voltaje<105 || voltaje>130){ // se valida el valor de V
        Serial.print("2");
        sensores.serialprint(); //Un 2 significa voltaje inadecuado
      }
      else{
        digitalWrite(led,HIGH); //pone en alto la salida 13
        Serial.print("1");
        sensores.serialprint(); //se envía un 1 más V y C leídos
      }
    }
  }
  if(dato=='7'){ //actualiza valores de V y C
    sensores.calcVI(20,2000);
    sensores.serialprint();
  }
}
}
```

A.2: Código Fuente.py

```
#!/usr/bin/python

from subprocess import call
from subprocess import check_output
import serial

#configuraciones
call(["./nomac_test", "-C", "1"]) #coding rate 4/5
call(["./nomac_test", "-S", "8"]) #spreading factor
call(["./nomac_test", "--mac_set", "0"]) #modo NoMac
call(["./nomac_test", "-f", "915000000"]) #Frecuencia
call(["./nomac_test", "-p", "26"]) #potencia
call(["./nomac_test", "-w", "1"]) #Bandwith 62.5 KHz
call(["./nomac_test", "--antenna_set", "1"]) #antena externa
call(["./nomac_test", "-a"]) #Save
call(["./nomac_test", "-g"]) #Comprobación de para
print("Accesando al sistema de control remoto...")
print("Esperando petición del centro de monitoreo")
#a=check_output(["./nomac_test", "-r"]) #Eschar al CC
a='inicio'
lon=len(a)
var=1
while var==1: #loop principal
    while(lon<=163):
        a=check_output(["./nomac_test", "-r"])
        lon=len(a)
        #print lon
        #print a
    codigo=a[163:164]
    if codigo=='0':
        info=serial.Serial('/dev/ttyACM0', 9600, timeout=3)
        info.write('0')
        retorno=info.readline()
        if retorno[0:1]=='0':
            call(["./nomac_test", "-t", "0"])
    if codigo=='1':
        info=serial.Serial('/dev/ttyACM0', 9600, timeout=3)
        info.write('1')
        retorno=info.readline()
        if retorno[0:1]=='1':
            call(["./nomac_test", "-t", retorno])
        if retorno[0:1]=='2':
            call(["./nomac_test", "-t", retorno])
    if codigo=='7':
        info=serial.Serial('/dev/ttyACM0', 9600, timeout=3)
        info.write('7')
        retorno=info.readline()
        call(["./nomac_test", "-t", retorno])
    a='inicio'
    lon=len(a)
```

A.3: Código Monitor_Center.py

```
#!/usr/bin/python

from subprocess import call
from subprocess import check_output
from sys import exit

call(["./nomac_test", "-C", "1"]) #seteamos coding rate a
4/5
call(["./nomac_test", "-S", "8"]) #Seteamos spreading factor
call(["./nomac_test", "--mac_set", "0"]) #Seteamos en modo NoMac
call(["./nomac_test", "-f", "915000000"]) #Seteamos la frecuencia de
operacion
call(["./nomac_test", "-p", "26"]) #Seteamos la potencia de
operacion
call(["./nomac_test", "-w", "1"]) #ancho de banda de 125 KHz

call(["./nomac_test", "--antenna_set", "1"]) #Para utilizar antena
externa
call(["./nomac_test", "-a"]) #Se guarda la configuración
call(["./nomac_test", "-g"]) #Comprobación de configuración
correcta

opcion=0
while opcion!=1 and opcion!=2:
    opcion=input("Bienvenido al centro de control, presione 1 para
encender el sistema, o 2 para salir \n")
if opcion==1:
    call(["./nomac_test", "-t", "1"]) #mandamos comando de
encendido
    a=check_output(["./nomac_test", "-r"]) #esperamos respuesta del
lado remoto por diez segundos
    respuesta=a[163:164]
    #print a
    #print respuesta
    val=len(a)
    while (val<=163):
        accion=input("Imposible conectar con el sistema remoto:
presione 0 para salir o 1 para reintentar")
        if accion==0:
            print("La sesion ha terminado, gracias")
            exit()
        if accion==1:
            call(["./nomac_test", "-t", "1"]) #mandamos
comando de encendido
            a=check_output(["./nomac_test", "-r"]) #esperamos
respuesta del lado remoto
            respuesta=a[163:164]
            val=len(a)
        else:
            print("Opción Invalida")
            exit()

    if respuesta=='1':
        #print a
        #opcion2=2
        print("El sistema ha iniciado correctamente")
```

