

UNIVERSIDAD NACIONAL DE INGENIERÍA
RECINTO UNIVERSITARIO SIMON BOLIVAR
FACULTAD DE ELECTROTECNIA Y COMPUTACION



INFORME FINAL DE TRABAJO MONOGRÁFICO PARA OPTAR AL TÍTULO
DE INGENIERO EN ELECTRÓNICA

“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE RIEGO POR GOTEO
AUTOMATIZADO PARA EL CULTIVO DEL CAFÉ USANDO TECNOLOGÍA
DE MICROCONTROLADOR Y ESTABLECIENDO UN SISTEMA DE
CONTROL INALÁMBRICO EN LA COMUNIDAD DE TOTOGALPA”

AUTORES:

Br. Maynor Antonio Flores Chávez.

Br. José Francisco Rodríguez Castro.

TUTOR:

Msc. Saúl Otoniel Núñez Zeledón.

ASESOR:

Ing. Álvaro Antonio Gaitán.

Febrero del 2018

Managua, Nicaragua

DEDICATORIA

A Dios por darme la oportunidad de vivir, una vida llena de experiencias y aventuras, por haberme permitido llegar hasta esta etapa de mi vida y ser fiel para conmigo en buenos y malos momentos de mi vida.

A mis padres Felipe Flores Martínez y Miriam Chávez Jarquín por haberme brindado su apoyo incondicional; a mis hermanos Cristhian Josué Flores Chávez, Miriam de los Ángeles Flores Chávez y Richard David Flores Chávez que han sido grandes pilares a mi vida.

Maynor Flores

Dedico este trabajo de tesis primeramente a Dios todo poderoso por proveerme sabiduría, perseverancia, fortaleza, habilidades intelectuales para lograr mis metas como culminar mi carrera.

A mis padres Francisco Rodríguez y María Castro. Por trabajar duro para lograr sacarme adelante y guiarme en la senda de la rectitud, enseñarme a ser humilde, honesto, disciplinado, perseverante y de muy buenos principios morales, por brindarme todo su amor y comprensión, por corregirme para lograr ser una buena persona y ayudar a quien me necesite por el apoyo brindado en los momentos adversos en mi vida y por creer en mí para poder lograr mis objetivos como finalizar mis estudios universitarios.

De manera muy especial a nuestro maestro Saúl Núñez por depositar su confianza, Por ser pacientes, tolerantes, y proveernos de conocimientos y experiencia que fueron las herramientas del éxito de nuestra tesis.

José Rodríguez

AGRADECIMIENTOS

A Dios sobre todas las cosas, por darnos la fortaleza para cumplir con esta meta, por llenarnos de sabiduría y por estar a nuestro lado cuando más lo necesitábamos.

A nuestros padres por sus buenos consejos, por ser padres ejemplares que nos han enseñado a no darse por vencido, ser personas persistentes y valientes.

A nuestras familias, por su gran apoyo incondicional que durante todo este tiempo de estudio fueron también una guía e inspiración para cumplir con esta meta.

A nuestro tutor de tesis Saúl Núñez y asesor el profesor Álvaro Gaitán, por el tiempo dedicado a la revisión de este trabajo.

Al profesor PhD. Pritpal Singh de la universidad de VILLANOVA, por todo el apoyo brindado durante la realización de este proyecto.

A nuestros compañeros de clases por sus apoyos en la realización de este proyecto.

A junior Chávez por proveernos de herramientas necesarios para la realización de este proyecto.

A nuestros profesores, por la enseñanza y esmero en la impartición de sus clases hacia todos nosotros.

Este proyecto es el resultado del esfuerzo en conjunto de quienes formamos el grupo de trabajo. Al ver hoy el resultado por la constancia y dedicación compartimos el logro alcanzado. A todos y cada una de las personas agradecemos de ante mano por el apoyo brindado en la realización de este trabajo.

Maynor Flores y José Castro.

RESUMEN

Este trabajo monográfico tiene el propósito de brindar una alternativa de solución al problema de riego del café que afecta al grupo Promotores Solares Agroecológicos en la comunidad de Totogalpa. El problema radica en la técnica de riego utilizada para el plantío, carencia del recurso hídrico en el sitio, además del derroche de este en los tanques. De las opciones evaluadas para dar solución al problema se optó por implementar un sistema de riego automatizado para el cultivo de café en la comunidad Totogalpa.

El sistema es capaz de monitorear: el estado de un tanque para el abastecimiento del recurso hídrico hacia el plantío, control del cierre-abierto de las electroválvulas del tanque, el estado del plantío de café a través de variables de entrada (humedad y temperatura) y enviadas por un sistema de radiofrecuencia entre las unidades, además el control de una bomba dc sumergible. Para el procesamiento de datos se utilizó el microprocesador atmega2560 incorporada en una placa Arduino.

Se presenta el diseño de cada uno de los sistemas de las unidades que conforman el prototipo. Esto incluye el desarrollo de hardware y software necesario. De igual forma se presenta el desarrollo de cálculo de sistema fotovoltaico para las mismas.

Para el desarrollo de este proyecto se conformó un grupo con miembros de la Universidad Nacional de Ingeniería (UNI) y la universidad de VILLANOVA con sede en Pensilvania, Estados Unidos.

LISTA DE FIGURAS

Figura 1. Representación esquemática de una estoma. (Allen, 2006)	8
Figura 2.Evapotranspiración del cultivo de referencia (ET_0), bajo condiciones estándar (ET_c) y bajo condiciones no estándar ($ET_{c_{aj}}$). (Allen, 2006)	9
Figura 3. Método de riego por goteo	10
Figura 4. ARDUINO MEGA 2560	12
Figura 5. Edición de un programa en el entorno de programación.....	13
Figura 6.Topologías Zigbee	18
Figura 7.Xbee Pro Serie 2.....	19
Figura 8. Pestaña Terminal.	20
Figura 9. Pestaña PC Settings	20
Figura 10. .Pestaña Modem Configuration.	20
Figura 11. Shett Size Configuration, ofrece cinco tamaños de hoja y un tamaño definido por el usuario.....	22
Figura 12. Herramienta de modo.	22
Figura 13. Pantalla LCD 16X2.....	23
Figura 14.Sensor de humedad YL-69, comparador de voltaje, cable hembra-hembra.....	25
Figura 15.Sensor de humedad YL-69 y ARDUINO UNO	25
Figura 16.Sensor de temperatura LM35 y Arduino UNO.....	26
Figura 17.Sensor Ultrasónico HC-SR04	27
Figura 18. Funcionamiento del Sensor Ultrasónico HC-SR04.....	28
Figura 19. Arreglo de un sistema fotovoltaico. (Gilbert, 2004).....	30
Figura 20. Célula fotovoltaica, módulo y arreglo. (Gilbert, 2004).....	31
Figura 21.Instalación del sistema fotovoltaico para el sistema.....	32
Figura 22.Panel solar de 160W	33
Figura 23. Panel solar 60W	33
Figura 24. Controlador de carga 5 Amperio marca Epsolar.	34
Figura 25. Sistema de riego del café en la comunidad de Totogalpa.....	37
Figura 26.Cultivo de café comunidad de Totogalpa.....	38
Figura 27.Diagrama de bloque general sistema de riego	41
Figura 28.Diagrama de bloque general Unidad Central (Local P.S.A)	42
Figura 29.Sistema adquisición y representación de datos	43
Figura 30.Teclado Matricial 4x4.	43
Figura 31.Pantalla LCD 16x2.	44
Figura 32.Circuito propuesto para el sistema de adquisición y representación de datos, elaborado en software Proteus	47
Figura 33.Conexión del módulo xbee pro a Arduino.....	49
Figura 34.Shield conectado al módulo xbee pro.....	49
Figura 35.Jumpers XBee y USB del Shield	50
Figura 36. Diagrama de flujo de adquisición y representación de datos	52

Figura 37.Sistema de llenado.....	53
Figura 38. Electroválvula 1 pulgada.....	54
Figura 39.Electroválvula 1/2 pulgada.....	54
Figura 40.Módulo Relé.....	55
Figura 41.Circuito propuesto para el sistema de llenado, elaborado en software proteus.....	55
Figura 42.Diagrama de bloque alimentación Unidad Central.....	56
Figura 43.Consumo de energía Unidad central (Local P.S.A).....	59
Figura 44.Temperatura ambiente comunidad de Totogalpa (NASA, 2016).....	62
Figura 45.Comparación energía solar y consumo de energía por mes.....	64
Figura 46. Días de almacenamiento de batería para un sistema independiente con 95% y 99% de disponibilidad del sistema. (Gilbert, 2004).....	65
Figura 47.Diagrama circuital distribuidor de carga UC.....	66
Figura 48. Unidad Remota de medición.....	67
Figura 49.Sensor LM35.....	69
Figura 50.Arduino UNO.....	72
Figura 51.Circuito propuesto para la unidad remota de medición, elaborado en software Proteus.....	72
Figura 52. Diagrama de flujo de temperatura propuesto para la unidad remota de medición.....	73
Figura 53. Diagrama de flujo de humedad propuesto para la unidad remota.....	74
Figura 54.Diagrama de bloque de alimentación Unidad Remota.....	75
Figura 55.Diagrama de bloque de la unidad de bombeo.....	81
Figura 56.Circuito propuesto para la unidad de bombeo, elaborado en software Proteus.....	82
Figura 57.Conexión sistema fotovoltaico.....	83
Figura 58.Diagrama de cableado de controlador PCA-30-M1 SERIES.....	83
Figura 59. Implementación Unidad Remota.....	86
Figura 60. Implementación Unidad Central.....	87
Figura 61.Sistema de bombeo.....	88
Figura 62. Instalación de los equipos en la comunidad.....	89

LISTA DE TABLAS

Tabla 1. Instrucciones, variables del lenguaje de programación de ARDUINO	14
Tabla 2. Comparación tecnología WI-FI, BLUETOOTH Y ZIGBEE.....	15
Tabla 3. Características de pantallas candidatos.	44
Tabla 4. Características de los módulos de radio frecuencia candidatos.....	45
Tabla 5. Cacterísticas de controladores candidatos para la unidad central.	46
Tabla 6. Módulos Xbee pro en las tarjetas Arduino.	48
Tabla 7. Características de sensores de movimiento candidatos.....	53
Tabla 8. Promedio mensual incidencia de radiación en una superficie inclinada apuntada al ecuador en (KWh/m ² /día). (NASA, 2016)	57
Tabla 9. Consumo de energía Unidad Central.....	58
Tabla 10. Energía generada por el panel solar.....	63
Tabla 11. Característica sensores de humedad.	68
Tabla 12. Característica sensores de temperatura candidatos.	68
Tabla 13. Características de controladores candidatos para la Unidad Remota.....	71
Tabla 14. Consumo de energía Unidad Remota.....	76
Tabla 15. Energía generada por el sistema fotovoltaico Unidad Remota.	79
Tabla 16. Consumo de energía unidad de bombeo.....	84
Tabla 17. Costo del prototipo.	92

ÍNDICE

I.	INTRODUCCIÓN.....	1
II.	OBJETIVOS	2
III.	JUSTIFICACIÓN	3
CAPITULO 1: MARCO TEORICO.....		5
1.1.	Origen del café	5
1.1.1.	Temperatura del café	5
1.1.2.	Luz para el cultivo de café.....	6
1.1.3.	Crecimiento vegetativo del café	6
1.2.	Evapotranspiración de una planta.....	7
1.2.1.	Clasificación de evapotranspiración.....	8
1.3.	Riego por goteo y su origen	9
1.3.1.	Método de riego por goteo	10
1.3.2.	Ventaja de riego por goteo	10
1.3.3.	Programación del riego	11
1.4.	Arduino Mega 2560	11
1.4.1.	Elementos de Arduino Mega 2560.....	12
1.4.2.	Microcontrolador ATMEL ATMEGA 2560	12
1.4.3.	Memoria de datos (RAM)	12
1.4.4.	Puertos de Entrada/Salida	12
1.4.5.	Lenguaje de programación del IDE de ARDUINO	13
1.5.	Tecnología ZIGBEE.....	15
1.5.1.	Comparación entre WIFI, BLUETOOTH Y ZIGBEE	15
1.5.2.	Características técnicas de zigbee.....	16
1.5.3.	Módulo Xbee	18
1.6.	Software Proteus.....	21
1.6.1.	Hoja de trabajo	21
1.6.2.	Herramientas de modo.....	22
1.7.	Pantalla LCD 16x2.....	23
1.7.1.	Características de la LCD 16X2.....	23
1.8.	Importancia de los sensores de humedad	24
1.8.1.	Sensor de humedad del suelo YL-69.....	24
1.9.	Sensor de temperatura LM35.....	26
1.10.	Sensor de ultrasonidos HC-SR04	27
1.10.1.	Características eléctricas	28
1.10.2.	Funcionamiento del sensor HC-SR04.....	28
1.10.3.	Ecuación para determinar el nivel de agua en un tanque.....	29
1.11.	Sistema Fotovoltaico	29
1.11.1.	Radiación solar.....	30
1.11.2.	Células fotovoltaicas.....	30

1.11.3. Sistema fotovoltaico aislado	31
1.11.4. Elementos de un sistema fotovoltaico	32
1.12. Electroválvula	35
CAPITULO 2: ANALISIS Y PRSENTACIÓN DE RESULTADOS	36
2.1. Metodología de desarrollo	36
2.1.1. Análisis	36
2.1.2. Diseño del sistema automatizado de riego	39
CAPITULO 3: CONCLUSIONES Y RECOMENDACIONES	94
3.1. Conclusiones	94
3.2. Recomendaciones.....	96
BIBLIOGRAFÍA.....	97
ANEXOS.....	A

I. INTRODUCCIÓN

En este documento se describe el desarrollo del trabajo monográfico que tiene como propósito brindar una alternativa de solución al problema de riego del café que afecta al grupo Promotores Solares Agroecológicos en la comunidad Totogalpa¹, además incluye las herramientas teóricas que se necesitaron para el desarrollo del proyecto.

En la comunidad Totogalpa se encuentra un grupo de personas llamado “PROMOTORES SOLARES AGROECOLÓGICOS” a quienes se le hará referencia en el presente documento como (P.S.A)². Este grupo tiene un plantío de café de 300 metros cuadrados con una cantidad de 520 plantas de café.

Actualmente, el sistema de riego de dicho plantío no está en buen estado por tanto el grupo P.S.A realiza el riego de manera manual sin conocer con exactitud datos relevantes como humedad y temperatura de las plantas para su respectivo riego. Esto puede provocar una inadecuada gestación de la planta. Por tanto, se considera que el plantío de café del grupo P.S.A tiene la necesidad de automatizar el control del riego para el plantío, también el llenado de los tanques debido al derroche del recurso hídrico.

La alternativa de solución propuesta es el desarrollo de un sistema automatizado de riego por goteo utilizando la tecnología de microcontrolador y un sistema de comunicación inalámbrica en la comunidad Totogalpa. Se estableció un control inalámbrico ubicado en Local P.S.A. Esto permite conocer el nivel de llenado de los tanques, la humedad del suelo del plantío, la temperatura ambiente y el estado de una bomba dc sumergible.

Para el desarrollo de este proyecto se conformó un grupo con miembros de la Universidad Nacional de Ingeniería (UNI) y la Universidad de VILLANOVA con sede en Pensilvania, Estados Unidos. Los miembros de este grupo son Maynor Flores y José Rodríguez de la UNI y Nicole Rotyliano y Kurt Heidemann de la Universidad de VILLANOVA. Los miembros están siendo coordinados por el Msc. Saúl Núñez y el PhD. Pritpal Singh.

¹Totogalpa: Municipio de Madriz a 211 Km de Managua.

²(P.S.A): Promotores Solares Agroecológicos.

II. OBJETIVOS

Objetivo General

Diseñar un prototipo de sistema de riego por goteo para el cultivo del café en la comunidad de Totogalpa utilizando la tecnología de microcontrolador y un sistema de comunicación inalámbrica con el fin de conocer el factor climático de temperatura y humedad del plantío.

Objetivo Específicos

1. Controlar el flujo de agua en los tanques de riego mediante el uso de microcontrolador y módulos de radio frecuencia para el sistema de riego en la comunidad Totogalpa.
2. Desarrollar un sistema de control para el riego del café en la comunidad Totogalpa tomando en cuenta los siguientes parámetros:
 - i. Control del flujo de llenado en los tanques de agua.
 - ii. Control de las electroválvulas del sistema de llenado en los tanques.
 - iii. Control de temperatura y humedad para el cultivo de café.
 - iv. Gestionar señales de datos y enviarlas al transmisor RF
3. Establecer un enlace RF en Local P.S.A para transmitir datos del sistema de automatización de riego utilizando módulos de radio frecuencia.

III. JUSTIFICACIÓN

El cultivo de café es una de las actividades del grupo P.S.A. El plantío de café existente tiene una infraestructura de riego por goteo ineficiente. De acuerdo a una entrevista³ aplicada a un miembro del grupo P.S.A el plantío no cuenta con un especialista para determinar los tiempos de riego de dicho plantío. Tampoco se toma en cuenta diversos factores como humedad y temperatura. De manera que, el grupo P.S.A se ve forzado a regar el plantío según su experiencia.

Las condiciones climáticas es uno de los factores que se debe tener en cuenta ya que no permanecen constantes. Debido al cambio climático enfermedades como la roya, el ojo de gallo, la antracnosis, broca, los nematodos y minadores de la hoja, han adquirido una mayor propagación y mayor resistencia a los agroquímicos utilizados por los agricultores. El grupo P.S.A según su experiencia determina los tiempos de riego al plantío sin tomar en cuenta las condiciones climáticas exponiendo el cultivo a estas plagas.

Además, el sistema de llenado de los tanques está en mal estado. Por consiguiente, el grupo P.S.A tiene que realizar el riego de manera manual. Esto provoca que no haya una irrigación uniforme ni consistente afectando el desarrollo de las plantas.

Ante esta problemática el grupo UNI-VILLANOVA ha venido investigando diferentes tecnologías que brinden una alternativa viable para dar solución a esta situación por lo que se plantea la implementación de un sistema automatizado de riego por goteo para el café utilizando la tecnología de Microcontrolador y un sistema de control inalámbrico.

El sistema de control elaborado en ARDUINO⁴ controla el llenado de los tanques, el estado abierto o cerrado de las electroválvulas, tiempo de riego del plantío basados en variables de humedad y temperatura a través de sensores que se comunican de manera remota con Local P.S.A, lugar donde se instaló un sistema de control.

³Entrevista: Ver anexo C.3

⁴ARDUINO: Plataforma de hardware de código

Con la realización de este proyecto se beneficia al grupo promotores solares agroecológicos en la comunidad Totogalpa: teniendo un mejor control de riego, uniformidad de riego, aprovechamiento del recurso hídrico y energía solar. Aunque no abarca todos los problemas existentes en el sitio, se espera que el grupo realice sus labores de una manera más eficiente, y que puedan contar con una alternativa solución con la problemática del riego del plantío.

CAPITULO 1: MARCO TEORICO

1.1. Origen del café

La planta del café es originaria del alto Egipto de Abisinia y en particular de Kaffa, lugar del cual es probable que haya tomado su nombre. Es fácil confundirse con el origen verdadero del café, ya que antiguas leyendas sobre el cultivo y la costumbre de tomar café proviene de Arabia. (Duran, 2008)

La leyenda dice que un pastor de Abisinia llamado Kaldi, observo el efecto tonificante de unos pequeños frutos rojos de arbustos en las cabras que lo habían consumido en los montes, efecto comprobado por el mismo al renovarse su energía.

Al parecer las tribus africanas que sabían del café desde la Antigüedad, molían los granos del café y elaboraban una pasta para alimentar a los animales y aumentar las fuerzas de los guerreros. Su agricultura se extendió en primer lugar en la vecina Arabia, llevado probablemente por prisioneros de guerra, donde se popularizó la prohibición de alcohol por el Islam.

Desde entonces los datos arqueológicos disponibles hoy en día sugieren que el café no fue “Domesticado” antes del siglo XV, el proceso de elaboración de la bebida es largo y complejo. Los recientes descubrimientos de un equipo arqueológico británico, aún por confirmar, dejan entrever la posibilidad de que el consumo comenzó a partir del siglo VII, en Arabia. (Peláez, 2012)

A continuación, se abordará los factores que influyen en el cultivo del café.

1.1.1. Temperatura del café

La temperatura es de vital importancia, es preferible que no varíe mucho de un extremo a otro de los límites deseables del café. El cultivo puede ser afectado por el aire, el suelo y la planta misma. Las diferentes especies son sensibles a este factor, cuando las temperaturas son excesivamente bajas de (7 °C), mueren millares de plantas. A temperaturas de -2°C mueren los tejidos foliares y retoños, lo que se traduce a una defoliación del arbusto y en la muerte de las extremidades de las bandolas.

La temperatura está relacionada con la altura donde el café está ubicado. El café requiere de una temperatura anual entre los (19 y 24 °C), ya que por encima de los (24 °C) se acelera el crecimiento vegetativo, limitando tanto la floración como el llenado de los frutos. Cuando la temperatura promedio es superior a los 26 °C la fructificación se caracteriza por la presencia de frutos maduros o próximo a la madurez acompañado de frutos verdes de tamaño mediano y pequeños en una misma rama. (Díaz, Flores y Montalbán, 2015)

1.1.2. Luz para el cultivo de café

La intensidad de luz y su duración son igualmente importantes, existiendo periodos en que la insolación y el tiempo seco sean esenciales para la formación de madera y yemas florales, o para permitir la libre distribución de polen seco cuando se abren las flores.

El hábitat natural del cultivo del café se halla en lugares sombríos o semi-sombreados. Su comportamiento ante la luz ha hecho que durante mucho tiempo se le considere como una planta heliófila, exigiendo en las plantaciones un cubrimiento medio. No obstante la práctica de sombras está siendo abandonada por medianos y grandes productores, los cultivos intensivos sin sombra, son capaces de dar rendimientos, pero la consecuencia de que se acorta el ciclo de vida de la planta por efectos de una función fisiológica más intensas.

En conclusión el café requiere para su desarrollo y su reproducción eficientemente un fotoperiodo corto de 4 a 5 horas luz al día, por lo que se hace necesario asociar el cultivo con especies de árboles de sombra que brinden una cobertura en el cafetal de 40 a 50 %. (Díaz, Flores y Montalbán, 2015)

1.1.3. Crecimiento vegetativo del café

El café toma 2 años para completar el ciclo fenológico de fructificación, al contrario de la mayoría de las plantas que contemplan el ciclo reproductivo en un año. Se caracteriza por tener 6 etapas fenológicas que toman 2 años y empiezan en septiembre de cada año. Estas fases son:

1. Vegetativa, con 7 meses de septiembre hasta marzo todos con días largos.
2. También vegetativa de abril a agosto con días cortos, cuando ocurre la transformación de las yemas vegetativas de los nudos formados en la primera etapa a yemas reproductivas. Al final de esta fase, de julio y agosto, las plantas entran en un estado relativo reposo con la formación de uno o dos pares de pequeñas hojas que generalmente no florecen. La maduración de las yemas reproductiva ocurre después de la acumulación de alrededor de 350 mm de evapotranspiración potencial, iniciando a principio de abril.
3. Floración y expansión de los frutos, de septiembre a diciembre. Usualmente la floración ocurre de 8 a 15 días después del aumento del potencial hídrico de las yemas florales causado por lluvia o irrigación.
4. Formación del grano, de enero a marzo.
5. Maduración del grano, cuando se acumula alrededor de 700 mm de evapotranspiración desde la floración principal.
6. Senescencia y muerte de las ramas productivas no primaria en julio y agosto. (Marín, 2003)

1.2. Evapotranspiración de una planta

Se conoce como evapotranspiración (ET), la combinación de dos procesos separados que conllevan a una pérdida de agua. (Allen, 2006)

Estos dos procesos son:

- ☞ Evaporación: Proceso por el cual el agua líquida se convierte en vapor de agua y se retira de la superficie evaporante.
- ☞ Transpiración: Se refiere a la vaporización del agua líquida contenida en los tejidos de la planta y su posterior remoción hacia la atmósfera. Los cultivos pierden agua predominantemente a través de las estomas,

los cuales controlan el intercambio de gases y vapor de agua de la planta hacia la atmosfera (ver Figura 1).

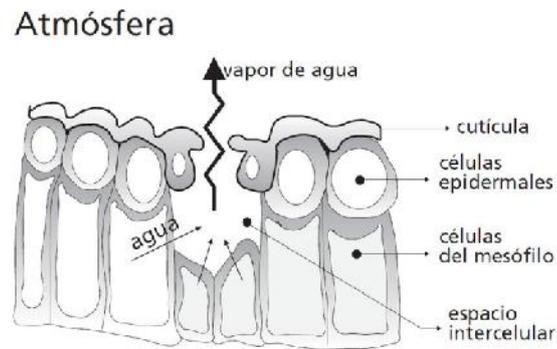


Figura 1. Representación esquemática de una estoma.
(Allen, 2006)

1.2.1. Clasificación de evapotranspiración

La evapotranspiración se clasifica en tres definiciones:

- ☞ Evapotranspiración del cultivo de referencia (ET₀).
- ☞ Evapotranspiración del cultivo bajo condiciones estándar (ETC).
- ☞ Evapotranspiración del cultivo bajo condiciones no estándar (ETC aj).

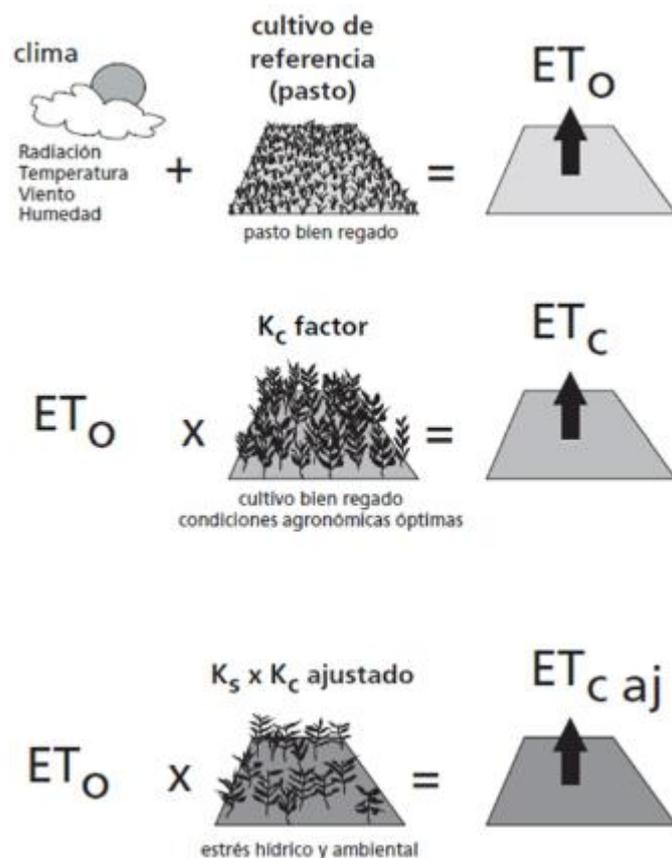


Figura 2. Evapotranspiración del cultivo de referencia (ET_0), bajo condiciones estándar (ET_c) y bajo condiciones no estándar ($ET_{c\ aj}$). (Allen, 2006)

1.3. Riego por goteo y su origen

El riego por goteo se ha utilizado desde la antigüedad cuando se enterraban vasijas de arcilla llenas de agua con el fin de que el agua se infiltrara gradualmente en el suelo. El riego gota a gota moderno se desarrolló en Israel en una empresa llamada NETAFIM por Simcha Blass y su colega Kibbutz Hatzerim en el año 1965. (NETAFIM, 1965)

Este sistema de riego es aplicado mayormente en campos agrícolas, invernadero y jardines, el de riego por goteo es un método de irrigación utilizado en las zonas áridas y época de pocas lluvias. (Valleybolivia, 1965)

1.3.1. Método de riego por goteo

El riego por goteo es un método de irrigación utilizado en las zonas áridas pues permite la utilización óptima de agua y abonos es decir consiste en la infiltración lentamente de agua hacia las raíces de las plantas e irrigando directamente la zona de influencia de las raíces, a través de un sistema de tuberías, consiguiendo así reducir al mínimo la utilización de agua. Esta técnica es la innovación más importante y de alto costo en la agricultura. Ver Figura 3. (Valleybolivia historia Netafim, 1965)



Figura 3. Método de riego por goteo

1.3.2. Ventaja de riego por goteo

El riego por goteo es un medio eficaz para aportar agua a la planta. El sistema de riego presenta diversas ventajas del punto de vista económico, técnico y agronómico, derivados de un uso más eficiente del agua y de la mano de obra. Tales como:

La posibilidad de automatizar completamente el sistema de riego, con los consiguientes ahorros en mano de obra. El control de las dosis de aplicación al cultivo es más fácil y completo. (Pirapey, 2014)

Adaptación a todo tipo de superficie y desniveles en su relieve natural sin inversión en la nivelación y transporte de tierra. (Sistema de riego, 2014)

Reduce la proliferación de malas hierbas debido a la fertilización directamente a la raíz de la planta, dando como resultado un significativo aumento en la cosecha. (Acerca de NETAFIM, 1965)

1.3.3. Programación del riego

El termino programación del riego es generalmente usado para describir el procedimiento mediante el cual se determina la dosis de agua a aplicar en cada riego. (Hillel, 1990)

La determinación del momento y cantidad de agua a aplicar en cada riego es un proceso de toma de decisiones. En nuestro país por lo general, se ha basado en la experiencia previa y no en procedimiento (científicos-técnico). De tal forma que si se desea que la conversión del agua en aplicaciones de riego sea una realidad, es necesario cada vez disponer de más información que permita caracterizar la necesidad de los cultivos para así programar los riegos de forma eficiente. (Federes, 2000)

A continuación se abordaran los dispositivos que se utilizaran en el sistema de riego con lo cual se desarrolla el proyecto.

1.4. Arduino Mega 2560

ARDUINO es una plataforma de hardware libre, basada en una placa con un microcontrolador ATMEL ATMEGA 2560 (ver figura 4). La cual está compuesta por 54 pines de entradas/salidas digitales de las cuales 14 pines pueden ser utilizadas como salidas PWM, 16 de entradas análogas, 4 de UARTS (puerto serial por hardware), 1 cristal de 16 MHz, conexión USB, Jack de alimentación, conector ICPS y botón de reset. Lo necesario para que el microcontrolador trabaje.



Figura 4. ARDUINO MEGA 2560

1.4.1. Elementos de Arduino Mega 2560

Aref = Pin de referencia analógica (naranja).

GND = Señal de tierra (verde claro).

Pines digitales = Del pin 2 hasta 53, Entrada y salida (verde claro).

Pin de TX/RX = Son 0-1, 14-15, 16-17, 18-19 y 20-21 (verde oscuro).

Botón de reset = Pulsador (azul oscuro).

Pines de entrada analógica = Del pin A0 hasta A15 (azul claro).

Fuente de alimentación = De 9-12VDC.

1.4.2. Microcontrolador ATMEL ATMEGA 2560

El Atmega 2560 tiene 256 KB de memoria flash para almacenar el código, la cual se utiliza 8KB para el cargador de arranque, 8KB de SRAM y 4KB de EEPROM.

1.4.3. Memoria de datos (RAM)

Se utiliza para almacenar las variables y los cambios de la información que se produzcan en el transcurso del programa, por lo tanto es necesario una memoria de lectura y escritura y que además soporte muchos ciclos de trabajo. La memoria RAM estática (SRAM) es la más utilizada, aunque en algunos casos también se utiliza memoria EPROM para poder almacenar datos una vez finalizado el programa.

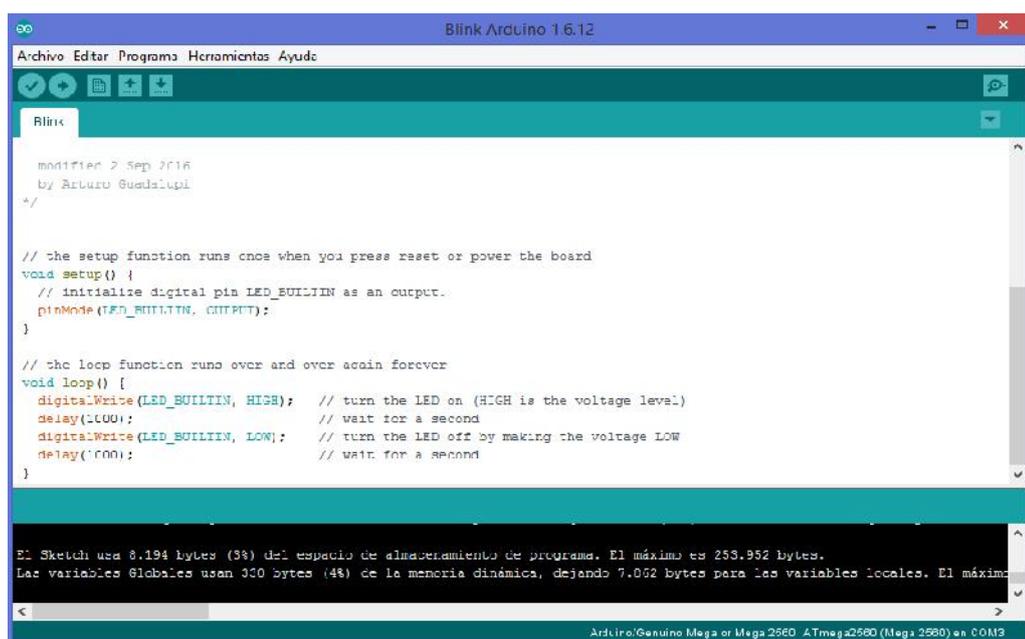
1.4.4. Puertos de Entrada/Salida

Este elemento permite al microprocesador comunicarse con el mundo exterior. A través de los pines del circuito integrado, puede emitir y recibir señales por medio de corrientes eléctricas. Mediante estas entradas y salidas se pueden efectuar muchísimas funciones, leer información de sensores, activar o desactivar diferente mecanismo.

1.4.5. Lenguaje de programación del IDE de ARDUINO

Para comunicarnos con el microcontrolador y cargarle los programas usamos IDE (Integrated Development Environment), sistema de desarrollo de ARDUINO, sirve para desarrollar el programa, compilarlos y descargar los programas a la placa de ARDUINO mediante el puerto USB. En el mismo portal de IDE hay disponible una serie de biblioteca estándar para ARDUINO, para manejar hardware o datos, para manejo de EEPROM, Ethernet, visualizadores LCD entre otras. (Ver figura 5)

El lenguaje que utiliza este entorno para escribir los programas es parecido al C++, todos los programas escritos para ARDUINO en el entorno IDE contienen las funciones *setup ()* y *loop ()*. La función *setup ()* se sitúa en primer lugar y es también la primera función en ejecutarse, habitualmente una sola vez. Esta función contiene la declaración de las variables de acuerdo al sistema a implementar. (Herrero y Sánchez, 2015)



```

Blink

modified 7 Sep 2016
by Arturo Guadalupe
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

El Sketch usa 8.194 bytes (3%) del espacio de almacenamiento de programa. El máximo es 253.952 bytes.
Las variables Globales usan 320 bytes (4%) de la memoria dinámica, dejando 7.052 bytes para las variables locales. El máximo

```

Figura 5. Edición de un programa en el entorno de programación.

La tabla 1 resume las instrucciones y variables del lenguaje de programación de Arduino.

Tabla 1.
Instrucciones, variables del lenguaje de programación de ARDUINO

Estructura de básica	Funciones de E/S digitales	Operadores de comparación	Operadores compuestos
void setup () void loop ()	pinMode(pin [INPUT, OUTPUT]) digitalWrite(pi n, valor) int digitalRead(pin)	== (igual que) != (no igual que) < (menor que) > (mayor que) <= (menor o igual que) >= (mayor o igual que)	++ (incremento) -- (decremento) += (suma compuesta) -= (resta compuesta) *= (multiplicacion compuesta) /= (division compuesta)
Estructura de control	Operador aritmético	Conversiones	Constantes
if () if ()..else () for () Switch ().. case () while () do ().. while () break continue return goto	= (asignacion) + (suma) - (resta) * (multiplicacion) / (division) % (modulo)	char () byte () int () word () long () float ()	HIGH / LOW INPUT / OUTPUT
Sintaxis	Tipo de datos	Comunicación serie	Funciones de tiempo
; (punto y coma) { } (llaves) // (comentario)	Void boolean char unsigned int word long unsigned long short float double string	Serial.begin () intSerial.avaible () intSerial.read () Serial.flush () Serial.print () Serial.println ()	/**/ (comentario multilinea) # define # include

1.5. Tecnología ZIGBEE

En el desarrollo del proyecto utilizaremos Zigbee como protocolo de comunicaciones para la implementación de los sensores inalámbricos que veremos más adelante. La tecnología Zigbee surge de la necesidad de desarrollar una tecnología inalámbrica fiable, pero de no muy alta frecuencia de datos. De esta forma, en 1998 un conjunto de empresas se asociaron para desarrollar un estándar de comunicaciones que complementara a WIFI y BLUETOOTH. Fue terminado y publicado por la IEEE en Mayo del 2003.

Zigbee es un estándar que define un conjunto de protocolos de comunicación de baja velocidad de datos de corto alcance de redes inalámbrica. Zigbee está basado en dispositivos inalámbricos que operan en la banda 868 MHz, 915 MHz y 2.4 GHz. (Girod, 2016)

1.5.1. Comparación entre WIFI, BLUETOOTH Y ZIGBEE

En resumen la tecnología bluetooth está orientada a la movilidad del usuario y eliminar el cableado a corta distancia como por ejemplo entre el teléfono móvil o un ordenador portátil mientras que zigbee está orientado hacia el control remoto y a la automatización por ejemplo una red de sensores inalámbricos que se conecten sin cables a una estación de control.

Tabla 2.
Comparación tecnología WI-FI, BLUETOOTH Y ZIGBEE.

