

Instituto Tecnológico de Tuxtla Gutiérrez

“Diseño de un medidor de niveles de
agua para estimar la evaporación
(Ev)”

ASESOR

Dr. Héctor Ricardo Hernández de León

REVISORES

Ing. Aldo E. Aguilar Castillejos

M. C. Osvaldo Brindis Velázquez

ALUMNOS

Juan Gómez Santiz

José Joaquín Díaz Ramírez

**INGENIERIA ELECTRONICA
RESIDENCIA PROFESIONAL**

Contenido

CAPÍTULO 1. GENERALIDADES	2
1.1 Introducción	2
1.2 Información de la institución donde se desarrolló el proyecto	3
1.2.1 Historia del ITTG	3
1.2.2 Misión	4
1.2.3 Visión	4
1.2.4 Localización	4
1.2.5 Área específica relacionada directamente con el proyecto	5
1.3 Antecedentes	5
1.4 Planteamiento del problema	6
1.5 Nombre del proyecto	6
1.6 Objetivos	6
1.6.1 Objetivo general	6
1.6.2 Objetivos específicos	6
1.7 Justificación del proyecto.	7
1.8 Delimitación del tema	7
1.9 Metodología para el desarrollo del proyecto.	8
CAPÍTULO 2. FUNDAMENTO TEÓRICO	9
2.1 Tanque evaporímetro clase A	9
2.1.1 Evapotranspiración (ETo) estimada con el tanque de evaporación	10
2.2 Variables medidas	11
2.2.1 Evaporación	11
2.3 Instrumentos de medición	12
2.4 Programas de apoyo	12
2.4.1 App Inventor	12
2.4.2 LabView	14
2.4.3 Utilizando LabVIEW	14
2.4.4 LabVIEW para el Medio Ambiente	15
2.4.5 Propiedad de los componentes	16
2.5 Arduino MEGA 2560	22
2.5.1 Estructura básica de un programa	23
2.5.2 Tipos de Datos	24
2.5.3 Funciones de Tiempo	24

2.5.4 Características físicas y compatibilidad de Shields	25
2.6 Módulo Bluetooth	25
2.6.1 Configuración del módulo Bluetooth	26
2.7 Sensor DTH11	27
2.7.1 Proceso de Comunicación general	28
2.7.2 Señal de inicio de MCU a DHT	29
2.8 Sensor ultrasónico HC-SR04	29
CAPÍTULO 3. DESARROLLO E IMPLEMENTACIÓN DEL PROYECTO	32
3.1 Diseño del sistema de adquisición de datos	32
3.2 Construcción de la etapa de potencia	33
3.2.1 Electroválvula 12 VCD y Electroválvula 120 VCA	33
3.3 Simulación del sistema de adquisición de datos	34
3.4 Implementación del sistema de adquisición de datos	35
3.5 Materialización del circuito medidor de nivel	38
3.6 Interfaz gráfica de LabVIEW	39
CAPÍTULO 4. PRUEBAS Y RESULTADOS	41
4.1 Pruebas realizadas	41
4.2 Resultados obtenidos	42
Conclusiones	45
Referencias	46
Anexos	47
A1. DataSheet, MOC3011	47
A2. Tabla de radiación solar del hemisferio norte a 10° latitud	49
A3. Circuito virtual del medidor de niveles	51
A4. Programa grabado en ATmega2560	52
A5. Programación (configuración) del módulo Bluetooth	54

CAPÍTULO 1. GENERALIDADES

1.1 Introducción

Se conoce como evapotranspiración (ET) a la combinación de dos procesos por separado, por los que el agua se pierde a través de la superficie del suelo por evaporación, y por otra parte por la transpiración del cultivo.

La evaporación es el proceso por el cual el agua líquida se convierte en vapor de agua (vaporización) y se retira de la superficie evaporante (remoción de vapor). El agua se evapora de una variedad de superficies, tales como lagos, ríos, caminos, suelos y la vegetación mojada.

La transpiración consiste en la vaporización del agua líquida contenido en los tejidos de la planta y su posterior remoción hacia la atmósfera. Los cultivos pierden agua predominantemente a través de las estomas. Estos son pequeñas aberturas en la hoja de la planta a través de las cuales atraviesan los gases y el vapor de agua de la planta hacia la atmósfera.

La evaporación y la transpiración ocurren simultáneamente y no hay alguna manera sencilla de distinguir entre estos dos procesos. La evapotranspiración se expresa en milímetros (mm) por unidad de tiempo. Esta medida expresa la cantidad de agua perdida de una superficie cultivada en unidades de altura de agua. La unidad de tiempo puede ser una hora, un día, 10 días, un mes o incluso un completo periodo de cultivo o un año.

Cabe mencionar que, el clima, las características del cultivo, el manejo y el medio de desarrollo son factores que afectan la evaporación y la transpiración.

Para el tanque evaporímetro de tipo o clase "A", se hablarán de sus especificaciones técnicas, su uso experimental y la comparación con las diversas ecuaciones que se usan para calcular la evapotranspiración, y de esta manera, poder tener tanto las pruebas teóricas, como las prácticas.

Posteriormente se abordarán los temas de evapotranspiración, su definición, las distintas expresiones o ecuaciones para el cálculo de ella, y también, uno de los métodos experimentales más usados para el cálculo de esta magnitud; hablamos del tanque evaporímetro, de tipo A, el cual ha sido utilizado en muchas partes del mundo para la medición de la evapotranspiración, y de esta manera aplicar los más precisos criterios al sistema de riego en cultivos.

1.2 Información de la institución donde se desarrolló el proyecto

1.2.1 Historia del ITTG

El Instituto Tecnológico de Tuxtla Gutiérrez es una institución pública dependiente de la Secretaría de Educación Pública. Imparte 8 licenciaturas y 2 programas de posgrado en las áreas de Ingeniería, Tecnología y Ciencias Administrativas.

El Instituto Tecnológico de Tuxtla Gutiérrez fue fundado el 24 de octubre de 1972, se encuentra ubicado en el centro del estado de Chiapas, que por su gran variedad de climas y suelos es propio para el cultivo de muy diversas especies vegetales nativas y adaptadas, con lo que se puede desarrollar la agroindustria, así como muchos otros procesos industriales a partir del gran potencial que ofrece este estado, en el cual el Instituto debe constituirse en el sujeto de cambio, al presentar alternativas de desarrollo sustentable del estado en forma multidisciplinaria.

La matrícula escolar en licenciatura se conforma de 1 982 estudiantes: 1 473 hombres y 509 mujeres. En el posgrado hay 78 alumnos: 68 hombres y 10 mujeres.

La población de egresados en el nivel de licenciatura es de 2 345 alumnos, de ellos 1 892 son hombres y 453 son mujeres.

Los principales laboratorios con los que cuenta el Instituto Tecnológico de Tuxtla Gutiérrez son:

- ✓ Microbiología
- ✓ Biotecnología
- ✓ Química
- ✓ Química pesada
- ✓ Mecánica
- ✓ Sistemas computacionales
- ✓ Ingeniería industrial
- ✓ Plantas piloto
- ✓ Polo Tecnológico Nacional

Proyectos de vinculación exitosos con la industria

- ✓ Aprovechamiento integral del timbre
- ✓ Incremento en el contenido de sacarosa en la caña de azúcar
- ✓ Efectos del boro en la producción del cacahuete
- ✓ Factibilidad técnica en la extracción de aceites esenciales de la flora chiapaneca

1.2.2 Misión

Formar de manera integral profesionistas de excelencia en el campo de la ciencia y la tecnología con actitud emprendedora, respeto al medio ambiente y apego a los valores éticos.

1.2.3 Visión

Ser una Institución de excelencia en la educación superior tecnológica del Sureste, comprometida con el desarrollo socioeconómico sustentable de la región.