```

    voltaje=a[164:170]
    corriente=a[170:174]
    print 'Voltaje inicial: %s' % (voltaje)
    print 'Corriente Inicial: %s' % (corriente)
    opcion2=input("\nPresione uno para actualizar valores o cero
para apagar: \n")
    while opcion2!=0 and opcion2==1:
        call(["./nomac_test","-t","7"])          #mandamos
comando de encendido
        a=check_output(["./nomac_test","-r"])    #esperamos
respuesta del lado remoto
        voltaje=a[163:169]
        corriente=a[169:174]
        print 'Voltaje: %s' % (voltaje)
        print 'Corriente: %s' % (corriente)
        opcion2=input("\nPresione uno para actualizar valores o
cero para apagar: \n")
        respuesta2=1
        while(respuesta2!='0'):
            call(["./nomac_test","-t","0"])
            a=check_output(["./nomac_test","-r"])    #esperamos
respuesta del lado remoto
            respuesta2=a[163:164]
            print("El sistema se ha apagado y cerrado correctamente")
            exit()
        if respuesta=='2':
            print("Por el momento no es posible arrancar el sistema,
voltaje inadecuado. Por favor intente más tarde")
            voltaje=a[164:170]
            print 'Voltaje= %s' % (voltaje)
            exit()

elif opcion==2:
    print("La sesion ha terminado, gracias")
    exit()

```

Anexo B: Costo de materiales

En las siguientes tablas se resumen el precio de cada uno de los sensores, módulos, periféricos, tarjeta Arduino, tarjeta Raspberry pi3 así como los dispositivos o elementos que conforman cada uno de las etapas que conforman el sistema de automatización en planta de bombeo. En la tabla A0, los asteriscos representan los elementos importados fuera del país estos ya incluyen el porcentaje del envío.

Cantidad	Descripción	Precio Unitario \$	Precio \$
1	Tarjeta Raspberry pi3*	67.0	67.0
1	Tarjeta Arduino uno*	41.0	41.0
1	Módulo LL-RXR-27	155.00	310.00
1	Mouse USB	4.00	4.00
1	Teclado USB	6.00	6.00
1	Case para Arduino Uno	5.00	5.00
1	Sensor de corriente no invasivo 30A*	25.00	25.0
1	Sensor de corriente invasivo ACS712 30A	14.00	14.0
1	Sensor de voltaje ATTOPILOT 180A*	35.00	35.00
1	Monitor 19" con entrada HDMI	90.00	90.0
1	Case para Raspberry	6.50	6.50
1	Arduino DUE*	39.00	39.00
TOTAL			\$ 642.50

Tabla B0. Precios de Tarjeta, Módulos, sensores y Periféricos.

Cantidad	Descripción	Precio Unitario \$	Precio \$
1	Antena yagi	63.73	63.73
10	Base para pines TER-1X8/HS	0.59	5.90
4	Capacitores de 100Nf	0.45	1.80
4	Resistencia de 4.7k X 1/4W	0.20	0.80
4	Resistencias de 220 ohm X 1/4W	0.20	0.80
2	Resistencias de 2k X 1/2W	0.27	0.54
12	Resistencias de 100k X 1/2W	0.22	2.64
2	Resistencias de 10k X 1/4W	0.20	0.40
2	Potenciómetro de 1K	1.12	2.24
1	Juego de cables para Arduino	19.00	19.00
TOTAL			\$ 98.25