Datos	WI-FI	BLUETOOTH	ZIGBEE
Velocidad	< 50 Mbps	1 Mbps	< 250 Kbps
Número nodos	32	8	255/ 65535
Duración de batería	Horas	Días	Años
Consumo de transmisión	400 ma	40 ma	30 ma
Consumo de reposo	20 ma	0,2 ma	3 µa
Precio	Caro	Medio	Barato
Configuración	Compleja	Compleja	Simple
Aplicaciones	Internet en edificios	Informática y móviles	Domótica y monitorización

1.5.2. Características técnicas de zigbee

1.5.2.1. Frecuencia de trabajo y rango de velocidad

En el estándar Zigbee se definen tres bandas de frecuencia distinta:

- ☞ 868- 868.6 MHz (banda de 868 MHz).
- ☞ 902- 928 MHz (banda de 915 MHz).
- ☞ 2400-2483.5 MHz (banda de 2.4 GHz).

La banda 2.4 GHz es usada en el mundo entero mientras que la de 868 y 915 MHz se usan en Europa y América del Norte respectivamente.

1.5.2.2. Capas de zigbee

Las capas del protocolo Zigbee están basada en el modelo OSI (Open Systems Interconnection). Aunque el modelo OSI especifica siete capas, Zigbee utiliza cuatro capas. Las cuales son:

- ☞ Capa física.
- ☞ Capa de control de acceso al medio.
- ☞ Capa de red
- ☞ Capa de aplicación

La ventaja de dividir un protocolo en capas es que, si el protocolo cambia, es más fácil cambiar una capa que el protocolo entero.

1.5.2.3. Tipos de Zigbee

Se definen tres tipos distintos de dispositivos Zigbee según su papel en la red:

- **Coordinador Zigbee:** El tipo de dispositivo más completo. Debe existir uno por red. Sus funciones son las de encargarse de controlar la red y los caminos que deben seguir los dispositivos para conectarse entre ellos, requiere memoria y capacidad de computación.

- **Router Zigbee:** Interconecta dispositivos separados en la topología de la red, además de ofrecer un nivel de aplicación para la ejecución de código de usuario.
- **Dispositivo final (End Device):** Posee la funcionalidad necesaria para comunicarse con su nodo padre (el coordinador o un router), pero no puede transmitir información destinada a otros dispositivos. De esta forma, este tipo de nodo puede estar dormido la mayor parte del tiempo, aumentando la vida media de sus baterías.

1.5.2.4. Topologías de Zigbee

En ZigBee existen tres tipos de topologías (ver figura 6): estrella, árbol y en red malla (Mesh Network), las cuales pueden observarse en la figura 6. Siempre hay un nodo de red que asume el papel de coordinador central encargado de centralizar la adquisición y las rutas de comunicación entre dispositivos. Además, si se aplica el concepto de Mesh Network, pueden existir coordinadores o routers, alimentados permanentemente en espera de recibir/repetir las tramas de los dispositivos o sensores.

A continuación, una de las mayores aportaciones del ZigBee y el que mayor interés está despertando a las empresas desarrolladoras de productos, es el concepto de Mesh Network por el que cualquier dispositivo ZigBee puede conectarse con otro dispositivo usando a varios de sus compañeros como repetidores. A este se le conoce como enrutador, primero hace llegar la información al nodo ZigBee vecino, el cual puede además ser coordinador de la red, para así llegar al nodo destino, pasando por todos los que sean necesarios. De esta manera cualquier nodo ZigBee puede hacer llegar los datos a cualquier parte de la red inalámbrica siempre y cuando todos los dispositivos tengan un vecino dentro de su rango de cobertura.

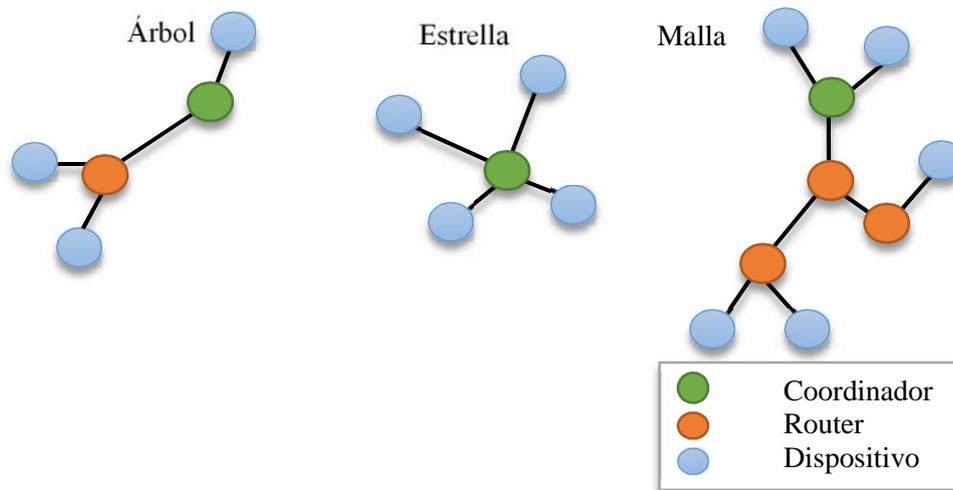


Figura 6. Topologías Zigbee

De todas las topologías antes analizadas, la más ventajosa y característica del protocolo Zigbee es la Mesh Network. El poder hacer un enrutamiento dinámico permite a este tipo de topología ganar sobretodo en fiabilidad de comunicación.

1.5.3. Módulo Xbee

Modulo basado en la tecnología inalámbrica ZigBee. Utiliza el protocolo IEEE 802.15.4 para comunicaciones punto a punto y punto a multipunto. También cuenta con una baja latencia de transmisión, bajo consumo energético y largo alcance. Además, se pueden configurar para implementar una red Mesh para futuros desarrollos.

Construidos y respaldados por la empresa Digi's International, que cuenta con toda la documentación en línea y también con el software gratuito compatible con Windows, para la configuración de sus parámetros.

Existen dos tipos de módulos XBee, denominados: serie 1 y serie 2. Estos no son compatibles entre sí y no pueden comunicarse debido a que la serie 1 está basado en el chipset de freescale y la serie 2 está basado en el chipset de Ember.

Los nodos XBee se pueden configurar de dos modos denominados: modo AT y API.

- **Modo AT (Attention).** En esta configuración los datos que se desean transmitir no reciben ningún tipo de modificación. El paquete puede ser enviado a un solo destino o a múltiples destinos.

En el proyecto se utilizara la serie 2 modo AT (ver Figura 7), el uso de los módulos xbee es para gestionar la transmisión y recepción de datos humedad del suelo, temperatura del ambiente entre los microcontroladores ARDUINO.



Figura 7. Xbee Pro Serie 2

1.5.3.1. Modo de operación AT

En modo transparente todo lo que ingresa por el pin 3 (DIN) se guarda en buffer de entrada y se transmite y aquello que ingresa como paquete RF se guarda en buffer de salida y se envía por el pin 2 (DOUT) inmediatamente o tras el paso de un tiempo (parámetro RO). Este modo se destina a comunicación punto a punto, donde no es necesario ningún tipo de control.

1.5.3.2. Programación de los módulos Xbee: X-CTU

X-CTU es el software oficial para programar radios XBee y permitir al usuario la posibilidad de bajar nuevo firmware de la página de Digi, programar parámetros a la radio o hacer pruebas de comunicación fácilmente. Básicamente el programa se compone de una serie de funciones muy útiles ver figura 8 y 9 que veremos a continuación.

1.6. Software Proteus

Proteus VSM es un sistema de diseño electrónico basado en la simulación analógica, digital o mixta de circuitos, que brinda la posibilidad de interacción con muchos de los elementos que integran el circuito. Incluye componentes animados para la visualización de su comportamiento en tiempo real, además de un completo sistema de generación y análisis de señales. También cuenta con un módulo para el diseño de circuitos impresos.

Las siglas VSM significan Virtual System Modelling, que en español podemos traducir como sistema de modelado virtual, ya que Proteus VSM permite modelar de forma virtual en la computadora prácticamente cualquier circuito.

La característica principal que hace de Proteus VSM uno de los simuladores preferidos por muchos aficionados y profesionales de la electrónica es la posibilidad de simular circuitos que incluyen microprocesadores o microcontroladores.

Aunque el nombre completo del programa es Proteus VSM, a partir de ahora nos referiremos a él solo como Proteus, para mayor facilidad.

1.6.1. Hoja de trabajo

En la ventana de edición tenemos dos elementos principales: la hoja de trabajo y la rejilla. La hoja de trabajo está delimitada por el cuadro de color azul. Cuando accedemos a Proteus, sin abrir ningún circuito, por defecto la ventana de edición está vacía y el tamaño de la hoja de trabajo es de 10 por 7 pulgadas.

Si necesitamos cambiar el tamaño de la hoja de trabajo, podemos hacerlo mediante el menú (System/Set Sheet Size). Al seleccionar esta opción, aparece una ventana con el título (Sheet Size Configuration), donde podemos elegir un tamaño de la lista o uno personalizado, llamado (user). En realidad, es posible seleccionar un tamaño de la lista y luego ajustarlo a la medida que deseamos (Ver figura 11).

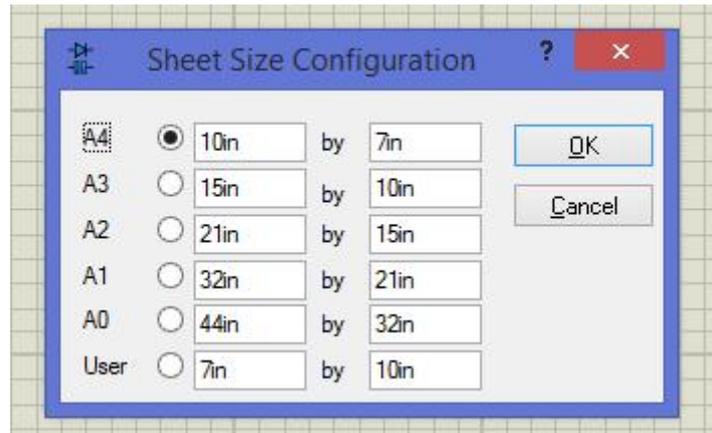


Figura 11. Shett Size Configuration, ofrece cinco tamaños de hoja y un tamaño definido por el usuario.

1.6.2. Herramientas de modo

La herramienta modo se encuentra dividida en tres partes: la superior presenta las herramientas principales, en la parte media tenemos herramienta de dispositivos y en la inferior, herramientas de dibujo en 2D. (Ver figura 12)

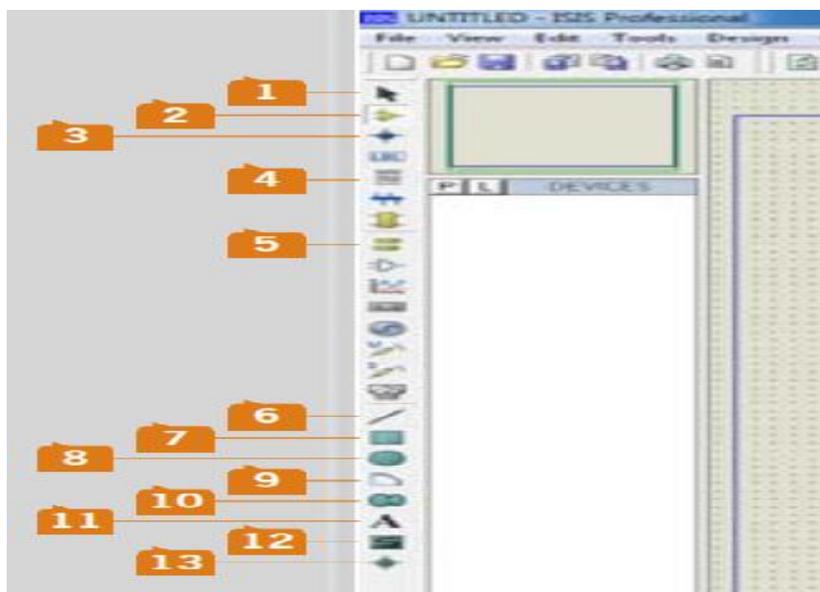


Figura 12. Herramienta de modo.

1.7. Pantalla LCD 16x2

La pantalla de cristal líquido o LCD (Liquid Crystal Display) es un dispositivo microcontrolador de visualización gráfica (ver figura 13) para la representación de caracteres, símbolos o incluso dibujos, en este caso el dispositivo LCD QC1602A V 2.0 dispone de 2 filas de 16 caracteres cada una y cada carácter dispone de una matriz de 5x7 puntos (píxeles) con una luz de fondo amarilla y color negro para los caracteres, lo que permite una correcta visualización desde un ángulo amplio.

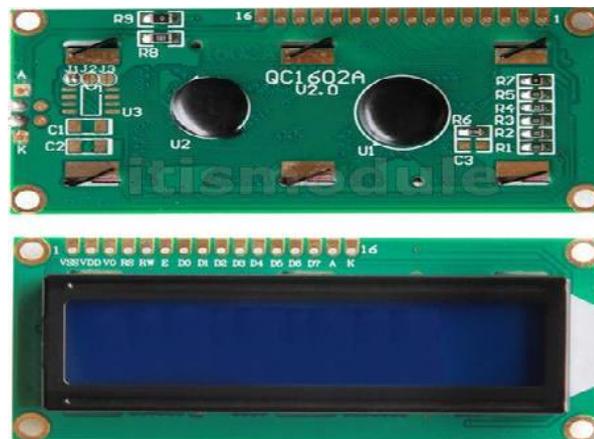


Figura 13. Pantalla LCD 16X2.

1.7.1. Características de la LCD 16X2

Pines de alimentación

- **Vss:** Gnd
- **Vdd:** +5 voltios
- **Vee:** corresponde al pin de contraste

Pines de control

- **RS**
- **RW**
- **E**

Pines de Bus de datos:

- ☞ El bus de datos bidireccional comprende desde los pines D0 a D7. Para realizar la comunicación con el LCD podemos hacerlo utilizando los 8 bits del bus de datos (D0 a D7) o empleando los 4 bits más significativos del bus de datos (D4 a D7).

La LCD 16X2 en el proyecto indica los parámetros del sistema tales como: porcentaje de la humedad del suelo, temperatura ambiente y nivel de llenado de tanque.

1.8. Importancia de los sensores de humedad

El monitorear el contenido de agua en el suelo es esencial para ayudar a los agricultores a optimizar la producción, conservar agua, reducir los impactos ambientales y ahorrar dinero. El monitorear la humedad del suelo le puede ayudar a tomar mejores decisiones en la programación del riego, tales como el determinar la cantidad de agua a aplicar y cuándo aplicarla. El riego insuficiente puede reducir la producción de las cosechas. (Enciso, Porter y Périès, 2008)

1.8.1. Sensor de humedad del suelo YL-69

Es un tipo de sensor resistivo, que utiliza dos sondas para medir la corriente que circula en el medio. Esta corriente determina la resistencia, con la cual se calcula la humedad del terreno y se conecta al ARDUINO mediante un módulo YI-38. La señal analógica evidentemente nos resultará más precisa, ya que emite un valor proporcional a la resistencia, e inversamente proporcional a la conductividad y humedad, del suelo. (My robot self, 2016)

Los valores siguientes son aproximados:

- La tierra está muy mojada: 0-350
- La tierra está humedad: 350-600
- La tierra está seca: 600-1023

El YL-69 es un sensor que puede usarse para detectar la humedad del suelo cuando el índice de humedad presente en el suelo es alto. Este sensor cuenta con una salida digital y una analógica, las mismas que pueden ser conectadas directamente a un microcontrolador.

La sensibilidad de la salida digital es regulada por el potenciómetro incluido en el circuito de control del sensor, en este circuito la señal es procesada por un comparador de voltaje LM393 cuando la señal de salida digital se encuentra dentro del rango establecido para la sensibilidad del sensor, esta se mantendrá en un estado alto, sin embargo, al superar el límite establecido

devolverá un estado bajo, este circuito se conecta al sensor YL-69 por medio de cables hembra-hembra como se muestra en la figura 14.

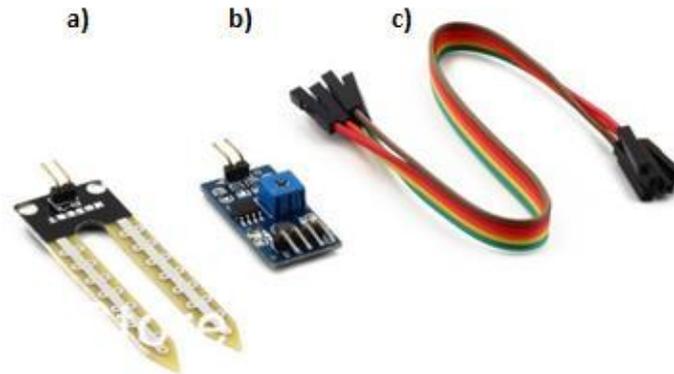


Figura 14. Sensor de humedad YL-69, comparador de voltaje, cable hembra-hembra.

El censado de la humedad por medio del sensor YL-69 se realiza en tiempo real con un nivel de detección que va desde el 0 hasta el 85% de humedad por protección al cultivo del café, con un voltaje de salida máximo igual al voltaje de alimentación.

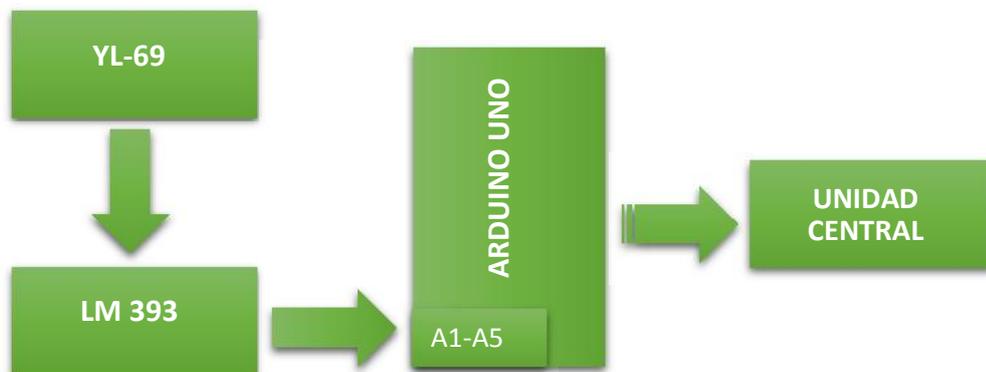


Figura 15. Sensor de humedad YL-69 y ARDUINO UNO

En la figura 15 se muestra el diagrama de bloque de para el censado del suelo usando sensores YI-69. Se ajusta a la placa de ARDUINO UNO para la recepción de datos por los puertos analógicos (A1-A5) activando a su vez el sensor de humedad del suelo YL-69. Al detectar la variable presente comienza a existir una variación de voltaje entre las dos puntas del sensor, estas variaciones son procesadas en el comparador LM393, el cual genera

la señal analógica y la transmite a la placa ARDUINO UNO, luego la información se envía a otra unidad y mostrada en pantalla con el objetivo que el usuario pueda observar en tiempo real la humedad del suelo. El programa se ejecuta de manera cíclica.

1.9. Sensor de temperatura LM35

El sensor de temperatura LM35 son dispositivos que transforman los cambios de temperatura en cambios en señales eléctricas que son procesados por equipo electrónico. Las series LM35 son dispositivos de temperatura de circuito integrado de precisión con un voltaje de salida linealmente proporcional a la temperatura centígrada. El dispositivo LM35 tiene una ventaja sobre los sensores de temperatura lineales calibrados en Kelvin, ya que no se requiere que el usuario sustraiga un voltaje constante grande de la salida para obtener un escalamiento centígrado conveniente.

El dispositivo LM35 está diseñado para operar en un rango de temperatura de -55°C a 150°C , pero para la implementación del proyecto resulta suficiente, dado que la temperatura deseada en nuestro sistema tiene que mantenerse en un rango de $(16$ y $21^{\circ}\text{C})$.

Adicionalmente el LM35 tiene otra ventaja, dado que tiene un bajo consumo de corriente, aproximadamente de $60\mu\text{A}$ y puede ser alimentado desde 4 hasta 30V. Otra ventaja muy importante que nos presenta el LM35 es su baja impedancia de salida, cerca de 0.1 para una carga que demande 1mA de corriente. Esto resulta muy importante dado que la señal proporcional a la temperatura es una señal de voltaje. (Texas Instruments, 1951)

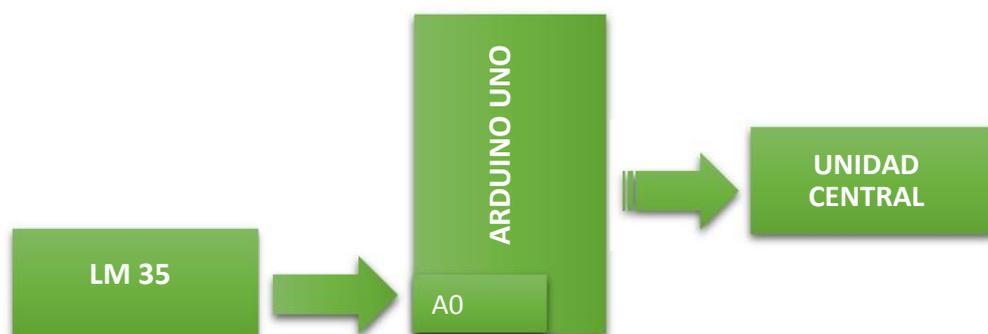


Figura 16. Sensor de temperatura LM35 y Arduino UNO

En la figura 16 se muestra el diagrama de bloque para el censado de la temperatura ambiente del cultivo del café. Se ajusta a la placa de ARDUINO UNO para la recepción de datos por el puerto analógico (A0) activando a su vez el sensor de temperatura LM35 y enviando los datos por medio de un módulo RF a otra unidad para ser mostrado en una pantalla LCD 16x2 con el objetivo que el usuario pueda observar en tiempo real la temperatura dada por el ambiente. El programa se ejecuta de manera cíclica.

1.10. Sensor de ultrasonidos HC-SR04

El sensor de distancia HC-SR04 tiene la capacidad de medir la distancia por medio de la diferencia en tiempo entre la transmisión y recepción de una serie de pulsos que el modulo envía y captura. Este sensor tiene un alcance de precisión entre 2 cm y 400 cm. (Micropik, 2008)

El sensor HC-SR04 (ver figura 17) permite controlar de una manera exacta el nivel del agua dentro del depósito. También permite ajustar los valores de nivel máximo y mínimo modificando software de control implementado en ARDUINO y no teniendo que mover el sensor o agujereando el tanque cada vez que se desea cambiar las alturas.



Figura 17. Sensor Ultrasónico HC-SR04

Este sensor HC-SR04 consta de cuatro pines, los cuales son:

- Alimentación (Vcc)
- Receptor (Echo)
- Disparador (Trigger)
- Tierra (GND)

1.10.1. Características eléctricas

- Voltaje de trabajo: 5Vcc
- Corriente de trabajo: 15mA
- Frecuencia de trabajo: 40 KHz
- Rango de funcionamiento: 2 a 400cm
- Ángulo de detección: 15 a 20 grados

1.10.2. Funcionamiento del sensor HC-SR04

El funcionamiento de este sensor es muy sencillo, tan solo hay que generar una onda acústica ultrasónica en el emisor mediante un pulso en la patilla Trigger, esta onda al encontrarse con algún obstáculo rebotara, volviendo al sensor y siendo registrada por el receptor, traduciéndose este en un pulso por la patilla Echo (Ver figura 18).

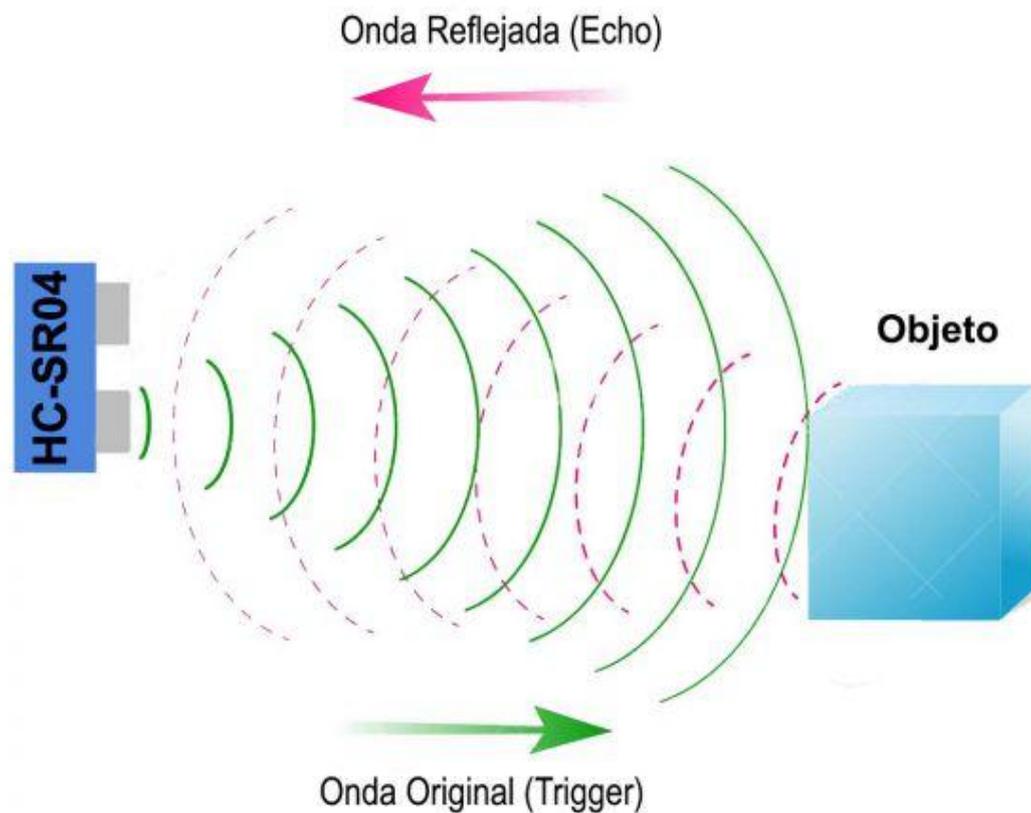


Figura 18. Funcionamiento del Sensor Ultrasónico HC-SR04.

1.10.3. Ecuación para determinar el nivel de agua en un tanque

Para el cálculo de nivel del agua en el tanque es necesario ser preciso para conocer la distancia en la que va estar ubicado este sensor. Por lo que, se usaron las siguientes ecuaciones para el cálculo dadas por el fabricante (ver anexo C).

$$Velocidad\ del\ sonido = \frac{Espacio}{Tiempo} \quad (1)$$

Donde:

$$Velocidad\ del\ sonido = 343\ M/S \approx 0.343\ Cm/\mu S$$

La onda ultrasónica recorre su trayectoria dos veces (ida y vuelta). Por lo que, hay que dividir la velocidad entre dos para conocer la distancia a la que se encuentra el objeto. De (1) podemos conocer el valor del espacio obteniendo (2)

$$Espacio = \frac{Velocidad}{2} \times Tiempo \quad (2)$$

Para determinar la distancia del nivel de agua en el tanque se tiene que determinar el espacio que es igual a la mitad de la velocidad del sonido multiplicándola por el tiempo transcurrido entre la emisión y recepción del pulso.

1.11. Sistema Fotovoltaico

Un sistema fotovoltaico es un conjunto de elementos que aprovechan la energía producida por el sol y la convierten en energía eléctrica (ver figura 19). Los sistemas fotovoltaicos se basan en la capacidad de las celdas fotovoltaicas de transformar energía solar en energía eléctrica (DC).

La generación de energía eléctrica dependerá de las horas que el sol brille sobre el panel solar teniendo en cuenta la marca del panel, los módulos instalados, orientación, inclinación, radiación solar que les llegue, calidad de la instalación y la potencia nominal

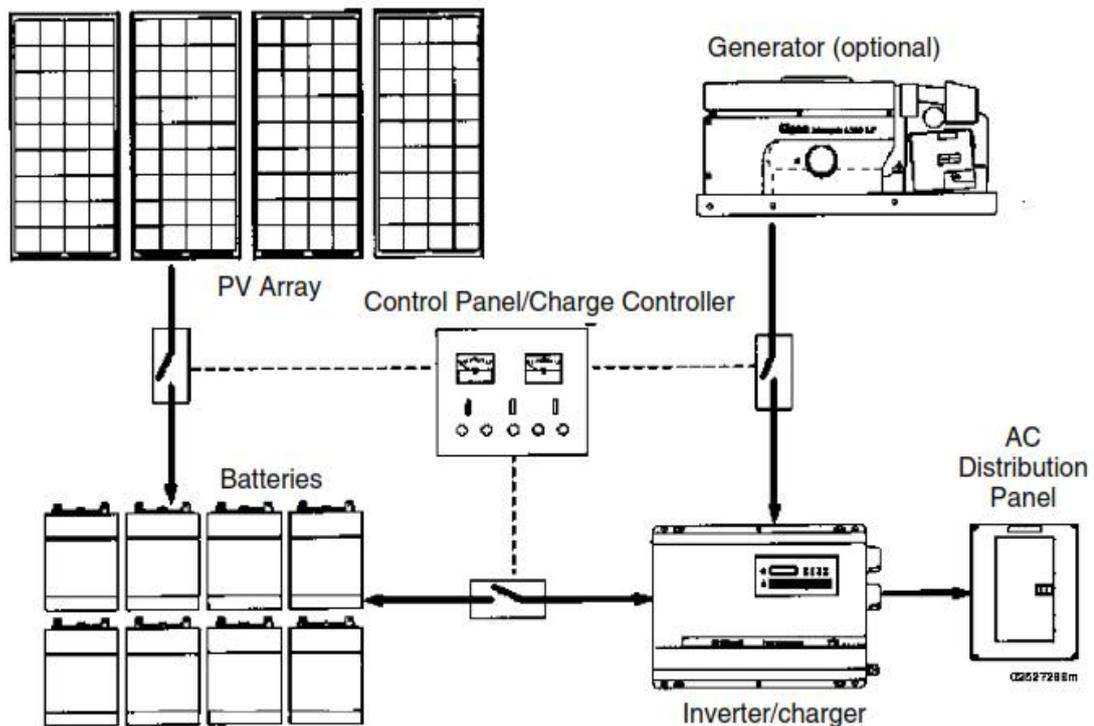


Figura 19. Arreglo de un sistema fotovoltaico. (Gilbert, 2004)

1.11.1. Radiación solar

Se conoce como radiación solar al conjunto de radiaciones electromagnéticas emitidas por el sol. El sol se comporta prácticamente como un cuerpo negro que emite energía siguiendo la ley de Planck a una temperatura de un 6000 k. La radiación solar se distribuye desde infrarrojo hasta ultravioleta.

No toda la radiación alcanza la superficie de la tierra, porque las ondas ultravioletas, más cortas, son absorbidas por los gases de la atmosfera fundamentalmente por el ozono. La magnitud que mide la radiación solar que llega a la tierra es la irradiación, que mide la energía por unidad de tiempo y área, alcanza a la tierra. Su unidad es W/m^2 (vatio x metro cuadrado). (Manzini, 2004)

1.11.2. Células fotovoltaicas

Los paneles fotovoltaicos destinados a generación energética están compuesto por células fotovoltaicas, que es donde realmente tiene lugar la

transformación de la energía luminosa (fotones) en electricidad (electrones en movimiento), aprovechando las propiedades de los materiales semiconductores. (Iter, 2008)

Dichas células están formadas por dispositivos tipo diodo constituido por materiales semiconductores en los que artificialmente se ha creado un campo eléctrico constante (mediante unión p-n), al recibir radiación solar, se excitan y provocan salto electrónico, generando una pequeña diferencia potencial en sus extremos.

Las celdas cuenta con dos terminales: la cara expuesta a la luz, posee un enrejado metálico muy fino (plata y aluminio), el cual colecta los electrones foto generados. Esta capa corresponde a la terminal negativa. La otra cara cuenta con una capa metálica, usualmente de aluminio. Esta corresponde a la terminal positiva ya que en ella se acumulan las cargas positivas.

También la celda está cubierta con anti reflejante para disminuir las perdidas por reflexión. Cada célula son capaces de generar (0.46 a 0.48V), luego las células se montan en serie para conseguir un voltaje adecuado a las aplicaciones correspondiente (ver figura 20).

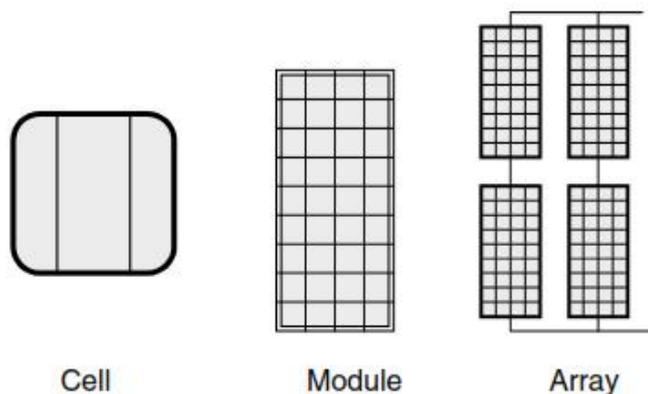


Figura 20. Célula fotovoltaica, módulo y arreglo. (Gilbert, 2004)

1.11.3. Sistema fotovoltaico aislado

Los sistemas fotovoltaicos aislados, por el hecho de no estar conectados a la red eléctrica, normalmente están equipados con sistema de acumulación de energía. La acumulación es necesaria porque el campo fotovoltaico puede

proporcionar energía solo en las horas diurnas, la mayor demanda por parte del usuario se concentra en las horas de la tarde y la noche. Durante la fase de insolación es necesario prever una acumulación de la energía no inmediatamente utilizada, que es proporcionada a la carga cuando la energía disponible es reducida o nula.

Una configuración de este tipo implica que el campo fotovoltaico debe estar dimensionado de forma que permita, durante las horas de insolación, la alimentación de la carga y de la recarga de las baterías de acumulación. (Iter, 2008)

1.11.4. Elementos de un sistema fotovoltaico

En el sistema solar fotovoltaico que se va a diseñar, la energía producida por los módulos fotovoltaicos es almacenada en batería de acumulación. La carga es alimentada, a través del regulador de carga, por la energía acumulada en las baterías.

El esquema básico de la instalación fotovoltaica para el sistema del proyecto es lo siguiente (Ver figura 21).

La unidad central está compuesta por ARDUINO MEGA 2560, electroválvulas, sensor HC-SR04 y módulo Xbee Pro.

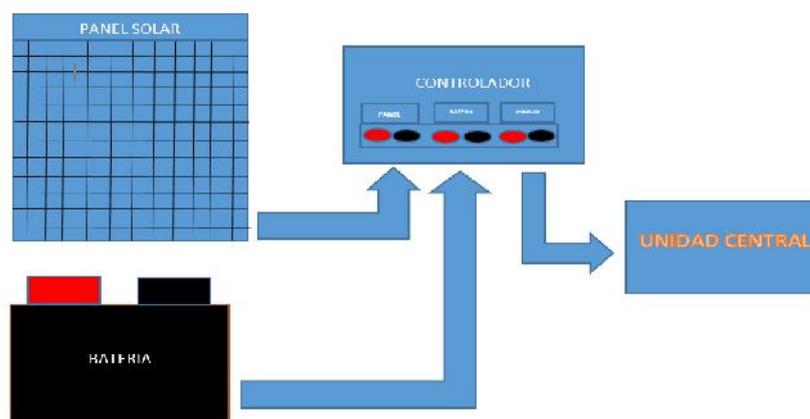


Figura 21. Instalación del sistema fotovoltaico para el sistema

1.11.4.1. Panel solar

El elemento principal de cualquier instalación de energía solar es el panel solar, convierte la energía del sol en energía eléctrica (DC). Su función se basa por convertir directamente en electricidad los fotones provenientes de la luz del sol. Su funcionamiento se basa en efecto fotovoltaico.

El efecto fotovoltaico es la propiedad que tienen determinados materiales para producir una corriente eléctrica cuando incide una radiación lumínica sobre ellos. En un panel fotovoltaico la producción de corriente depende del nivel de iluminación, de tal forma que a medida que aumenta la iluminación o irradiación, aumenta la intensidad a través de la célula.

La vida útil de un panel fotovoltaico puede llegar hasta 30 años y los fabricantes generalmente otorgan garantías de 20 a más años. El mantenimiento del panel es sencillo solo basta con darle limpieza al vidrio para prevenir sombras en las celdas.

La potencia del panel solar a utilizar son:

Comercialmente el panel solar seleccionado es de 160W (Ver figura 22) y un panel de 60W (Ver figura 23).

Estos tipos de paneles solares tienen vidrio templado con bajo hierro para dar mayor rigidez y resistencia al impacto. Para mayor información ver anexo C.



Figura 22. Panel solar de 160W



Figura 23. Panel solar 60W

1.11.4.2. Regulador de carga

El regulador de carga tiene como función fundamental impedir que la batería continúe recibiendo energía del colector solar una vez que ha alcanzado su carga máxima. Si una vez que se ha alcanzado la carga máxima, se intenta seguir introduciendo energía, se inicia en la batería un proceso de gasificación que es la hidrólisis del agua en hidrogeno y oxígeno, esto puede llegar a ser de mucho peligro y acortaría la vida útil de la batería.

Otra función del regulador es la prevención de la sobre carga, con el fin de evitar que se agote en exceso la carga de la batería, por la cual puede provocar una sensible disminución en la capacidad de carga de la batería en sucesivos ciclos. (Iter, 2008)

El regulador de carga que se utilizara en el proyecto será de 5 Amperios marca Epsolar. Debido que, en el comercio no se encuentra controladores de menor capacidad a los 5 Amperios (Ver figura 24).



Figura 24. Controlador de carga 5 Amperio marca Epsolar.