1.2.4 Localización

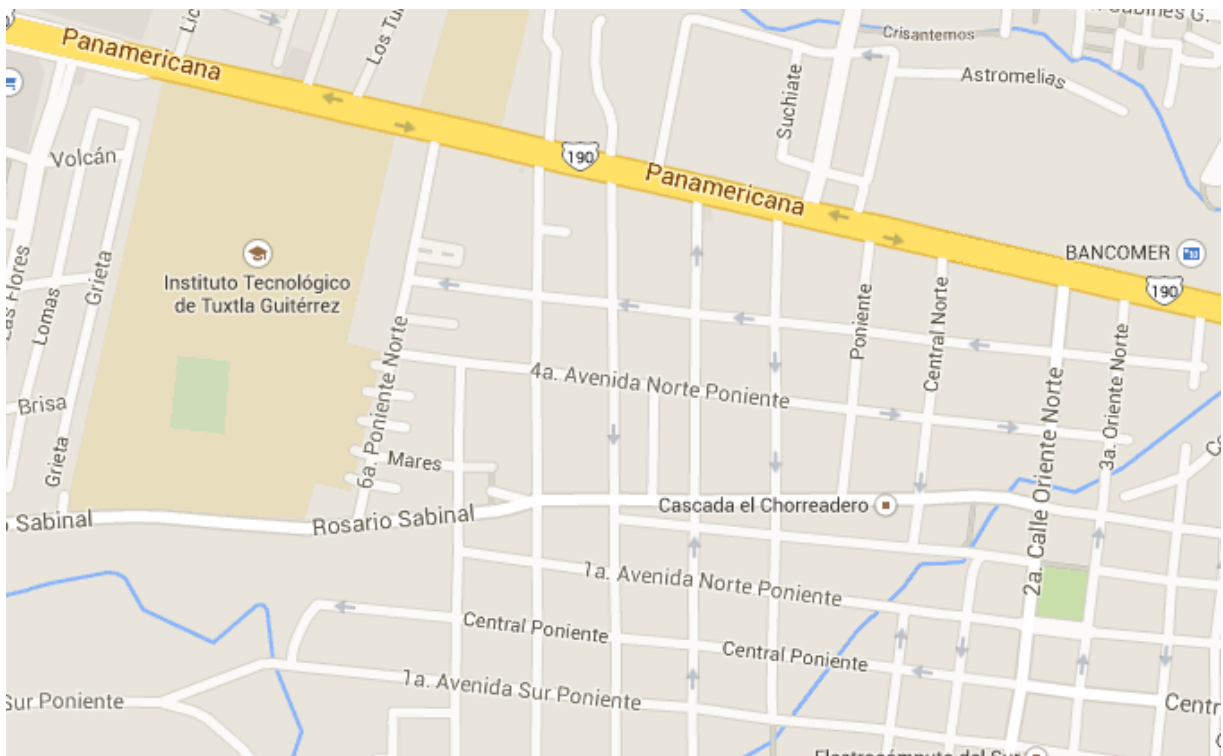


Fig. 1.1 Carretera Panamericana Km 1080. Tuxtla Gutiérrez Chiapas, C.P. 29050.

1.2.5 Área específica relacionada directamente con el proyecto

El laboratorio de Ingeniería Electrónica cuenta con 8 áreas destinadas a la investigación, desarrollo e implementación de proyectos escolares, 5 de las cuales son educativas, en las que los docentes imparten clases de: electrónica digital, PLC, electrónica de potencia, instrumentación, mediciones eléctricas, programación en c y electrónicas analógicas. Dos más son para investigación, una pertenece al grupo de IEEE y la otra es de desarrollo de circuitos electrónicos. La última área está destinada para que los alumnos desarrollen sus actividades en extra clase. Parte del área que se ocupó en la realización de éste proyecto fueron las de IEEE, las de Electrónicas Analógicas y desarrollo de circuitos electrónicos y el almacén de Mecatrónica.

La carrera de Ingeniería Electrónica tiene como misión “formar profesionales de excelencia con competencias en el ámbito de la Ingeniería Electrónica, motivados para la promoción del desarrollo profesional y el conocimiento científico y tecnológico, con actitud emprendedora, respeto al medio ambiente y apego a los valores cívicos y éticos”.

1.3 Antecedentes

En Venezuela se realizó el artículo denominado “Estimación de evapotranspiración de referencia a nivel mensual en Venezuela. ¿Qué método utilizar?”, cuyo autor es Ricardo Trezza, con fecha de publicación, marzo 3 de 2008 en la revista Bioagro.

En *centro interdisciplinario de investigación para el desarrollo integral regional CIIDIR Michoacán* se desarrolló un proyecto sobre la estimación de la evaporación de cultivo y requerimientos hídricos del tomate (*Solanum lycopersicum* Mill.cv. El cid) en invernadero el año de 2009.

En Bogotá, en la Pontificia Universidad Javeriana en el departamento de Ingeniería Civil se desarrolló un trabajo de grado “Evaluación de la relación entre la evapotranspiración potencial teórica y la evaporación registrada en los departamentos de Cundinamarca y Valle del Cauca” por Valentina Marín Valencia en el año 2010.

1.4 Planteamiento del problema

La importancia que tendrá realizar estas mediciones es pronosticar con una mejor anticipación al riego de los cultivos, sin que las plantas se deterioren por la excesiva evapotranspiración.

Con el tiempo el ambiente cambia y mientras el cultivo está en desarrollo hay momentos en que las radiaciones solares aumentan de tal manera, la tierra se reseca aún más rápido y la planta demanda más agua (Evapotranspira) que la proporcionada por el agricultor y es ahí donde se presentan las pérdidas más altas de producción.

La importancia radica en que al realizar este tipo de mediciones en el tanque evaporímetro, en base a la evaporación que se presente en el tanque, sabremos cuánto líquido se ha evaporado en la tierra, y así sabremos en que momento regar el cultivo, y como habrán indicadores de riego el agricultor tendrá una mejor visión al aplicar el vital líquido, y ya no tendrá la necesidad de estar revisando continuamente en qué estado se encuentran las plantas ni verificando que la tierra marque signos de resequeidad.

1.5 Nombre del proyecto

Diseño de un medidor de niveles de agua para estimar la evaporación (Ev)

1.6 Objetivos

1.6.1 Objetivo general

Diseñar, implementar y calibrar un circuito digital que permita la medición precisa de niveles de agua en un tanque evaporímetro, con la finalidad de verificar la cantidad de agua evaporada por día.

1.6.2 Objetivos específicos

1. Diseñar el medidor de niveles para adaptarlo a un tanque de 1.10 m de diámetros, por 55 cm de alto aproximadamente con el fin de censar niveles de agua en dicho tanque.
2. Adaptar el sensor (Ultrasónico) según las necesidades del proyecto para medir niveles de agua en el tanque.

3. Diseñar, el circuito digital que permita el proceso de las variables de entrada y obtener una salida que permita visualizar los resultados
4. Acoplarlo al tanque y según las mediciones de E_v , convertir Evapotranspiración (E_{To}) conectar el medidor de E_v a un sistema tipo semáforo para indicar: rojo (riego de la parcela), verde (no riego de la parcela) y amarillo (a tres días de riego de la parcela).
5. Crear el prototipo físico aplicable, como medidor de nivel de agua, para la creación de otros medidores.

1.7 Justificación del proyecto.

Debido a que el agua es la primera necesidad de las plantas, para la producción de buenos frutos (árboles frutales), es por eso que se ha tratado de ir modificando los métodos de riego para obtener una mejor producción, es por eso que se han ido creando sistemas que les permitan saber el momento adecuado para cada riego.