Tabla B1. Precios para amplificadores de señal.

Cantidad	Descripción	Precio Unitario \$	Precio \$
1	Cable video HDMI	2.95	2.95
1	Memoria USB 8GB	3.75	3.75
4	Capacitores electrolíticos 10 μ F X 25V	0.30	1.20
2	Capacitores electrolíticos 1 μ F X 25V	0.30	0.60
1	Regulador de Voltaje LM7906	0.18	0.18
1	Regulador de Voltaje LM7806	1.10	1.10
1	Tabla para elaboración de maqueta de pruebas	10.00	10.00
		TOTAL	\$19.88

Tabla B2. Precios para la instalación de equipo.

Anexo C: Datasheet

Anexo c1: Datasheet de módulos RF



SPEC SHEET

Symphony Module LL-RXR-27

The LL-RXR-27 is a highly integrated, bidirectional, multi-band radio transceiver module. It is compatible with LoRaWAN 1.0 public networks and Link Labs' Symphony Link™ system for wide area, private networking. The modules use Semtech's LoRa™ modulation to maximize range while minimizing power consumption. The network stack and command interface is controlled by a low power 32-bit microcontroller.

Key Features

- >-155 dBm sensitivity
- Low Power
- Small form factor
- FCC/IC/CE/ETSI compliance
- Integrated PCB antenna and U.FL connector
- Multi-band operation
- Supports both Symphony Link and LoRaWAN
- Small Host driver - C source available
- Window GUI for evaluation

Model Numbers

w/ Trace Antenna	w/o Trace Antenna
Symphony Link	
LL-RXR-27-915-SYM-A	LL-RXR-27-915-SYM-B
LL-RXR-27-868-SYM-A	LL-RXR-27-868-SYM-B
LoRaWAN	
LL-RXR-27-915-LRW-A	LL-RXR-27-915-LRW-B
LL-RXR-27-868-LRW-A	LL-RXR-27-868-LRW-B

For ordering, or additional questions:
info.link-labs.com/contact
 +1 (202) 524-1390

Additional product info:
<http://docs.link-labs.com>

Warranty and repair info:
support@link-labs.com



Communication Protocols	LoRaWAN 1.0; Symphony Link
Interfaces	UART (Rx, Tx)
Dimensions	24.5 x 29 mm (with trace antenna); 15.8 x 29 mm (without antenna)
Modulation	LoRa Digital Spread Spectrum; FSK (future use)
Frequency	902-928 MHz ISM (US); 868 MHz ETSI (EU)
MCU	Renesas - R5F5116ADNE
Sleep Current	<1uA
Max Clock	32 MHz
Flash Memory	256 KB
RAM	32 KB
Max. Transmitter Power Output	23 dBm
Max. Receive Sensitivity	-137 dBm (spreading factor 10 – 125 kHz)
Link Budget	160 dB point to multi-point
Maximum EIRP	24.9 dBm (dipole antenna), 20 dBm (trace antenna)
Operating Temperature	-20° to +70°C
Storage Temperature	-40° to +85°C
Compliance	FCC Part 15 Class B, Etsi certified, IC
Other	UL/cUL 60950-1 2nd Ed., CE

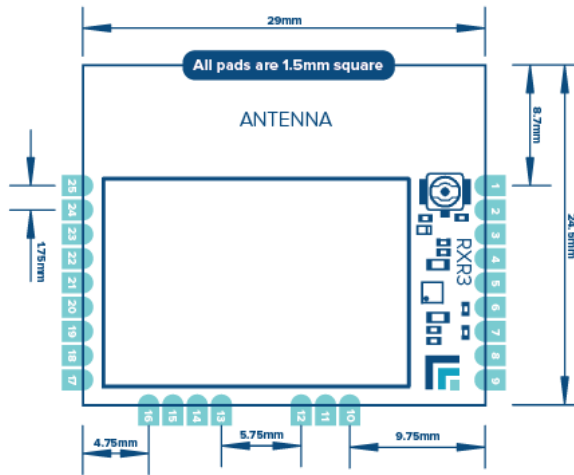
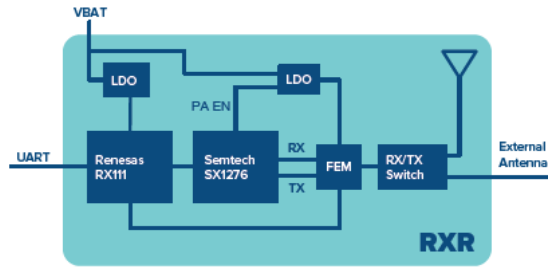


PINOUT

Symphony Module LL-RXR-27

Notes:

- RF connection is via U.FL connector on upper left side of the module or Pin 2
- RF output is 50 Ω , DC blocking capacitor not required
- ALL I/O signals must be 3.3 V. If necessary, pin 4 (3.3 OUT) may be used to supply an off-module level shifter