1.11.4.3. Banco de batería

Las baterías fotovoltaicas son un componente muy importante de todo el sistema, proveen un suministro de energía eléctrica estable y adecuada para la utilización de aparatos eléctricos.

En las instalaciones fotovoltaica lo más habitual es utilizar es un conjunto de baterías asociadas en serie o paralelo para almacenar la energía eléctrica generada durante las horas de radiación, para su utilización posterior en los momentos de baja o nula insolación. La fiabilidad de la instalación depende en gran medida del sistema de acumulación. Su unidad de medida Amperios-hora (Ah).

A la hora de comprar baterías para sistemas fotovoltaicos se debe tener en cuenta la capacidad, eficiencia de carga, auto-descarga y profundidad de descarga

1.12. Electroválvula

La electroválvula tiene como funcionamiento abrirse o cerrarse en un tiempo determinado. La función de la electroválvula en el sistema es de la siguiente manera:

- ❖ La electroválvula que está en la entrada del tanque, es para controlar en nivel del tanque y otra electroválvula para el riego del cultivo.

CAPITULO 2: ANALISIS Y PRSENTACIÓN DE RESULTADOS

En este capítulo se presentan los aspectos relacionados con el diseño e implementación del sistema de riego que se instaló en la comunidad de Totogalpa para el grupo P.S.A. Se describe la metodología de desarrollo del prototipo, pruebas y resultados obtenidos.

2.1. Metodología de desarrollo

Para el diseño del prototipo sistema automatizado de riego se realizó investigación aplicada, debido que se utilizó nuestros conocimientos para resolver el problema del riego del café, además se aplicó el método experimental orientado a actividades de investigación. Este comprende las siguientes etapas:

2.1.1. Análisis

En esta etapa se realizó una visita de campo en la comunidad Totogalpa para conocer acerca de las problemáticas que afectan al grupo P.S.A (ver figura 27). Se realizó una entrevista⁵ al señor Juliam Martinez quien lo hizo de manera espontánea y quien forma parte del grupo P.S.A y que además es uno de los que está a cargo del cuidado del plantío, en la cual exponía la situación del grupo con respecto al cultivo de café. Según lo conversado don Julian exponía que el riego del plantío es una de las problemáticas que aún no se ha podido resolver.

Susan Kinne quien es la coordinadora de las actividades en el grupo P.S.A expresó la labor que han estado haciendo los miembros del grupo con la problemática del plantío de café y que un nuevo sistema de riego puede ser beneficioso para el cultivo y para el grupo, además los autores han venido trabajando⁶ con ella por lo que se considera que la realización de este proyecto vendría a beneficiar al grupo P.S.A.

⁵Entrevista: Ver anexo C.3.

⁶trabajando: Ver anexo C.2.

A continuación según las observaciones por los autores y por don Juliam, se describen las condiciones del sistema de riego del plantío, infraestructura. Asimismo, la alternativa solución propuesta por los autores para mejorar las condiciones de riego del plantío en la comunidad.

2.1.1.1. Condiciones del sistema de riego del café.

El cultivo de café en la comunidad de Totogalpa está al cuidado de Susan kinne. Ella está como coordinadora de todas las actividades en el centro solar en conjunto con el grupo P.S.A.

Actualmente, el método de riego de dicho plantío responde al siguiente funcionamiento. Existen dos tanques de 2500 y 1100 litros respectivamente. El primer tanque (2500 litros) está conectado aproximadamente a 30 metros de un pozo de donde se sirve para extraer agua por medio de una bomba dc sumergible. Una boya interna en el primer tanque cierra el pase del agua en la tubería cuando el tanque está lleno. Este tanque está conectado también a un segundo tanque (1100 litros) a 500 metros de distancia. Luego el segundo tanque abastece de agua a todo el plantío de café el cual está a 200 metros de distancia (ver figura 25). Este tanque tiene en su interior un mecanismo para desactivar la bomba dc por medio de un cableado conectado hacia el controlador que se encuentra a las cercanías del primer tanque. Actualmente, este sistema está en mal estado. Por consiguiente, tiene que estar un usuario en el sitio para desactivar la bomba una vez que se halla llenado el segundo tanque.

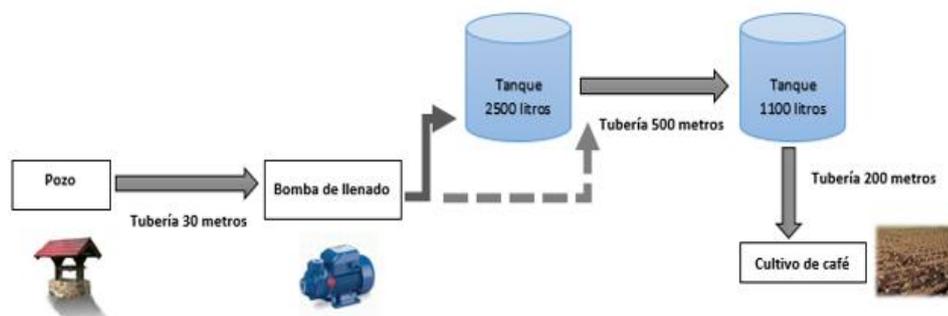


Figura 25. Sistema de riego del café en la comunidad de Totogalpa

El plantío de café tiene un área de $\frac{1}{4}$ de manzana con una cantidad aproximadamente de 300 plantas. Según sus experiencias, los encargados

del riego utilizan el agua del segundo tanque solo para 111 plantas, porque la capacidad del segundo tanque no abastece para todo el plantío. De manera que, los encargados realizan el riego de manera manual para el resto de las plantas.

A causa del descuido y la falta del recurso hídrico han sido un factor que ha afectado drásticamente al cultivo (ver figura 26). Igualmente, se suman a estas algunas enfermedades como la roya y el ojo de gallo. El plantío utiliza el método de riego por goteo con la caída de una gota por segundo a la planta y sembrada sobre limo arcillosa. Las plantas tienen una distancia de aproximadamente 1.28 metros y la producción por temporada es entre 25 y 30 baldes.



*Figura 26.*Cultivo de café comunidad de Tototalpa

2.1.1.2. Requerimientos del sistema

En base a las observaciones por los autores y entrevista realizada al miembro del grupo P.S.A en la comunidad de Totogalpa, se determinan los requerimientos finales para el diseño del prototipo.

- El prototipo debe ser capaz de automatizar el sistema de riego para el cultivo de café.
- El sistema debe de priorizar el llenado del tanque del cultivo de café para su respectivo riego.
- El prototipo debe ser capaz de enviar y recibir de manera inalámbrica los datos de los sensores ubicados en el terreno.
- El prototipo debe ser capaz de presentar toda la actividad relacionada con el cultivo en una pantalla, además del estado del segundo tanque.
- El prototipo debe integrar paneles solares, batería y controlador en conjunto para el suministro de energía de las unidades.
- El prototipo debe ser diseñado de la manera sencilla posible para que el usuario pueda interactuar con este.

2.1.2. Diseño del sistema automatizado de riego

2.1.2.1. Consideraciones del diseño

Modularmente el sistema de riego para el plantío está compuesto por tres unidades que son la unidad central local (P.S.A), unidad remota y la unidad de bombeo. Estas unidades en conjunto con los componentes seleccionados se abordan con más detalles en las siguientes secciones. Referirse a las secciones (2.1.2.3), (2.1.2.4) y (2.1.2.5)

A continuación, se abordan algunos parámetros de diseño propuesto por los autores en conjunto con el tutor que debe cumplir el prototipo:

- El microcontrolador para la Unidad Central debe ser capaz de determinar los tiempos de riego del plantío en base a los datos de los sensores de humedad y temperatura.

La elección de este microcontrolador debe de cumplir las siguientes prestaciones:

- 15 entradas analógicas
 - 30 entradas/salidas digitales
 - 14 salidas PWM
 - Comunicación USB (emulando RS-232)
 - 256k de memoria flash
- El microcontrolador para la Unidad Remota debe ser capaz de recibir y procesar los datos de los sensores de humedad y temperatura del plantío de café.

La elección de este microcontrolador debe de cumplir las siguientes prestaciones:

- 6 entradas analógicas
 - 5 entradas/salidas digitales
 - Comunicación USB (emulando RS-232)
 - Comunicación inalámbrica
- El sistema debe de recibir y presentar los datos de las variables cada 3 segundos.
 - El prototipo debe estar protegido ante cualquier intento de daño, robo o intento de desenergizar las unidades.
 - La alimentación de las unidades debe de contar con un banco de baterías para suplir al menos tres días máximos sin luz solar.

2.1.2.2. Descripción general del prototipo sistema de riego

El prototipo de sistema de riego automatizado está conformado por tres unidades llamadas: Unidad control central (Local P.S.A) en conjunto con la unidad sistema de llenado, unidad sistema de bombeo y unidad sensores remotas. Estas unidades están en constante comunicación con el propósito de automatizar el riego por goteo. La figura 27 muestra el diagrama de bloque general del prototipo.

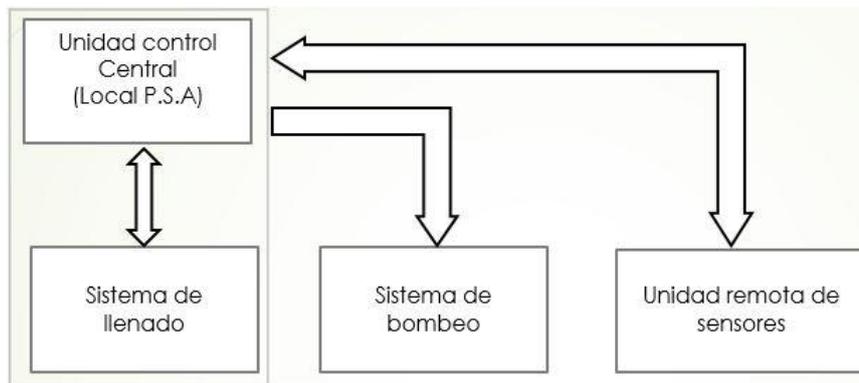


Figura 27. Diagrama de bloque general sistema de riego

El sistema cuenta con tres unidades. Dichas unidades envían información del plantío, estado de una bomba DC sumergible y estado del tanque hacia la Unidad control central (Local P.S.A) para su debido procesamiento, a continuación se describe en términos generales el funcionamiento de cada unidad.

La unidad central (Local P.S.A) es la encargada de recibir la información proveniente de las demás unidades por medio de dispositivo inalámbrico a excepción de la unidad de sistema de llenado y bombeo. Por lo tanto, estas unidades se conectan de manera alámbrica a la unidad central. Por último, la unidad central determina el llenado de los tanques, estado de la bomba y los tiempos de riego para el plantío de café en función de las variables de entrada (humedad, temperatura) y mostrarlos en pantalla para que el usuario puede ver el estado del segundo tanque y el plantío.

El sistema de llenado proporciona información acerca del llenado del segundo tanque y el cierre-abierto de las electroválvulas a la Unidad central. Por consiguiente, la unidad central se encarga de procesar esta información. En fin, la comunicación entre estas dos unidades es de manera alámbrica.

La unidad sistema de bombeo tiene la función del cierre-abierto de la electroválvula para el llenado del primer tanque y de la activación de una bomba DC sumergible ubicada en un pozo a las cercanías del primer tanque. Finalmente, La unidad central por medio de un programa elaborado en Arduino controla esta unidad y están comunicada de manera alámbrica.

La unidad de medición o unidad de sensores remotos se encarga de recolectar datos de temperatura relativa y humedad de suelo del plantío de café y enviarlos hacia la unidad central (Local P.S.A) usando módulos de transmisión XBEE PRO. Por último, la Unidad central (Local P.S.A) determina los tiempos de riego necesario para el plantío en función de las variables de entrada.

El uso de paneles solares es una buena alternativa para alimentar un sistema independiente. Por consiguiente, se ha usado para las unidades paneles solares y baterías en conjunto.

2.1.2.3. Unidad de control central (Local P.S.A)

La unidad central tiene la función de determinar los tiempos de riego para el plantío de café y el estado del tanque. La figura 28 muestra el diagrama de bloque general de la unidad central (Local P.S.A), el cual está integrado el sistema de llenado, sistema de adquisición y representación de datos y alimentación.

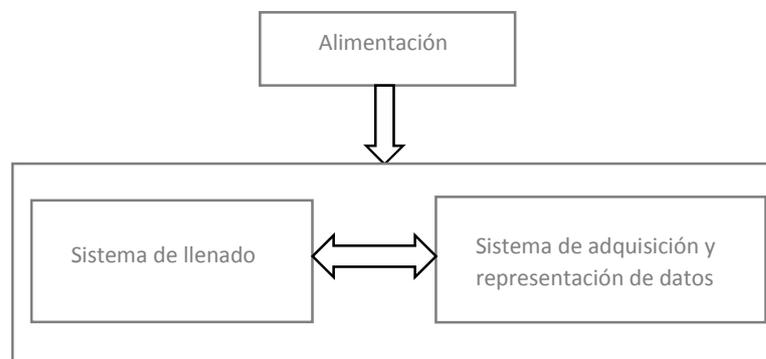


Figura 28. Diagrama de bloque general Unidad Central (Local P.S.A)

2.1.2.3.1. Sistema de adquisición y presentación de datos

Se propone el diagrama de bloque de la figura 29 para el sistema de adquisición y representación de datos. El diagrama está integrado como elemento principal un microcontrolador que tendrá una función importante en el sistema de riego. Igualmente, un módulo de radiofrecuencia, teclado numérico, pantalla cristal líquido y fuente de alimentación.

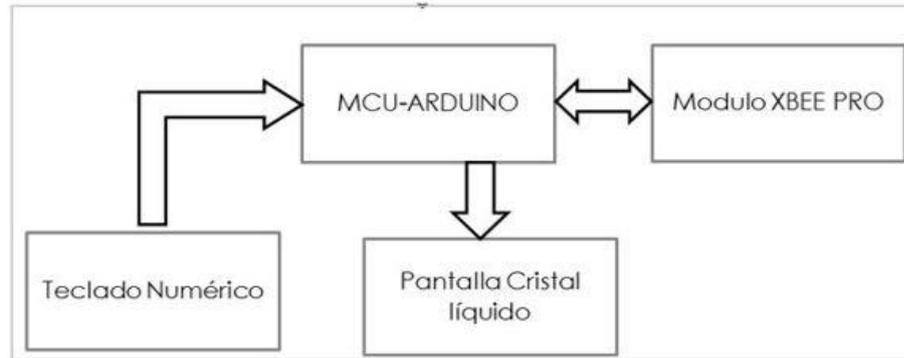


Figura 29.Sistema adquisición y representación de datos

Selección de componentes principales

• Teclado

Para la elección de este dispositivo se decidió utilizar un teclado numérico 4x4 (ver figura 30) como principal elemento de control para el sistema. Un teclado nos permite tomar control de las operaciones del sistema, además permite una interacción entre el sistema y el usuario. Se escogió el teclado Matricial 4x4 por las siguientes razones:

- Su tamaño es adecuado para el sistema de riego.
- Su programación con la plataforma ARDUINO es muy amigable.
- Costo del teclado relativamente bajo.
- Usado en diversas aplicaciones.



Figura 30.Teclado Matricial 4x4.

- **Pantalla**

Para la visualización de datos, se analizaron una serie de pantallas. Dicho análisis permitió obtener dos candidatos para la elección de uno de ellos. La tabla 3 muestra las características más importante de estos dispositivos.

Tabla 3.
Características de pantallas candidatos.

Característica	Pantalla LCD 16x2	LCD-016M004B 16x4
Alimentación	5V _{DC}	5V _{DC}
Corriente	1.30 mA	1.65 mA
Temperatura de operación	50 °C	70 °C
Pantalla táctil	No	No
Costo	\$7	\$14
Dimensiones	80 x 36 mm	70.6 x 60 mm
Lenguaje de Programación	Arduino	Arduino

Se seleccionó la pantalla LCD 16x2 (ver figura 31) por las siguientes razones:

- Su resolución y tamaño es adecuado para el sistema.
- Su programación con la plataforma ARDUINO es muy amigable.
- Dispositivo de bajo costo.

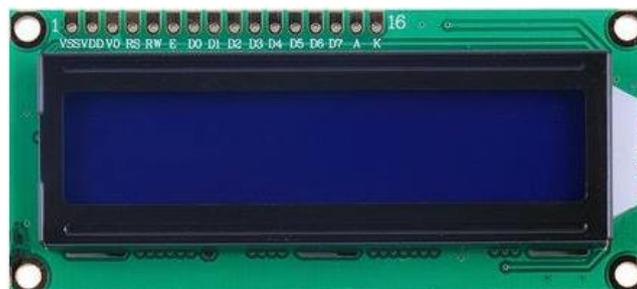


Figura 31. Pantalla LCD 16x2.

- **Módulo de radiofrecuencia**

Para la elección de este módulo se analizaron una serie de módulos de radiofrecuencia. Dicho análisis permitió obtener tres módulos de radiofrecuencia para luego escoger uno de ellos. La tabla 4 muestra las características más importante de estos módulos.

Tabla 4.
Características de los módulos de radio frecuencia candidatos

Característica	SRFW-2500	Xbee zigbee	DR4100
Alimentación	3.0 - 5 V _{DC}	3.3 V _{DC}	2.7 – 3.5 V _{DC}
Corriente de transmisión	30 mA	295 mA	30 mA
Esquema de modulación	FSK	O-KPSK	ASK
Frecuencia de operación	2.4-2.4853 GHz ISM Band	2.4 GHz	433.92MHz
Transferencia de datos	9600bps	250bps	115.2Kbps
Número de canales	16/32 canales	16 Canales	--
Configuración	Software comunicación serial	Comandos AT o API	Software comunicación serial
Antena incorporada	SI	SI	No
Sensibilidad	-105dBm	-102dBm	--
Distancia Máx.	70metros	1500metros	50 metros
Temperatura de operación	-20 - 60°C	-20 - 55°C	--
Dimensión	28x23x8mm	32.94x24.38x25mm	25x22x15mm
Costo	\$20	\$70	\$30

Se seleccionó el módulo xbee Zigbee (ver figura 7) por las siguientes razones:

- Nivel de potencia
- Distancia máxima de transmisión excelente para los requerimientos del sistema.
- Costo es bajo.

- Posee diversas aplicaciones tales como: Domótica, sistema de seguridad y alarma entre otras.
- Antena incorporada

- **Microcontrolador de la Unidad Central**

El microcontrolador para esta unidad debe de cumplir con los requerimientos necesarios basado en las consideraciones de diseño. En la tabla 5 se muestran los microcontroladores candidatos para esta unidad.

Tabla 5.
Características de controladores candidatos para la unidad central.

Característica	PIC18F46k22-I	Atmega 2560	PIC16C74B-20/P
Alimentación	1.8 - 5.5Vcc	4.5 -5.5Vcc	2.5 – 5.5Vcc
Costo	\$6.62	\$5.11	\$10
Frecuencia	32KHz,16 MHz	16 MHz	20MHz
Puertos E/S	36	54	33
Ancho de bus de datos	8 bits	8 bits	8 bits
Tipo de interfaz	USART x2	I2C, SPI, USART	USART,SSP
Temperatura de trabajo	+ 75°	+ 85°	+ 80°

El microcontrolador escogido es el atmega 2560, escogido por su bajo costo, tiene suficientes puertos I/O, además de su comunicación con módulos inalámbricos. El microcontrolador atmega 2560 viene incorporada en una placa ARDUINO llamada ARDUINO Mega 2560 (ver figura 4). El Arduino Mega 2560 es una tarjeta de hardware libre basada en una placa que tiene como elemento principal el microcontrolador Atmega 2560 y un entorno de desarrollo.

El Arduino Mega 2560 es una de las tarjetas más recientes dentro de la familia Arduino, el cual soporta protocolo de comunicación (I2C(TWI), SPI,UART), un cristal oscilador de 16MHz, conector ICSP, puertos digitales y analógicas, además contiene una serie de elementos pasivos y activos.

El microcontralador en la placa Arduino se programa mediante el lenguaje de programación Arduino basado en Wiring (vea figura 5). Es lo suficientemente flexible para el usuario. Por lo tanto, no necesita de un programador con profundos conocimientos en el tema para realizar su programación. Se debe de instalar el software Arduino a la pc. Este programa es gratis y fácil de utilizar, pero se debe de tener conocimientos básicos de programación en C C++, porque está basado en este lenguaje de programación.

Para la transferencia del archivo .hex (programa de control en código decimal) de la pc a la placa Arduino solamente se utiliza un cable USB, además es una de las maneras de alimentar la placa.

Existen una gran variedad de placas Arduino y para saber que tarjeta utilizar dependerá de la magnitud del proyecto.

Hardware propuesto por el sistema de adquisición y representación de datos

Se propone el siguiente diagrama circuital (ver figura 32) para el sistema de adquisición y representación de datos.

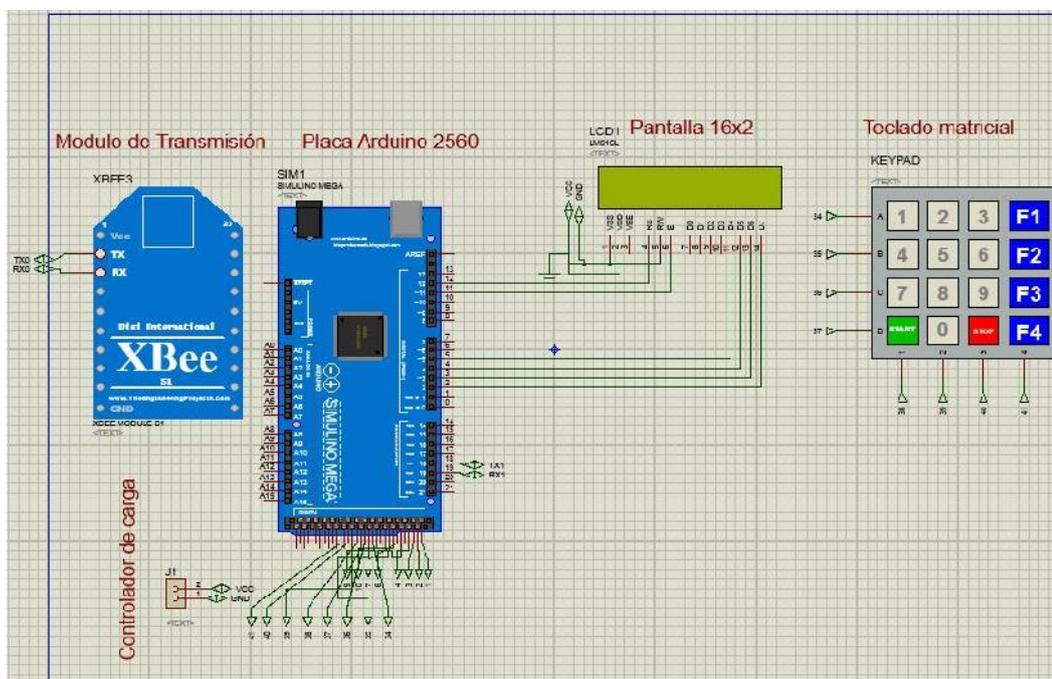


Figura 32. Circuito propuesto para el sistema de adquisición y representación de datos, elaborado en software Proteus

A continuación se menciona las características de la figura 32

- La pantalla LCD 16x2 es alimentada por una circuitería llamada circuito distribuidor de carga para la unidad central (ver figura 48) que proporciona 5V. El módulo de radiofrecuencia es alimentado por la tarjeta Arduino que proporciona 3.3V.
- La tarjeta Arduino es alimentada por el circuito distribuidor de carga que proporciona 12V.
- Los módulos de radiofrecuencia configurados en el programa X-CTU están conectados a las tarjetas Arduinos. En la tarjeta Unidad Central el módulo RF llamado Coordinador. En la tarjeta de la Unidad Remota el módulo RF llamado Router. La tabla 6 muestra los pines que van conectados los módulos de radiofrecuencia a las diferentes tarjetas Arduinos.

Tabla 6.
Módulos Xbee pro en las tarjetas Arduino.

Tarjeta Arduino	Módulo xbee Coordinador	Módulo xbee Router
Unidad Central	TX1 y RX1	
Unidad Unidad Remota		TX y RX

Configuración del módulo Xbee pro S2 Coordinador

El X-CTU (ver figura 8) es el software que se utilizó para configurar los módulos xbee. El programa está compuesto por cuatro pestañas en la cual se usan para programar parámetros de los módulos, testear la comunicación, configurar los módulos en modo sleep para ahorrar energía, visualizar el número de paquetes enviados entre otros.

A continuación se presenta el desarrollo de la configuración del módulo xbee pro para la resección de datos.

Se propone para este trabajo el uso del modo estándar para un enlace directo punto a punto para el envío de datos de la Unidad remota a la Unidad Central (Local P.S.A) y el uso de un shield (ver figura 34) para la conexión del módulo con el Arduino. Para la configuración del Módulo xbee pro Coordinador hemos usado una placa Arduino UNO. El Arduino uno sirvió como puente para comunicar el módulo RF con la pc (ver figura 33).

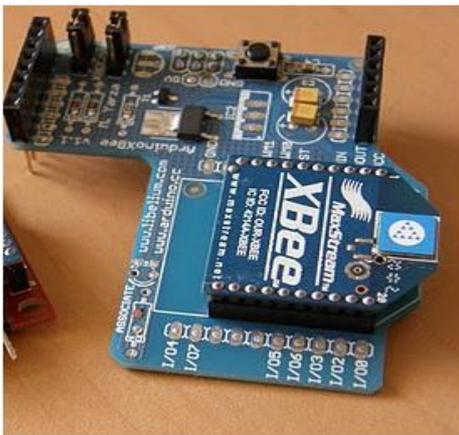


Figura 34. Shield conectado al módulo xbee pro



Figura 33. Conexión del módulo xbee pro a Arduino

Para el uso del shield se debe tener en cuenta a la hora de la práctica seleccionar el modo de trabajo. Esta placa posee dos Jumpers etiquetados con XBee y USB (ver figura 35). A la hora de configurar el módulo se debe colocar los dos jumper en el modo USB y extraer de la tarjeta Arduino uno el microprocesador. Esto es, para conectar directamente el módulo RF a la pc por medio de la placa Arduino uno y acceder al sistema del XBee.

Los jumpers en la posición XBee son para la transmisión o recepción de datos. Una vez configurado el módulo XBee se coloca nuevamente el microcontrolador. El microcontrolador solo será capaz de recibir datos desde el módulo XBee y no desde el USB del ordenador.

Accesar al sistema de los XBee se requiere de una comunicación serial. De igual forma, la tarjeta Arduino requiere de una comunicación serial para programar su microprocesador. Es por eso que, extrayendo el microprocesador de la tarjeta Arduino hará que el módulo XBee se conecte directamente a la pc.

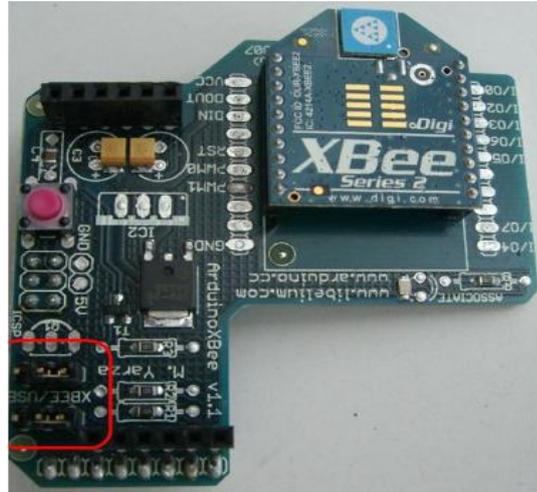


Figura 35. Jumpers XBee y USB del Shield

A continuación se presentan los pasos a seguir durante el proceso de configuración del módulo RF coordinador:

1. Conectar el módulo al shield en modo jumper USB.
2. Extraer el microcontrolador del Arduino uno y conectar el shield.
3. Instalar drivers FTDI para nuestro sistema para que el módulo aparezca conectado como un puerto COM.
4. Abrir software X-CTU.
5. Seleccionamos el puerto disponible en el X-CTU y hacemos un Test/Query para comprobar si el software X-CTU se pueda comunicar con el módulo.
6. Si el software se comunica con el módulo seleccionamos la pestaña "Modem Configuration"
7. En la lista "Módem" Para módulos serie 2 seleccionamos XB24-ZB
8. En la lista "Function set" configuramos nuestro módulo.
9. Seleccionamos la opción ZIGBEE COORDINATOR AT
10. En la carpeta Networking ingresamos un ID en la opción "PAN ID" en nuestro caso es 1234. Este código será compartido con el módulo Router. Vale decir, que solo podemos ingresar valores hexadecimales de 0 FFFE.
11. En la carpeta Addressing ingresamos en la opción "Destination Address High" o DH el valor 13A200. Este valor lo encontramos en la parte de atrás de los módulos XBee.

12. En la misma carpeta seleccionamos la opción “Destination Address Low” o DL el valor del módulo Router. En nuestro caso es 40A62191. Este valor se encuentra en la parte de abajo del módulo XBee y es único.
13. En la carpeta “RF Interfacing” seleccionamos la opción “Power level” o PL. Escogemos la opción de nivel de potencia en nuestro caso HIGHEST.
14. En la carpeta “Serial Interfacing” seleccionamos la opción “BAUD RATE”. Seleccionamos la opción 3 - 9600. Este valor debe de ser igual para el módulo router.
15. En la misma pestaña “Modem Configuration” damos clic en la opción “write” para guardar estos cambios en el módulo coordinador.
16. Una vez hecho todo esto desconectamos el Arduino UNO de la pc para extraer el shield del módulo XBee.

Algoritmo propuesto por el sistema de adquisición y representación de datos

Se propone el siguiente diagrama de flujo (ver figura 36) para la unidad de adquisición y representación de datos.

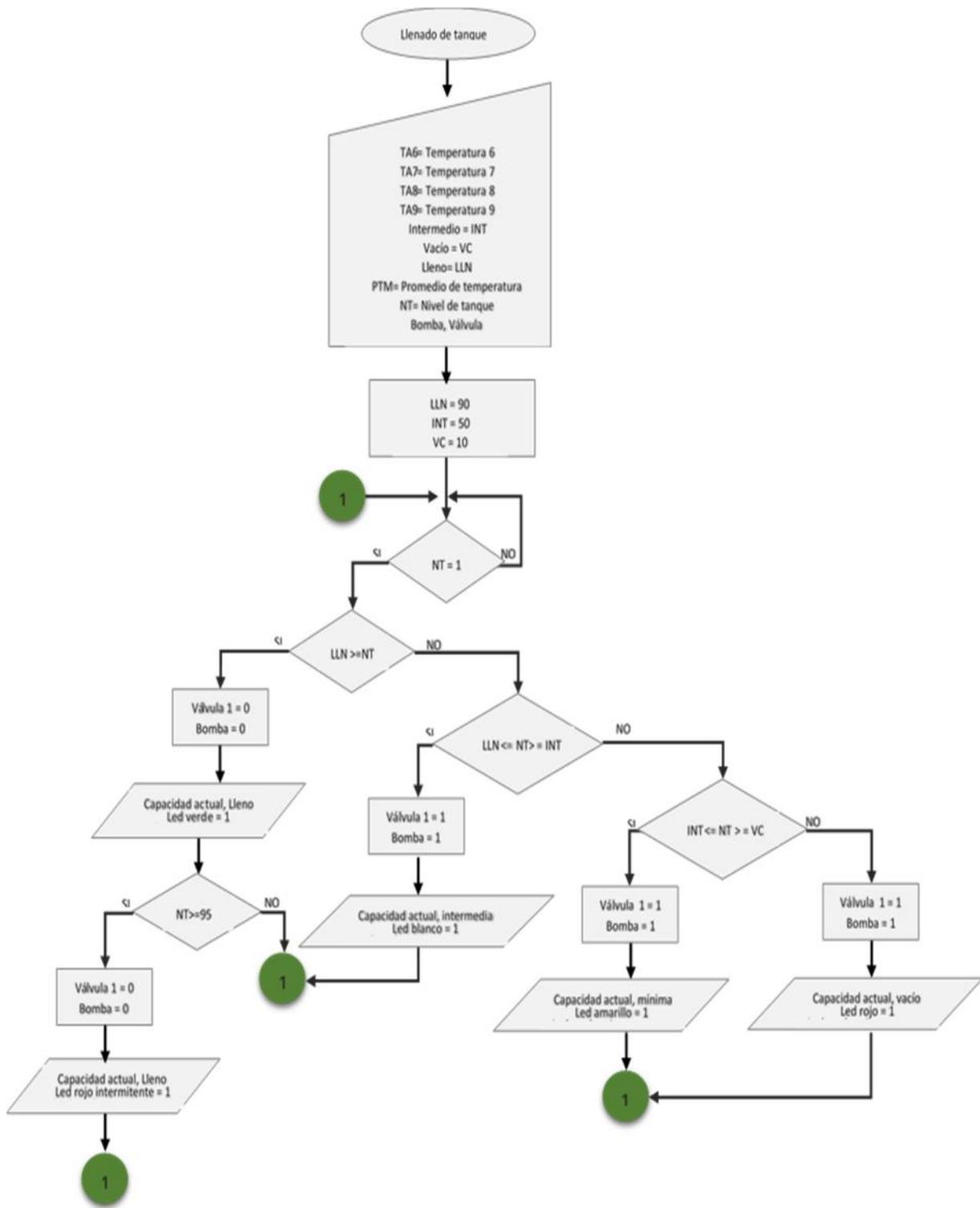


Figura 36. Diagrama de flujo de adquisición y representación de datos

2.1.2.3.2. Sistema de llenado

Se propone el diagrama de bloque para el sistema de llenado (ver figura 37). Esta unidad en conjunto con la unidad de adquisición y representación de datos están encargadas del monitoreo del llenado del tanque para el abastecimiento del recurso hídrico hacia el cultivo de café, también el cierre-abierto de las electroválvulas del segundo tanque. La figura 38 muestra el diagrama de bloque de la unidad sistema de llenado y están integrados un sensor de proximidad, módulo relé y electroválvulas.

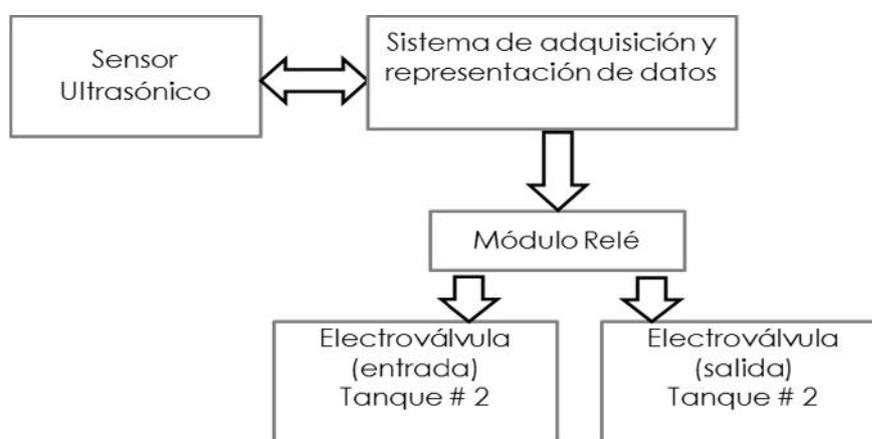


Figura 37. Sistema de llenado

Selección de componentes principales

- **Sensor de distancia**

Para la elección de este módulo se analizaron una serie de módulos de proximidad. Dicho análisis permitió obtener tres modelos para luego escoger uno de ellos. La tabla 7 muestra las características más importante de estos módulos.

Tabla 7.

Características de sensores de movimiento candidatos.

Característica	Sensor HC-SR04	Sensor PIR BV-300
Alimentación	5 V _{DC}	12 V _{DC}
Costo	\$9.79	\$25
Frecuencia	40 KHz	--
Atraviesa objetos	NO	NO
Alcance Max.	400 cm	300 cm
Dimensión	45x20x15mm	3.9x2.8x2.6
Corriente trabajo	15mA	--

Se seleccionó el sensor HC-SR04 (ver figura 18) por las siguientes razones:

- Costo es bajo
- Su programación con la plataforma ARDUINO es muy amigable.
- Distancia máxima de operación excelente para los requerimientos del sistema

• **Electroválvula**

Para este trabajo se proponen el uso de las electroválvulas 2W-250-25 de una pulgada (ver figura 38) y D9152 (ver figura 39) de media pulgada por los siguientes motivos:

- Operan a 12 VDC
- Simple instalación
- Rango de presión de 0-1 Mpa para electroválvula 1”
- Temperatura de fluidos de -5 80°C para electroválvula 1”
- Ideal para conectarse con baterías
- Bajo nivel de mantenimiento
- Resistentes



Figura 38. Electroválvula 1 pulgada



Figura 39. Electroválvula 1/2 pulgada

• **Módulo relé**

Se propone el uso de un módulo relé para la activación de las electroválvulas. El módulo provee dos canales con relés de 5VDC. Este módulo es un interruptor operado eléctricamente que activa un circuito usando un voltaje

mayor que el del microcontrolador a las salidas del relé. La figura 40 muestra el módulo relé y se escogió por las siguientes razones:

- Voltaje de operación 5VDC.
- 2 canales independientes con opto acopladores.
- 2 señales de entrada con corriente activación 15 20mA.
- V_{máx} Salida 10Amp-220 VAC y 10Amp-30VDC.
- Tamaño 45.8mm x32.4mm.



Figura 40. Módulo Relé

Hardware propuesto para el sistema de llenado

Se propone el siguiente diagrama circuital (ver figura 41) para el sistema de llenado.