Existen distintos tipos de plantas de riego en nuestro país, tales como las frutales y las ornamentales y cada una de ellas necesitan distintas cantidades de agua y a determinado tiempo según el tipo de planta.

En México existen 6 000 000 hectáreas de bajo riego la producción de alimentos del país procede esta superficie el rendimiento agrícola de estos cultivos depende de la frecuencia y la fecha apropiada de riego los agricultores utilizan su “experiencia” para decidir cuándo se debe regar estos cultivos. Por lo que es necesario desarrollar un sistema que les permita decidir cuándo regar y de esta forma incrementar la eficiencia de producción por unidad de superficie. [2]

1.8 Delimitación del tema

El proyecto se ha contemplado como un medidor de niveles, en un tanque evaporímetro a través del censado de los niveles y así poder calcular la evaporación por día según los datos que proporcione el sensor (sensor ultrasónico) para medir en nivel en el tanque evaporímetro, así mismo se toman lecturas de las temperatura ambiente y la humedad relativa del sitio para tener una mejor aproximación en la estimación de la evapotranspiración. Cabe mencionar que este proyecto se hará en primer plano como un prototipo.

1.9 Metodología para el desarrollo del proyecto.

Primeramente, la base de procesos y cálculos es el microcontrolador Arduino Mega 2560 o ADK, en donde se llevará a cabo la programación para controlar los dispositivos de entrada y salida, también se cuenta con un sensor ultrasónico(HC-SR04) para medición de nivel y otro sensor de temperatura y humedad (HMZ-433A1), además, de una pantalla LCD donde se podrá visualizar la altura del agua y la humedad relativa del ambiente, para un mejor pronóstico del riego en la parcela, se tendrán indicadores (rojo, verde, azul), y dispositivos actuadores en este caso las electroválvulas quienes se encargarán del desagüe del tanque o llenado del mismo, el envío de datos se realizará por medio de módulo Bluetooth, que mediante una aplicación Android se estarán recibiendo los datos, de la misma manera se enviarán los datos a la computadora en donde se graficarán y analizarán datos.

Cabe mencionar que para este proyecto se utilizará la plataforma de programación de Arduino Mega 2560 o ADK para programar, el motivo de utilizar este Arduino fue la necesidad de utilizar más entradas y salidas que el Arduino Uno está más limitado en ese aspecto

CAPÍTULO 2. FUNDAMENTO TEÓRICO

2.1 Tanque evaporímetro clase A

Es un depósito circular que contiene agua y está expuesto al ambiente, de donde se obtiene lecturas de evaporación, (E_v), con el propósito de simular la evapotranspiración de referencia (E_{To}) y estimar la evapotranspiración de un cultivo determinado (E_{Tc}).

El tanque evaporímetro más conocido es el de clase "A", este tanque es circular, tiene un diámetro de 120.7 cm, una profundidad de 25 cm y el nivel del agua se mantiene 5 a 7.5 cm debajo del borde. El tanque está construido de hierro galvanizado de 0.8 mm de espesor y está montado 15 cm arriba de la superficie del suelo sobre una tarima o plataforma de madera. Los cultivos que se encuentran alrededor del tanque evaporímetro no deben ser más altos de 1 m. [5]

Características del tanque clase A:

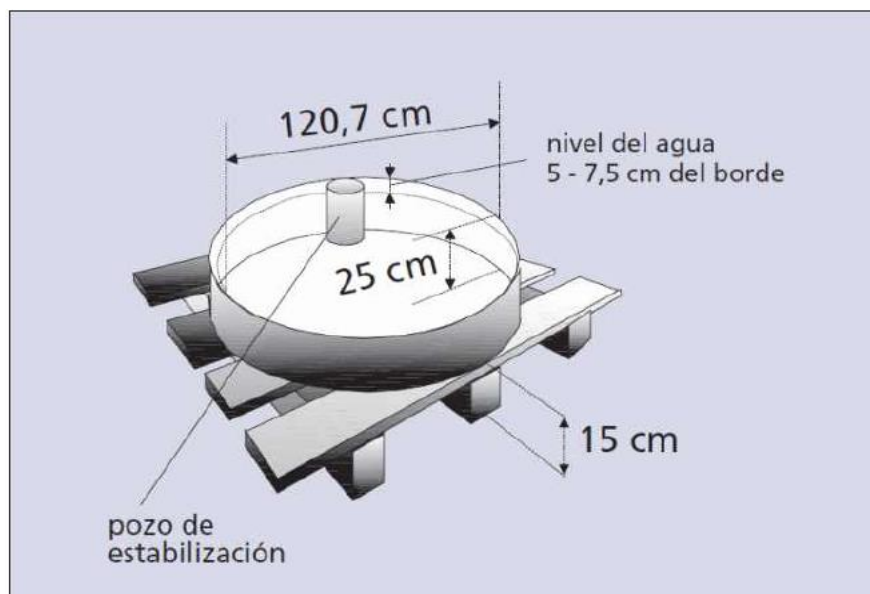


Fig.2.1 Características del tanque clase A

2.1.1 Evapotranspiración (ETo) estimada con el tanque de evaporación

El método FAO Penman-Monteith se mantiene como método estándar único para el cómputo de la ETo de datos meteorológicos. Sin embargo, la evaporación de una superficie libre de agua, proporciona un índice del efecto integrado de la radiación, la temperatura del aire, la humedad y del viento en la evapotranspiración. Sin embargo, las diferencias entre la superficie del agua y de las superficies cultivadas producen diferencias significativas entre la pérdida de agua de una superficie libre de agua y una superficie cultivada.

El tanque ha probado su valor práctico y ha sido utilizado con éxito para estimar la evapotranspiración de referencia observando la pérdida por evaporación de una superficie y aplicando coeficientes empíricos para relacionar la evaporación del tanque con ETo.

Doorenbos y pruit (1977) proponen un método para la estimación de la ETo a partir de los datos medidos de evaporación de una lámina libre de agua en el tanque de tipo A. la ecuación correspondiente, toma la forma siguiente:

$$ETo = Kp(Eo).....Ec. 1$$

Donde

ETo Evapotranspiración de referencia (mm/día)

Eo Evaporación del tanque tipo A (mm/día)

Kp Coeficiente del tanque (adim)

El coeficiente del tanque está en función de la humedad relativa ambiental, del recorrido del viento tomado a 2 m sobre el nivel del terreno y de la distancia en metros desde el tanque hasta el extremo de la cubierta sobre la que se ubica. Los valores de *Kp* fluctúan alrededor de 0.80 por considerar que hay diferencias significativas en la pérdida de agua de una superficie libre en comparación a la de un cultivo, siendo mayor en la superficie libre que al del cultivo. El almacenaje de calor dentro del tanque puede ser apreciable y puede causar una significativa evaporación durante la noche mientras la mayoría de los cultivos transpiran solamente durante el día. [6]

2.2 Variables medidas

2.2.1 Evaporación

La evaporación es el proceso físico por el cual el agua que se encuentra en estado líquido pasa al estado gaseoso. Para que el cambio de estado se produzca, se necesita de un calor de aproximadamente 600 calorías por cada gramo de agua evaporada.

El agua se evapora de una variedad de superficies, tales como lagos, ríos, caminos, suelos y la vegetación mojada (Allen et al., 1998).

Algunos de los factores meteorológicos más importantes que afectan a la evaporación son:

- *Radiación Solar:* La evaporación es la conversión de agua en vapor de agua. Este proceso se lleva a cabo casi sin interrupción durante las horas de luz del día y, a menudo también durante la noche. Para cambiar el estado de las moléculas de agua de líquido a gas se requiere de energía (conocida como el calor latente de vaporización), por lo cual el proceso es más activo bajo la radiación directa del sol.