- Default UART baud rate is 115,200 8n1
- If possible, keep area under module free of copper. If not possible, ensure all features are masked
- If using VBAT, you must tie LDO_OUT to VCC



Pin #	Name	Type	Description
2	ANT	RF I/O	RF path if not using U.FL connector
3	VBAT	Input Voltage	Optional unregulated supply voltage 3.5 to 5.5V
4	LDO_OUT	Output Voltage	Power digital. 3.3V
5	VCC	Input Voltage	Regulated supply voltage. 2.5 - 3.3V
7	nRESET	I	External reset pin, active low
8	HOST_J00	O	Status message indicator

Pin #	Name	Type	Description
17	BOOT	I	NC = Boot normally, GND = Bootloader
23	HOST_RXIN	I	UART interface: module Rx
24	HOST_TXOUT	O	UART interface: module Tx
1, 9, 25	GND	Ground	-
6, 10, 11-16, 18-22	Reserved	-	-

For ordering, or additional questions:

info.link-labs.com/contact
+1 (202) 524-1390

Additional product info:
<http://docs.link-labs.com>

Warranty and repair info:
support@link-labs.com

P2N2222A

Amplifier Transistors

NPN Silicon

Features

- These are Pb-Free Devices*

MAXIMUM RATINGS (T_A = 25°C unless otherwise noted)

Characteristic	Symbol	Value	Unit
Collector - Emitter Voltage	V _{CEO}	40	Vdc
Collector - Base Voltage	V _{CBO}	75	Vdc
Emitter - Base Voltage	V _{EBO}	6.0	Vdc
Collector Current - Continuous	I _C	600	mAdc
Total Device Dissipation @ T _A = 25°C Derate above 25°C	P _D	625 5.0	mW mW/°C
Total Device Dissipation @ T _C = 25°C Derate above 25°C	P _D	1.5 12	W mW/°C
Operating and Storage Junction Temperature Range	T _J , T _{stg}	-55 to +150	°C

THERMAL CHARACTERISTICS

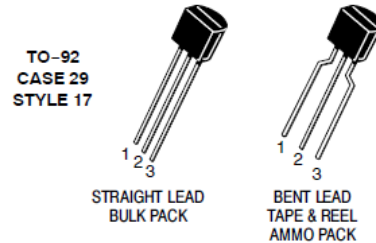
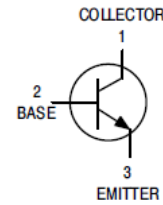
Characteristic	Symbol	Max	Unit
Thermal Resistance, Junction to Ambient	R _{θJA}	200	°C/W
Thermal Resistance, Junction to Case	R _{θJC}	83.3	°C/W

Stresses exceeding Maximum Ratings may damage the device. Maximum Ratings are stress ratings only. Functional operation above the Recommended Operating Conditions is not implied. Extended exposure to stresses above the Recommended Operating Conditions may affect device reliability.

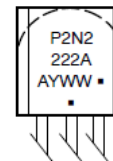


ON Semiconductor®

<http://onsemi.com>



MARKING DIAGRAM



- A = Assembly Location
 - Y = Year
 - WW = Work Week
 - = Pb-Free Package
- (Note: Microdot may be in either location)

ORDERING INFORMATION

Device	Package	Shipping†
P2N2222AG	TO-92 (Pb-Free)	5000 Units/Bulk
P2N2222ARL1G	TO-92 (Pb-Free)	2000/Tape & Ammo

†For information on tape and reel specifications, including part orientation and tape sizes, please refer to our Tape and Reel Packaging Specification Brochure, BRD8011/D.

*For additional information on our Pb-Free strategy and soldering details, please download the ON Semiconductor Soldering and Mounting Techniques Reference Manual, SOLDERRM/D.