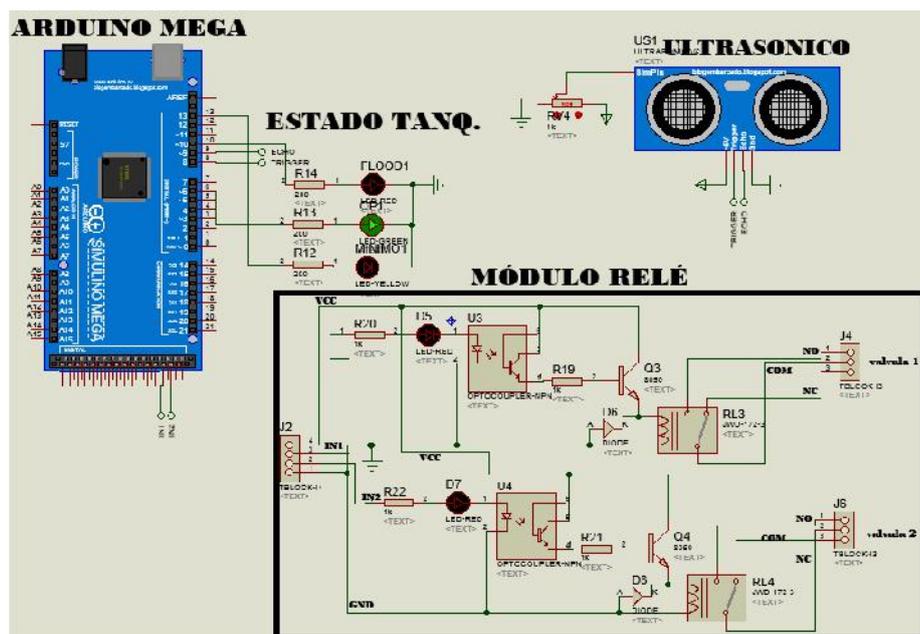


Figura 41. Circuito propuesto para el sistema de llenado, elaborado en software

Funcionamiento.

El sensor ultrasónico es utilizado para monitorear el nivel de llenado del tanque 2 del cultivo de café. La placa Arduino de la unidad central se encarga de recibir los datos provenientes del sensor ultrasónico. Este sensor se encuentra ubicado en la parte superior del tanque y envía información del tanque cada 10us a la unidad central. Por otro lado, los indicadores de led indican los niveles del tanque en tres estados (vacío, óptimo y desborde).

El módulo relé controlado por la microcontrolador Atmega 2560 de la unidad central se encarga de activar dos elementos a su salida que son una electroválvula de una pulgada y una electroválvula de media pulgada del tanque 2. La alimentación de este sistema es por medio de un circuito distribuidor de carga que se aborda más adelante.

2.1.2.3.3. Alimentación Unidad de control central (Local P.S.A)

La alimentación general de la unidad Central (Local P.S.A) está constituida por un panel solar, controlador de carga y una batería diseñadas en función de demanda de consumo de energía para la unidad central. Se propone el diagrama de bloque de la figura 42 para dicha unidad.

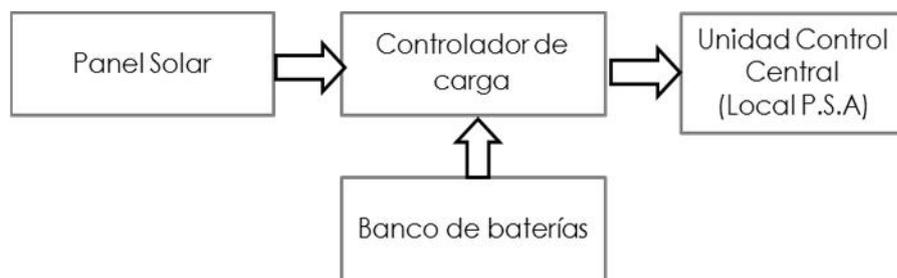


Figura 42. Diagrama de bloque alimentación Unidad Central

- **Diseño sistema fotovoltaico**

A continuación se presenta el desarrollo de cálculo para el diseño sistema fotovoltaico para la Unidad Central (Local P.S.A).

Para el desarrollo de estos cálculos se debe tener en cuenta:

- ❖ Localización del sitio (Latitud y Longitud) donde se va a instalar el sistema (ver tabla 8).
- ❖ Tabla de radiación del sitio (ver tabla 8).
- ❖ Consumo de energía de la unidad.
- ❖ Disponibilidad de nuestro sistema al 95% o 99%.

Tabla 8.

Promedio mensual incidencia de radiación en una superficie inclinada apuntada al ecuador en (KWh/m²/día). (NASA, 1958)

Lat. 13,527 Long 86,48 Ángulo 13°	Mes (KWh/m²/día)
Enero	6.06
Febrero	6.7
Marzo	7.26
Abril	6.87
Mayo	5.9
Junio	4.57
Julio	4.53
Agosto	4.44
Septiembre	5.01
Octubre	5.25
Noviembre	4.97
Diciembre	5.43
Promedio Anual	5.58

Para conocer el consumo de energía es necesario saber que dispositivos vamos a alimentar. La tabla 9 muestra el consumo de energía para la Unidad Central (Local P.S.A).

Tabla 9.
Consumo de energía Unidad Central.

Cantidad	Componente	Corriente A	Voltaje V	Potencia unitaria (W)	Potencia General (W)	Horas de trabajo	Energía kwh/día	Energía Kwh/Mes	Energía Kwh/Año
1	ARDUINO Kit	2	12	24	24	24	0.58	17.57	210.24
1	Sensor Ultrasónico	0.015	5	0.075	0.075	24	0.0018	0.05	0.66
1	Electroválvula 1/2 pulgada	0.25	12	3	3	7	0.02	0.64	7.67
1	Electroválvula 1 pulgada	0.25	12	3	3	7	0.02	0.64	7.67
1	Pantalla de cristal liquido	0.025	5	0.125	0.125	8	0.001	0.03	0.37
2	Módulos Relé	0.04	5	0.2	0.4	8	0.0032	0.10	1.17
1	Bombillo DC	0.58	12	7	7	3	0.02	0.64	7.67
							0.65	19.67	235.43

Con respecto a la tabla 8, el consumo de energía para la Unidad Central es de 19.67 Kwh/mes y 235.43 kwh/año. Estos datos son de gran importancia para el cálculo de nuestro panel y batería. La figura 43 muestra en porcentaje el consumo de cada dispositivo de manera gráfica siendo el Arduino el dispositivo de mayor consumo de energía para ésta unidad. Esto es debido que, ciertos dispositivos están conectados a el. Ya que el Arduino también provee voltajes de 3.3V_{DC} y 5V_{DC}, además este está encargado de todas las operaciones y procesos del sistema de riego.

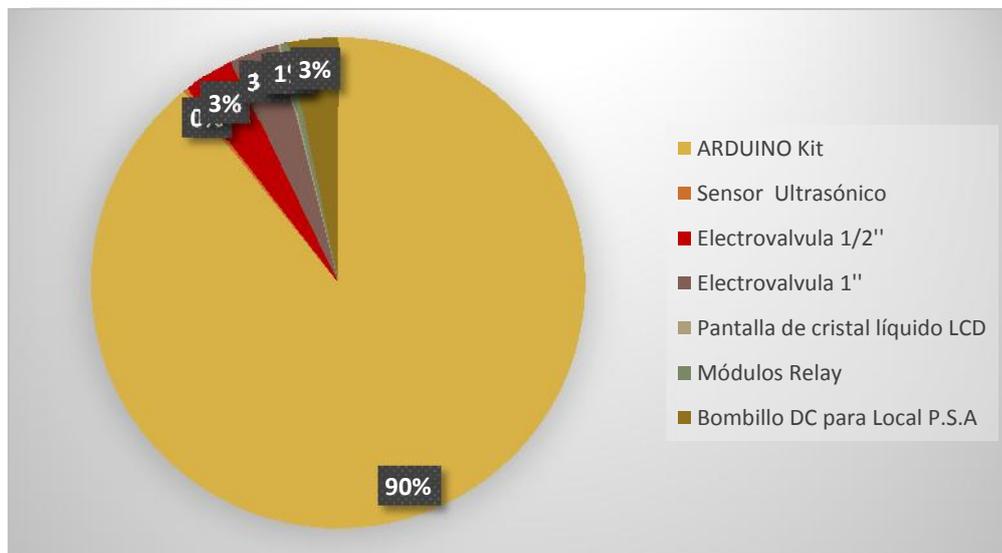


Figura 43. Consumo de energía Unidad central (Local P.S.A)

- **Panel solar Unidad Central**

El proceso de calcular la eficiencia de nuestro panel para esta unidad se basa en seleccionar un arreglo de módulos (paneles) para calcular en base a las características de estos módulos y las ecuaciones que se usaron del libro de Sistema de energía eléctrica renovable y eficiente, la potencia requerida para la unidad central. Por lo tanto, el panel seleccionado para esta unidad es de 160W 12V_{DC}.

Un factor de pérdida de 0.85 es usado. Esto es basado en los siguientes parámetros: Factor de sombra 95%, eficiencia del inversor 100%, cableado DC 98%, cableado AC 99%, factor pérdida por suelo 95%, factor pérdida por desacoplamiento 98%, factor disponibilidad del sistema 95%.

Características eléctricas del panel seleccionado.

- Potencia máxima a STC(Pmax): 160 W
- Operación nominal temperatura de celda (NOCT): 45C±2C
- Corriente de corto circuito (Isc): 8.91 A
- Coeficiente de temperatura de Pmax: -0.44%C
- Voltaje de circuito abierto (Voc): 22.3 V
- Coeficiente de temperatura de Voc: -0.34%C
- Coeficiente de temperatura de Isc: 0.06%C

Cálculos para el número de paneles Unidad Central

Conociendo los datos de irradiación solar y el consumo de energía por los equipos para la unidad central partimos para calcular el valor de P_{ac} en (3).

$$P_{ac} = \frac{\text{Energía(Kwh/año)}}{\left(\frac{h}{\text{día}}\right) \text{ de " insolución solar" }} \quad (3)$$

Donde:

Energía: Consumo de energía anual en (Kwh/año) de los equipos.

P_{ac} : Potencia ac de un sistema fotovoltaico ubicado en el campo en W.

$P_{dc,STC}$: Potencia dc en W de un arreglo obtenido bajo condiciones de prueba estándar:

Insolación $\left(1 \frac{\text{Kwh/m}^2}{\text{día}}\right)$, *Aire de masa*: 1.5, *Temperatura de la celda*: 25°C.

Introducimos los valores de la tabla 7 y 8 en (3). El valor de P_{ac} es el siguiente:

$$P_{ac} = 115.59 \text{ W}$$

Para conocer el valor de $P_{dc,STC}$ se usó (4)

$$P_{dc,STC} = \frac{P_{ac}}{\text{Factor de pérdida}} \quad (4)$$

Introducimos los valores en (4). El valor de $P_{dc,STC}$ es el siguiente:

En (4). El valor de P_{ac}

Para conocer el número de paneles tomamos en cuenta la potencia del módulo. Introducimos los datos en (5)

$$\text{Número de módulo} = \frac{P_{dc,STC}}{\text{Potencia módulo}} \quad (5)$$

Para la Unidad Central hemos elegido un panel de 160 W. Introducimos los valores en (5). El valor de *Número de módulo* es el siguiente:

$$\text{Número de módulo} = 0.85 \approx 1$$

Para conocer la potencia generada por el módulo utilizamos (6)

$$P_{dc} = \text{Potencia módulo} \times \text{Número de módulos} \quad (6)$$

Para esta ecuación se introducen los valores en (6) teniendo en cuenta la potencia del panel. El valor de P_{dc} es el siguiente:

$$P_{dc} = 160 \text{ W}$$

- **Calculo de temperatura de la celda**

Para saber el valor de la temperatura de la celda, según el tipo de panel el fabricante provee una serie de características que debemos tener en cuenta. Entre ellas tenemos el valor del NOCT (Nominal Operating Cell Temperature), temperatura de la celda, luminosidad solar entre otras bajo condiciones estándar. Este dato es útil debido que entre mayor sea la temperatura ambiente el voltaje del panel disminuye. Por consiguiente, será necesario conocer este dato para conocer la energía generada por nuestro panel bajo estas condiciones. Para nuestro panel tenemos un valor de NOCT=45 Este valor se calcula conociendo el valor de la temperatura en el sitio (ver figura 45) y usando (7).

$$T_{cell} = T_{amb} + \left(\frac{NOCT - 25^\circ}{0.8} \right) \times 1 \frac{kW}{m^2} \quad (7)$$

Los resultados de la temperatura de la celda se reflejan en la tabla 9.

La temperatura ambiente en la comunidad de Totogalpa (ver figura 44) se muestra con un promedio mensual y varía según los meses del año.

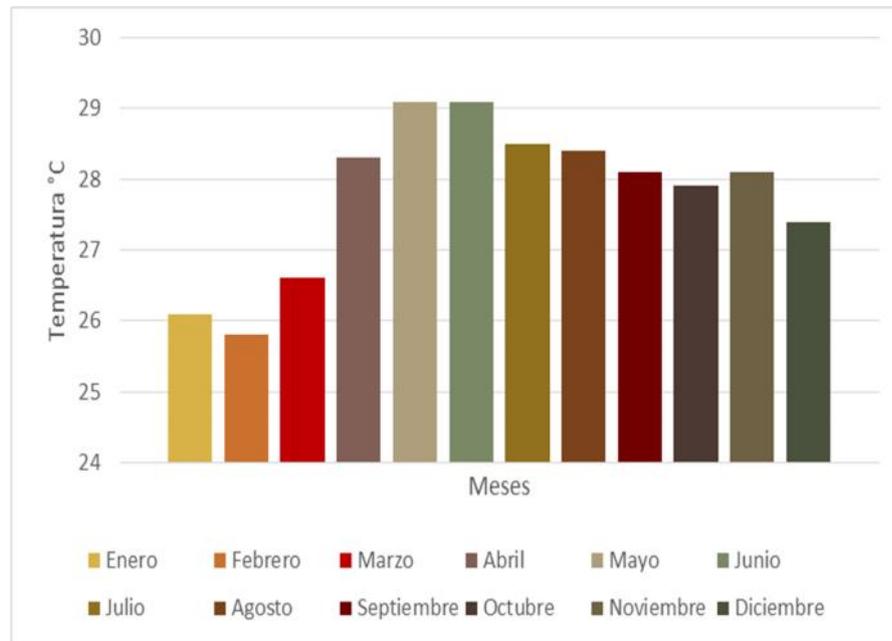


Figura 44. Temperatura ambiente comunidad de Totogalpa (NASA, 2016)

La energía generada por el sistema fotovoltaico se conoce usando (8) tomando en cuenta la temperatura de la celda.

$$Energía : P_{dc} \times Factor\ de\ pérdida \times Insolación\ mensual \times (1 - T_{cell} - P_{máx} (T_{amb} - 25)) \quad (8)$$

La tabla número 9 muestra la energía generada mensualmente por el sistema fotovoltaico y la energía total por un año tomando en cuenta la temperatura de la celda, horas de sol al día en el sitio, temperatura ambiente y factor de pérdida.

La producción anual de energía por el sistema fotovoltaico (ver tabla 10) es de 239.21 Kwh/año lo cual es suficiente para cumplir con la demanda de consumo de energía para la unidad central (ver tabla 9).

En síntesis, según los cálculos realizados se propone el uso de un panel de 12 VDC 160 W para la unidad central (Local P.S.A).

Tabla 10.
Energía generada por el panel solar.

Potencia (W)	Factor pérdida	Incidencia de radiación solar-Tilt 13 kWh/m ² /d	Mes	Días del mes	Energía Kwh/mes	Temperatura °C	T de la celda	Energía Kwh/Mes
160	0.85	6.06	Enero	31	25.25	26.1	57.35	22.24
160	0.85	6.7	Febrero	28	25.51	25.8	57.05	22.24
160	0.85	7.26	Marzo	31	30.61	26.6	57.85	26.59
160	0.85	6.87	Abril	30	28.03	28.3	59.55	24.16
160	0.85	5.9	Mayo	31	24.87	29.1	60.35	21.36
160	0.85	4.57	Junio	30	18.65	29.1	60.35	16.01
160	0.85	4.53	Julio	31	19.10	28.5	59.75	16.44
160	0.85	4.44	Agosto	31	18.72	28.4	59.65	16.12
160	0.85	5.01	Septiembre	30	20.44	28.1	59.35	17.63
160	0.85	5.25	Octubre	31	22.13	27.9	59.15	19.11
160	0.85	4.97	Noviembre	30	20.28	28.1	59.35	17.49
160	0.85	5.43	Diciembre	31	22.89	27.4	58.65	19.81
					276.28			239.21

La figura 45 compara el consumo de energía generada a un año de la unidad central con la energía producida por nuestro panel.



Figura 45. Comparación energía solar y consumo de energía por mes

- **Controlador de carga Unidad Central**

Para proteger la batería frente a sobrecargas y descargas profundas se propone el uso de un controlador de carga. El controlador seleccionado para esta unidad es de 5A (ver figura 27). Debido que, comercialmente no se encuentran controladores de menor carga.

- **Cálculo de capacidad de la batería Unidad Central**

Para conocer la capacidad de la batería para esta unidad se desarrolló a través de las ecuaciones del libro Sistema de energía eléctrica renovable y eficiente, por lo que, es necesario tener en cuenta:

- ❖ Insolación solar en el sitio (ver tabla 6).
- ❖ Disponibilidad de nuestro sistema 95% (ver figura 46).
- ❖ Consumo de energía por día (ver tabla 7).

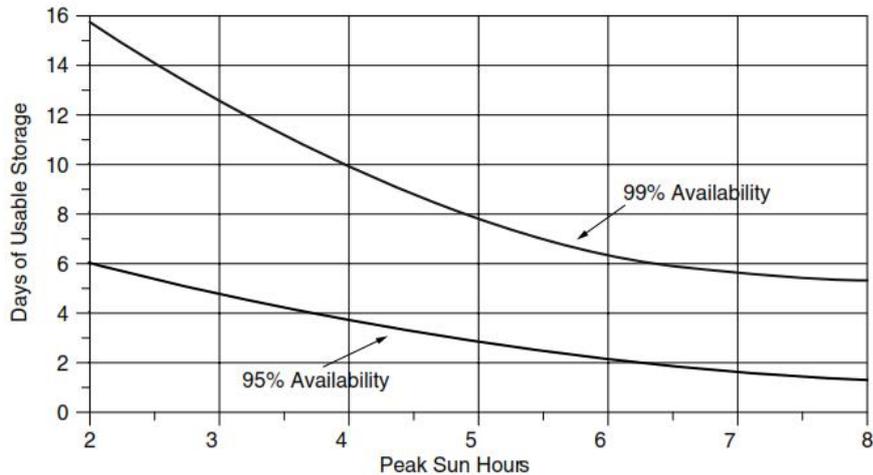


Figura 46. Días de almacenamiento de batería para un sistema independiente con 95% y 99% de disponibilidad del sistema. (Gilbert, 2004)

Para una disponibilidad de nuestro sistema al 95% agosto es el mes con condiciones más desfavorables de irradiación. Por tanto, es nuestro mes de diseño con 4.44 h/día "máxima irradiación" (ver tabla 7). Comparando este dato con la figura 46 se estimó que la batería puede entregar energía a un máximo de tres días a la unidad central. Es decir, nuestra batería tiene que suministrar energía para cubrir la demanda en peores casos sin luz solar a la unidad por tres días máximo.

Para el cálculo del diseño de la batería para ésta unidad, se escogió una batería a 12 VDC en conjunto con los valores de la tabla 9. El valor de la carga se calcula usando (9).

$$Carga_{DC} = \frac{\text{Consumo } Wh/día}{\text{Voltaje batería}} \quad (9)$$

El valor de la carga es el siguiente:

$$Carga_{DC} = 54.16 \frac{Ah}{día} \text{ a } 12V$$

Para conocer el valor de almacenamiento de la batería incluyendo los días máximo sin sol usamos (10)

$$Almacenamiento = Carga DC \times \text{días de almacenamientos} \quad (10)$$

El resultado esperado es:

$$Almacenamiento = 162.48 Ah$$

Aunque se ha tomado en cuenta los días máximos de almacenamiento de energía para la unidad, es necesario también tomar en cuenta otros factores como factor de máximo descarga (F.M.D = 80%) y factor de tasa de descarga y temperatura (F.D.T = 97%) bajo condiciones estándar de pruebas (C/20 a 25°C) con el objetivo de asegurar la capacidad de la batería a los tres días. Se calcula usando (11)

$$Nominal \left(\frac{C}{20}, 25^\circ C \right) \text{ Capacidad de la batería} = \frac{Almacenamiento (Ah)}{0.80 \times 0.97} \quad (11)$$

La capacidad real de la batería para la Unidad Central es:

$$Nominal \left(\frac{C}{20}, 25^\circ C \right) \text{ Capacidad de la batería} = 209.38 Ah a 12V$$

2.1.2.3.4. Circuito distribuidor de carga

Para la distribución de carga para la unidad central (Local P.S.A) se propone el siguiente diagrama circuital (ver figura 48).

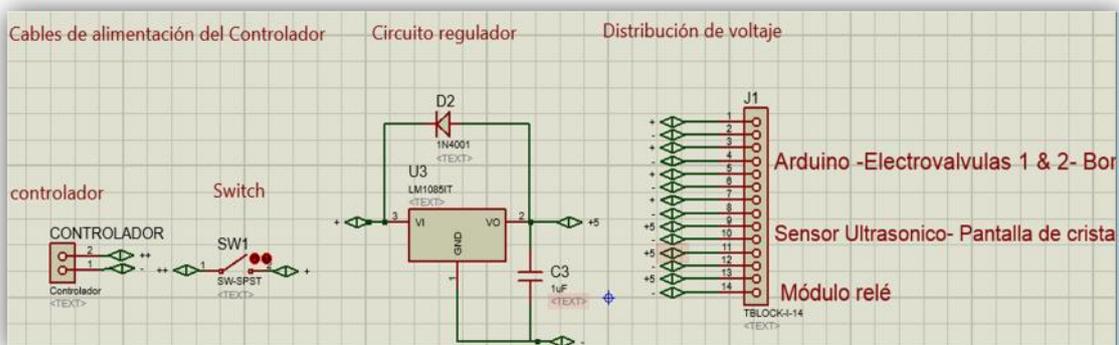


Figura 47. Diagrama circuital distribuidor de carga UC

A continuación se mencionan las características de la figura 48

- La pantalla LCD 16x2, módulo relé y el sensor ultrasónico son alimentados por un regulador de voltaje LM1085IT que proporciona a su salida un valor de $5V_{DC}$.
- La placa Arduino Mega, un bombillo y las electroválvulas de una y media pulgada son alimentados directamente del controlador.

2.1.2.4. Unidad remota de medición

Esta unidad está en contacto de manera inalámbrica con la unidad central (Local P.S.A). La función de esta unidad es enviar datos relevantes tales como humedad y temperatura del plantío de café. La figura 48 muestra el diagrama de bloque general de la unidad remota de medición. En esta unidad está integrado como elemento principal una tarjeta Arduino UNO, además un módulo de transmisión de datos xbee, sensor de humedad, sensor de temperatura y su respectiva alimentación.

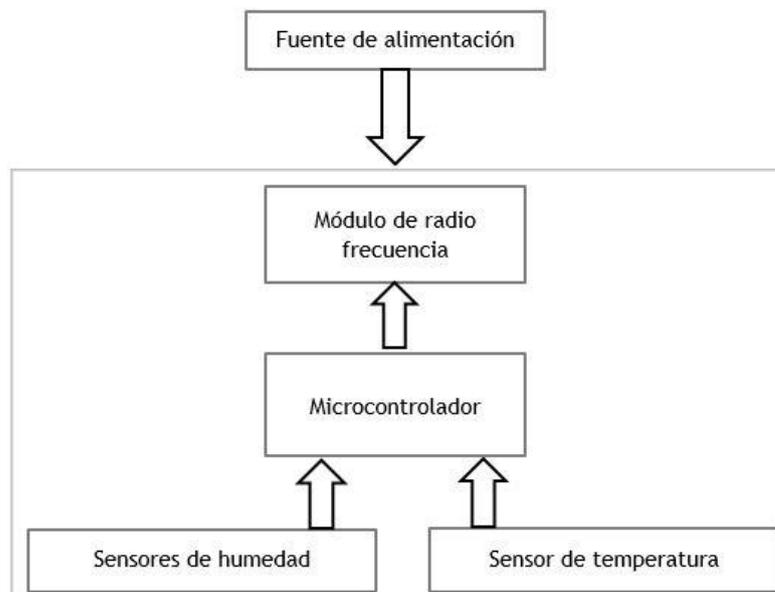


Figura 48. Unidad Remota de medición

Selección de componentes

- **Sensor de humedad**

Para la selección de este sensor se analizaron una serie de sensores de humedad. El análisis permitió obtener 3 sensores (ver tabla 11) mostrando las características importantes de los candidatos.

Tabla 11.
Característica sensores de humedad.

Característica	Sensor MAS-1	Sensor YL-69	Sensor DTH11
Alimentación	12~32 V _{DC}	3.3~5 V _{DC}	5 V _{DC}
Max. Corriente	20mA	35 mA	--
Dimensión	8.9 x 1.8 x 0.7 cm	60 x 30 mm	0.2x0.2x0.2 cm
Temperatura de operación	-40 a 50 °C	0 a 50 °C	0 a 50 °C
Costo	\$5	\$2	\$4.60

Las razones por la cual se escogió el sensor YL-69 (ver figura 15) son:

- Su programación con la plataforma ARDUINO es muy amigable.
- Costo del sensor relativamente bajo.
- Contiene un módulo YL-38 con un circuito comparador LM393.
- Sensor resistivo de bajo consumo.
- Uso para diversas aplicaciones.

• Sensor de temperatura

Para la elección de este sensor de temperatura se analizaron una serie de sensores. Dicho análisis permitió obtener tres modelos para luego escoger uno de ellos. La tabla 12 muestra las características más importante de estos sensores.

Tabla 12.
Característica sensores de temperatura candidatos.

Característica	SHT11	LM35	TMP20
Alimentación	5.5 V _{DC}	4 - 20 V _{DC}	1.8 - 5.5 V _{DC}
Tipo de salida	Lineal	Lineal	Lineal
Dimensión	6.9 x7.6 cm	60 x 30 mm	40 x 25 mm
Temperatura de operación	-40 a 123 °C	-55 a 150 °C	-55 a 130 °C
Costo	\$20	\$2.67	\$15

3. Instalar drivers FTDI para nuestro sistema para que el módulo aparezca conectado como un puerto COM.
4. Abrir software X-CTU
5. Seleccionamos el puerto disponible en el X-CTU y hacemos un Test/Query para comprobar si el software X-CTU se pueda comunicar con el módulo.
6. Si el software se comunica con el módulo seleccionamos la pestaña “Modem Configuration”
7. En la lista “Módem” Para módulos serie 2 seleccionamos XB24-ZB
8. En la lista “Function set” configuramos nuestro módulo.
9. Seleccionamos la opción ZIGBEE ROUTER AT
10. En la carpeta Networking ingresamos un ID en la opción “PAN ID” en nuestro caso es 1234. Este código será compartido con el módulo Coordinador. Vale decir, que solo podemos ingresar valores hexadecimales de 0 FFFE.
11. En la carpeta Addressing ingresamos en la opción “Destination Address High” o DH el valor 13A200. Este valor lo encontramos en la parte de atrás de los módulos XBee.
12. En la misma carpeta seleccionamos la opción “Destination Address Low” o DL el valor del módulo Coordinador en nuestro caso es 40A621CF. Este valor se encuentra en la parte de abajo del módulo XBee y es único.
13. En la carpeta “RF Interfacing” seleccionamos la opción “Power level” o PL. Escogemos la opción de nivel de potencia en nuestro caso HIGHEST.
14. En la carpeta “Serial Interfacing” seleccionamos la opción “BAUD RATE”. Seleccionamos la opción 3 - 9600. Este valor debe de ser igual para el módulo router.
15. En la misma pestaña “Modem Configuration” damos clic en la opción “write” para guardar estos cambios en el módulo router.
16. Una vez hecho todo esto desconectamos el Arduino UNO de la pc para extraer el shield del módulo XBee.

- **Microcontrolador de la Unidad Remota**

El microcontrolador para esta unidad debe de cumplir con los requerimientos necesarios basado en las consideraciones de diseño. En la tabla 13 se muestran los microcontroladores candidatos para esta unidad.

Tabla 13.

Características de controladores candidatos para la Unidad Remota.

Característica	PIC16C54-XT/P	Atmega 328	PIC16F84A-20/P
Alimentación	2.5 - 5.5Vcc	1.8 - 5Vcc	4 – 5.5Vcc
Costo	\$7	\$2.46	\$6.25
Frecuencia	40 MHz	20 MHz	20MHz
Puertos E/S	12	14	13
Ancho de bus de datos	8 bits	8 bits	8 bits
Tipo de interfaz	I2C, SPI	I2C, SPI,, USART	--
Temperatura de trabajo	+ 85° C	+ 85° C	+ 80° C

El microcontrolador escogido es el atmega 328, escogido por su bajo costo, tiene suficientes puertos I/O digitales, además de su comunicación con módulos inalámbricos. El microcontrolador atmega 328 viene incorporada en una placa ARDUINO llamada ARDUINO UNO (ver figura 50). El Arduino UNO es una tarjeta de hardware libre basada en una placa que tiene como elemento principal el microcontrolador Atmega328 y un entorno de desarrollo.

Siendo de la misma familia Arduino, el microcontrolador atmega 328 en la placa Arduino se programa de modo similar usando el mismo lenguaje de programación para el Arduino Mega 2560 de la Unidad Central (ver figura 5).

Para la transferencia del archivo .hex (programa de control en código decimal) de la pc a la placa Arduino solamente se utiliza un cable USB, además es una de las maneras de alimentar la placa.

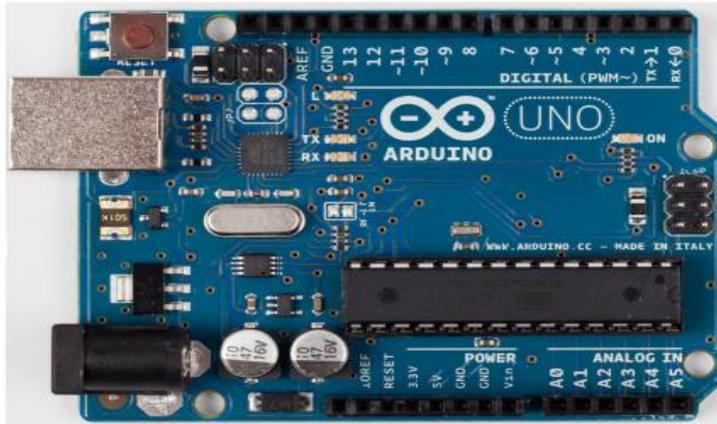


Figura 50.Arduino UNO

2.1.2.4.1. Hardware propuesto para la unidad remota de medición

Se propone el siguiente diagrama circuital (ver figura 51) para la Unidad Remota de medición.

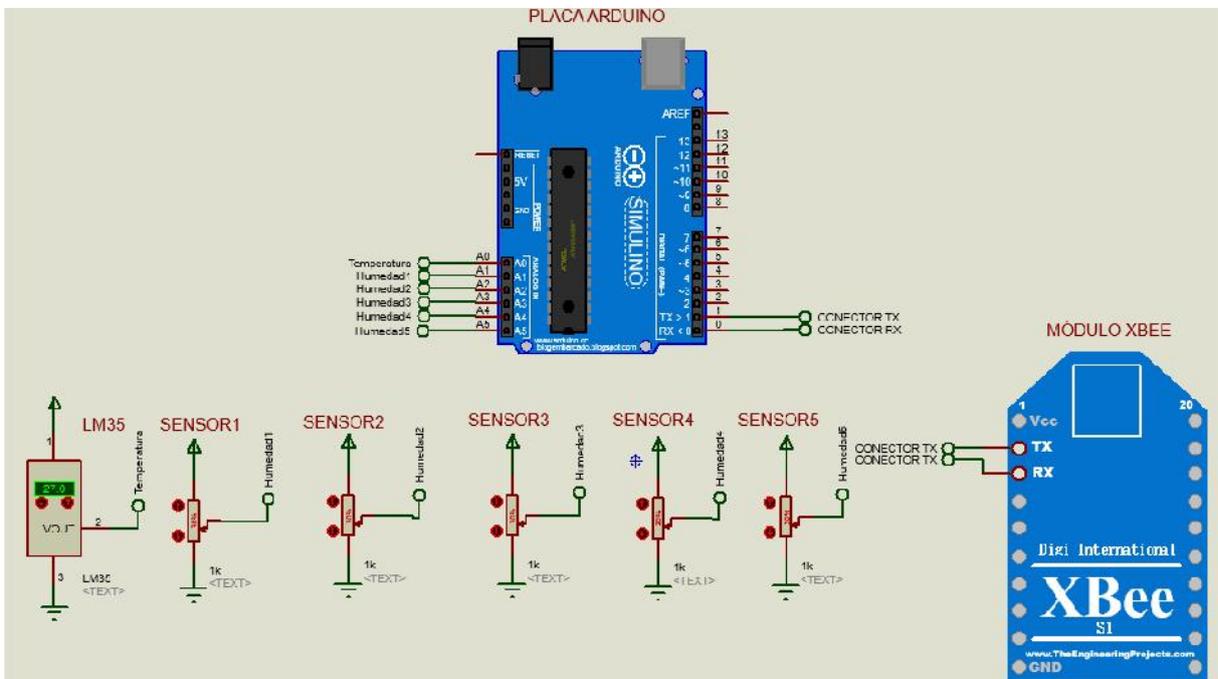


Figura 51.Circuito propuesto para la unidad remota de medición, elaborado en software Proteus

A continuación se menciona las características de la figura 50

- ❖ Los sensores de humedad y temperatura son alimentados por la placa Arduino. El módulo de radiofrecuencia es alimentado por la placa Arduino UNO que proporciona 3.3V.
- ❖ La tarjeta Arduino es alimentada por el controlador que proporciona 12VDC.
- ❖ Los módulos de radiofrecuencia configurados en el programa X-CTU están conectados a las tarjetas Arduinos. En la tarjeta Unidad Central el módulo RF llamado Coordinador. En la tarjeta de la Unidad Remota el módulo RF llamado Router (ver tabla 4).

2.1.2.4.2. Algoritmo propuesto para la unidad remota

Se propone el siguiente diagrama de flujo (ver figura 52) para la unidad remota.