De ello se deduce que los días nublados el proceso de evaporación serán más lento, debido a que las nubes no permiten el paso total de la radiación solar hacia la superficie terrestre.

- *Viento:* Cuando el agua se evapora en la atmosfera la capa límite entre la tierra y el aire se satura completamente. Esta capa debe ser removida y continuamente reemplazada por aire seco para que pueda continuar el proceso de evaporación. Este movimiento del aire en la capa limite depende del viento y de su velocidad.
- *Humedad Relativa:* A medida que aumenta la humedad del aire, su capacidad de absorber más vapor de agua disminuye por lo que la tasa de evaporación disminuye. Si la capa límite de aire saturado es reemplazada por aire igualmente saturado o con niveles altos de humedad no se podrá mantener la velocidad de evaporación; solo se mantendrá si el aire entrante es más seco que el aire que se desplaza.
- *Temperatura:* Como se mencionó anteriormente para que suceda la evaporación es necesario una entrada de energía. De ello se deduce que si las temperaturas ambientes del aire y del suelo son altas, la evaporación

sucedirá con mayor rapidez que si son bajas. Dado de que la capacidad del aire para absorber vapor de agua aumenta a medida que la temperatura se eleva podemos decir que la temperatura del aire tiene un efecto doble en el proceso de evaporación, mientras que la temperatura del suelo y del agua tiene solo efectos directos (Viessman & Lewis, 2003). [5]

2.3 Instrumentos de medición

Los instrumentos comunes y variables para la medición de un tanque evaporímetro de tipo a son las siguientes:

- Sensor de temperatura
- Sensores de distancia
- Sensor de humedad
- Anemómetro

2.4 Programas de apoyo

2.4.1 App Inventor

App Inventor es una aplicación web de código abierto originalmente proporcionado por Google, y ahora mantenido por el Instituto de Tecnología de Massachusetts (MIT).

La aplicación se puso a disposición a través de la solicitud el 12 de julio de 2010 y lanzado públicamente el 15 de diciembre de 2010. La App Inventor equipo fue dirigido por Hal Abelson y Mark Friedman. En el segundo semestre de 2011, Google lanzó El código fuente, puesto fin a su servidor, y proporcionó fondos para la creación del Centro del MIT para el Aprendizaje Móvil, dirigida por el creador de App Inventor Hal Abelson y profesores del MIT compañeros Eric Klopfer y Mitchel Resnick. La versión del MIT se puso en marcha en marzo de 2012.

El 6 de diciembre de 2013 (el comienzo de la Hora de Código), MIT lanzó App Inventor 2, cambiar el nombre de la versión original de "App Inventor Classic" La principal diferencia es:

- El editor de bloques en la versión original corrió en un proceso Java independiente, utilizando la biblioteca de bloques abiertos de Java para crear bloques visuales lenguajes de programación y programación

Bloques Abiertas se distribuye por el Instituto de Tecnología de Massachusetts 's Programa de Formación Docente Scheller (PASO) y se deriva de la investigación de tesis de maestría por Ricarose Roque. Profesor Eric Klopfer y Daniel Wendel del Programa Scheller apoyaron la distribución de bloques abiertos bajo una licencia MIT .Los bloques abiertos de programación visual está estrechamente relacionado con StarLogo TNG, un proyecto de STEP y arañazos, un proyecto del MIT Media Laboratory permanente Grupo Kindergarten. App Inventor 2 reemplazado bloques abiertos con Blockly, un editor de bloques que se ejecuta dentro del navegador. [8]

2.4.2.1 Propiedades de App Inventor

App Inventor incluye:

- Un diseñador, en el que se especifican los componentes de un programa. Esto incluye componentes visibles, como botones e imágenes, que se colocan en una pantalla simulada, y componentes no visibles, tales como sensores y conexiones web.
- Un editor de bloques, en el que se crea la lógica del programa.
- Un compilador basado en el marco idioma Kawa y el dialecto de Kawa del Esquema lenguaje de programación, desarrollado por Per Bothner y se distribuye como parte de la GNU sistema operativo por la Free Software Foundation.
- Una aplicación para la depuración en tiempo real en un dispositivo Android conectado. [8]

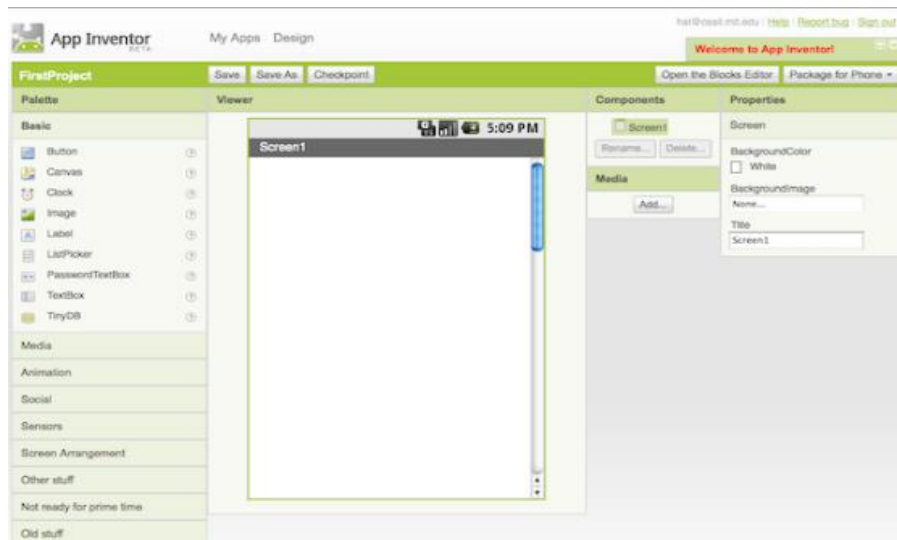


Fig. 2.2 Área de trabajo de App Inventor

2.4.2 LabView

El software LabVIEW es ideal para cualquier sistema de medidas y control, es una plataforma y entorno de desarrollo para diseñar sistemas, resolver problemas, productividad acelerada y constante innovación. Con un lenguaje de programación visual gráfico. Recomendado para sistemas hardware y software de pruebas, control y diseño, simulado o real y embebido, pues acelera la productividad. El lenguaje que usa se llama lenguaje G, donde la G simboliza que es lenguaje Gráfico.

Los programas desarrollados con LabVIEW se llaman Instrumentos Virtuales, o VIs, y su origen provenía del control de instrumentos, aunque hoy en día se ha expandido ampliamente no sólo al control de todo tipo de electrónica (Instrumentación electrónica) sino también a su programación embebida, comunicaciones, matemáticas, etc. Un lema tradicional de LabVIEW es: "La potencia está en el Software", que con la aparición de los sistemas multinúcleo se ha hecho aún más potente. Entre sus objetivos están el reducir el tiempo de desarrollo de aplicaciones de todo tipo (no sólo en ámbitos de Pruebas, Control y Diseño) y el permitir la entrada a la informática a profesionales de cualquier otro campo. LabVIEW consigue combinarse con todo tipo de software y hardware, tanto del propio fabricante -tarjetas de adquisición de datos, PAC, Visión, instrumentos y otro Hardware- como de otros fabricantes. [3]

2.4.3 Utilizando LabVIEW

En este capítulo se explican los conceptos básicos en LabVIEW.

Los temas son los siguientes:

- El LabVIEW Medio Ambiente
- Panel Frontal y Diagrama de Bloques
- Paletas: Paleta de control, funciones de la paleta, paleta de Herramientas
- Tipos de datos
- Nodos de Propiedad

2.4.4 LabVIEW para el Medio Ambiente

Programas de LabVIEW se llaman Instrumentos Virtuales o VIs, porque su apariencia y funcionamiento imitan los instrumentos físicos, como osciloscopios y multímetros. LabVIEW contiene un amplio conjunto de herramientas para la adquisición de analizar, visualizar y almacenar los datos, así como herramientas para ayudarle a solucionar su código.