P2N2222A

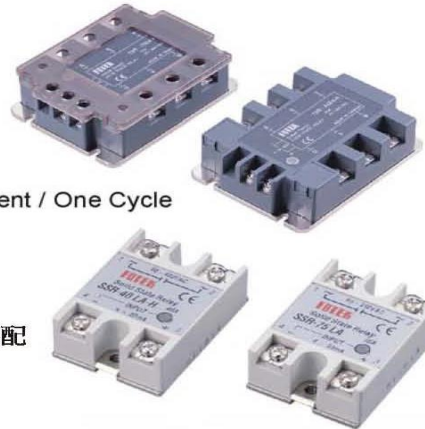
ELECTRICAL CHARACTERISTICS (T_A = 25°C unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
OFF CHARACTERISTICS				
Collector - Emitter Breakdown Voltage (I _C = 10 mA _{dc} , I _B = 0)	V _{(BR)CEO}	40	-	V _{dc}
Collector - Base Breakdown Voltage (I _C = 10 μA _{dc} , I _E = 0)	V _{(BR)CBO}	75	-	V _{dc}
Emitter - Base Breakdown Voltage (I _E = 10 μA _{dc} , I _C = 0)	V _{(BR)EBO}	6.0	-	V _{dc}
Collector Cutoff Current (V _{CE} = 60 V _{dc} , V _{EB(off)} = 3.0 V _{dc})	I _{CEX}	-	10	nA _{dc}
Collector Cutoff Current (V _{CB} = 60 V _{dc} , I _E = 0) (V _{CB} = 60 V _{dc} , I _E = 0, T _A = 150°C)	I _{CBO}	-	0.01 10	μA _{dc}
Emitter Cutoff Current (V _{EB} = 3.0 V _{dc} , I _C = 0)	I _{EBO}	-	10	nA _{dc}
Collector Cutoff Current (V _{CE} = 10 V)	I _{CEO}	-	10	nA _{dc}
Base Cutoff Current (V _{CE} = 60 V _{dc} , V _{EB(off)} = 3.0 V _{dc})	I _{BEX}	-	20	nA _{dc}
ON CHARACTERISTICS				
DC Current Gain (I _C = 0.1 mA _{dc} , V _{CE} = 10 V _{dc}) (I _C = 1.0 mA _{dc} , V _{CE} = 10 V _{dc}) (I _C = 10 mA _{dc} , V _{CE} = 10 V _{dc}) (I _C = 10 mA _{dc} , V _{CE} = 10 V _{dc} , T _A = -55°C) (I _C = 150 mA _{dc} , V _{CE} = 10 V _{dc}) (Note 1) (I _C = 150 mA _{dc} , V _{CE} = 1.0 V _{dc}) (Note 1) (I _C = 500 mA _{dc} , V _{CE} = 10 V _{dc}) (Note 1)	h _{FE}	35 50 75 35 100 50 40	- - - - 300 - -	-
Collector - Emitter Saturation Voltage (Note 1) (I _C = 150 mA _{dc} , I _B = 15 mA _{dc}) (I _C = 500 mA _{dc} , I _B = 50 mA _{dc})	V _{CE(sat)}	- -	0.3 1.0	V _{dc}
Base - Emitter Saturation Voltage (Note 1) (I _C = 150 mA _{dc} , I _B = 15 mA _{dc}) (I _C = 500 mA _{dc} , I _B = 50 mA _{dc})	V _{BE(sat)}	0.6 -	1.2 2.0	V _{dc}
SMALL-SIGNAL CHARACTERISTICS				
Current - Gain - Bandwidth Product (Note 2) (I _C = 20 mA _{dc} , V _{CE} = 20 V _{dc} , f = 100 MHz) _C	f _T	300	-	MHz
Output Capacitance (V _{CB} = 10 V _{dc} , I _E = 0, f = 1.0 MHz)	C _{obo}	-	8.0	pF
Input Capacitance (V _{EB} = 0.5 V _{dc} , I _C = 0, f = 1.0 MHz)	C _{ibo}	-	25	pF
Input Impedance (I _C = 1.0 mA _{dc} , V _{CE} = 10 V _{dc} , f = 1.0 kHz) (I _C = 10 mA _{dc} , V _{CE} = 10 V _{dc} , f = 1.0 kHz)	h _{ie}	2.0 0.25	8.0 1.25	kΩ
Voltage Feedback Ratio (I _C = 1.0 mA _{dc} , V _{CE} = 10 V _{dc} , f = 1.0 kHz) (I _C = 10 mA _{dc} , V _{CE} = 10 V _{dc} , f = 1.0 kHz)	h _{re}	- -	8.0 4.0	X 10 ⁻⁴
Small-Signal Current Gain (I _C = 1.0 mA _{dc} , V _{CE} = 10 V _{dc} , f = 1.0 kHz) (I _C = 10 mA _{dc} , V _{CE} = 10 V _{dc} , f = 1.0 kHz)	h _{fe}	50 75	300 375	-
Output Admittance (I _C = 1.0 mA _{dc} , V _{CE} = 10 V _{dc} , f = 1.0 kHz) (I _C = 10 mA _{dc} , V _{CE} = 10 V _{dc} , f = 1.0 kHz)	h _{oe}	5.0 25	35 200	μMhos