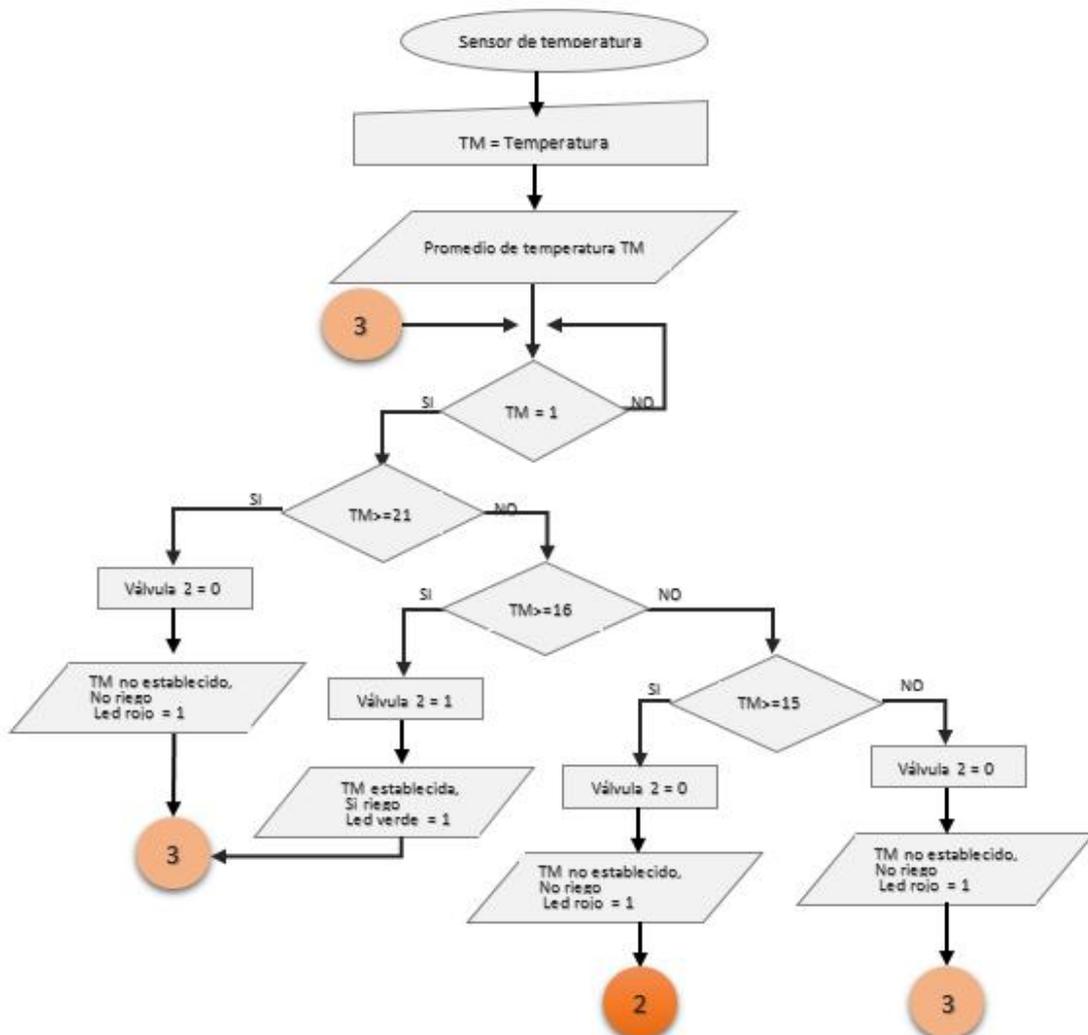


Figura 52. Diagrama de flujo de temperatura propuesto para la unidad remota de medición

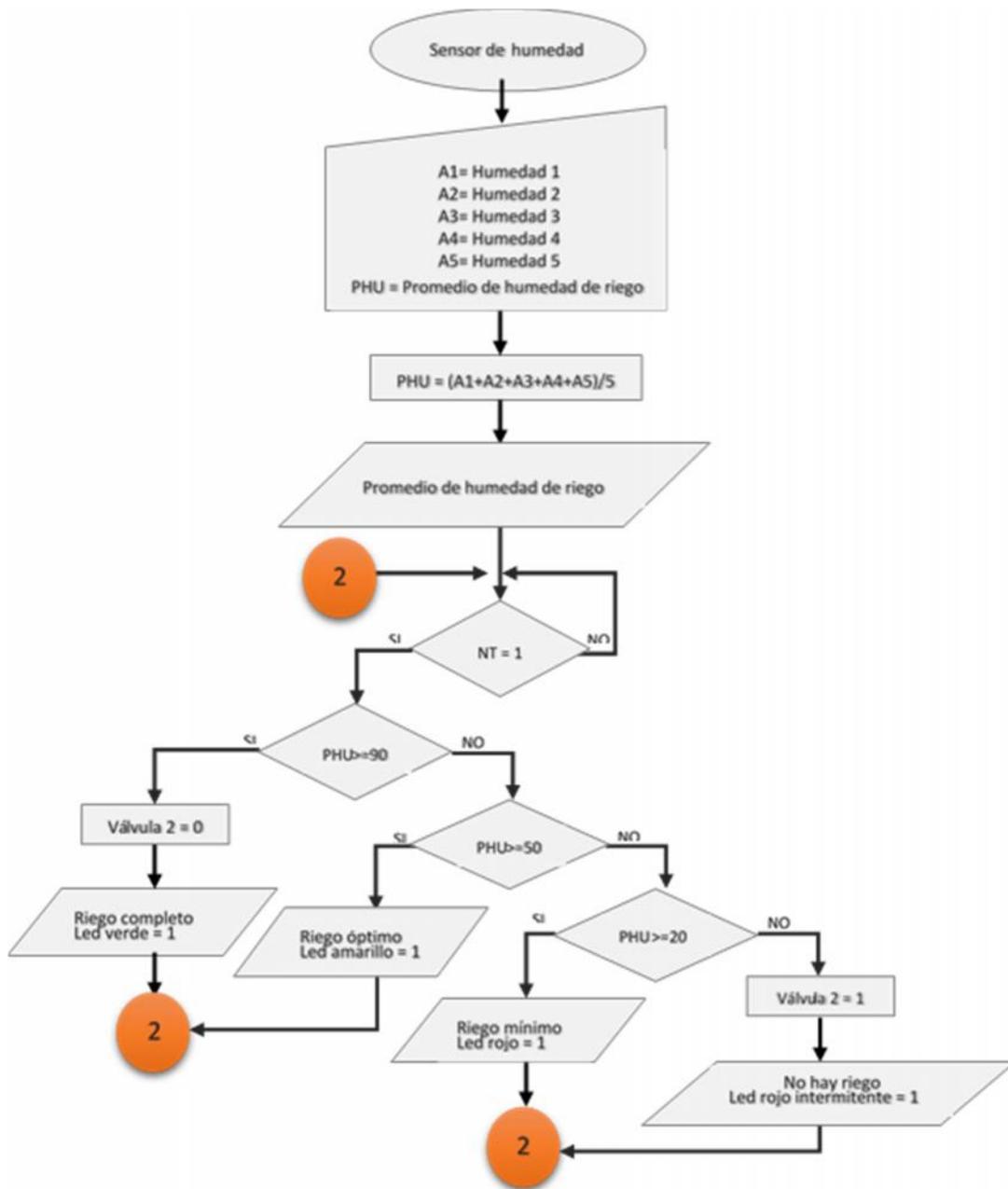


Figura 53. Diagrama de flujo de humedad propuesto para la unidad remota.

2.1.2.4.3. Alimentación Unidad remota

La alimentación general para la Unidad Remota está constituida por un panel solar, controlador de carga y una batería diseñadas en función de demanda de consumo de energía para la unidad central. Se propone el diagrama de bloque de la figura 54 para dicha unidad.

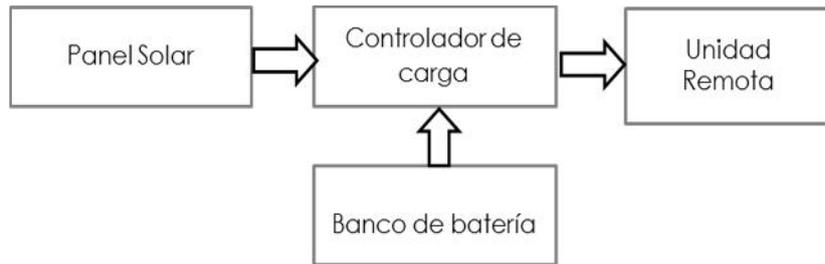


Figura 54. Diagrama de bloque de alimentación Unidad Remota

- **Diseño sistema fotovoltaico para la unidad remota de medición**

A continuación se presenta el desarrollo de cálculo para el diseño sistema fotovoltaico de la Unidad Remota. Para estos cálculos se procede utilizando las mismas formulas empleadas para la Unidad Central tomando en cuenta la localización del sitio (Longitud y Latitud), tabla de insolación del sitio (ver tabla 8), consumo de energía y disponibilidad de nuestro sistema.

La tabla 14 muestra el consumo de energía para la Unidad Remota. Debido que, el módulo de transmisión y los sensores de humedad y temperatura están conectados directamente a la placa Arduino estos componentes no se reflejan en la tabla 14. Por ende, la placa Arduino se encarga de suministrar los voltajes requeridos para estos dispositivos.

Con respecto a la tabla 14, el consumo de energía para la Unidad Remota es de 8.78 Kwh/mes y 105.12 kwh/año. Estos datos son de gran importancia para el cálculo de nuestro panel y batería.

Tabla 14.
Consumo de energía Unidad Remota

Cantidad	Componente	Corriente A	Voltaje V	Potencia unitaria (W)	Potencia General (W)	Horas de Trabajo	Energía kwh/día	Energía Kwh/Mes	Energía Kwh/Año
1	ARDUINO UNO	1	12	12	12	24	0.29	8.78	105.12
							0.29	8.78	105.12

- **Panel solar Unidad Remota**

Para el desarrollo de cálculo del panel para esta unidad se aplicó el mismo procedimiento anterior de la Unidad Central seleccionando uno o un grupo de paneles para luego conocer si la potencia generada por el panel cubre las necesidades de dicha unidad. Por lo tanto, el panel seleccionado para esta unidad es de 12VDC 70W. Para mayor información del módulo referirse al anexo C.

Un factor de pérdida de 0.85 es usado.

Características eléctricas del panel seleccionado.

- | | |
|--|--|
| -Potencia máxima a STC(Pmax): 70 W | -Operación nominal temperatura de celda (NOCT): 47C±2C |
| -Corriente de corto circuito (Isc): 3.31 A | -Coeficiente de temperatura de Pmax: -0.48%C |
| -Voltaje de circuito abierto (Voc): 21.5 V | -Coeficiente de temperatura de Voc: -0.34%C |
| | -Coeficiente de temperatura de Isc: 0.037%C |

Cálculos de número de paneles Unidad Remota

Para conocer el valor de P_{ac} usamos (3). El valor obtenido de P_{ac} es:

$$P_{ac} = 51.61 W$$

Para conocer el valor de $P_{dc,STC}$ usamos (4). El valor obtenido de $P_{dc,STC}$ es:

$$P_{dc,STC} = 60.71 W$$

El número de paneles para ésta unidad usamos (5). El número de paneles es:

$$\text{Número de módulo} = 0.86 \approx 1$$

Para la potencia generada por el módulo usamos (6) dando como resultado:

$$P_{dc} = 70 W$$

Para el cálculo de la temperatura de la celda se usó (5) la cual fue aplicada para la unidad central. También recurrimos a las características del panel

para esta unidad para tener en cuenta ciertos parámetros bajo condiciones de prueba estándar (STC), además se tomó en cuenta la temperatura del sitio (ver figura 43). Los resultados se muestran en la tabla 15.

NOCT=47

Pmax=70W

La energía generada por el sistema fotovoltaico para esta unidad se conoce usando (6). La tabla numero 15 muestra la energía generada mensualmente por el sistema fotovoltaico, también por un año teniendo en cuenta la temperatura, horas de sol al día en el sitio y factor de pérdida. La producción anual de energía por el sistema fotovoltaico (ver tabla 15) es de 105.01 Kwh/año lo cual es suficiente para suplir con la demanda de consumo de energía para la unidad Remota (ver tabla 14). En síntesis, según los cálculos realizados se propone el uso de un panel de 12 VDC 70 W para la unidad Remota.

Tabla 15.
Energía generada por el sistema fotovoltaico Unidad Remota.

Potencia (W)	Factor pérdida	Incidencia de radiación solar-Tilt 13 kWh/m ² /d	Mes	Días del mes	Energía Kwh/mes	Temperatura °C	T de la celda	Energía Kwh/Mes
70	0.85	6.06	Enero	31	11.18	26.1	57.35	9.73
70	0.85	6.7	Febrero	28	11.16	25.8	57.05	9.73
70	0.85	7.26	Marzo	31	13.39	26.6	57.85	11.63
70	0.85	6.87	Abril	30	12.26	28.3	58.03	10.64
70	0.85	5.9	Mayo	31	10.88	29.1	60.35	9.34
70	0.85	4.57	Junio	30	8.16	29.1	59.25	7.04
70	0.85	4.53	Julio	31	8.36	28.5	57.25	7.28
70	0.85	4.44	Agosto	31	8.19	28.4	58.35	7.10
70	0.85	5.01	Septiembre	30	8.94	28.1	58.35	7.75
70	0.85	5.25	Octubre	31	9.68	27.9	59.15	7.36
70	0.85	4.97	Noviembre	30	8.87	28.1	58.12	7.70
70	0.85	5.43	Diciembre	31	10.02	27.4	57.65	8.71
					121.09			105.01

- **Controlador de carga**

El controlador de carga utilizado para esta unidad es el mismo de la unidad central (ver figura 23). Referirse a la sección 1.2.3.3

- **Cálculo de capacidad de la batería Unidad Remota**

Se escogió una de batería de 12V para realizar el cálculo de la capacidad de la batería de la Unidad Remota. Se debe tener en cuenta el consumo de energía por día (ver tabla 14). A continuación se procede el desarrollo utilizando las formulas empleadas por la Unidad Central.

El valor de la carga se calcula empleando (9) dando como resultado:

$$Carga_{dc} = 24.16 \frac{Ah}{día} a 12 V$$

La unidad remota al igual que la unidad central están ubicadas en la misma comunidad comparten los valores de la tabla 7 (Insolación solar). Por lo que, los días máximo de carga para esta unidad son tres días. Empleamos (10) obteniendo como resultado:

$$Almacenamiento = 72.48 Ah$$

Se toma en cuenta factor de máximo descarga (F.M.D = 80%) y factor de tasa de descarga y temperatura (F.D.T = 97%) bajo condiciones estándar de pruebas (C/20 a 25°C) con el objetivo de asegurar la capacidad de la batería a los tres días. Se calcula usando (9) obteniendo como resultado:

$$Nominal\left(\frac{C}{20}, 25^{\circ}C\right) \text{ Capacidad de la batería} = 93.40 Ah a 12V$$

2.1.2.5. Unidad sistema de bombeo

La unidad sistema de bombeo tiene la función del cierre-abierto de la electroválvula para el llenado del primer tanque y de la activación de una bomba DC sumergible ubicada en un pozo a las cercanías del primer tanque. La unidad central por medio de un programa elaborado en Arduino controla esta unidad y están comunicada de manera alámbrica.

Se propone el siguiente diagrama de bloque general de la unidad sistema de bombeo (ver figura 55). En esta unidad está integrado un módulo relé, una electroválvula de media pulgada y su respectiva alimentación.

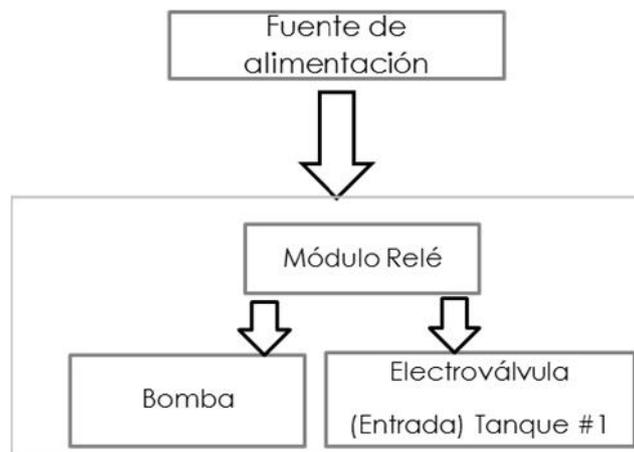


Figura 55. Diagrama de bloque de la unidad de bombeo

Selección de componentes

- **Módulo relé**

El módulo seleccionado para esta unidad es el mismo módulo relé escogido para la unidad sistema de llenado. Referirse a la sección 2.4.3

- **Electroválvula**

La electroválvula seleccionada para esta unidad es la misma electroválvula D9152 escogida para la unidad sistema de llenado. Referirse a la sección 2.4.2

2.1.2.5.1. Hardware propuesto para unidad de bombeo

Se propone el siguiente diagrama circuital (ver figura 56) para la Unidad de bombeo.

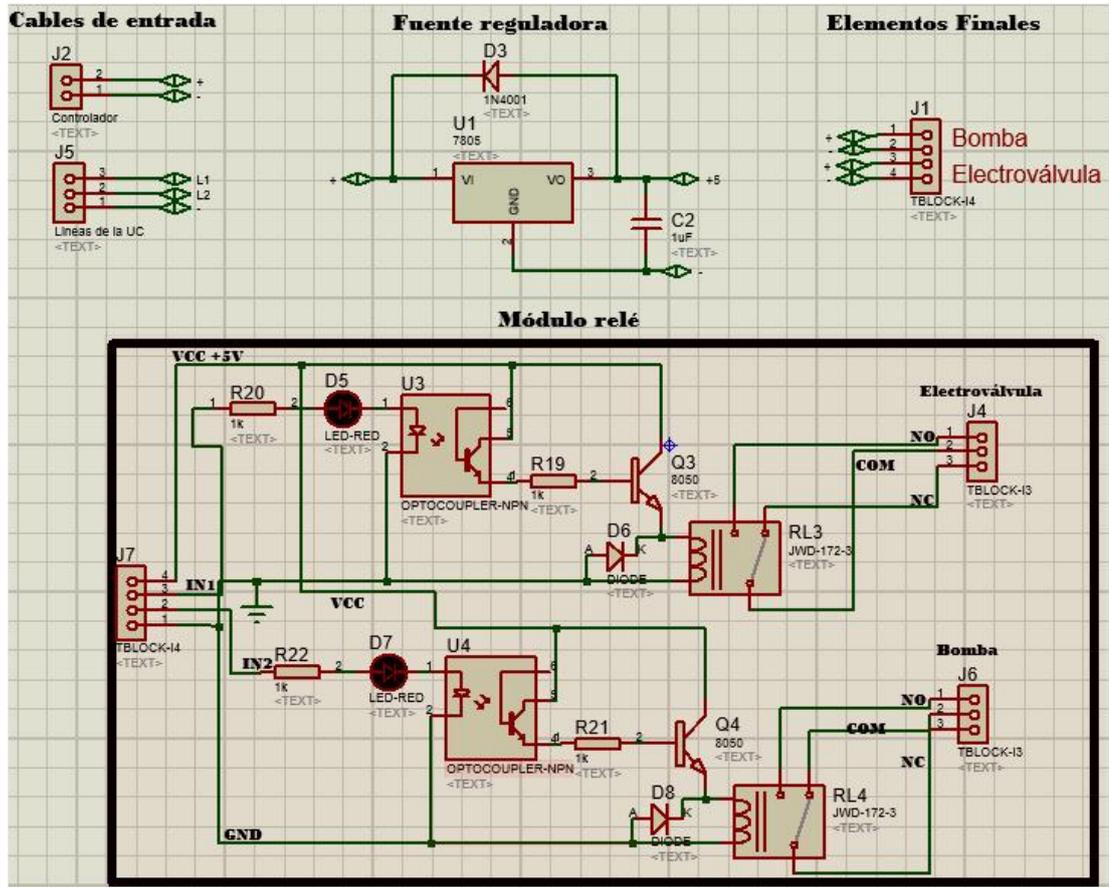


Figura 56. Circuito propuesto para la unidad de bombeo, elaborado en software Proteus

2.1.2.5.2. Alimentación Unidad de bombeo

Para la alimentación de la unidad de bombeo se cuenta con un sistema fotovoltaico ya instalado en el sitio. Este sistema tiene como objetivo alimentar una bomba DC sumergible para el llenado de los tanques (2500 y 1100 litros) en la comunidad.

Este diseño utiliza cuatro módulos (ver figura 57) con una potencia de 120 Watts, además cuenta con un controlador (ver figura 58) especial para la bomba entre otras aplicaciones. Sin embargo, este sistema no cuenta con un banco de baterías. Por lo que, el sistema funciona solo cuando hay luz solar.

- ✓ Voltaje: 24 VDC
- ✓ Corriente: 6A
- ✓ Potencia: 120W

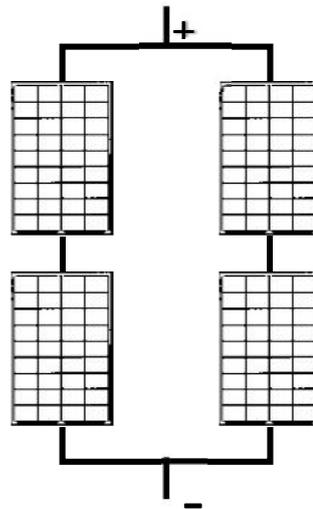


Figura 57. Conexión sistema fotovoltaico

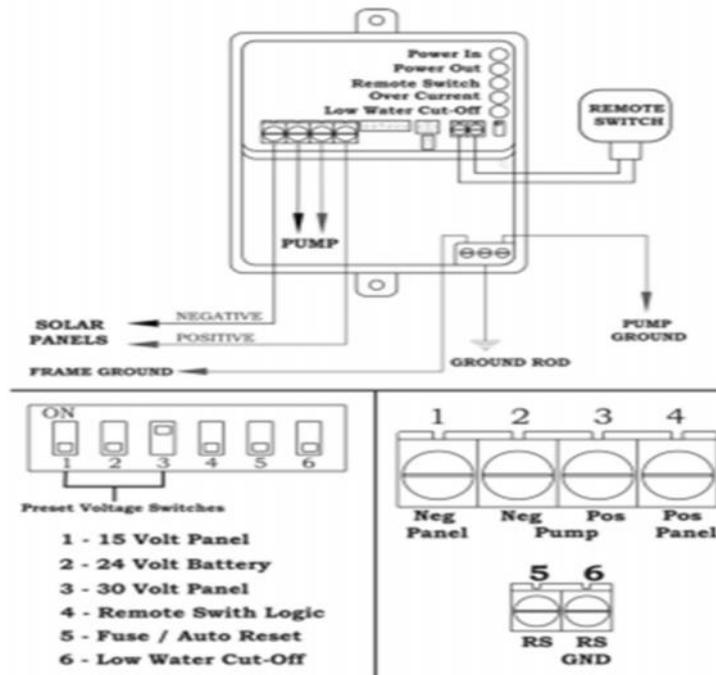


Figura 58. Diagrama de cableado de controlador PCA-30-M1 SERIES

Tabla 16.
Consumo de energía unidad de bombeo.

Cantidad	Componente	Corriente A	Voltaje V	Potencia unitaria (W)	Potencia General (W)	Horas de trabajo	Energía kwh/día	Energía Kwh/Mes	Energía Kwh/Año
1	Módulo Relé	0.04	5	0.2	0.2	12	0.0024	0.07	0.88
1	Electroválvula 1/2 pulgada	0.25	12	3	3	7	0.02	0.64	7.67
1	Bomba DC	4.2	24	100.2	100.2	4	0.4	12.30	147.17
							0.43	13.01	155.71

Para realizar el cálculo de la batería para la unidad de bombeo se procede utilizando las mismas fórmulas utilizadas para la unidad central y los datos de la tabla 16.

Cálculo de la capacidad de la batería Unidad de bombeo

El valor de la carga se calcula empleando (9) dando como resultado:

$$Carga_{dc} = 35.83 \frac{Ah}{día} a 12V$$

Los días máximos de almacenamiento para esta unidad son tres días. Empleamos (10) obteniendo como resultado:

$$Almacenamiento = 107.5 Ah$$

Se toma en cuenta factor de máximo descarga (F.M.D = 80%) y factor de tasa de descarga y temperatura (F.D.T = 97%) para esta unidad bajo condiciones estándar de pruebas (C/20 a 25°C) con el objetivo de asegurar la capacidad de la batería a los tres días. Se calcula usando (11) obteniendo como resultado:

$$Nominal \left(\frac{C}{20}, 25^{\circ}C \right) \text{ Capacidad de la batería} = 138.52 Ah a 12v$$

2.1.2.6. Implementación del prototipo

El sistema automatizado de riego se implementó en la comunidad de Totogalpa usando los equipos descritos en las secciones anteriores. Para la implementación de este proyecto se llevó a cabo varios procesos: Diseño de tarjetas electrónicas, diseño de estructuras para el soporte de los paneles solares y cajas para las diferentes unidades, programación de las placas Arduino, instalación de los equipos en el sitio y pruebas.

En la figura 59, 60, 61 y 62 se demuestran el resultado de este proceso para las unidades.

Para la unidad remota se demuestra el resultado en la figura 59.



Figura 59. Implementación Unidad Remota

Para la unidad de central se demuestra el resultado en la figura 60.



Figura 60. Implementación Unidad Central

Para la unidad de bombeo se demuestra el resultado en la figura 61.



Figura 61. Sistema de bombeo.

Para las instalaciones de las torres de los paneles, unidades, tarjetas, equipos entre otros se demuestra el resultado en la figura 62.

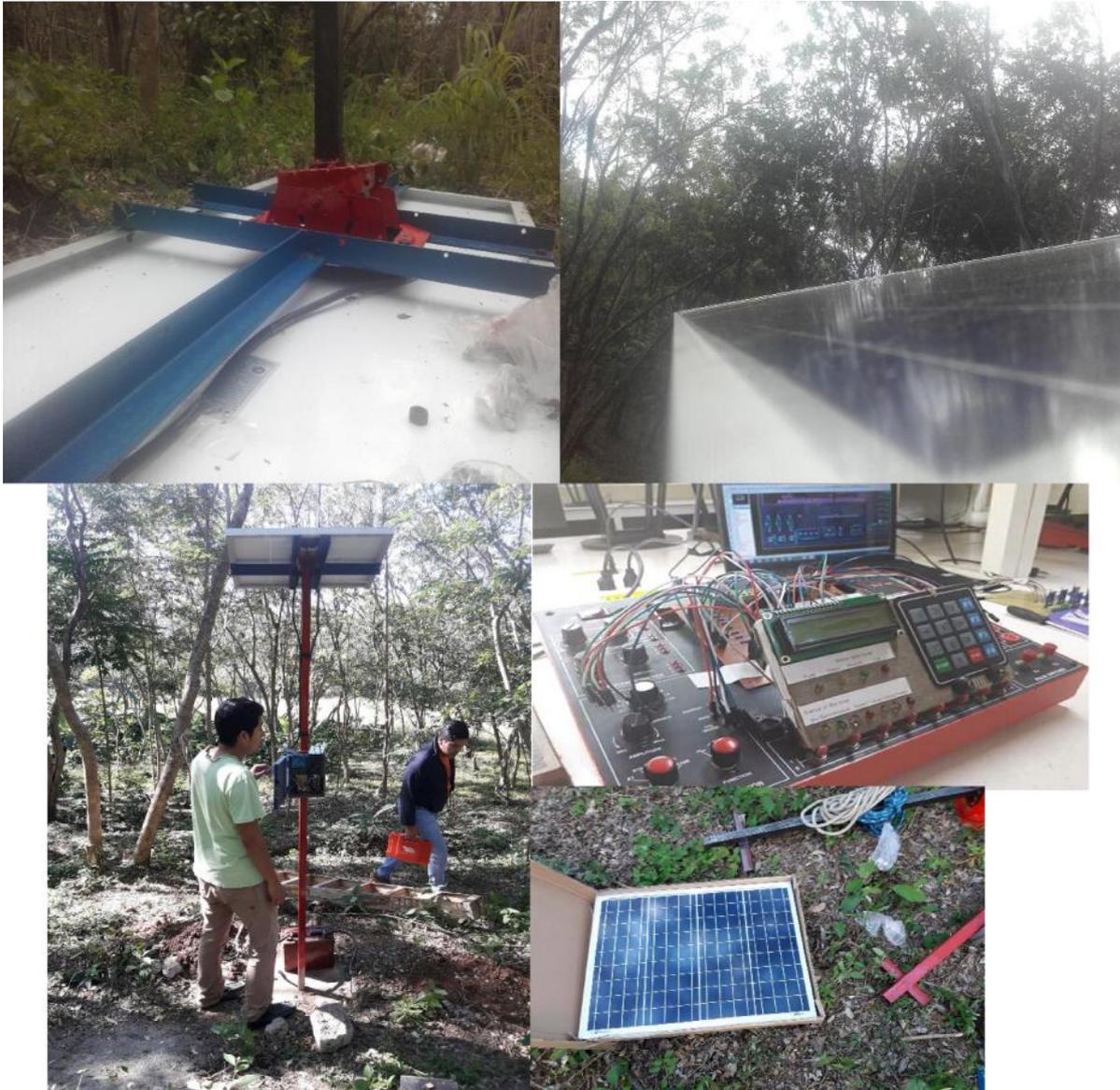


Figura 62. Instalación de los equipos en la comunidad

2.1.2.7. Pruebas y corrección de errores

Durante esta fase se realizaron pruebas de operación entre las diferentes unidades que conforman el prototipo en el sitio. Con el propósito de probar si el prototipo trabajaba de forma correcta o identificar oportunidades de mejoras.

A continuación se presentan los problemas identificados y como se corrigieron.

El módulo coordinador no recibe datos del módulo router.

Una vez establecidas las unidades en el sitio se empezaron a hacer pruebas de comunicación entre la unidad remota y unidad central a una distancia de 180 metros, pero la unidad central no recibía datos correctos del plantío. Este problema se logró resolver después de darnos cuenta que entre los módulos de radiofrecuencia no estaban a línea de vista.

El módulo relé de la unidad de bombeo no activa la bomba y electroválvula

Las pruebas realizadas entre la unidad de bombeo y unidad central no fueron satisfactoria al principio. Las pruebas consistían en mandar a activar una bomba dc sumergible para el llenado de los tanques más la activación de una electroválvula. El problema era que ninguno de estos dos elementos respondía cuando se mandaban a activar. Esto se logró resolver compartiendo conexión (GND) entre las unidades.

Sensor ultrasónico manda datos errados

Las pruebas realizadas con el sensor ultrasónico consistían en que este sensor enviara datos del estado del tanque número 2 a la unidad central. Los resultados no resultaron satisfactorio. Debido que, presentaba en pantalla datos erróneos. Este problema se logró resolver ajustando la distancia del sensor con respecto al objeto en la programación.

2.1.2.7.1. Resultados

El resultado de este proyecto fue un sistema de riego automatizado para el cultivo de café en la comunidad de Totogalpa respondiendo a la necesidad planteada. A continuación se listan las características del prototipo elaborado:

- Como elemento principal se utiliza la placa Arduino 2560 que determina los tiempos de riego en función de las variables de entrada: temperatura y humedad.
- El prototipo hace uso de indicadores de LED para cada evento a ocurrir en caso de fallar la pantalla.
- El prototipo hace uso de sensores calibrados para la medición de variables (humedad, temperatura y nivel de agua).
- El prototipo cuenta con interruptores encendido y apagado para las diferentes unidades que conforman el sistema de riego.
- El sistema envía información del cultivo cada 3 segundos.
- El prototipo cuenta con el uso de paneles solares, controlador y baterías para las diferentes unidades convirtiéndolo en un sistema autónomo.
- El prototipo tiene la posibilidad de conectar más módulos de radio frecuencia Xbee S2 creando una red malla asumiendo que haiga una expansión del cultivo de café en el sitio o se desea controlar otros equipos con la misma finalidad.
- El prototipo dispone de baterías de descargas profundas para las diferentes unidades permitiendo almacenamiento de energía por un máximo de tres días sin luz solar.
- El prototipo cuenta con una estructura de hierro para la sostención de los paneles solares y protección para cada unidad

2.1.2.7.2. Costo del prototipo

A continuación se presentan los costos de los equipos de las diferentes unidades utilizados para la realización de este proyecto. Vale decir, que en la tabla 15 no incluyen otros factores (herramientas de diseño, instalación, bibliografía entre otros) que verdaderamente influyen en el costo del prototipo.

Tabla 17.
Costo del prototipo.

Descripción	Cantidad	Costo Unitario (\$)	Costo total(\$)
Arduino Kit - Uno R3 Atmega328p	1	17	17
Arduino Kit - Mega 2560 R3	1	25	25
Sensor Ultrasónico HSCR04	2	9	18
Módulo Xbee pro 50 mW wire antenna - series 2 (Zigbee Mesh_	2	70	140
Electraválvula 1 pulgada	2	34	68
Electroválvula 1/2 pulgada	2	6	12
LCD 16x2 screen	1	7	7
Módulo Relé SRD-5VDC-DL-C	2	2.31	4.62
LED Diode (Blue, Red and Green)	15	0.5	7.5
Resistencia 200 Ω	15	0.15	2.25
Alambres conectores hembra y macho 7,8"	2	4.99	9.98
SainSmart Xbee Shield Module for Arduino UNO Mega Nano DUE Duemilanove	2	12	24
Teclado matricial	1	28.27	28.27
Tarjeta virgen 20x15 cm	2	5.06	10.12
Tarjeta virgen 10x15 cm	2	1.66	3.32
Terminal bloque 2 salida	17	0.23	3.91
Terminal bloque 3 salida	2	0.23	0.46
Regulador LM7805	1	0.66	0.66
Regulador LM1875	1	1.56	1.56
Capacitor 1uF	2	0.5	1
Pasta térmica	1	0.33	0.33
sensor de temperatura	3	2.33	6.99
sensor de YL-69	5	2.272	11.36
Conector de alimentación	2	0.33	0.66
Caja unidad remota	1	3.33	3.33
Caja unidad central	1	3.33	3.33

Interruptor	1	0.33	0.33
70 pin IC socket	1	2.33	2.33
Panel solar 160 watt	1	114	114
Panel solar 70 watt	1	62	62
Batería trojan 105 Ah	2	100	100
Batería trojan 100 Ah	1	100	100
controlador 5 A	2	15	30
Gabinete batería	3	20	60
Estructura Panel Solar	2	10	20
		Total	999.31

CAPITULO 3: CONCLUSIONES Y RECOMENDACIONES

3.1. Conclusiones

En base a los objetivos propuestos que se presentaron al comienzo de este proyecto, además de los resultados obtenidos se llegaron a las siguientes conclusiones:

- El sistema de riego automatizado responde a los requerimientos planteados, de forma que integra características y funcionalidades descritas en los objetivos propuesto.
- El prototipo responde a la necesidad planteada de automatizar el riego del cultivo en la comunidad de Totogalpa.
- El mecanismo de monitoreo para el cultivo de café implica bajo costos de hardware. Es decir, en la placa Arduino se pueden agregar más sensores, sustituir o eliminar, además en la programación en Arduino no implicaría mucho trabajo.
- La determinación de los tiempos de riego fue desarrollado con éxito gracias a la tecnología Arduino que permitió también el control de llenado de los tanques de agua.
- La integración de paneles solares para el suministro de energía para las unidades contribuyó de manera positiva con el medio ambiente, además de generar energía limpia a los equipos que conforman el prototipo.
- El desarrollo de este trabajo monográfico permitió a los autores del proyecto la obtención de nuevas competencias en el diseño e implementación de: sistema automatizado de riego, sistemas fotovoltaicos autónomos, además de adquirir habilidades en la creación de tarjetas de circuito impresos, transmisión y recepción de datos por medio de radiofrecuencia y programación de tarjetas Arduino.

- El desarrollo de este proyecto puede servir de ejemplo para otros estudiantes de ingeniería que deseen investigar o hacer uso acerca de diseños, basados en la tecnología Arduino.

3.2. Recomendaciones

En base al desarrollo de proyecto se deben hacer las siguientes recomendaciones:

- Es muy importante realizar el estudio minucioso del uso consecutivo del plantío, debido a que se definen la cantidad de agua necesaria que se le debe suministrar al cultivo.
- Intensificar y dar información necesaria a los usuarios del uso del sistema de riego por goteo, mediante talleres u otras actividades demostrando las ventajas y beneficios que pueden obtener.
- Realizar un estudio de la tecnología Zigbee para la ampliación de una nueva red.
- Abocarse con un ingeniero agrónomo para la aplicación de abonos al plantío de café y conocer si se debe de ajustar ciertos parámetros del sistema para el riego de las plantas y mejorar la calidad de estas.
- Incorporación de nuevos módulos hardware para el monitoreo de nuevas plantas.
- Integración de nuevos sensores para monitoreo del plantío de café a mayor escala.
- Diseño de un sistema para el registro de datos del plantío y almacenarlos para futuros estudios o investigaciones.
- En énfasis el éxito del proyecto del sistema de riego automatizado por goteo se tiene que cumplir con los siguientes parámetros:
 - Establecer un método de programación.
 - Realizar pruebas de uniformidad de riego.
 - Realizar mantenimiento exhaustivo de todo el sistema.
 - Cuantificar el consumo de energía del sistema.

BIBLIOGRAFÍA

Allen, R. G. (2006). Riego y Drenaje. Guías prácticas para la determinación de los requerimientos de agua de los cultivos. Roma.

Duran, F. (2008). Guía de cultivo de café. Colombia: Grupo latino

Díaz Valenzuela, M.G, Flores Quezada, E.J., y Montalbán Castro, Z.R. (2015). Efectos de los abonos orgánicos a base de la pulpa de café, compost, gallinaza en plántulas de café (*coffea arabica*) en la finca “El bosque” comunidad buena vista, Municipio de San Juan Rio Coco, departamento de Madriz, octubre 2012-julio 2013. Recuperado de: <http://riul.unanleon.edu.ni:8080/jspui/bitstream/123456789/3856/1/227737.pdf>

Federes. (2000). Avances recientes en la programación de los riegos. Ingeniería del Agua.

Gilbert, M. (2004). Renewable and efficient electric power systems. Hoboken, New Jersey. John Wiley and Sons.

Hillel, D. (1990). Role of Irrigation in agricultural systems. Wisconsin.

Herrero Herranz, J.C. y Sánchez Allende, J. (2015). Una mirada al mundo arduino. *Tecnología@ y desarrollo*, XIII. Recuperado de: www.uax.es/publicacion/una-mirada-al-mundo-arduino.pdf

Enciso, J.M., Porter, D., y Périès, X. (10 de noviembre de 2016). Uso de sensores de humedad del suelo para eficientizar el riego. Recuperado de: <http://www.euskobaratza.com/wp-content/uploads/2016/02/Enciso-et-al-2008.-Uso-de-sensores-de-humedad-para-optimizar-el-riego.pdf>

Iter. (2008). Instalación Solar Fotovoltaica.

Manzini, F. (2004). Nuevas energías renovables: una alternativa energética sustentable para México (análisis y propuesta). México.

Marín, R. (2003). Manual de café orgánico. Estelí, Nicaragua.