Al abrir LabVIEW, primera vez que vienen a la ventana "Getting Started"

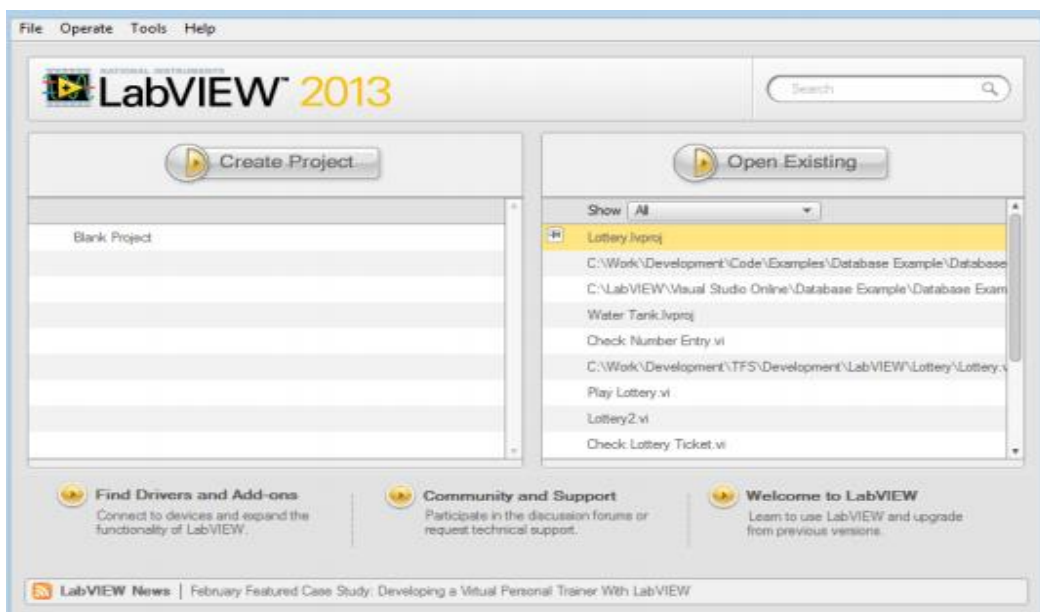


Fig. 2.3 Software LabVIEW

Con el fin de crear un nuevo VI, seleccione "En blanco VI" o con el fin de crear un nuevo proyecto de LabVIEW, seleccione "proyecto vacío".

Al abrir un VI en blanco, aparece una ventana de panel frontal sin título. Esta ventana muestra el panel frontal y es una de las dos ventanas de LabVIEW que se utilizan para construir un VI. La otra ventana contiene el diagrama de bloques. Las secciones siguientes describen el panel frontal y el diagrama de bloques. [3]

2.4.5 Propiedad de los componentes

Panel frontal

Cuando haya creado un nuevo VI o seleccionado existente, aparecerá el panel frontal y el diagrama de bloques para que especifica VI.

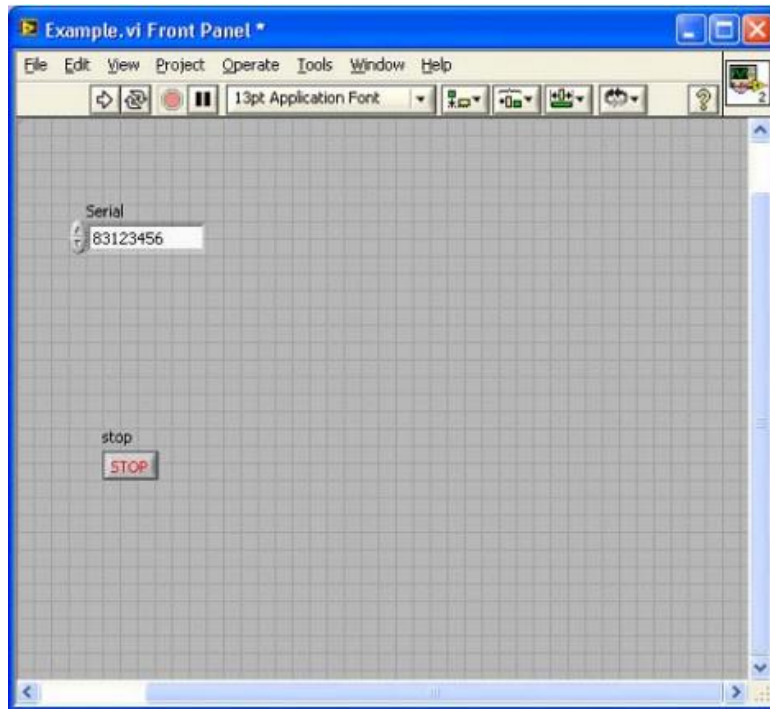


Fig.2.4 Panel Frontal

En LabVIEW, usted construye una interfaz de usuario, o el panel frontal, con controles e indicadores, que son los terminales de entrada y salida interactivas. Los controles son perillas, botones, diales y otros dispositivos de entrada. Los indicadores son gráficas, LEDs y otras pantallas.

Los controles simulan dispositivos de entrada del instrumento y de suministro de datos al diagrama de bloques. Los indicadores simulan los dispositivos de salida de instrumentos y datos de visualización del diagrama de bloques adquiere o genera.

Diagrama de bloques

Después de construir la interfaz de usuario, se agrega código usando VIs y estructuras para controlar los objetos del panel frontal. El diagrama de bloques contiene el código. En cierto modo, el diagrama de bloques se asemeja a un diagrama de flujo.

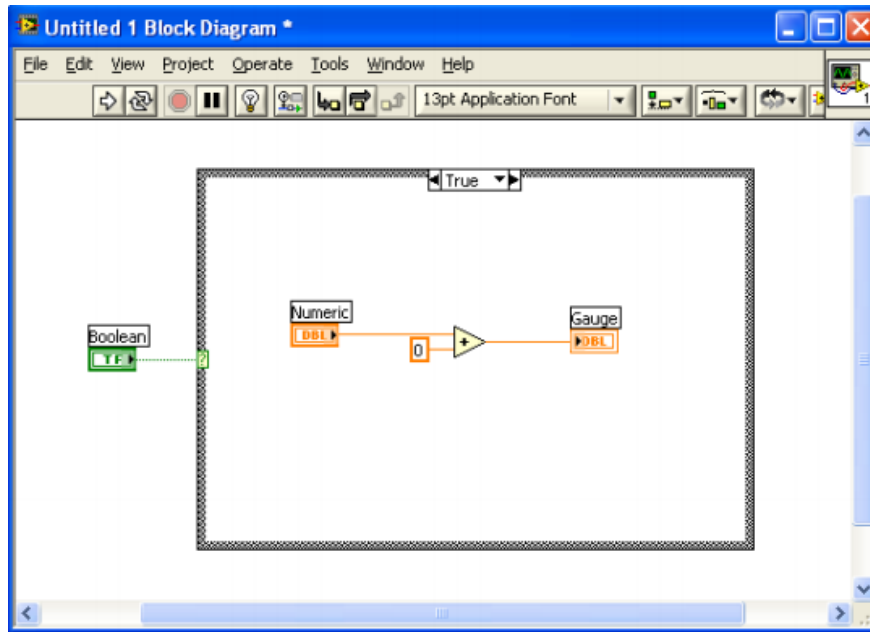


Fig. 2.5 Diagrama de bloque

Después de construir el panel frontal, se agrega código utilizando representaciones gráficas de funciones para controlar los objetos del panel frontal. El diagrama de bloques contiene el código fuente gráfica. Objetos del panel frontal aparecen como terminales, en el diagrama de bloques. Objetos de diagrama de bloques incluyen terminales, subVIs, funciones, constantes, estructuras y cables, que transfieren datos entre otros objetos de diagrama de bloques.