Collector Base Time Constant (I _E = 20 mA _{dc} , V _{CB} = 20 V _{dc} , f = 31.8 MHz)	rb'C _c	-	150	ps
Noise Figure (I _C = 100 μA _{dc} , V _{CE} = 10 V _{dc} , R _S = 1.0 kΩ, f = 1.0 kHz)	N _F	-	4.0	dB

1. Pulse Test: Pulse Width ≤ 300 μs, Duty Cycle ≤ 2.0%.
2. f_T is defined as the frequency at which |h_{fe}| extrapolates to unity.

FOTEK SSR SERIES SOLID STATE RELAY

- ★ High Reliability by S.M.T. & TQC. (Surface Mounting Technology)
- ★ High Isolation over than 50MΩ / 500VDC
- ★ High Dielectric over than 2.5KV
- ★ Low Enable Current less than 7.5mA / 12VDC
C MOS IC or TTL Compatible.
- ★ Low EMI / EFI & Surge by Zero Cross Trigger Method.
- ★ High Surge Current Duration Over 10 Times of Rated Current / One Cycle
- ★ High Surge Voltage Duration by Snubber Circuit.
- ★ 表面貼焊技術及全面品管，提供最可靠的 SSR
- ★ 高絕緣阻抗超過 50MΩ / 500VDC
- ★ 高耐壓超過 2.5KV
- ★ 低觸發電流小於 7.5mA / 12VDC，可與 CMOS IC及 TTL 匹配
- ★ 零點觸發方式避免電磁 / 高頻干擾
- ★ 高耐過電流，超過10倍額定電流/一週期
- ★ 具緩衝迴路可吸收瞬間突破



Guiding of Model

Terminal Type

Ex. **SSR- 40 D A-H**

① ② ③ ④ ⑤

- ① **Product 產品別**
SSR:SINGLE PHASE SOLID STATE RELAY
單相固態繼電器
STR:THREE PHASE SOLID STATE RELAY
三相固態繼電器
- ② **Output Current 輸出電流**
10 : 10A 25 : 25A
40 : 40A 50 : 50A
75 : 75A
- ③ **Input Voltage 輸入方式**
D:DC3W~32V<ON/OFF>
A:AC80~250V<ON/OFF>
L:4~20mA(linear)
V:VARIABLE RESISTER
- ④ **Output Voltage 輸出電壓**
A:AC VOLTAGE
D:DC VOLTAGE
- ⑤ **Output Voltage Range 輸出電壓範圍**
H : High Voltage Type < 90 ~ 480VAC >
Non: Standard Type < 24 ~ 380VAC >

PCB Type

Ex. **SSR -P 03 D A**

① ② ③ ④ ⑤

- ① **Product 產品別**
SSR : SINGLE PHASE SOLID ST
- ② **Mounting Method 固定方式**
PCB:PCB TYPE
- ③ **Output Current 輸出電流**
03 = 3A
05 = 5A
- ④ **Input Method 輸入方式**
D : DC VOLTAGE
- ⑤ **Output Voltage 輸出電壓**
D : DC 5 ~ 60V
A : AC 24 ~ 280V

Type	Terminal Type					PCB Type
Model	SSR-10DA	SSR-25DA	SSR-40DA	SSR-25DA-H	SSR-40DA-H	SSR-P03DA
Rated Load Current	10A	25A	40A	25A	40A	3A
Input Data						
Operating Voltage	3~32VDC					
Min. ON / OFF Voltage	ON > 2.4V , OFF < 1.0V					
Trigger Current	7.5mA / 12V					
Control Method	Zero Cross Trigger					
Output Data						
Operating Voltage	24~380VAC		90~480VAC		24~380VAC	
Min. Black Voltage	600 VAC < Repetive >					
Voltage Drop	1.6 V / 25 C					
Max. Durated Current	135A	275A	410A	275A	410A	135A
Leakage Current	3.0mA	3.0mA	3.0mA	5.0mA	5.0mA	3.0mA
Response Time	ON < 10ms , OFF < 10ms					
General Data						
Dielectric Strength	Over 2.5KVAC / 1min.					
Isolation Strength	Over 50MΩ / 500VDC					
Operating Temperature	-20 C ~+80 C					
Housing Material	Intensive ABS					
Weight	Appr. 105g					Appr. 15g
Connection Diagram						

Application Hints

Input	NPN	Output	NO	Input	PNP	Output	NO	Input	L.S.	Output	NO
Input	NPN	Output	NC	Input	PNP	Output	NC	Latch Circuit (AC to AC)			