My robot self. . (14 de noviembre de 2016). Sensor de humedad de suelo YL-69. [Mensaje en un blog]. Recuperado de: <https://myrobotself.com/2016/07/23/sensor-de-humedad-del-suelo-modulos-yl-69-y-yl-38/>

Micropik. (15 de noviembre de 2016). Ultrasonic Ranging Module HC - SR04. Recuperado de: <http://www.micropik.com/PDF/HCSR04.pdf>

NETAFIM. (23 de agosto de 2016). Acerca de NETAFIM. Recuperado de: <http://www.pirapey.com.py/archivos/products/acerca-de-netafim>

Valleybolivia. (23 de agosto de 2016). Valleybolivia historia Netafim. Recuperado de: http://www.valleybolivia.com/?page_id=1010

Pirapey. (23 de agosto de 2016). Sistema de riego. Recuperado de: <http://www.pirapey.com.py/wp-content/uploads/2014/08/triptico-netafim-sistema-riegos.pdf>

Peláez, M. (2012). La semilla del café y el proceso germinativo. Recuperado de <http://politecnicovirtual.edu.com/prod-agricola/asp-fisio-cafe.htm>

Texas Instruments. (21 de noviembre de 2016). LM35 DATA. Recuperado de: <http://pdf.datasheetcatalog.net/datasheet/texasinstruments2/lm35.pdf>

NASA. (10 de julio de 2016). Surface meteorology and Solar Energy. Recuperado de: <https://eosweb.larc.nasa.gov/cgi-bin/sse/grid.cgi?email=skip@larc.nasa.gov>

ANEXOS

ANEXO A: CÓDIGO FUENTE

A.1. Código para la unidad central y módulo coordinador en arduino.

```
#include <Ultrasonic.h>
#include <LiquidCrystal.h> /*Librería del LCD*/
#include <Keypad.h> //llama a la librería Keypad.h

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); /*Entradas/salidas del LCD*/

const byte ROWS = 4; //Numero de filas del teclado que se esta usando
const byte COLS = 3; //Numero de columnas del teclado que se esta usando

char Keys[ROWS][COLS] = //Aquí pondremos la disposición de los caracteres tal cual están
en nuestro teclado
{
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};

//byte rowPins[ROWS] = {22, 23, 24, 25}; //Seleccionamos los pines en el arduino donde iran
conectadas las columnas
//byte colPins[COLS] = {26, 27, 28}; //Seleccionamos los pines en el arduino donde iran
conectadas las filas

//byte rowPins[ROWS] = {34, 35, 36, 37}; //Seleccionamos los pines en el arduino donde iran
conectadas las columnas
//byte colPins[COLS] = {38, 39, 40}; //Seleccionamos los pines en el arduino donde iran
conectadas las filas

/*MAYNOR*/
byte rowPins[ROWS] = {47, 49, 51, 53}; //Seleccionamos los pines en el arduino donde iran
conectadas las columnas
byte colPins[COLS] = {39, 41, 43}; //Seleccionamos los pines en el arduino donde iran
conectadas las filas

Keypad teclado = Keypad(makeKeymap(Keys), rowPins, colPins, ROWS, COLS); //inicializa
el teclado

int eco = 9;
int trigger = 8;
float distancia,tiempo,nivel;
int capacidad;
int variable,tanque;

///DEFINIMOS LA ENTRAD DE NT */
int led1ybuzzer = 10; /*LED ROJO DE NT FLOOD WARNING AND EMPTY*/
int led2 = 6; /*LED VERDE DE NT LLENO*/
int led13 = 13; /*LED AMARILLO DE NT MIMINO*/
int led22 = 23; /*LED VERDE DE BOMBA */
int led23 = 25; /*LED AZUL DE VALVE 1 */
int led24= 27; /*LED AZUL DE VALVE 2*/
int led25= 29; /*LED AMARILLO ESTADO MINIMO DEL CULTIVO*/
int led27= 31; /*LED VERDE ESTADO COMPLETO DEL CULTIVO*/
int led28= 33; /*LED ROJO ESTADO SECO DEL CULTIVO*/
int led26= 35; /*LED VERDE INDICANDO ABIERTO DE LA VALVULA*/
float pinA0,pinA1,pinA2,pinA3,pinA4,pinA5;
int PHU;
```

```

//int led24 =24; /*LED AZUL DE VALVE 2*/
int led29= 29; /*LED ROJO TEMPERATURA 0-15 Y 23-30 NO REGAR*/
int led30= 37; /*LED VERDE TEMPERATURA EN LA CUAL SE PUEDE REGAR 16-22*/
float TM6,TM7,TM8,TM9,TM10,PTM;

void setup ( )
{
  Serial.begin (9600);
  lcd.begin(16,2);
  lcd.setCursor(1,0);
  lcd.print(" TOTOGALPA");
  pinMode(trigger,OUTPUT);
  pinMode(eco,INPUT);

  /*DEFINIMOS LA SALIDAD DE NT*/
  pinMode(led2, OUTPUT); /*LED VERDE DE NT LLENO*/
  pinMode(led1ybuzzer, OUTPUT); /*LED ROJO DE NT FLOOD WARNING AND EMPTY*/
  pinMode(led13, OUTPUT); /*LED AMARILLO DE NT MIMINO*/
  pinMode(led22, OUTPUT); /*LED VERDE DE BOMBA */
  pinMode(led23, OUTPUT); /*LED AZUL DE VALVE 1 */
  pinMode(led26, OUTPUT); /* LED VALVULA DE TANQUE QUE ESTA CERCA AL APZO*/

  /*DEFINIMOS LA SALIDA DE HUMEDAD*/
  pinMode(led24, OUTPUT); /*LED AZUL DE VALVE 2 */
  pinMode(led25, OUTPUT); /*LED AMARILLO ESTADO MINIMO DEL CULTIVO*/
  pinMode(led27, OUTPUT); /*LED VERDE ESTADO COMPLETO DEL CULTIVO*/
  pinMode(led28, OUTPUT); /*LED ROJO ESTADO SECO DEL CULTIVO*/
  /*DEFINIMOS LA SALIDA DE TEMP*/
  pinMode(led29, OUTPUT); /*LED ROJO TEMPERATURA 0-15 Y 23-30 NO REGAR*/
  pinMode(led30, OUTPUT); /*LED VERDE TEMPERATURA EN LA CUAL SE PUEDE
  REGAR 16-22*/
}
/*Variables como salida o entrada del sensor de ultrasonidos*/
////////////////////////////////// INICIO //////////////////////////////////////

void dos ()
{
  // lcd .clear ();
  ////////////////////////////////////ciclo loop//
  //Serial.begin(9600);
  // for(int i=0; i<100; i++)
  // {
  // pinMode(i,OUTPUT);

  ////////////////////////////////////fin ciclo loop//

  /*Comandos para controlar el sensor de ultrasonidos*/
  digitalWrite(trigger, LOW); /*Emitimos ultrasonidos*/
  delayMicroseconds(2); /*tiempo de espera*/
  digitalWrite(trigger, HIGH); /*Paramos la emisión*/
  delayMicroseconds(10); /*tiempo de espera*/
  digitalWrite(trigger, LOW); /*Emitimos ultrasonidos*/
  delayMicroseconds(10); /*tiempo de espera*/
  tiempo= pulseIn(eco, HIGH); /*Lo que tarda en recibir el eco*/
  distancia= int(0.017*tiempo);
  nivel =-3 + distancia; /*Pues he situado el medido 3cm por encima del depósito*/
}

```

```

    capacidad = (100 - ((nivel)/1095) * 100);

/*Calculamos la distancia como ya se ha dicho en el apartado de material necesario
//"Sensor de ultrasonidos"*/

/* /200 es porque mi depósito mide 27cm de altura, así sacaremos el porcentaje
//*100. Después, se lo restamos al 100% puesto que al medir desde arriba estamos
//midiendo la distancia en la cual no hay agua */

/*Para las condiciones de NT he usado un if, else if, else*/

    if (capacidad <= 10) { /* INICIO VACIO*/

        Serial.begin (9600);
        lcd.begin(16,2);
        lcd.setCursor(0,0);
        lcd.print("TP=  C");
        lcd.setCursor(3,0);
        lcd.print(PTM);
        lcd.setCursor(0,1);
        lcd.print("H1= % NT= %");
        lcd.setCursor(3,1);
        lcd.print(PHU);
        lcd.setCursor(10,1);
        lcd.print(capacidad);
        //delay(10);

        /*INDICADORES DE NT CUANDO ESTA VACIO ACCIONAR EL LED ROJO*/
        digitalWrite(led2, LOW); /*Led verde2 nivel BAJO*/
        digitalWrite(led13, LOW); /*Led amarillo nivel BAJO*/
        digitalWrite(led1ybuzzer, HIGH); /*Led rojo + buzzer nivel ALTO*/
        digitalWrite(led22, HIGH);

        /*INDICADORES DE LA ELECTROVALVULA 1 Y BOMBA */
        digitalWrite(led26, LOW); /*VALVULA DEL POZO */
        digitalWrite(led23, HIGH); /*ENCENDER VALVE 1 INDICADOR DE COLOR AZUL*/
        delay(500);
        digitalWrite(led22, HIGH); /*ENCENDER bomba INDICADOR DE COLOR VERDE*/

    } else if (capacidad <= 50) { /*INICIO MINIMO*/

        Serial.begin (9600);
        lcd.begin(16,2);
        lcd.setCursor(0,0);
        lcd.print("TP=  C");
        lcd.setCursor(3,0);
        lcd.print(PTM);
        lcd.setCursor(0,1);
        lcd.print("H1= % NT= %");
        lcd.setCursor(3,1);
        lcd.print(PHU);
        lcd.setCursor(10,1);
        lcd.print(capacidad);
        delayMicroseconds(10); /*tiempo de espera*/
    }

```

```

/*INDICADORES DE NT CUANDO ESTA MINIMO ACCIONAR EL LED AMARILLO*/
digitalWrite(led2, LOW); /*Led verde2 nivel BAJO*/
digitalWrite(led13, HIGH); /*Led amarillo nivel ALTO*/
digitalWrite(led1ybuzzer, LOW); /*Led rojo + buzzer nivel ALTO*/

/*INDICADORES DE LA ELECTROVALVULA 1 Y BOMBA */
digitalWrite(led26, LOW); /*VALVULA DEL POZO */
digitalWrite(led23, HIGH); /*ENCENDER VALVE 1 INDICADOR DE COLOR AZUL*/
delay(500);
digitalWrite(led22, HIGH); /*ENCENDER bomba INDICADOR DE COLOR VERDE*/

} else if (capacidad <= 90) { /*INICIO OPTIMO*/

Serial.begin (9600);
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print("TP=  C");
lcd.setCursor(3,0);
lcd.print(PTM);
lcd.setCursor(0,1);
lcd.print("H1= % NT= %");
lcd.setCursor(3,1);
lcd.print(PHU);
lcd.setCursor(10,1);
lcd.print(capacidad);
delayMicroseconds(10); /*tiempo de espera*/

/*INDICADORES DE NT CUANDO ESTA OPTIMO ACCIONAR EL LED AZUL*/
digitalWrite(led2, HIGH); /*Led verde2 nivel BAJO*/
digitalWrite(led13, LOW); /*Led amarillo nivel BAJO*/
digitalWrite(led1ybuzzer, LOW); /*Led rojo + buzzer nivel ALTO*/

/*INDICADORES DE LA ELECTROVALVULA 1 Y BOMBA */
digitalWrite(led26, LOW); /*VALVULA DEL POZO */
digitalWrite(led23, HIGH); /*ENCENDER VALVE 1 INDICADOR DE COLOR AZUL*/
delay(500);
digitalWrite(led22, HIGH); /*ENCENDER bomba INDICADOR DE COLOR VERDE*/

} else if (capacidad <= 95) { /*INICIO LLENO */

Serial.begin (9600);
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print("TP=  C");
lcd.setCursor(3,0);
lcd.print(PTM);
lcd.setCursor(0,1);
lcd.print("H1= % NT= %");
lcd.setCursor(3,1);
lcd.print(PHU);
lcd.setCursor(10,1);
lcd.print(capacidad);
delayMicroseconds(10); /*tiempo de espera*/

/*INDICADORES DE NT CUANDO ESTA LLENO ACCIONAR EL LED VERDE*/
digitalWrite(led2, HIGH); /*Led verde nivel BAJO*/
digitalWrite(led13, LOW); /*Led AMARILLO nivel BAJO*/

```

```

digitalWrite(led1ybuzzer, LOW); /*Led rojo + buzzer nivel ALTO*/

/*INDICADORES DE LA ELECTROVALVULA 1 Y BOMBA */
digitalWrite(led22, HIGH); /*APAGAR BOMBA INDICADOR DE COLOR VERDE*/
delay(500);
digitalWrite(led23, LOW); /*APAGAR VALVE 1 INDICADOR DE COLOR AZUL*/
digitalWrite(led26, HIGH);

} else { /* INICIO PELIGRO*/

Serial.begin (9600);
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print("TP=  C");
lcd.setCursor(3,0);
lcd.print(PTM);
lcd.setCursor(0,1);
lcd.print("H1= % NT= %");
lcd.setCursor(3,1);
lcd.print(PHU);
lcd.setCursor(10,1);
lcd.print(capacidad);
// delay(10);

/*INDICADORES DE NT CUANDO ESTA DESEMBORDAMIENTO ACCIONAR EL LED
ROJO BLINK*/
digitalWrite(led2, LOW); /*Led verde2 nivel BAJO*/
digitalWrite(led13, LOW); /*Led amarillo verde1 nivel BAJO*/
digitalWrite(led1ybuzzer, HIGH); // turn the LED on (HIGH is the voltage level)
delay(100); /*tiempo de espera*/ // wait for a second
digitalWrite(led1ybuzzer, LOW); // turn the LED off by making the voltage LOW
delayMicroseconds(10); /*tiempo de espera*/

digitalWrite(led26, LOW);
/*INDICADORES DE LA ELECTROVALVULA 1 Y BOMBA */
digitalWrite(led22, LOW); /*APAGAR BOMBA INDICADOR DE COLOR VERDE*/
delay(500);
digitalWrite(led23, LOW); /*APAGAR VALVE 1 INDICADOR DE COLOR AZUL*/
}

/*Para las condiciones he usado un if, else if, else*/

if (PHU <= 20 && PTM <= 16) { /*INICIO VACIO*/ /*NO REGAR */
Serial.begin (9600);
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print("TP=  C");
lcd.setCursor(3,0);
lcd.print(PTM);
lcd.setCursor(0,1);
lcd.print("H1= % NT= %");
lcd.setCursor(3,1);
lcd.print(PHU);
lcd.setCursor(10,1);
lcd.print(capacidad);
delayMicroseconds(10); /*tiempo de espera*/

```

```

/*INDICADORES DE CULTIVO CUANDO ESTA VACIO ACCIONAR EL LED ROJO*/
digitalWrite(led28, HIGH); /*LED ROJO ESTADO SECO DEL CULTIVO*/
digitalWrite(led25, LOW); /*LED AMARILLO ESTADO MINIMO DEL CULTIVO*/
digitalWrite(led27, LOW); /*LED VERDE ESTADO COMPLETO DEL CULTIVO*/
/*INDICADORES DE TEMPERATURA CON RESPECTO AL CULTIVO*/
digitalWrite(led29, HIGH); /*LED ROJO TEMPERATURA 0-15 Y 23-30 NO REGAR*/
digitalWrite(led30, LOW); /*LED VERDE TEMPERATURA EN LA CUAL SE PUEDE
REGAR 16-22*/
/*INDICADOR DE LA VALVE 2*/
digitalWrite(led24, LOW); /*ENCENDER VALVE 2 INDICADOR DE COLOR AZUL*/

} else if (PHU <= 50 && PTM <= 16) { /*INICIO MINIMO NO REGAR CUANDO LA
TEMPERATURA ES MENOR A 16*/
Serial.begin (9600);
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print("TP= C");
lcd.setCursor(3,0);
lcd.print(PTM);
lcd.setCursor(0,1);
lcd.print("H1= % NT= %");
lcd.setCursor(3,1);
lcd.print(PHU);
lcd.setCursor(10,1);
lcd.print(capacidad);
delayMicroseconds(10); /*tiempo de espera*/
/*INDICADORES DE NT CUANDO ESTA MINIMO ACCIONAR EL LED AMARILLO*/
digitalWrite(led28, LOW); /*LED ROJO ESTADO SECO DEL CULTIVO*/
digitalWrite(led25, HIGH); /*LED AMARILLO ESTADO MINIMO DEL CULTIVO*/
digitalWrite(led27, LOW); /*LED VERDE ESTADO COMPLETO DEL CULTIVO*/
/*INDICADORES DE TEMPERATURA CON RESPECTO AL CULTIVO*/
digitalWrite(led29, HIGH); /*LED ROJO TEMPERATURA 0-15 Y 23-30 NO REGAR*/
digitalWrite(led30, LOW); /*LED VERDE TEMPERATURA EN LA CUAL SE PUEDE
REGAR 16-22*/
/*INDICADOR DE LA VALVE 2*/
digitalWrite(led24, LOW); /*ENCENDER VALVE 2 INDICADOR DE COLOR AZUL*/

} else if (PHU <= 85 && PTM <= 16) { /*INICIO OPTIMO NO REGAR CUANDO LA
TEMPERATURA ES MENOR A 16*/
Serial.begin (9600);
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print("TP= C");
lcd.setCursor(3,0);
lcd.print(PTM);
lcd.setCursor(0,1);
lcd.print("H1= % NT= %");
lcd.setCursor(3,1);
lcd.print(PHU);
lcd.setCursor(10,1);
lcd.print(capacidad);
delayMicroseconds(10); /*tiempo de espera*/
/*INDICADORES DE NT CUANDO ESTA MINIMO ACCIONAR EL LED AMARILLO*/
digitalWrite(led28, LOW); /*LED ROJO ESTADO SECO DEL CULTIVO*/
digitalWrite(led25, LOW); /*LED AMARILLO ESTADO MINIMO DEL CULTIVO*/
digitalWrite(led27, HIGH); /*LED VERDE ESTADO COMPLETO DEL CULTIVO*/
/*INDICADORES DE TEMPERATURA CON RESPECTO AL CULTIVO*/

```

```

digitalWrite(led29, HIGH); /*LED ROJO TEMPERATURA 0-15 Y 23-30 NO REGAR*/
digitalWrite(led30, LOW); /*LED VERDE TEMPERATURA EN LA CUAL SE PUEDE
REGAR 16-22*/
/*INDICADOR DE LA VALVE 2*/
digitalWrite(led24, LOW); /*ENCENDER VALVE 2 INDICADOR DE COLOR AZUL*/

} else if (PHU <= 20 && PTM <= 21) { /*INICIO MINIMO REGAR CUANDO LA
TEMPERATURA ES MENOR A 21*/
Serial.begin (9600);
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print("TP= C");
lcd.setCursor(3,0);
lcd.print(PTM);
lcd.setCursor(0,1);
lcd.print("H1= % NT= %");
lcd.setCursor(3,1);
lcd.print(PHU);
lcd.setCursor(10,1);
lcd.print(capacidad);
delayMicroseconds(10); /*tiempo de espera*/
/*INDICADORES DE NT CUANDO ESTA MINIMO ACCIONAR EL LED AMARILLO*/
digitalWrite(led28, HIGH); /*LED ROJO ESTADO SECO DEL CULTIVO*/
digitalWrite(led25, LOW); /*LED AMARILLO ESTADO MINIMO DEL CULTIVO*/
digitalWrite(led27, LOW); /*LED VERDE ESTADO COMPLETO DEL CULTIVO*/
/*INDICADORES DE TEMPERATURA CON RESPECTO AL CULTIVO*/
digitalWrite(led29, LOW); /*LED ROJO TEMPERATURA 0-15 Y 23-30 NO REGAR*/
digitalWrite(led30, HIGH); /*LED VERDE TEMPERATURA EN LA CUAL SE PUEDE
REGAR 16-22*/
/*INDICADOR DE LA VALVE 2*/
digitalWrite(led24, HIGH); /*ENCENDER VALVE 2 INDICADOR DE COLOR AZUL*/

} else if (PHU <= 50 && PTM <= 21) { /*INICIO MINIMO REGAR CUANDO LA
TEMPERATURA ES MENOR A 21*/
Serial.begin (9600);
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print("TP= C");
lcd.setCursor(3,0);
lcd.print(PTM);
lcd.setCursor(0,1);
lcd.print("H1= % NT= %");
lcd.setCursor(3,1);
lcd.print(PHU);
lcd.setCursor(10,1);
lcd.print(capacidad);
delayMicroseconds(10); /*tiempo de espera*/
/*INDICADORES DE NT CUANDO ESTA MINIMO ACCIONAR EL LED AMARILLO*/
digitalWrite(led28, LOW); /*LED ROJO ESTADO SECO DEL CULTIVO*/
digitalWrite(led25, HIGH); /*LED AMARILLO ESTADO MINIMO DEL CULTIVO*/
digitalWrite(led27, LOW); /*LED VERDE ESTADO COMPLETO DEL CULTIVO*/
/*INDICADORES DE TEMPERATURA CON RESPECTO AL CULTIVO*/
digitalWrite(led29, LOW); /*LED ROJO TEMPERATURA 0-15 Y 23-30 NO REGAR*/
digitalWrite(led30, HIGH); /*LED VERDE TEMPERATURA EN LA CUAL SE PUEDE
REGAR 16-22*/
/*INDICADOR DE LA VALVE 2*/
digitalWrite(led24, HIGH); /*ENCENDER VALVE 2 INDICADOR DE COLOR AZUL*/

```

```

} else if (PHU <= 85 && PTM <= 21) { /*INICIO LLENO REGAR CUANDO LA
TEMPERATURA ES MENOR A 21*/
  Serial.begin (9600);
  lcd.begin(16,2);
  lcd.setCursor(0,0);
  lcd.print("TP= C");
  lcd.setCursor(3,0);
  lcd.print(PTM);
  lcd.setCursor(0,1);
  lcd.print("H1= % NT= %");
  lcd.setCursor(3,1);
  lcd.print(PHU);
  lcd.setCursor(10,1);
  lcd.print(capacidad);
  delayMicroseconds(10); /*tiempo de espera*/
  /*INDICADORES DE NT CUANDO ESTA MINIMO ACCIONAR EL LED AMARILLO*/
  digitalWrite(led28, LOW); /*LED ROJO ESTADO SECO DEL CULTIVO*/
  digitalWrite(led25, LOW); /*LED AMARILLO ESTADO MINIMO DEL CULTIVO*/
  digitalWrite(led27, HIGH); /*LED VERDE ESTADO COMPLETO DEL CULTIVO*/
  /*INDICADORES DE TEMPERATURA CON RESPECTO AL CULTIVO*/
  digitalWrite(led29, LOW); /*LED ROJO TEMPERATURA 0-15 Y 23-30 NO REGAR*/
  digitalWrite(led30, HIGH); /*LED VERDE TEMPERATURA EN LA CUAL SE PUEDE
REGAR 16-22*/
  /*INDICADOR DE LA VALVE 2*/
  digitalWrite(led24, HIGH); /*ENCENDER VALVE 2 INDICADOR DE COLOR AZUL*/

} else if (PHU <= 20 && PTM <= 30) { /*INICIO VACIO*/ /*NO REGAR CUANDO LA
TEMPERATURA SOBREPASA LOS 21*/
  Serial.begin (9600);
  lcd.begin(16,2);
  lcd.setCursor(0,0);
  lcd.print("TP= C");
  lcd.setCursor(3,0);
  lcd.print(PTM);
  lcd.setCursor(0,1);
  lcd.print("H1= % NT= %");
  lcd.setCursor(3,1);
  lcd.print(PHU);
  lcd.setCursor(10,1);
  lcd.print(capacidad);
  delayMicroseconds(10); /*tiempo de espera*/

  /*INDICADORES DE CULTIVO CUANDO ESTA VACIO ACCIONAR EL LED ROJO*/
  digitalWrite(led28, HIGH); /*LED ROJO ESTADO SECO DEL CULTIVO*/
  digitalWrite(led25, LOW); /*LED AMARILLO ESTADO MINIMO DEL CULTIVO*/
  digitalWrite(led27, LOW); /*LED VERDE ESTADO COMPLETO DEL CULTIVO*/
  /*INDICADORES DE TEMPERATURA CON RESPECTO AL CULTIVO*/
  digitalWrite(led29, HIGH); /*LED ROJO TEMPERATURA 0-15 Y 23-30 NO REGAR*/
  digitalWrite(led30, LOW); /*LED VERDE TEMPERATURA EN LA CUAL SE PUEDE
REGAR 16-22*/
  /*INDICADOR DE LA VALVE 2*/
  digitalWrite(led24, LOW); /*ENCENDER VALVE 2 INDICADOR DE COLOR AZUL*/

} else if (PHU <= 50 && PTM <= 30) { /*INICIO MINIMO NO REGAR CUANDO LA
TEMPERATURA SOBREPASA LOS 21*/

```

```

Serial.begin (9600);
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print("TP=  C");
lcd.setCursor(3,0);
lcd.print(PTM);
lcd.setCursor(0,1);
lcd.print("H1= % NT= %");
lcd.setCursor(3,1);
lcd.print(PHU);
lcd.setCursor(10,1);
lcd.print(capacidad);
delayMicroseconds(10); /*tiempo de espera*/
/*INDICADORES DE NT CUANDO ESTA MINIMO ACCIONAR EL LED AMARILLO*/
digitalWrite(led28, LOW); /*LED ROJO ESTADO SECO DEL CULTIVO*/
digitalWrite(led25, HIGH); /*LED AMARILLO ESTADO MINIMO DEL CULTIVO*/
digitalWrite(led27, LOW); /*LED VERDE ESTADO COMPLETO DEL CULTIVO*/
/*INDICADORES DE TEMPERATURA CON RESPECTO AL CULTIVO*/
digitalWrite(led29, HIGH); /*LED ROJO TEMPERATURA 0-15 Y 23-30 NO REGAR*/
digitalWrite(led30, LOW); /*LED VERDE TEMPERATURA EN LA CUAL SE PUEDE
REGAR 16-22*/
/*INDICADOR DE LA VALVE 2*/
digitalWrite(led24, LOW); /*ENCENDER VALVE 2 INDICADOR DE COLOR AZUL*/

} else if (PHU <= 85 && PTM <= 30) { /*INICIO OPTIMO NO REGAR CUANDO LA
TEMPERATURA SOBREPASA LOS 21*/
Serial.begin (9600);
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print("TP=  C");
lcd.setCursor(3,0);
lcd.print(PTM);
lcd.setCursor(0,1);
lcd.print("H1= % NT= %");
lcd.setCursor(3,1);
lcd.print(PHU);
lcd.setCursor(10,1);
lcd.print(capacidad);
delayMicroseconds(10); /*tiempo de espera*/
/*INDICADORES DE NT CUANDO ESTA MINIMO ACCIONAR EL LED AMARILLO*/
digitalWrite(led28, LOW); /*LED ROJO ESTADO SECO DEL CULTIVO*/
digitalWrite(led25, LOW); /*LED AMARILLO ESTADO MINIMO DEL CULTIVO*/
digitalWrite(led27, HIGH); /*LED VERDE ESTADO COMPLETO DEL CULTIVO*/
/*INDICADORES DE TEMPERATURA CON RESPECTO AL CULTIVO*/
digitalWrite(led29, HIGH); /*LED ROJO TEMPERATURA 0-15 Y 23-30 NO REGAR*/
digitalWrite(led30, LOW); /*LED VERDE TEMPERATURA EN LA CUAL SE PUEDE
REGAR 16-22*/
/*INDICADOR DE LA VALVE 2*/
digitalWrite(led24, LOW); /*ENCENDER VALVE 2 INDICADOR DE COLOR AZUL*/

} else { /*CUANDO LA HUMEDA ESTA COMPLETA CERRAR ELECTROVALVULA2*/

Serial.begin (9600);
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print("TP=  C");

```

```

lcd.setCursor(3,0);
lcd.print(PTM);
lcd.setCursor(0,1);
lcd.print("H1= % NT= %");
lcd.setCursor(3,1);
lcd.print(PHU);
lcd.setCursor(10,1);
lcd.print(capacidad);
delayMicroseconds(10); /*tiempo de espera*/
/*INDICADORES DE NT CUANDO ESTA MINIMO ACCIONAR EL LED AMARILLO*/
digitalWrite(led28, LOW); /*LED ROJO ESTADO SECO DEL CULTIVO*/
digitalWrite(led25, LOW); /*LED AMARILLO ESTADO MINIMO DEL CULTIVO*/
digitalWrite(led27, HIGH); /*LED VERDE ESTADO COMPLETO DEL CULTIVO*/
/*INDICADORES DE TEMPERATURA CON RESPECTO AL CULTIVO*/
digitalWrite(led29, HIGH); /*LED ROJO TEMPERATURA 0-15 Y 23-30 NO REGAR*/
digitalWrite(led30, LOW); /*LED VERDE TEMPERATURA EN LA CUAL SE PUEDE
REGAR 16-22*/
/*INDICADOR DE LA VALVE 2*/
digitalWrite(led24, LOW); /*ENCENDER VALVE 2 INDICADOR DE COLOR AZUL*/
}
//}
}
void cinco()
{

```

/*Para las condiciones he usado un if, else if, else*/

```

if (PHU <= 20 && PTM <= 16) { /*INICIO VACIO*/ /*NO REGAR */
// Serial.begin (9600);
// lcd.begin(16,2);
// lcd.setCursor(0,0);
// lcd.print("TP= C");
// lcd.setCursor(3,0);
// lcd.print(PTM);
// lcd.setCursor(0,1);
// lcd.print("H1= % NT= %");
// lcd.setCursor(3,1);
// lcd.print(PHU);
// lcd.setCursor(10,1);
// lcd.print(capacidad);
// delay(100); /*tiempo de espera*/

```

```

////////////////////////prueba inicio////////////////////
Serial.begin (9600);
lcd .clear ();
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print(" CULTIVO");
lcd.setCursor(0,1);
lcd.print("H1= % TP= C");
lcd.setCursor(3,1);
lcd.print(PHU);
lcd.setCursor(10,1);
lcd.print(PTM);
delay(100); /*tiempo de espera*/
////////////////////////fin prueba inicii////////////////////

```

```

/*INDICADORES DE CULTIVO CUANDO ESTA VACIO ACCIONAR EL LED ROJO*/
digitalWrite(led28, HIGH); /*LED ROJO ESTADO SECO DEL CULTIVO*/
digitalWrite(led25, LOW); /*LED AMARILLO ESTADO MINIMO DEL CULTIVO*/
digitalWrite(led27, LOW); /*LED VERDE ESTADO COMPLETO DEL CULTIVO*/
/*INDICADORES DE TEMPERATURA CON RESPECTO AL CULTIVO*/
digitalWrite(led29, HIGH); /*LED ROJO TEMPERATURA 0-15 Y 23-30 NO REGAR*/
digitalWrite(led30, LOW); /*LED VERDE TEMPERATURA EN LA CUAL SE PUEDE
REGAR 16-22*/
/*INDICADOR DE LA VALVE 2*/
digitalWrite(led24, LOW); /*ENCENDER VALVE 2 INDICADOR DE COLOR AZUL*/

} else if (PHU <= 50 && PTM <= 16) { /*INICIO MINIMO NO REGAR CUANDO LA
TEMPERATURA ES MENOR A 16*/
Serial.begin (9600);
lcd .clear ();
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print(" CULTIVO");
lcd.setCursor(0,1);
lcd.print("H1= % TP= C");
lcd.setCursor(3,1);
lcd.print(PHU);
lcd.setCursor(10,1);
lcd.print(PTM);
delay(100); /*tiempo de espera*/
/*INDICADORES DE NT CUANDO ESTA MINIMO ACCIONAR EL LED AMARILLO*/
digitalWrite(led28, LOW); /*LED ROJO ESTADO SECO DEL CULTIVO*/
digitalWrite(led25, HIGH); /*LED AMARILLO ESTADO MINIMO DEL CULTIVO*/
digitalWrite(led27, LOW); /*LED VERDE ESTADO COMPLETO DEL CULTIVO*/
/*INDICADORES DE TEMPERATURA CON RESPECTO AL CULTIVO*/
digitalWrite(led29, HIGH); /*LED ROJO TEMPERATURA 0-15 Y 23-30 NO REGAR*/
digitalWrite(led30, LOW); /*LED VERDE TEMPERATURA EN LA CUAL SE PUEDE
REGAR 16-22*/
/*INDICADOR DE LA VALVE 2*/
digitalWrite(led24, LOW); /*ENCENDER VALVE 2 INDICADOR DE COLOR AZUL*/

} else if (PHU <= 85 && PTM <= 16) { /*INICIO OPTIMO NO REGAR CUANDO LA
TEMPERATURA ES MENOR A 16*/
Serial.begin (9600);
lcd .clear ();
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print(" CULTIVO");
lcd.setCursor(0,1);
lcd.print("H1= % TP= C");
lcd.setCursor(3,1);
lcd.print(PHU);
lcd.setCursor(10,1);
lcd.print(PTM);
delay(100); /*tiempo de espera*/
/*INDICADORES DE NT CUANDO ESTA MINIMO ACCIONAR EL LED AMARILLO*/
digitalWrite(led28, LOW); /*LED ROJO ESTADO SECO DEL CULTIVO*/
digitalWrite(led25, LOW); /*LED AMARILLO ESTADO MINIMO DEL CULTIVO*/
digitalWrite(led27, HIGH); /*LED VERDE ESTADO COMPLETO DEL CULTIVO*/
/*INDICADORES DE TEMPERATURA CON RESPECTO AL CULTIVO*/
digitalWrite(led29, HIGH); /*LED ROJO TEMPERATURA 0-15 Y 23-30 NO REGAR*/

```

```

digitalWrite(led30, LOW); /*LED VERDE TEMPERATURA EN LA CUAL SE PUEDE
REGAR 16-22*/
/*INDICADOR DE LA VALVE 2*/
digitalWrite(led24, LOW); /*ENCENDER VALVE 2 INDICADOR DE COLOR AZUL*/

} else if (PHU <= 20 && PTM <= 21) { /*INICIO MINIMO REGAR CUANDO LA
TEMPERATURA ES MENOR A 21*/
Serial.begin (9600);
lcd .clear ();
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print(" CULTIVO");
lcd.setCursor(0,1);
lcd.print("H1= % TP= C");
lcd.setCursor(3,1);
lcd.print(PHU);
lcd.setCursor(10,1);
lcd.print(PTM);
delay(100); /*tiempo de espera*/
/*INDICADORES DE NT CUANDO ESTA MINIMO ACCIONAR EL LED AMARILLO*/
digitalWrite(led28, HIGH); /*LED ROJO ESTADO SECO DEL CULTIVO*/
digitalWrite(led25, LOW); /*LED AMARILLO ESTADO MINIMO DEL CULTIVO*/
digitalWrite(led27, LOW); /*LED VERDE ESTADO COMPLETO DEL CULTIVO*/
/*INDICADORES DE TEMPERATURA CON RESPECTO AL CULTIVO*/
digitalWrite(led29, LOW); /*LED ROJO TEMPERATURA 0-15 Y 23-30 NO REGAR*/
digitalWrite(led30, HIGH); /*LED VERDE TEMPERATURA EN LA CUAL SE PUEDE
REGAR 16-22*/
/*INDICADOR DE LA VALVE 2*/
digitalWrite(led24, HIGH); /*ENCENDER VALVE 2 INDICADOR DE COLOR AZUL*/

} else if (PHU <= 50 && PTM <= 21) { /*INICIO MINIMO REGAR CUANDO LA
TEMPERATURA ES MENOR A 21*/
Serial.begin (9600);
lcd .clear ();
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print(" CULTIVO");
lcd.setCursor(0,1);
lcd.print("H1= % TP= C");
lcd.setCursor(3,1);
lcd.print(PHU);
lcd.setCursor(10,1);
lcd.print(PTM);
delay(100); /*tiempo de espera*/
/*INDICADORES DE NT CUANDO ESTA MINIMO ACCIONAR EL LED AMARILLO*/
digitalWrite(led28, LOW); /*LED ROJO ESTADO SECO DEL CULTIVO*/
digitalWrite(led25, HIGH); /*LED AMARILLO ESTADO MINIMO DEL CULTIVO*/
digitalWrite(led27, LOW); /*LED VERDE ESTADO COMPLETO DEL CULTIVO*/
/*INDICADORES DE TEMPERATURA CON RESPECTO AL CULTIVO*/
digitalWrite(led29, LOW); /*LED ROJO TEMPERATURA 0-15 Y 23-30 NO REGAR*/
digitalWrite(led30, HIGH); /*LED VERDE TEMPERATURA EN LA CUAL SE PUEDE
REGAR 16-22*/
/*INDICADOR DE LA VALVE 2*/
digitalWrite(led24, HIGH); /*ENCENDER VALVE 2 INDICADOR DE COLOR AZUL*/

} else if (PHU <= 85 && PTM <= 21) { /*INICIO LLENO REGAR CUANDO LA
TEMPERATURA ES MENOR A 21*/

```

```

Serial.begin (9600);
lcd .clear ();
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print(" CULTIVO");
lcd.setCursor(0,1);
lcd.print("H1= % TP= C");
lcd.setCursor(3,1);
lcd.print(PHU);
lcd.setCursor(10,1);
lcd.print(PTM);
delay(100); /*tiempo de espera*/
/*INDICADORES DE NT CUANDO ESTA MINIMO ACCIONAR EL LED AMARILLO*/
digitalWrite(led28, LOW); /*LED ROJO ESTADO SECO DEL CULTIVO*/
digitalWrite(led25, LOW); /*LED AMARILLO ESTADO MINIMO DEL CULTIVO*/
digitalWrite(led27, HIGH); /*LED VERDE ESTADO COMPLETO DEL CULTIVO*/
/*INDICADORES DE TEMPERATURA CON RESPECTO AL CULTIVO*/
digitalWrite(led29, LOW); /*LED ROJO TEMPERATURA 0-15 Y 23-30 NO REGAR*/
digitalWrite(led30, HIGH); /*LED VERDE TEMPERATURA EN LA CUAL SE PUEDE
REGAR 16-22*/
/*INDICADOR DE LA VALVE 2*/
digitalWrite(led24, HIGH); /*ENCENDER VALVE 2 INDICADOR DE COLOR AZUL*/

} else if (PHU <= 20 && PTM <= 30) { /*INICIO VACIO*/ /*NO REGAR CUANDO LA
TEMPERATURA SOBREPASA LOS 21*/
Serial.begin (9600);
lcd .clear ();
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print(" CULTIVO");
lcd.setCursor(0,1);
lcd.print("H1= % TP= C");
lcd.setCursor(3,1);
lcd.print(PHU);
lcd.setCursor(10,1);
lcd.print(PTM);
delay(100); /*tiempo de espera*/

/*INDICADORES DE CULTIVO CUANDO ESTA VACIO ACCIONAR EL LED ROJO*/
digitalWrite(led28, HIGH); /*LED ROJO ESTADO SECO DEL CULTIVO*/
digitalWrite(led25, LOW); /*LED AMARILLO ESTADO MINIMO DEL CULTIVO*/
digitalWrite(led27, LOW); /*LED VERDE ESTADO COMPLETO DEL CULTIVO*/
/*INDICADORES DE TEMPERATURA CON RESPECTO AL CULTIVO*/
digitalWrite(led29, HIGH); /*LED ROJO TEMPERATURA 0-15 Y 23-30 NO REGAR*/
digitalWrite(led30, LOW); /*LED VERDE TEMPERATURA EN LA CUAL SE PUEDE
REGAR 16-22*/
/*INDICADOR DE LA VALVE 2*/
digitalWrite(led24, LOW); /*ENCENDER VALVE 2 INDICADOR DE COLOR AZUL*/

} else if (PHU <= 50 && PTM <= 30) { /*INICIO MINIMO NO REGAR CUANDO LA
TEMPERATURA SOBREPASA LOS 21*/
Serial.begin (9600);
lcd .clear ();
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print(" CULTIVO");