Controla la gama de colores

Los controles y las funciones paletas contienen sub paletas de objetos que puede utilizar para crear un VI. Al hacer clic en un icono de la paleta de sub, la totalidad de los cambios en la gama de colores de la paleta sub seleccionado. Para utilizar un objeto en las gamas de colores, haga clic en el objeto y colocarlo en el panel frontal o diagrama de bloques. La paleta de controles sólo está disponible en el panel frontal. La paleta de controles contiene los controles e indicadores que se utilizan para construir el panel frontal. [4]

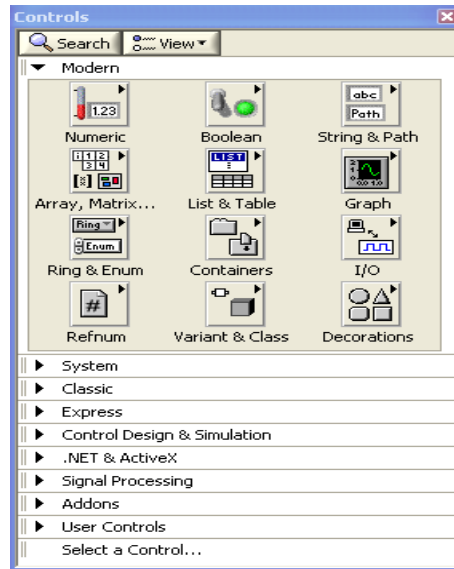


Fig. 2.6 LabVIEW incluye diferentes controles para interactuar con la aplicación.

Paleta de Función

La paleta de funciones está disponible sólo en el diagrama de bloques. La paleta de funciones contiene los VIs y funciones que utiliza para construir el diagrama de bloques.

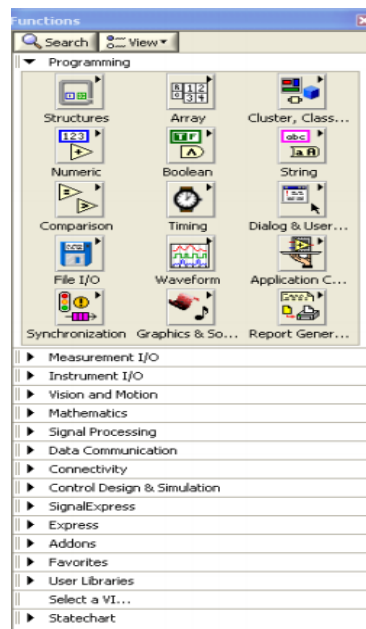








Fig. 2.7 Paleta de función

Barra de herramientas


A continuación vemos la Barra de Herramientas de LabVIEW:



Los comportamientos de los diferentes botones son los siguientes:

-  Haga clic en el botón Ejecutar para ejecutar un VI. LabVIEW compila el VI, si es necesario. Puede ejecutar un VI si el botón Run aparece como un sólido flecha blanca. La flecha blanca sólida, que se muestra más arriba, también indica que puede utilizar el VI como un subVI si crea un panel conector para el VI.
-  Mientras el VI se ejecuta, el botón Run aparece como se muestra a la izquierda si el VI es un VI de alto nivel, lo que significa que no tiene personas que llaman y por lo tanto no es un subVI.
-  Si el VI que se está ejecutando es un subVI, el botón Run aparece como se muestra a la izquierda.
-  El botón Run aparece roto, muestra a la izquierda, cuando el VI que está creando o editando contiene errores. Si el botón Ejecutar todavía aparece roto después de acabado el cableado del diagrama de bloques, el VI está roto y no se puede ejecutar. Haga clic en este botón para mostrar la ventana de la lista de errores, que enumera todos los errores y advertencias.
-  Haga clic en el botón Ejecutar Continualmente, se muestra a la izquierda, para ejecutar el VI hasta que se cancela o hacer una pausa en la ejecución. También puede hacer clic en el botón de nuevo para desactivar el funcionamiento continuo.
-  Mientras el VI se ejecuta, el botón Abort Execution, muestra a la izquierda, aparece. Haga clic en este botón para detener el VI inmediatamente si no hay otra forma de detener el VI. Si más de un nivel superior corriendo VI utiliza el VI, se atenúa el botón.

Nota: Evite utilizar el botón Abort Execution para detener un VI. O deje que el VI complete sus datos de flujo o el diseño de un método para detener el VI mediante programación. Al hacerlo, el VI se encuentra en un estado conocido. Por ejemplo, colocar un botón en el panel frontal que detiene el VI cuando hace clic en él.

-  Haga clic en el botón de pausa, se muestra a la izquierda, para hacer una pausa en un VI corriendo. Al hacer clic en el botón Pausa, LabVIEW destaca en el diagrama de bloques de la ubicación donde se detuvo la ejecución, y el botón

Pausa aparece en rojo. Haga clic en el botón de nuevo para continuar ejecutando el VI.

Paleta Sub Numérico

"Control Numérico" y "indicador numérico" son los objetos más utilizados en la paleta sub numérico. [4]

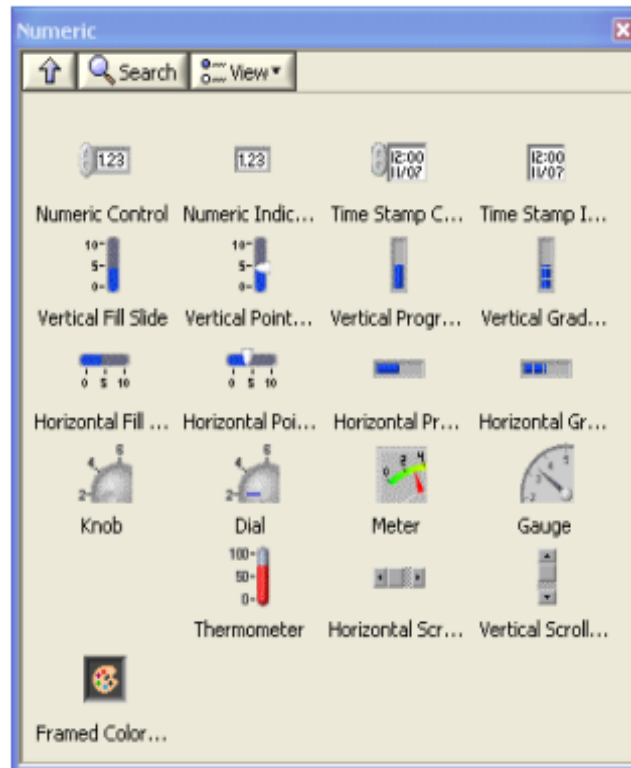


Fig. 2.8 Paleta Sub-Numérico

Interfaz de usuario

Después de programar en el software NI LabVIEW, generalmente se construyó una interfaz de usuario. Visualizando los datos de entrada adquiridos en tiempo real, la siguiente figura muestra el diseño y programación (diagrama de bloques) como se fue desarrollando la interfaz de usuario (datos que captura en la pantalla).