```

```

lcd.setCursor(0,1);
lcd.print("H1= % TP=  C");
lcd.setCursor(3,1);
lcd.print(PHU);
lcd.setCursor(10,1);
lcd.print(PTM);
delay(100); /*tiempo de espera*/
/*INDICADORES DE NT CUANDO ESTA MINIMO ACCIONAR EL LED AMARILLO*/
digitalWrite(led28, LOW); /*LED ROJO ESTADO SECO DEL CULTIVO*/
digitalWrite(led25, HIGH); /*LED AMARILLO ESTADO MINIMO DEL CULTIVO*/
digitalWrite(led27, LOW); /*LED VERDE ESTADO COMPLETO DEL CULTIVO*/
/*INDICADORES DE TEMPERATURA CON RESPECTO AL CULTIVO*/
digitalWrite(led29, HIGH); /*LED ROJO TEMPERATURA 0-15 Y 23-30 NO REGAR*/
digitalWrite(led30, LOW); /*LED VERDE TEMPERATURA EN LA CUAL SE PUEDE
REGAR 16-22*/
/*INDICADOR DE LA VALVE 2*/
digitalWrite(led24, LOW); /*ENCENDER VALVE 2 INDICADOR DE COLOR AZUL*/

} else if (PHU <= 85 && PTM <= 30) { /*INICIO OPTIMO NO REGAR CUANDO LA
TEMPERATURA SOBREPASA LOS 21*/
Serial.begin (9600);
lcd .clear ();
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print(" CULTIVO");
lcd.setCursor(0,1);
lcd.print("H1= % TP=  C");
lcd.setCursor(3,1);
lcd.print(PHU);
lcd.setCursor(10,1);
lcd.print(PTM);
delay(100); /*tiempo de espera*/
/*INDICADORES DE NT CUANDO ESTA MINIMO ACCIONAR EL LED AMARILLO*/
digitalWrite(led28, LOW); /*LED ROJO ESTADO SECO DEL CULTIVO*/
digitalWrite(led25, LOW); /*LED AMARILLO ESTADO MINIMO DEL CULTIVO*/
digitalWrite(led27, HIGH); /*LED VERDE ESTADO COMPLETO DEL CULTIVO*/
/*INDICADORES DE TEMPERATURA CON RESPECTO AL CULTIVO*/
digitalWrite(led29, HIGH); /*LED ROJO TEMPERATURA 0-15 Y 23-30 NO REGAR*/
digitalWrite(led30, LOW); /*LED VERDE TEMPERATURA EN LA CUAL SE PUEDE
REGAR 16-22*/
/*INDICADOR DE LA VALVE 2*/
digitalWrite(led24, LOW); /*ENCENDER VALVE 2 INDICADOR DE COLOR AZUL*/

} else { /*CUANDO LA HUMEDA ESTA COMPLETA CERRAR ELECTROVALVULA2*/

Serial.begin (9600);
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print(" CULTIVO");
lcd.setCursor(0,1);
lcd.print("H1= % TP=  C");
lcd.setCursor(3,1);
lcd.print(PHU);
lcd.setCursor(10,1);
lcd.print(PTM);
delay(100); /*tiempo de espera*/

```

```

/*INDICADORES DE NT CUANDO ESTA MINIMO ACCIONAR EL LED AMARILLO*/
digitalWrite(led28, LOW); /*LED ROJO ESTADO SECO DEL CULTIVO*/
digitalWrite(led25, LOW); /*LED AMARILLO ESTADO MINIMO DEL CULTIVO*/
digitalWrite(led27, HIGH); /*LED VERDE ESTADO COMPLETO DEL CULTIVO*/
/*INDICADORES DE TEMPERATURA CON RESPECTO AL CULTIVO*/
digitalWrite(led29, HIGH); /*LED ROJO TEMPERATURA 0-15 Y 23-30 NO REGAR*/
digitalWrite(led30, LOW); /*LED VERDE TEMPERATURA EN LA CUAL SE PUEDE
REGAR 16-22*/
/*INDICADOR DE LA VALVE 2*/
digitalWrite(led24, LOW); /*ENCENDER VALVE 2 INDICADOR DE COLOR AZUL*/
}
// }
}

```

```

////////////////////////////////UNO////////////////////////////////

```

```

void cero()
{

digitalWrite(eco,LOW);
digitalWrite(trigger,LOW);
digitalWrite(led1ybuzzer,LOW);
digitalWrite(led2,LOW);
digitalWrite(led13,LOW);
digitalWrite(led22,LOW);
digitalWrite(led23,LOW);
digitalWrite(HA0,LOW);
digitalWrite(HA1,LOW);
digitalWrite(HA2,LOW);
digitalWrite(HA3,LOW);
digitalWrite(HA4,LOW);
digitalWrite(HA5,LOW);
digitalWrite(led24,LOW);
digitalWrite(led25,LOW);
digitalWrite(led27,LOW);
digitalWrite(led28,LOW);
digitalWrite(TA6,LOW);
digitalWrite(TA7,LOW);
digitalWrite(TA8,LOW);
digitalWrite(TA9,LOW);
digitalWrite(TA10,LOW);
digitalWrite(led29,LOW);
digitalWrite(led30,LOW);
digitalWrite(led26,LOW);

```

```

Serial.begin (9600);
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print(" TOTOGALPA");
delay(1000);

```

```

}
////////////////////////////////FIN UNO////////////////////////////////

```

```

void uno ()
{
////////////////////////////////INICIO CICLO2////////////////////////////////
Serial.begin(9600);

```

```

//while (true)
//{

//////FIN CICLO 2//////////
//////////INICIO ECUACION//////////
/*Comandos para controlar el sensor de ultrasonidos*/
digitalWrite(trigger, LOW); /*Emitimos ultrasonidos*/
delayMicroseconds(2); /*tiempo de espera*/
digitalWrite(trigger, HIGH); /*Paramos la emisión*/
delayMicroseconds(10); /*tiempo de espera*/
digitalWrite(trigger, LOW); /*Emitimos ultrasonidos*/
delayMicroseconds(10); /*tiempo de espera*/
tiempo= pulseIn(eco, HIGH); /*Lo que tarda en recibir el eco*/
distancia= int(0.017*tiempo);
nivel =-3 + distancia; /*Pues he situado el medido 3cm por encima del depósito*/
capacidad = (100 - ((nivel) /1095) * 100);
//////////FIN ECUACION//////////
/*Para las condiciones de NT he usado un if, else if, else*/

if (capacidad <= 10) { /* INICIO VACIO*/

// Serial.begin (9600);
// lcd.begin(16,2);
// lcd.setCursor(0,0);
// lcd.print("TP= C");
// lcd.setCursor(3,0);
// lcd.print(PTM);
// lcd.setCursor(0,1);
// lcd.print("H1= % NT= %");
// lcd.setCursor(3,1);
// lcd.print(PHU);
// lcd.setCursor(10,1);
// lcd.print(capacidad);
// //delay(10);

Serial.begin (9600);
lcd .clear ();
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print("NIVEL DEL TANQUE");
lcd.setCursor(0,1);
lcd.print("NT= % ");
lcd.setCursor(3,1);
lcd.print(capacidad);
delay(10); /*tiempo de espera*/

/*INDICADORES DE NT CUANDO ESTA VACIO ACCIONAR EL LED ROJO*/
digitalWrite(led2, LOW); /*Led verde2 nivel BAJO*/
digitalWrite(led13, LOW); /*Led amarillo nivel BAJO*/
digitalWrite(led1ybuzzer, HIGH); /*Led rojo + buzzer nivel ALTO*/
digitalWrite(led22, HIGH);

/*INDICADORES DE LA ELECTROVALVULA 1 Y BOMBA */
digitalWrite(led26, LOW); /*VALVULA DEL POZO */
digitalWrite(led23, HIGH); /*ENCENDER VALVE 1 INDICADOR DE COLOR AZUL*/
delay(500);
digitalWrite(led22, HIGH); /*ENCENDER bomba INDICADOR DE COLOR VERDE*/

```

```

} else if (capacidad <= 50) { /*INICIO MINIMO*/

Serial.begin (9600);
lcd .clear ();
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print("NIVEL DEL TANQUE");
lcd.setCursor(0,1);
lcd.print("NT= % ");
lcd.setCursor(3,1);
lcd.print(capacidad);
delay(10); /*tiempo de espera*/

/*INDICADORES DE NT CUANDO ESTA MINIMO ACCIONAR EL LED AMARILLO*/
digitalWrite(led2, LOW); /*Led verde2 nivel BAJO*/
digitalWrite(led13, HIGH); /*Led amarillo nivel ALTO*/
digitalWrite(led1ybuzzer, LOW); /*Led rojo + buzzer nivel ALTO*/

/*INDICADORES DE LA ELECTROVALVULA 1 Y BOMBA */
digitalWrite(led26, LOW); /*VALVULA DEL POZO */
digitalWrite(led23, HIGH); /*ENCENDER VALVE 1 INDICADOR DE COLOR AZUL*/
delay(500);
digitalWrite(led22, HIGH); /*ENCENDER bomba INDICADOR DE COLOR VERDE*/

} else if (capacidad <= 90) { /*INICIO OPTIMO*/

Serial.begin (9600);
lcd .clear ();
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print("NIVEL DEL TANQUE");
lcd.setCursor(0,1);
lcd.print("NT= % ");
lcd.setCursor(3,1);
lcd.print(capacidad);
Delay(10); /*tiempo de espera*/

/*INDICADORES DE NT CUANDO ESTA OPTIMO ACCIONAR EL LED AZUL*/
digitalWrite(led2, HIGH); /*Led verde2 nivel BAJO*/
digitalWrite(led13, LOW); /*Led amarillo nivel BAJO*/
digitalWrite(led1ybuzzer, LOW); /*Led rojo + buzzer nivel ALTO*/

/*INDICADORES DE LA ELECTROVALVULA 1 Y BOMBA */
digitalWrite(led26, LOW); /*VALVULA DEL POZO */
digitalWrite(led23, HIGH); /*ENCENDER VALVE 1 INDICADOR DE COLOR AZUL*/
delay(500);
digitalWrite(led22, HIGH); /*ENCENDER bomba INDICADOR DE COLOR VERDE*/

} else if (capacidad <= 95) { /*INICIO LLENO */

Serial.begin (9600);
lcd .clear ();
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print("NIVEL DEL TANQUE");
lcd.setCursor(0,1);

```



```

digitalWrite(led26, HIGH); /*PARA ENCENDER LA VALVULA DEL TANQUE QUE ESTA
CERCA DEL POZO*/
delay (500);
digitalWrite(led22, HIGH); /*ENCIENDE LA BOMBA DESPUES DE UN TIEMPO
DETERMINDANO*/
}else {
digitalWrite(led22, LOW); /*APAGA LA BOMBA DESPUES DE UN TIEMPO
DETERMINDANO*/
digitalWrite(led26, LOW); /*APAGA LA BOMBA DESPUES DE UN TIEMPO
DETERMINDANO*/
}

lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print("LLENADO TANQUE 1");/*LLENANDO EL TANQUE I*/
delay(10);
}
//////////////////////////////////////////////////////////////////FIN VOID TRES//////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////// FASE LOOP//////////////////////////////////////////////////////////////////

void loop(){
char customKey = teclado.getKey();

if (customKey){
switch (customKey)
{
case '#': // sistema de reinicio stop//
variable=0;
break;

case '1': //sistema de llenado de tanque//
variable=1;
break;

case '2': // sistema de cultivo//
variable=3;
break;

case '*': // sistema automatico start//
variable=2;
break;

case '3': // sistema de cultivo//
variable=5;
break;

}
}
if (variable==0)
{
cero();
}
else if (variable==1)
{
uno();
}
}

```

```
else if (variable==2)
{
  dos();
}
else if (variable==3)
{
  tres();
}

else if(variable==5)
{
  cinco();
}
}
```

A.2. Código para la unidad remota y módulo router en arduino.

```
/*
  XBEE - Emisor
*/

#include <Wire.h>

int temperature; // asignación para el valor real de la temperatura
int Humidity; // asignación para el valor de la humedad
int TP = 0; // asignación para el valor del puerto de la temperatura
int H,Hm = 0; // asignación para el valor del puerto de la humedad
int A,E,I,O,U = 0; // asignación para el valor del puerto A0 de la moisture
int B,C = 0; // asignación para el valor del puerto A1 de la humedad
char buffer[10]; // variable para crear una cadena de caracteres para poder mostrar el valor
de humedad
int rojo = 7;
int verde = 8;

unsigned long tiempo_tx; // variable de tiempo para la transmisión con condiciones de tiempo
//unsigned long tiempo_ttx; // variable de tiempo para la transmisión con condiciones de
tiempo

void setup(){

  Serial.begin(9600); //Iniciar el Serial
  delay(1000); // retardo
  // It will use the pin six to control the sleep of the module
  pinMode(6, OUTPUT);

  // it configure the digital port. It will use red pin to know if arduino board are communicating
  pinMode(rojo,OUTPUT);
  pinMode(8, OUTPUT);
  // it begin the port 7 in low.
  digitalWrite(rojo,LOW);
  digitalWrite(verde,LOW);

  tiempo_tx = millis();
  // tiempo_ttx = millis();
```

```

    }
void loop(){

    B = analogRead(A0); // puerto para obtener valor de temperatura
    A = analogRead(A1); // puerto para obtener valor de humedad
    E = analogRead(A2); // puerto para obtener valor de humedad
    I = analogRead(A3); // puerto para obtener valor de humedad
    O = analogRead(A4); // puerto para obtener valor de humedad
    U = analogRead(A5); // puerto para obtener valor de humedad
    /*Ecuación para determinar la temperatura ambiente y humedad*/
    //TP = (((5*A)*100)/1023);
    TP = (B*0.5);
    H = (A + E + I + O + U)/5; //(A + E + I + O + U)/5;
    Hm= map(H,1023,0,0,90); //290-265 // //(100-(B * 0.09775171));

    // tierra seca 0-40
    // tierra optima 41-74
    // tierra saturada 75-100

    temperature = (int)TP; // valor de la temperatura con variable tipo entero
    Humidity = (int)Hm; //(int)buffer; // valor de la temperatura con variable tipo entero
    sprintf(buffer,"%02d",Humidity); // se crea una cadena para mostrar solo 3 digitos en
pantalla.

    // probamos si el el sensor está procesando datos

    //if (isnan(temperature) || isnan(Humidity)){
    //
    //Serial.println("check the sensor");
    //digitalWrite(verde,HIGH);
    //}

    //else
    digitalWrite(verde,LOW);

    if ((millis()-tiempo_tx) > 3000){ // condición de transmisión cada 2 segundos de la
temperatura
        digitalWrite(verde,HIGH);

    /* Pin 6 configurado como pin de hibernación en modo salida

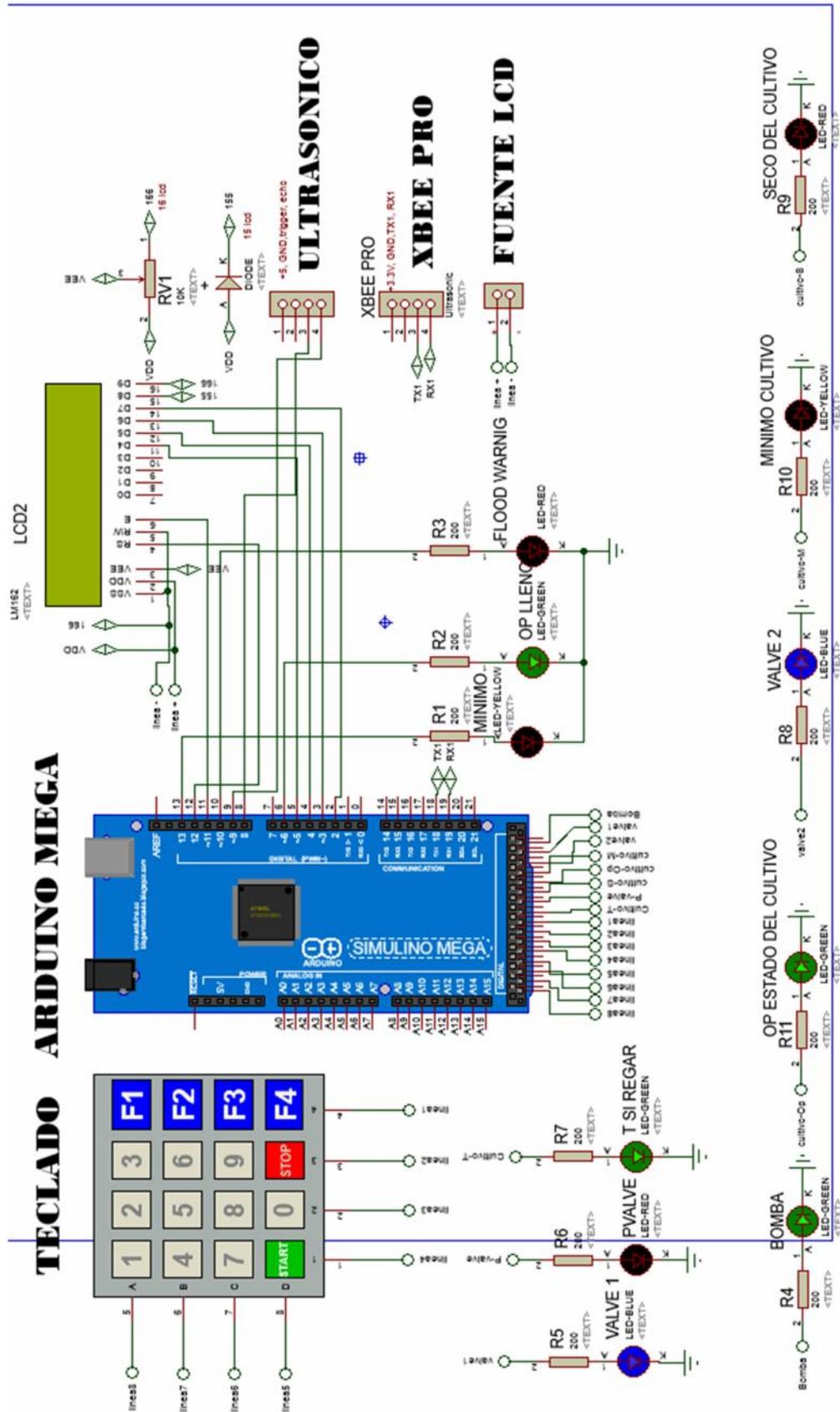
```

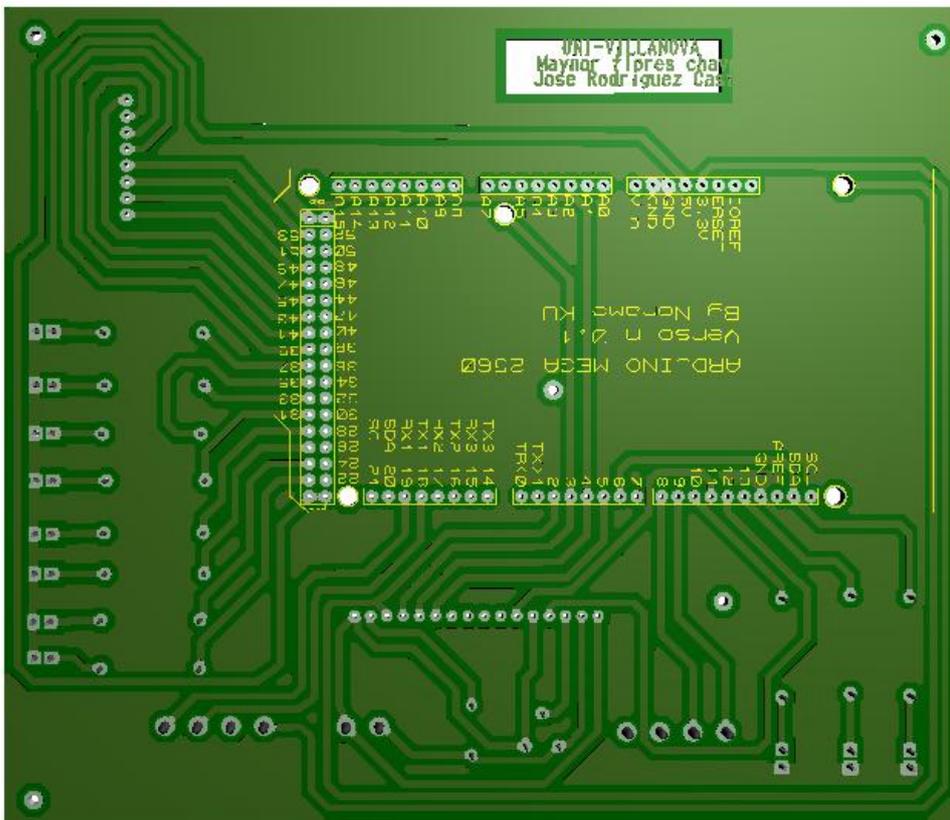
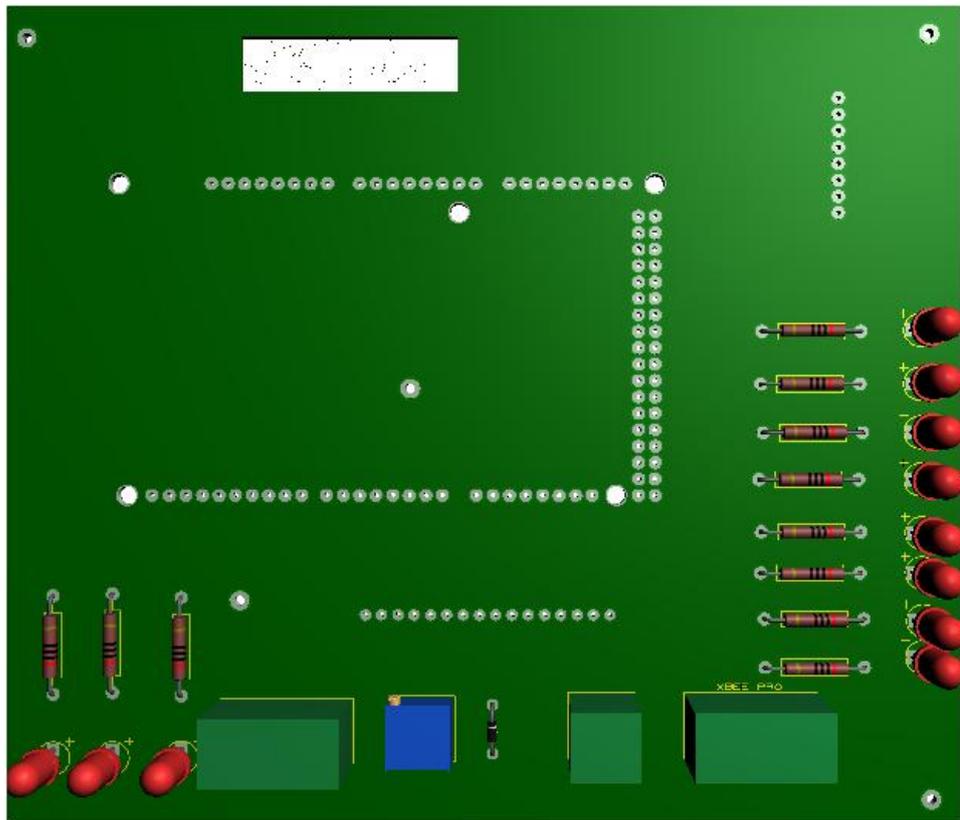
prescindir del pin 6 si no se desea modo de bajo consumo)
Se desactiva el modo de bajo consumo para enviar */

```
// digitalWrite(6, LOW);  
// delay(1000); // Tiempo de guarda  
  
Serial.print("T"); // se envía la unidad hacia el coordinador  
Serial.print(temperature); // valor de la temperatura  
Serial.print("H"); // se envía la unidad hacia el coordinador  
Serial.print(buffer); // valor de la temperatura  
tiempo_tx = millis(); // tiempo de actualización.  
}  
  
delay(1000); // Tiempo de guarda  
  
}
```

ANEXO B. Diseño, fabricación y visualización 3D de tarjetas de circuito impreso.

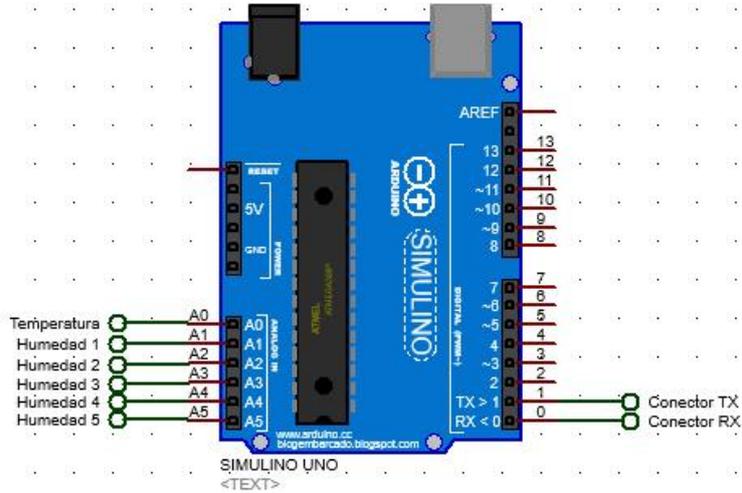
B.1. Diagrama circuital de los sistemas de la unidad central.



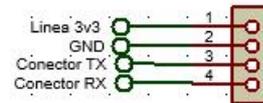


B.3. Diagrama circuital de la unidad remota.

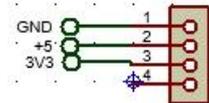
ARDUINO UNO



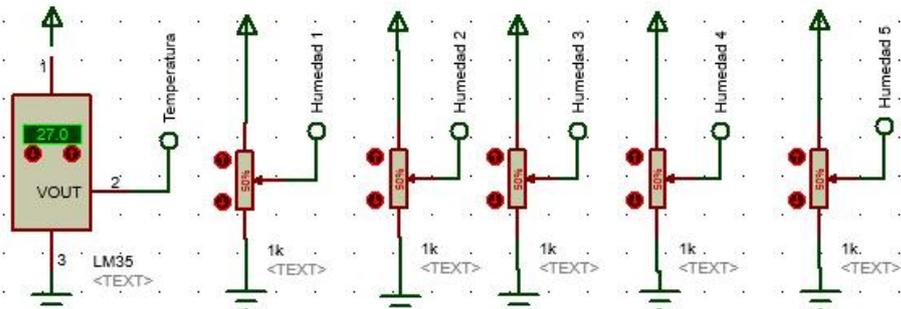
XBEE PRO



ARDUI-SENSOR

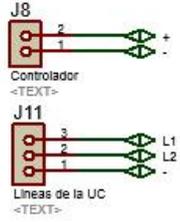


SENSORES

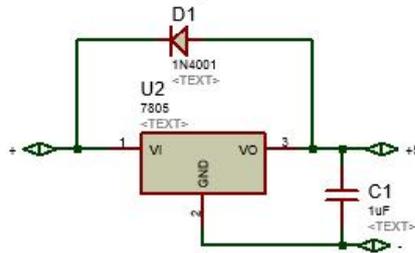


B.4. Diagrama circuital de la unidad de bombeo.

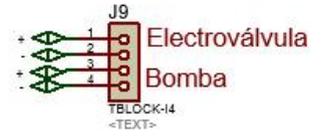
Cables de entrada



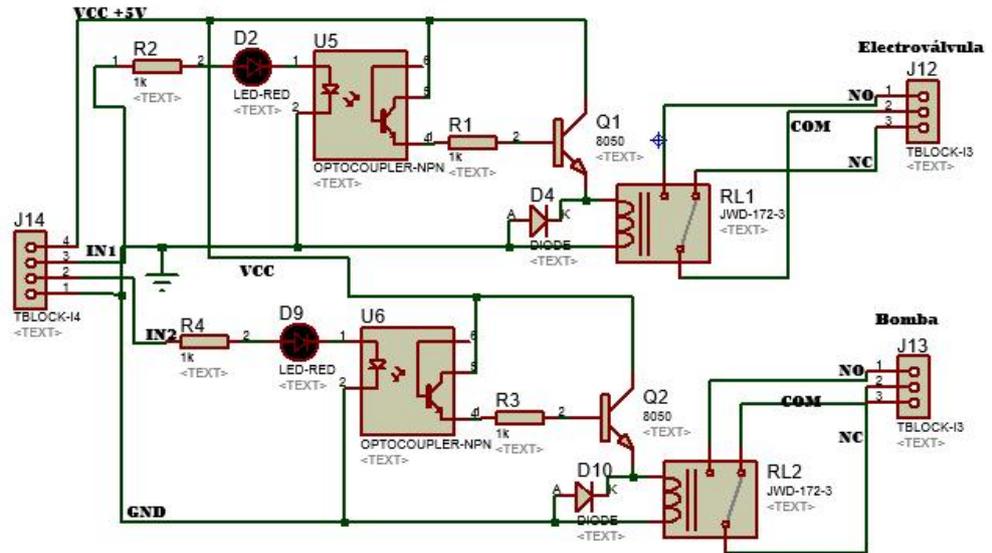
Fuente reguladora



Elementos Finales



Módulo relé



ANEXO C. Correos y cuestionarios para el grupo Totogalpa

C.1. Correo para la líder del grupo en la comunidad de Tototalpa

Fwd: Informe final de proyecto en tototalpa



Jose Castro
Ayer, 03:06 p.m.
Usted ↕



Responder | ▾

----- Mensaje reenviado -----

De: **Jose Castro** <josefrancisorodriguezcastro@gmail.com>
Fecha: 16 de noviembre de 2017, 23:00
Asunto: Re: Informe final de proyecto en tototalpa
Para: Susan Kinne <susankinnefenix@gmail.com>

¡Gracias a ti! Cuando lo tenga mas información precisa se lo hare saber

El 16 nov. 2017 10:25 PM, "Susan Kinne" <susankinnefenix@gmail.com> escribió:
Muchas gracias Jose,
Justo a tiempo para bajarlos.

Adelante con el Sol,;)
Susan

On Nov 16, 2017 5:17 PM, "Jose Castro" <josefrancisorodriguezcastro@gmail.com> wrote:
Hola susan. Le adjunto unos de los avances que hemos tenido para el proyecto del sistema de riego.

Saludos

C.2. Correo de avance del funcionamiento de los equipos en los laboratorios para el líder del grupo

Re: Avances de proyecto en la comunidad de Totogalpa

SK

Susan Kinne <susankinnefenix@gmail.com>

mié 05/04, 06:37 p.m.

Usted ↵

Respondiste el 06/04/2017 06:06 p.m.

Muchas gracias Maynor,
Dice un amigo que trabaja con sistemas de riego que podríamos ocupar "Linear Current Boosters" y conectar todos nuestros paneles y ocupar las baterías. Es interesante. Voy a preguntar a otros.

Adelante con el Sol,

Susan Kinne

Presidente JD, Grupo Fenix,

www.grupofenix.org

Cell +505 8624 9350

El 3 de abril de 2017, 13:03, maynor flores <maynorflores82@hotmail.com>

Estimada Susan

Quiero compartir con usted algunos avances que José y yo hemos hecho en el sistema de riego de café en la comunidad de Totogalpa. voy a agregar un video con información relevante del proyecto, con la visita de campo que se realizó en nuestro proyecto. Tendremos una reunión con nuestro tutor la otra semana para una presentación más explicativa para que pueda compartirla con el grupo.

Link

<https://www.youtube.com/watch?v=M3TCuAk0pRI&feature=youtu.be>

<https://www.youtube.com/watch?v=3LWS67kxr8A>

Saludos,

Maynor Flores

José Rodríguez

C.3. Cuestionario realizado durante la visita de campo al señor Juliam Martinez en la comunidad de Totogalpa.

¿Quién es el encargado de las actividades en el centro solar?

R: La señora Susan en el centro solar

¿Cuántos tanques llenan usando el pozo?

R: Solo dos tanques

¿Que problemas han venido presentando con los cultivos?

R: El tanque cerca del pozo es de mayor capacidad. Por lo que, no hay mucho problema a la hora de regar los cultivos con el agua que demandan, pero hay un tanque cerca del cultivo de café que es el que se utiliza para regar todas las plantas de café pero como es de menor capacidad no hay para regar todas las plantas y si lo queremos volver a llenar tenemos que esperar que el primer tanque se llene porque es el que está más cerca del pozo.

¿Cuántas plantas de café tienen?

R: Hasta el momento hay 703 plantas y vamos a seguir sembrando.

¿Que tipo de tierra estan sembrando el café?

R: Es lima arcillosa

¿Que han hecho para mejorar el riego?

R: Regar por la mañana y en la noche. Las plantas que no se riegan con mangueras las tenemos que regar con baldes.

¿Que técnicas utilizan para el riego?

R: El sistema que tenemos aquí es por goteo

¿Cuanto es la producción de café por temporada?

R: Debido al tiempo y por falta de suficiente agua a las plantas no recogemos mucho.

¿Como está distribuido la tubería?

R: Se llena primero el tanque cerca del pozo, luego este que está cerca del cultivo y para regar el plantío solo abrimos llaves de paso.

¿Han pensado en instalar un nuevo sistema de riego?

R: Ya habían instalado un sistema que cuando se llenara el segundo tanque apagara la bomba, pero este sistema ya no está funcionando y hasta el momento no se ha pensado en instalar otro sistema para el riego del café.

¿Qué pasó con ese sistema?

R: Ya no está funcionando

¿Han pensado en repararlo?

R: No

¿Que enfermedades ha presentado las plantas?

R: Algunas como el ojo de gallo entre otras.

¿El grupo está preparado para tratar este tipo de enfermedades?

R: si

ANEXO D. HOJA DE DATOS

D.1. Hoja de dato del sensor ultrasónico



Tech Support: services@elecfreaks.com

Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal,
- (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- (3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time×velocity of sound (340M/S) / 2,

Wire connecting direct as following:

5V Supply
Trigger Pulse Input
Echo Pulse Output
0V Ground

Electric Parameter

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse

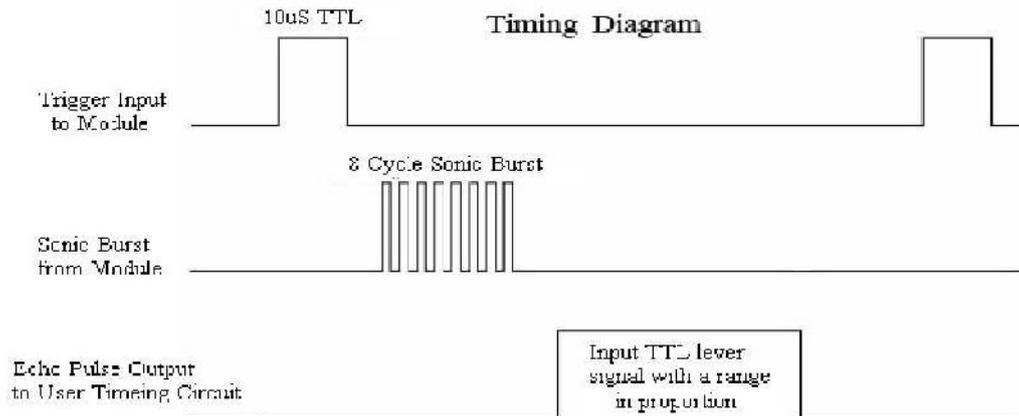
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm



Vcc Trig Echo GND

Timing diagram

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion .You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: $\mu\text{S} / 58 = \text{centimeters}$ or $\mu\text{S} / 148 = \text{inch}$; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



Attention:

The module is not suggested to connect directly to electric, if connected electric, the GND terminal should be connected the module first, otherwise, it will affect the normal work of the module.

When tested objects, the range of area is not less than 0.5 square meters and the plane requests as smooth as possible, otherwise, it will affect the results of measuring.

www.Electfreaks.com



D.2. Hoja de dato del teclado matricial 4x4

4x4 Matrix Membrane Keypad (#27899)

Office: (916) 624-8333
Fax: (916) 624-8003
Sales: (888) 512-1024
Tech Support: (888) 997-8267

This 16-button keypad provides a useful human interface component for microcontroller projects. Convenient adhesive backing provides a simple way to mount the keypad in a variety of applications.

Features

-

Ultra-thin design

- Adhesive backing
- Excellent price/performance ratio
- Easy interface to any microcontroller
- Example programs provided for the BASIC Stamp 2 and Propeller P8X32A microcontrollers

Key Specifications

-

Maximum Rating: 24 VDC, 30 mA

- Interface: 8-pin access to 4x4 matrix
- Operating temperature: 32 to 122 °F (0 to 50°C)
- Dimensions:

Keypad: 2.7 x 3.0 in (6.9 x 7.6 cm)

Cable: 0.78 x 3.5 in (2.0 x 8.8 cm)

Application Ideas

-

Security systems

- Menu selection
- Data entry for embedded systems

Copyright © Parallax Inc. 4x4 Matrix Membrane Keypad (#27899) v1.2 12/16/2011 Page 1 of 5

How it Works

Matrix keypads use a combination of four rows and four columns to provide button states to the host device, typically a microcontroller. Underneath each key is a pushbutton, with one end connected to one row, and the other end connected to one column. These connections are shown in Figure 1.