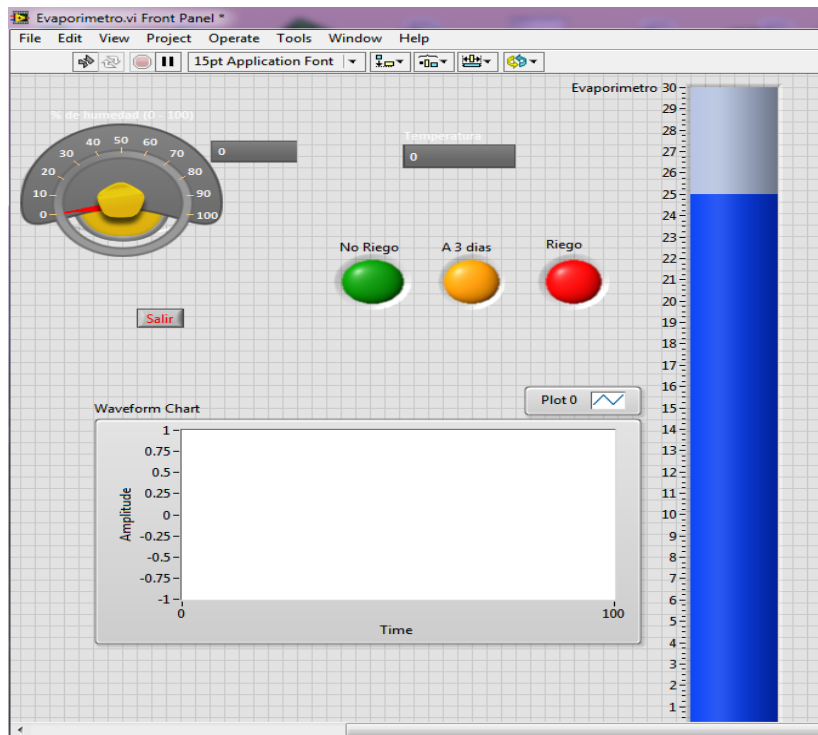
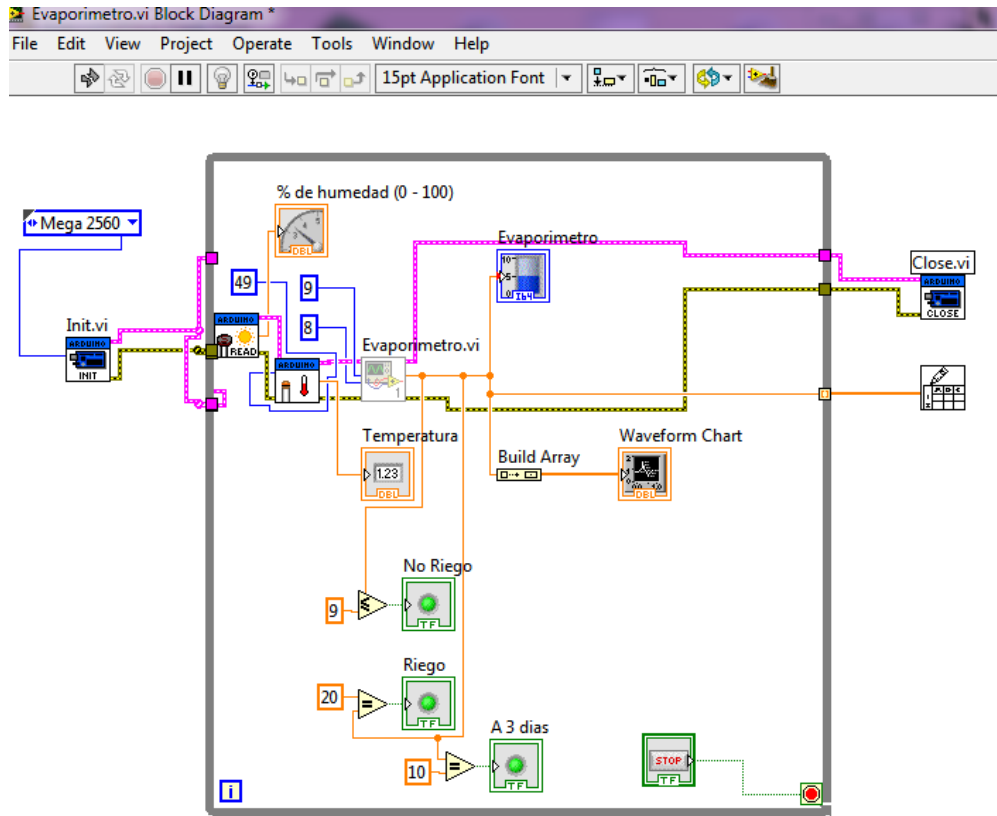


Fig. 2.9 Programas en el plano frontal.

2.5 Arduino MEGA 2560

Tomando en cuenta las características de los diferentes microcontroladores se seleccionó el que tiene más ventajas para su uso. Este circuito integrado forma parte de las placas Arduino mega, que pertenece a la categoría de los microcontroladores.

Presenta características muy versátiles que le permiten aplicaciones en un gran número de variedades, además de contar con módulos Ethernet con puerto RJ-45 para la conexión a internet y la posibilidad de enviar datos. Arduino mega 2560. [1]

Arduino es una plataforma para computación física de código abierto basada en una simple tarjeta I/O y un entorno de desarrollo que implementa el procesamiento/idioma de cableado (Processing/Wiring). Arduino puede ser usado para desarrollar objetos autónomos interactivos o puede ser conectado a software en su ordenador (por ejemplo, Flash, Processing, MaxMSP). El IDE de código abierto puede ser descargado de forma gratuita (en la actualidad para Mac OS X, Windows y Linux).

Características:

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Tabla 2.1 Características eléctricas del Arduino MEGA 2560

A continuación se muestra en la Fig. 2.10, donde están ubicados los elementos más importantes que componen la placa Arduino MEGA 2560 que son descritos de arriba abajo y de izquierda a derecha: V

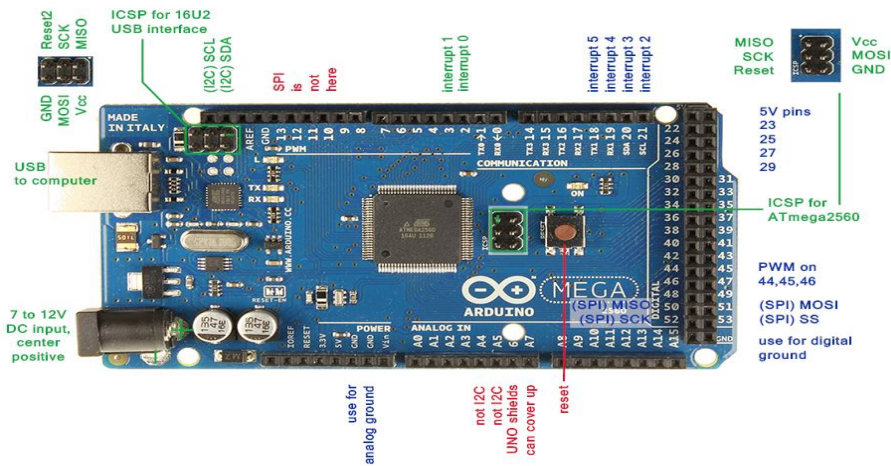


Fig. 2.10 Estructura del Arduino Mega 2560

2.5.1 Estructura básica de un programa

La estructura básica de programación de Arduino es bastante simple y divide la ejecución en dos partes:

```
void setup()
{
  //Preparacion del Programa
}

void loop()
{
  //Ejecución
}
```

Setup() constituye la preparación del programa y loop() es la ejecución. En la función Setup() se incluye la declaración de variables y se trata de la primera función que se ejecuta en el programa. Esta función se ejecuta una única vez y es empleada para configurar las terminales de salida mediante pinMode(). La función loop() incluye el código a ser ejecutado continuamente (leyendo las entradas de la placa, salidas, etc.).