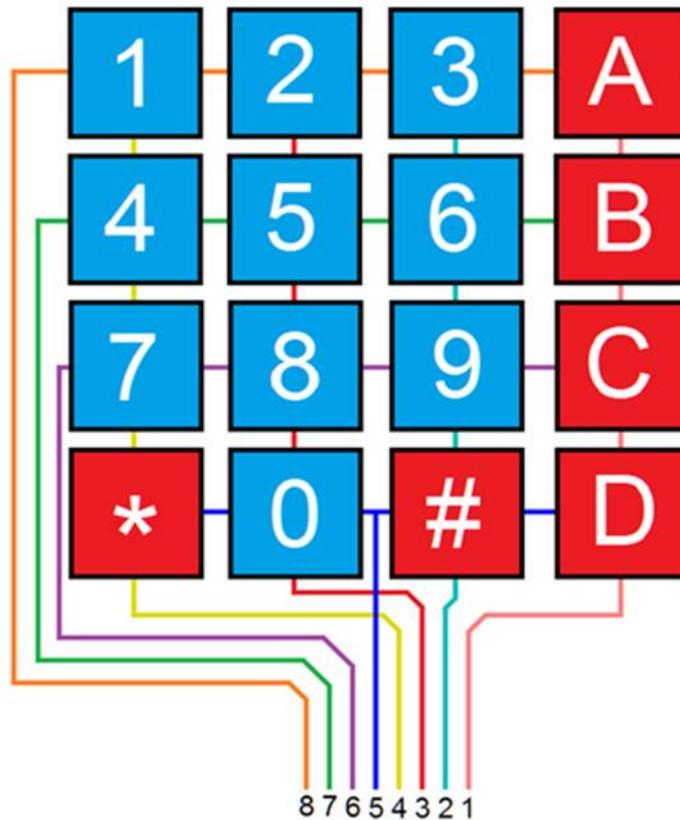


Figure 1: Matrix Keypad Connections

In order for the microcontroller to determine which button is pressed, it first needs to pull each of the four columns (pins 1-4) either low or high one at a time, and then poll the states of the four rows (pins 5-8). Depending on the states of the columns, the microcontroller can tell which button is pressed.

For example, say your program pulls all four columns low and then pulls the first row high. It then reads the input states of each column, and reads pin 1 high. This means that a contact has been made between column 4 and row 1, so button 'A' has been pressed.

Copyright © Parallax Inc. 4x4 Matrix Membrane Keypad (#27899) v1.2 12/16/2011 Page 2 of 5

Connection Diagrams

Figure 2

For use with the BASIC Stamp example program listed below.

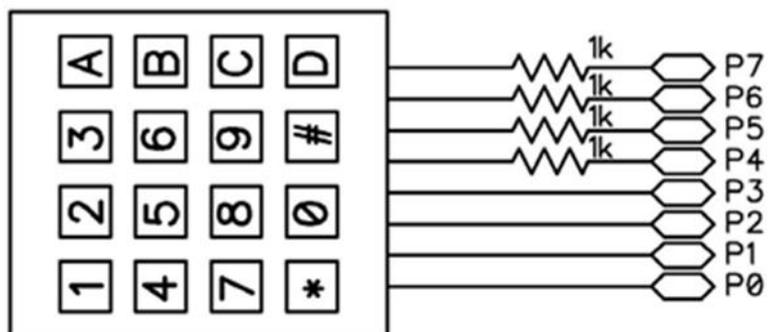
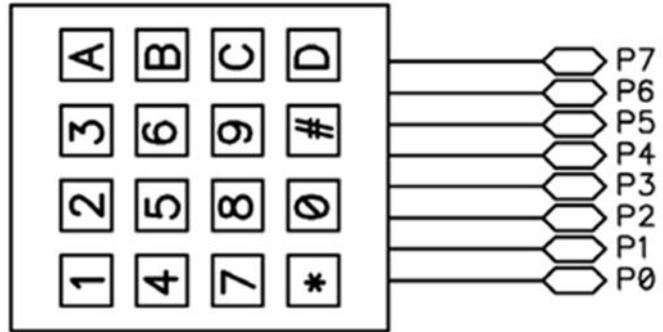


Figure 3

For use with the Propeller P8X32A example program listed below.



BASIC Stamp® Example Code

The example code below displays the button states of the 4x4 Matrix Membrane Keypad. It uses the Debug Terminal, which is built into the BASIC Stamp Editor software. The software is a free download from www.parallax.com/basicstampsoftware.

```
' 4x4MatrixKeypad_Demo.bs2
' Display buttons pressed on the 4x4 Matrix Membrane Keypad
' Author: Parallax HK Engineering

' {$STAMP BS2}
' {$PBASIC 2.5}

row          VAR  Nib          ' Variable space for row counting
column      VAR  Nib          ' Variable space for column counting
keypad      VAR  Word         ' Variable space to store keypad
output
keypadOld   VAR  Word         ' Variable space to store old keypad
output
temp        VAR  Nib          ' Variable space for polling column
states

DEBUG CLS          ' Clear Debug Terminal
GOSUB Update      ' Display keypad graphic

DO
  GOSUB ReadKeypad      ' Read keypad button states
  DEBUG HOME, BIN16 keypad, CR, CR, ' Display 16-bit keypad value
  BIN4 keypad >> 12, CR, ' Display 1st row 4-bit keypad value
  BIN4 keypad >> 8, CR, ' Display 2nd row 4-bit keypad value
  BIN4 keypad >> 4, CR, ' Display 3rd row 4-bit keypad value
  BIN4 keypad ' Display 4th row 4-bit keypad value

Copyright © Parallax Inc. 4x4 Matrix Membrane Keypad (#27899) v1.2 12/16/2011 Page 3 of 5

  IF keypad <> keypadOld THEN ' If different button is pressed,
    GOSUB Update ' update the keypad graphic to clear
  ENDIF ' old display

  IF keypad THEN ' Display button pressed in graphic
    GOSUB display
  ENDIF

  keypadOld = keypad ' Store keypad value in variable
keypadOld
LOOP

' -----[ Subroutine - ReadKeypad ]-----
' Read keypad button states
ReadKeypad:
  keypad = 0
```

```

OUTL  = %00000000      ' Initialize IO
DIRL  = %00000000

FOR row = 0 TO 3
  DIRB = %1111        ' Set columns (P7-P4) as outputs
  OUTB = %0000        ' Pull columns low (act as pull
down)
  OUTA = 1 << row     ' Set rows high one by one
  DIRA = 1 << row

  temp = 0            ' Reset temp variable to 0
  FOR column = 0 TO 3
    INPUT (column + 4) ' Set columns as inputs
    temp = temp | (INB & (1 << column)) ' Poll column state and store in
temp
  NEXT
  keypad = keypad << 4 | (Temp REV 4) ' Store keypad value
NEXT
RETURN

' -----[ Subroutine - Update ]-----
'
' Graphical depiction of keypad
Update:
DEBUG CRSRXY,0,7,
"+---+---+---+---+",CR,
"|   |   |   |   |",CR,
"+---+---+---+---+"
RETURN

' -----[ Subroutine - Display ]-----
'
' Display button pressed in keypad graphic
Display:
IF KeyPad.BIT15 THEN  DEBUG CRSRXY, 02,08,"1"
IF KeyPad.BIT14 THEN  DEBUG CRSRXY, 06,08,"2"
IF KeyPad.BIT13 THEN  DEBUG CRSRXY, 10,08,"3"
IF KeyPad.BIT12 THEN  DEBUG CRSRXY, 14,08,"A"
IF KeyPad.BIT11 THEN  DEBUG CRSRXY, 02,10,"4"
IF KeyPad.BIT10 THEN  DEBUG CRSRXY, 06,10,"5"
IF KeyPad.BIT9  THEN  DEBUG CRSRXY, 10,10,"6"
IF KeyPad.BIT8  THEN  DEBUG CRSRXY, 14,10,"B"
IF KeyPad.BIT7  THEN  DEBUG CRSRXY, 02,12,"7"
IF KeyPad.BIT6  THEN  DEBUG CRSRXY, 06,12,"8"
IF KeyPad.BIT5  THEN  DEBUG CRSRXY, 10,12,"9"
IF KeyPad.BIT4  THEN  DEBUG CRSRXY, 14,12,"C"
IF KeyPad.BIT3  THEN  DEBUG CRSRXY, 02,14,"*"
IF KeyPad.BIT2  THEN  DEBUG CRSRXY, 06,14,"0"
IF KeyPad.BIT1  THEN  DEBUG CRSRXY, 10,14,"#"
IF KeyPad.BIT0  THEN  DEBUG CRSRXY, 14,14,"D"
RETURN

```

Copyright © Parallax Inc. 4x4 Matrix Membrane Keypad (#27899) v1.2 12/16/2011 Page 4 of 5

Propeller™ P8X32A Example Code

The example code below displays the button states of the 4x4 Matrix Membrane Keypad, and is a modified version of the 4x4 Keypad Reader DEMO object by Beau Schwabe.

Note: This application uses the 4x4 Keypad Reader.spin object. It also uses the Parallax Serial Terminal to display the device output. Both objects and the Parallax Serial Terminal itself are included with the with the Propeller Tool v1.2.7 or higher, which is available from the Downloads link at www.parallax.com/Propeller.

```
{{ 4x4 Keypad Reader PST.spin Returns the entire 4x4 keypad matrix into a single WORD variable indicating which buttons are pressed. }}
```

CON

```
_clkmode = xtal1 + pll16x  
_xinfreq = 5_000_000
```

OBJ

```
text : "Parallax Serial Terminal"  
KP : "4x4 Keypad Reader"
```

VAR

```
word keypad
```

PUB start

```
'start term  
text.start(115200)  
text.str(string(13,"4x4 Keypad Demo..."))  
text.position(1, 7)  
text.str(string(13,"RAW keypad value 'word'"))
```

```
text.position(1, 13)  
text.str(string(13,"Note: Try pressing multiple keys"))
```

repeat

```
keypad := KP.ReadKeyPad ' <-- One line command to read the 4x4 keypad  
text.position(5, 2)  
text.bin(keypad>>0, 4) 'Display 1st ROW  
text.position(5,3)  
text.bin(keypad>>4, 4) 'Display 2nd ROW  
text.position(5, 4)  
text.bin(keypad>>8, 4) 'Display 3rd ROW  
text.position(5, 5)  
text.bin(keypad>>12, 4) 'Display 4th ROW  
text.position(5, 9)  
text.bin(keypad, 16) 'Display RAW keypad value
```

Revision History

v1.0: original document

v1.1: Updated Figure 1 on page 2

v1.2: Updated Figure 1 on page 2 (again); updated BS2 comments

Copyright © Parallax Inc. 4x4 Matrix Membrane Keypad (#27899) v1.2 12/16/2011 Page 5 of 5

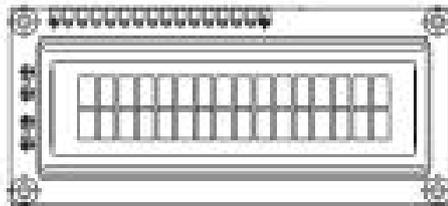
D.4. Hoja de dato de pantalla lcd 16x4



LCD-016M002B

Vishay

16 x 2 Character LCD



FEATURES

- 5 x 8 dots with cursor
- Built-in controller (KS 0066 or Equivalent)
- + 5V power supply (Also available for + 3V)
- 1/16 duty cycle
- EL to be driven by pin 1, pin 2 or pin 15, pin 16 or A-K (LED)
- N.V. optional for + 3V power supply

MECHANICAL DATA		
ITEM	STANDARD VALUE	UNIT
Module Dimension	83.0 x 38.0	mm
Viewing Area	68.0 x 18.0	mm
Dot Size	0.56 x 0.66	mm
Character Size	2.95 x 5.56	mm

ABSOLUTE MAXIMUM RATING					
ITEM	SYMBOL	STANDARD VALUE			UNIT
		MIN.	TYR.	MAX.	
Power Supply	VDD-VSS	-0.3	-	7.0	V
Input Voltage	VI	-0.3	-	VDD	V

NOTE: VSS = 0 Volt, VDD = 5.0 Volt

ELECTRICAL SPECIFICATIONS							
ITEM	SYMBOL	CONDITION	STANDARD VALUE			UNIT	
			MIN.	TYR.	MAX.		
Input Voltage	VDD	VDD = + 3V	4.7	5.0	5.5	V	
		VDD = + 5V	2.7	3.0	3.3	V	
Supply Current	IDC	VDD = 5V	-	1.2	3.0	mA	
Recommended LC Driving Voltage for Normal Temp. Version Module	VDD - VO	-25 °C	-	-	-	V	
		0 °C	4.2	4.8	5.1		
		25 °C	3.8	4.2	4.6		
		50 °C	3.6	4.0	4.4		
LED Forward Voltage	VF	25 °C	-	4.2	4.6	V	
LED Forward Current	IF	25 °C	Array	-	100	200	mA
			Edge	-	20	40	
EL Power Supply Current	IEL	VI = 110VAC-600Hz	-	-	3.0	mA	

DISPLAY CHARACTER ADDRESS CODE:																
Display Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	00	01														0F
DD RAM Address	40	41														4F

LCD-016M002B

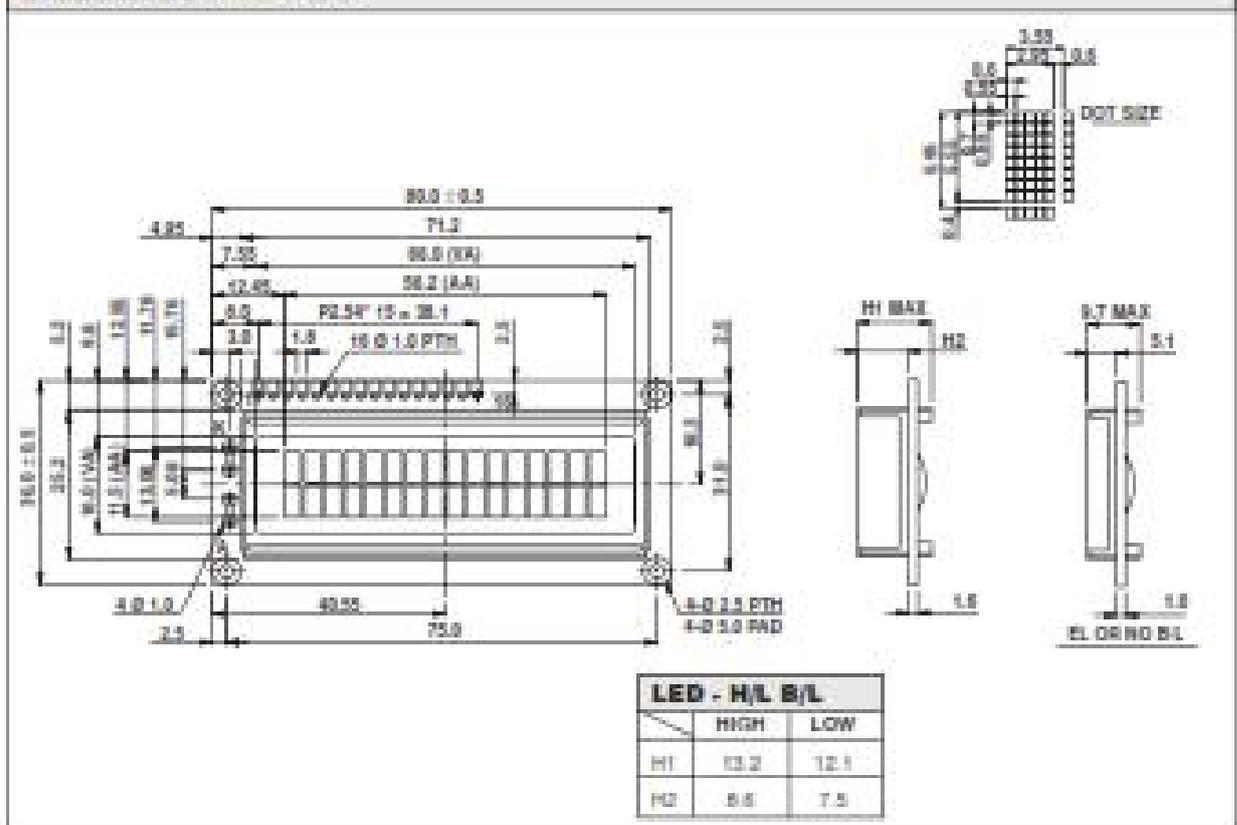


Vishay

16 x 2 Character LCD

PIN NUMBER	SYMBOL	FUNCTION
1	Vss	GND
2	Vdd	+3V or +5V
3	Vc	Contrast Adjustment
4	RS	HL Register Select Signal
5	RW	HL Read/Write Signal
6	E	H -L Enable Signal
7	DB0	HL Data Bus Line
8	DB1	HL Data Bus Line
9	DB2	HL Data Bus Line
10	DB3	HL Data Bus Line
11	DB4	HL Data Bus Line
12	DB5	HL Data Bus Line
13	DB6	HL Data Bus Line
14	DB7	HL Data Bus Line
15	A/Vss	+4.2V for LCD/Negative Voltage Output
16	K	Power Supply for B/L (CV)

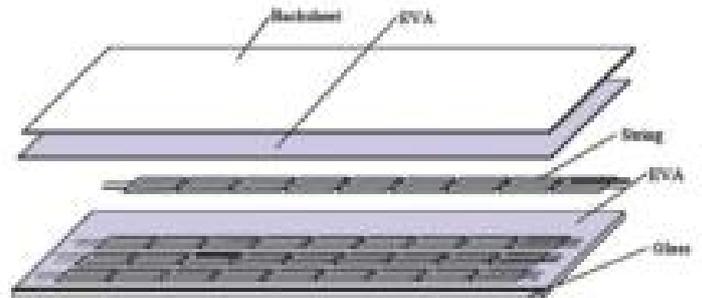
DIMENSIONS in millimeters



D.5 Hoja de dato del panel de 150w

We focus on precise production and provide Customer with best cost-effective products to create Market Value for Customer

CHN 36P 150W/155W/160W



Power tolerance (0→+3%) to ensure the high reliability of power output



Modules certified by TÜV to with stand high level of wind loads (2400 Pa) and snow loads(5400 Pa)



Special PV Module Insurance by world leading insurance company guarantees the benefit of PV Investors and PV module users



Junction box and bypass diodes guarantees the modules free of overheating and Hot spot effect

Guaranteed Performance**

10 Years Manufacturing Warranty

12 Years Warranty 90% Power Output

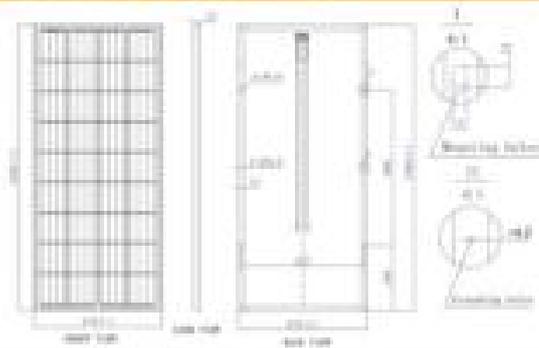
25 Years Warranty 80% Power Output

*Free module recycling through membership in the Green Cycle Association

choosing chinland solar

- 👉 The most vertically integrated solar manufacturer in the industry with production of ingots,wafers,solar cells and modules using both mono crystalline and multi crystalline technology.
- 👉 Global distribution with local warehousing,delivery and after sales services.
- 👉 Modules certified by global testing facilities: IEC61215, IEC61730, CE, ROHS.
- 👉 Manufacturing with international quality standards and environment management system ISO9001 and ISO14001.
- 👉 Most updated design with drainage holes in the frame ensures the modules to withstand various weather conditions.
- 👉 Minimal wiring effort required as the module has high reverse current resistance.

MECHANICAL DRAWINGS



MECHANICAL SPECIFICATION

Cell Type	Poly Crystalline 156x156mm
Number of cells	60(6x10)
Dimensions(WxHxD)	1480x670x35mm
Weight	13kg
Front Glass	3.2mm tempered low iron glass
Frame	Clear anodized aluminum alloy
Junction Box	IP65, with bypass diodes
Connector	MC4 compatible
Output Cable	TUV, length 900mm, 4.0mm ²

ELECTRICAL CHARACTERISTICS

	150W	150W	160W
Maximum Power at STC(P _{max})	150W	150W	160W
Short Circuit Current(I _{sc})	8.71A	8.89A	8.91A
Open Circuit Voltage(V _{oc})	21.8V	22.0V	22.3V
Maximum Power Current(I _{mp})	8.17A	8.34A	8.51A
Maximum Power Voltage(V _{mp})	18.4V	18.2V	18.8V
Module Efficiency	15.13%	15.63%	16.14%
Power Tolerance	0~+3%	0~+3%	0~+3%

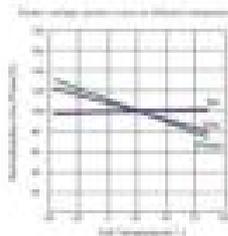
STC: irradiance: 1000W/m² Module temperature: 25°C AM1.5

TEMPERATURE CHARACTERISTICS

Nominal Operating Cell Temperature (NOCT)	45°C ±2°C
Temperature Coefficient of P _{max}	-0.44%/°C
Temperature Coefficient of V _{oc}	-0.34%/°C
Temperature Coefficient of I _{sc}	0.06%/°C

PACKING CONFIGURATION

Container	20'GP	40'GP
Pieces per pallet	30	30
Pallets per container	14	30
Pieces per container	420	1050



SYSTEM INTEGRATION PARAMETERS

Maximum system voltage	VDC 1000V
Maximum Series Fuse	15A
Increased snowload acc. to IEC 61215	5400Pa
Operating Temperature	-40to+85°C
Number of bypass diodes	3

WARRANTY PERCENTAGE BASED ON NOMINAL POWER



- Based on nominal power(P_{nom})
- 25-year transferable power output warranty: 5 years/95%, 12 years/90%, 18 years/85%, 25 years/80%

QUALIFICATIONS AND CERTIFICATES



D.6. Hoja de dato del controlador ls0512e



PWM Solar charge controller

LS-E series



LS-E economic solar controller is new member of LandStar family and ideal for small solar system especially home system. It has the beautiful appearance, and simple operation. And the industrial grade design ensures its reliability.



Model overview:

LS0512E 5A,12V

LS1012E 10A,12V

LS1024E 10A,12/24V

LS2024E 20A,12/24V





Technical specifications

Model	LS0012E	LS1012E	LS1024E	LS2024E
Nominal system voltage	12VDC		12/24VDC	
Rated battery current	5A	10A	10A	20A
Max. battery voltage	16V		32V	
Max. PV open circuit voltage	30V		50V	
Equalize charging voltage	Gel: ---, Sealed: 14.0V, Flooded: 14.0V			
Boost charging voltage	Gel: 14.2V, Sealed: 14.4V, Flooded: 14.6V			
Float charging voltage	Gel / Sealed / Flooded: 13.5V			
Low voltage reconnect voltage	Gel / Sealed / Flooded: 12.6V			
Low voltage disconnect voltage	Gel / Sealed / Flooded: 11.1V			
Charge circuit voltage drop	≤0.20V			
Discharge circuit voltage drop	≤0.15V			
Self-consumption	≤6mA			
Temp. compensation	-5mV/°C/2V (ref)			
Overall dimension(mm)	92.6x65x20.2	101.2x67x21.6	101.2x67x21.6	126x65.6x34.6
Mounting dimension	64.4mm	92.7mm	92.7mm	110mm
Mounting hole size	Φ4.5	Φ4.5	Φ4.5	Φ4.5
Terminal	2.5mm ²	4mm ²	4mm ²	6mm ²
Weight	74g	62.5g	62g	151.6g
Working temperature	-30°C ~ +55°C			
Storage temperature	-30°C ~ +50°C			
Humidity	≤95% N.C.			
Enclosure	IP30			



BEIJING EPSOLAR TECHNOLOGY CO.,LTD.

Add: BLDG #16, CO PARK, NO.6 HEYING ROAD, CHANGPING DISTRICT, BEIJING, CHINA
 Tel: 010-62664962 / 62694112
 Fax: 010-62694862

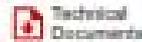
D.7. Hoja de dato del sensor de temperatura lm35



Product Folder



Sample & Buy



Technical Documents



Tools & Software



Support & Community



LM35

SNVS159G – AUGUST 1999 – REVISED AUGUST 2016

LM35 Precision Centigrade Temperature Sensors

1 Features

- Calibrated Directly in Celsius (Centigrade)
- Linear + 10-mV/°C Scale Factor
- 0.5°C Ensured Accuracy (at 25°C)
- Rated for Full -55°C to 150°C Range
- Suitable for Remote Applications
- Low-Cost Due to Wafer-Level Trimming
- Operates from 4 V to 30 V
- Less than 60-µA Current Drain
- Low Self-Heating, 0.06°C in Still Air
- Non-Linearity Only ±¼°C Typical
- Low-Impedance Output, 0.1 Ω for 1-mA Load

2 Applications

- Power Supplies
- Battery Management
- HVAC
- Appliances

3 Description

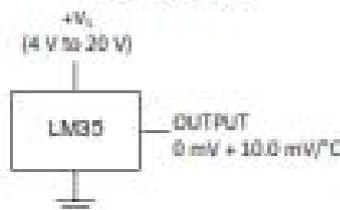
The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly-proportional to the Centigrade temperature. The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±½°C over a full -55°C to 150°C temperature range. Lower cost is assured by trimming and calibration at the wafer level. The low-output impedance, linear output, and precise inherent calibration of the LM35 device makes interfacing to readout or control circuitry especially easy. The device is used with single power supplies, or with plus and minus supplies. As the LM35 device draws only 60 µA from the supply, it has very low self-heating of less than 0.1°C in still air. The LM35 device is rated to operate over a -55°C to 150°C temperature range, while the LM35C device is rated for a -40°C to 110°C range (-10° with improved accuracy). The LM35-series devices are available packaged in hermetic TO transistor packages, while the LM35C, LM35CA, and LM35D devices are available in the plastic TO-92 transistor package. The LM35D device is available in an 8-lead surface-mount small-outline package and a plastic TO-220 package.

Device Information⁽¹⁾

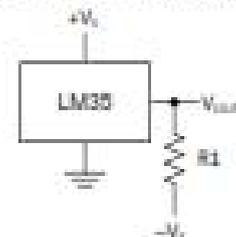
PART NUMBER	PACKAGE	BODY SIZE (NOM)
LM35	TO-CAN (3)	4.699 mm × 4.699 mm
	TO-92 (3)	4.30 mm × 4.30 mm
	SO8C (8)	4.90 mm × 3.91 mm
	TO-220 (3)	14.965 mm × 10.16 mm

(1) For all available packages, see the orderable addendum at the end of the datasheet.

Basic Centigrade Temperature Sensor (2°C to 150°C)



Full-Range Centigrade Temperature Sensor



Choose $R_L = -V_{S-} / 50 \mu\text{A}$
 $V_{OUT} = 1500 \text{ mV at } 150^\circ\text{C}$
 $V_{OUT} = 250 \text{ mV at } 25^\circ\text{C}$
 $V_{OUT} = -550 \text{ mV at } -55^\circ\text{C}$

6.5 Electrical Characteristics: LM35A, LM35CA Limits

Unless otherwise noted, these specifications apply: $-55^{\circ}\text{C} \leq T_J \leq 150^{\circ}\text{C}$ for the LM35 and LM35A; $-40^{\circ}\text{C} \leq T_J \leq 110^{\circ}\text{C}$ for the LM35C and LM35CA; and $0^{\circ}\text{C} \leq T_J \leq 100^{\circ}\text{C}$ for the LM35D. $V_S = 5\text{ Vdc}$ and $I_{\text{LOAD}} = 50\ \mu\text{A}$, in the circuit of [Full-Range Centigrade Temperature Sensor](#). These specifications also apply from 2°C to T_{MAX} in the circuit of [Figure 14](#).

PARAMETER	TEST CONDITIONS	LM35A			LM35CA			UNIT
		TYP	TESTED LIMIT ⁽¹⁾	DESIGN LIMIT ⁽²⁾	TYP	TESTED LIMIT ⁽¹⁾	DESIGN LIMIT ⁽²⁾	
Accuracy ⁽³⁾	$T_A = 25^{\circ}\text{C}$	± 0.2	± 0.5		± 0.2	± 0.5		$^{\circ}\text{C}$
	$T_A = -10^{\circ}\text{C}$	± 0.3			± 0.3		± 1	
	$T_A = T_{\text{MAX}}$	± 0.4	± 1		± 0.4	± 1		
	$T_A = T_{\text{MIN}}$	± 0.4	± 1		± 0.4		± 1.5	
Nonlinearity ⁽⁴⁾	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$, $-40^{\circ}\text{C} \leq T_J \leq 125^{\circ}\text{C}$	± 0.16		± 0.35	± 0.15		± 0.3	$^{\circ}\text{C}$
Sensor gain (average slope)	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	10	9.9		10		9.9	$\text{mV}/^{\circ}\text{C}$
	$-40^{\circ}\text{C} \leq T_J \leq 125^{\circ}\text{C}$	10	10.1		10		10.1	
Load regulation ⁽⁵⁾ $0 \leq I_L \leq 1\ \text{mA}$	$T_A = 25^{\circ}\text{C}$	± 0.4	± 1		± 0.4	± 1		mV/mA
	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$, $-40^{\circ}\text{C} \leq T_J \leq 125^{\circ}\text{C}$	± 0.5		± 3	± 0.5		± 3	
Line regulation ⁽⁵⁾	$T_A = 25^{\circ}\text{C}$	± 0.01	± 0.05		± 0.01	± 0.05		mV/V
	$4\ \text{V} \leq V_S \leq 30\ \text{V}$, $-40^{\circ}\text{C} \leq T_J \leq 125^{\circ}\text{C}$	± 0.02		± 0.1	± 0.02		± 0.1	
Quiescent current ⁽⁶⁾	$V_S = 5\ \text{V}$, 25°C	56	67		56	67		μA
	$V_S = 5\ \text{V}$, $-40^{\circ}\text{C} \leq T_J \leq 125^{\circ}\text{C}$	105		131	91		114	
	$V_S = 30\ \text{V}$, 25°C	56.2	66		56.2	66		
	$V_S = 30\ \text{V}$, $-40^{\circ}\text{C} \leq T_J \leq 125^{\circ}\text{C}$	105.5		133	91.5		116	
Change of quiescent current ⁽⁵⁾	$4\ \text{V} \leq V_S \leq 30\ \text{V}$, 25°C	0.2	1		0.2	1		μA
	$4\ \text{V} \leq V_S \leq 30\ \text{V}$, $-40^{\circ}\text{C} \leq T_J \leq 125^{\circ}\text{C}$	0.5		2	0.5		2	
Temperature coefficient of quiescent current	$-40^{\circ}\text{C} \leq T_J \leq 125^{\circ}\text{C}$	0.39		0.5	0.39		0.5	$\mu\text{A}/^{\circ}\text{C}$
Minimum temperature for rate accuracy	In circuit of Figure 14 , $I_L = 0$	1.5		2	1.5		2	$^{\circ}\text{C}$
Long term stability	$T_J = T_{\text{MAX}}$ for 1000 hours	± 0.05			± 0.05			$^{\circ}\text{C}$

(1) Tested Limits are ensured and 100% tested in production.

(2) Design Limits are ensured (but not 100% production tested) over the indicated temperature and supply voltage ranges. These limits are not used to calculate outgoing quality levels.

(3) Accuracy is defined as the error between the output voltage and $10\ \text{mV}/^{\circ}\text{C}$ times the case temperature of the device, at specified conditions of voltage, current, and temperature (expressed in $^{\circ}\text{C}$).

(4) Non-linearity is defined as the deviation of the output-voltage-versus-temperature curve from the best-fit straight line, over the rated temperature range of the device.

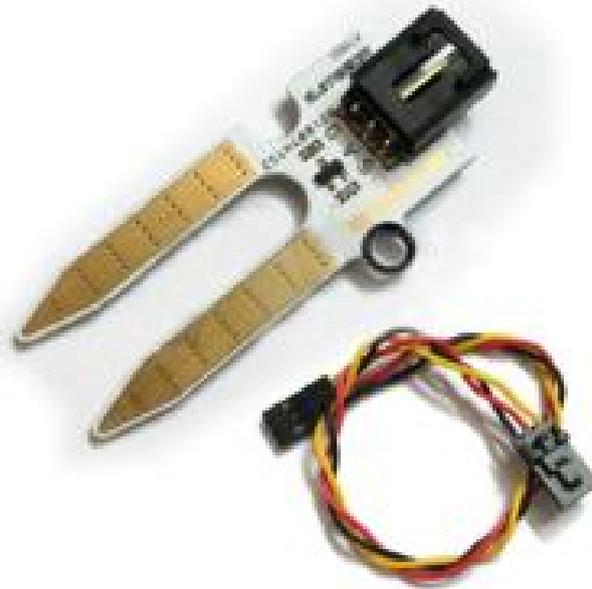
(5) Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the internal dissipation by the thermal resistance.

(6) Quiescent current is defined in the circuit of [Figure 14](#).

D.8. Hoja de dato del sensor de humedad YI-69



OBSoil-01 Octopus Soil Moisture Sensor Brick



Octopus Soil Moisture Sensor Brick can read the amount of moisture present in the soil surrounding it. Ideal for monitoring an urban garden, or your pet plant's water level. This is a must have tool for a connected garden!

This sensor uses the two probes to pass current through the soil, and then it reads that resistance to get the moisture level. More water makes the soil conduct electricity more easily (less resistance), while dry soil conducts electricity poorly (more resistance).

This sensor isn't hardened against contamination or exposure of the control circuitry to water and may be prone to electrolytic corrosion across the probes (Also it can be switched on, take the reading and switched off to minimize electrolytic corrosion), so it isn't well suited to being left in place or used outdoors.

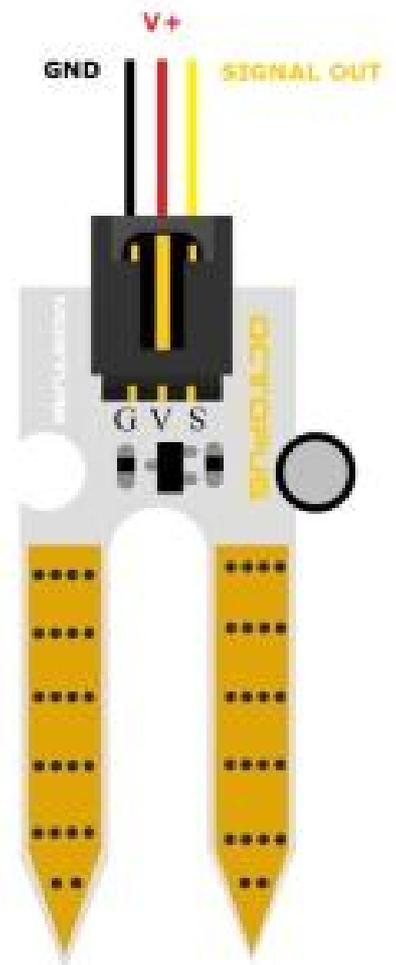
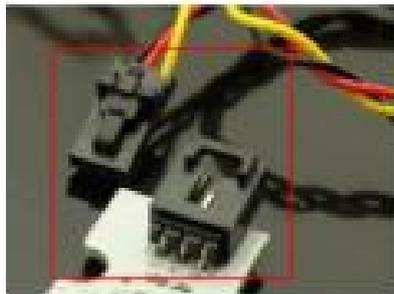
Specification

Item	Condition	Min	Typical	Max	Unit
Voltage	-	3.3	/	5	V
Current	-	0	/	35	mA
Output Voltage	Supply Voltage 5 V	0	∞	4.2	V
Output Value	Sensor in dry soil	0	∞	300	/
	Sensor in humid soil	300	∞	700	/
	Sensor in water	700	∞	950	/

Pinout:

- **G:** GND (Black wire)
- **V:** Voltage Supply (+) (Red wire)
- **S:** Signal out (Yellow wire)

3P buckled wires connector and cable included:



Applications

- Moisture sensing
- Botanical gardening
- Flood detection
- Liquid level detection

D.9. Hoja de dato del panel solar 50W



50 Watt Poly -Crystalline Solar Module

EGE-50P

Features

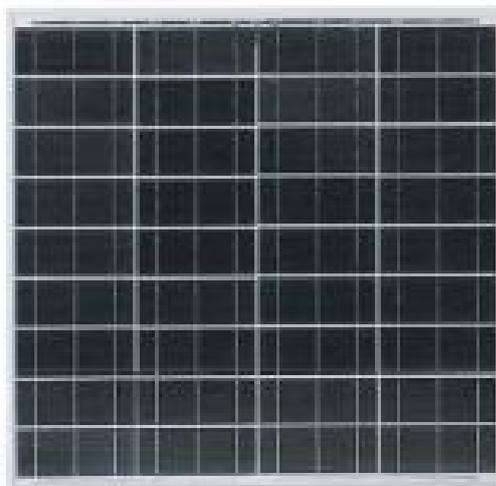
- Strong frame with higher resistance to snow and wind (mechanical load test of 5400Pa passed, instead of 2400Pa)
- High reliability with guaranteed $\pm 5\%$ power output tolerance, ensuring return on investment
- Specification can be customized

Quality and Safety

- Rigorous quality control meeting the highest international standards
- High-transmissivity low-iron 3.2mm tempered glass, extremely strong aluminium frame
- Secure embedding of sensitive glass edge using UV-resistant silicon
- IEC 61215 Certified / ISO 9001:2000 and ISO 14001:2004 certified
- Conformity to CE

Warranties

- 2 years limited product warranty
- 10 years at 80% of the minimal rated power output
- 25 years at 80% of the minimal rated power output





EGE

Electrical Characteristics

Maximum Power at STC (P _{max})	50W
Optimum Operating Voltage (V _{mp})	17.5V
Optimum Operating Current (I _{mp})	3.11A
Open-Circuit Voltage (V _{oc})	21.50V
Short-Circuit Current (I _{sc})	3.31A
Operating Temperature	-40 to 55 °C
Maximum System Voltage	DC1000
Maximum Series Fuse Rating	15A
Power Tolerance	+/-3%

Mechanical Characteristics

Dimensions	740mm*510mm*35mm
Front glass	3.2mm tempered glass
Frame	Anodized aluminium alloy
Junction box	PV junction box IP65

Temperature Coefficients

Nominal Operating Cell Temperature (NOCT)	47 °C +/-2 °C
Temperature Coefficient of P _{max}	-0.48%/°C
Temperature Coefficient of V _{OC}	-0.34V/°C
Temperature Coefficient of I _{SC}	+0.037%/°C