Nota: Al igual que en C se pueden introducir bloques de comentarios con /* ... */.
Entradas/Salidas digitales y analógicas.

digitalRead(pin). Lee el valor desde un pin digital específico. Devuelve un valor HIGH o LOW.

digitalWrite(pin, value). Introduce un nivel alto (HIGH) o bajo (LOW) en el pin digital especificado. De nuevo, el pin puede ser especificado con una variable o una constante.

analogRead(pin). Lee el valor desde el pin analógico especificado con una resolución de 10 bits. Esta función solo funciona en los pines analógicos (0-5). El valor resultante es un entero de 0 a 1023. Los pines analógicos, a diferencia de los digitales no necesitan declararse previamente como INPUT o OUTPUT.

analogWrite(pin, value). Escribe un valor pseudo-analógico usando modulación por ancho de pulso (PWM) en un pin de salida marcado como PWM.

2.5.2 Tipos de Datos

Arduino permite manejar los siguientes tipos de datos:

Byte. Almacena un valor numérico de 8 bits. Tienen un rango de 0-255.

Int. Almacena un valor entero de 16 bits con un rango de 32,767 a -32,768.

Long. Valor entero almacenado en 32 bits con un rango de 2,147,483,647 a -2,147,483, 648.

Float. Tipo coma flotante almacenado en 32 bits con un rango de 3.4028235E+38 a -3.4028235E+38.

Arrays. Se trata de una colección de valores que pueden ser accedidos con un numero de índice (el primer valor del índice es 0).

2.5.3 Funciones de Tiempo

delay(ms). Realiza una pausa en el programa la cantidad de tiempo en milisegundos especificada en el parámetro (máximo 1000, mínimo 1).

millis(). Devuelve la cantidad de milisegundos que lleva la placa Arduino ejecutando el programa actual como un valor long unsigned. Después de 9 horas el contador vuelve a 0. [1]

2.5.4 Características físicas y compatibilidad de Shields

La longitud y amplitud máxima de la placa Mega 2560 son de 4 y 2.1 pulgadas respectivamente, con el conector USB y la conexión de alimentación sobresaliendo de estas dimensiones. Tres agujeros para fijación con tornillos permiten colocar la placa en superficies y cajas. Tener en cuenta que la distancia entre los pines digitales 7 y 8 es 160 mil (0,16”), no es múltiplo de la separación de 100 mil entre los otros pines.

2.6 Módulo Bluetooth

El módulo HC-06 es prácticamente idéntico a simple vista con los demás módulos que existen en el mercado.

Una simple diferencia es que el módulo HC-06 funciona como Slave y el HC-05 como Master y Slave (lo que podría confundir a algunos).

Físicamente se diferencian por el número de pines. En el HC-06 tiene un conector de 4 pines (imagen Izquierda) mientras que el HC-05 trae uno de 6 pines (Imagen Derecha)

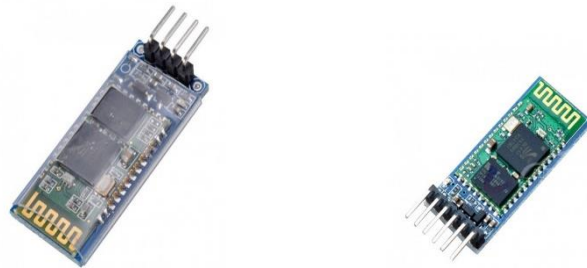


Fig. 2.11 Módulo de comunicación Bluetooth

Modulo Bluetooth Slave & Master HC-06	
Protocolo bluetooth	Bluetooth especificación V2.0+EDR
Modulación	GFSK (Gaussian Frequency Shift Keying)

Potencia de emisión	<4dBm, clase 2
Sensibilidad	<-84dBm a 0.1% VER
Temperatura de trabajo	-20°C a +75°C
Frecuencia	2.4Ghz ISM Band
Rango de baudios ajustable	9600 a 115200bps
Voltaje alimentación	3.3V a 6VDC
Distancia Bluetooth	10 metros

Tabla 2.2 Características del módulo bluetooth HC-06

Este módulo HC-06 se encuentra en modo Esclavo, es decir puede ser conectado hacia una PDA, PC, Teléfono, pero no a otros módulos Bluetooth

2.6.1 Configuración del módulo Bluetooth

Utilizando la librería Software Serial, se puede usar cualquier pin digital del Arduino como TX y RX. De no usar esta librería, vamos a tener que usar los pines 0 y 1 para la comunicación. Esto nos puede traer el inconveniente de tener que desconectar el Bluetooth para cargar el programa y conectarlo de nuevo una vez que compilo. Esto se debe a que los pines de comunicación 0 y 1 son los que se utiliza el Arduino para la conexión USB.

Recordar que la conexión es siempre cruzada, El pin RX del módulo Bluetooth va al pin TX del Arduino y viceversa. El resto de la conexión (pines) corresponde a la alimentación y está indicada en el módulo, vcc y gnd.

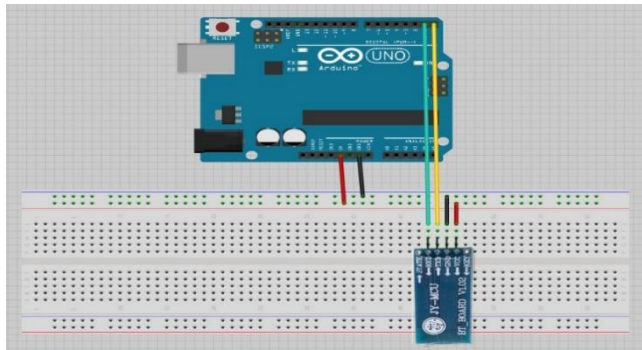


Fig. 2.12 Conexión del módulo Bluetooth

Podemos cambiar las configuraciones de velocidad de transmisión de datos, la contraseña del módulo, etc. mediante un protocolo llamado comandos AT. Para poder hacerlo es necesario que el HC-06 no este emparejado ni siendo usado por ningún dispositivo.

Una vez conectado y configurado el modulo, lo único que nos falta es un programa para Android que envíe datos vía Bluetooth y un sketch para Arduino que nos muestre en la pc los datos que recibimos vía Bluetooth o viceversa, que envíe lo que nosotros le escribimos en el puerto serie.

2.7 Sensor DTH11

DHT11 ofrece un complejo sensor de temperatura y humedad con una salida de señal digital calibrado. Mediante el uso de la señal digital de adquisición exclusiva técnica y la tecnología de detección de temperatura y humedad, que garantiza una alta fiabilidad y una excelente estabilidad a largo plazo. Este sensor incluye un componente de medición de humedad de tipo resistivo y un componente de medición de temperatura NTC, y se conecta a un alto rendimiento de 8 bits microcontrolador, que ofrece una excelente calidad, respuesta rápida, capacidad anti-interferencia y la rentabilidad.

Item	Measurement Range	Humidity Accuracy	Temperature Accuracy	Resolution	Package
DHT11	20-90%RH 0-50 C	+ 5% RH	± 2C	1	4 Pin Single Row
Parameters	Conditions	Minimum	Typical	Maximum	
Humidity Resolution		1%RH	1%RH	1%RH	
Repeatability			8 Bit		
Accuracy	25 C		± 1%RH		
	0-50 °C		± 4%RH		± 5%RH
Interchangeability	Fully Interchangeable				
Measurement Range	0C	30%RH			90%RH
	25 C	20%RH			90%RH
	50 C	20%RH			80%RH
Response Time (Seconds)	1/e(63%)25C , 1m/s Air	6 S	10 S		15 S
Hysteresis			± 1%RH		
Long-Term Stability	Typical		± 1%RH/year		
Temperature Resolution		1C	1C		1C
		8 Bit	8 Bit		8 Bit
Repeatability			± 1C		
Accuracy		± 1C			± 2C
Measurement Range		0C			50 C
Response Time (Seconds)	1/e(63%)	6 S			30 S

Tabla 2.3 Especificaciones técnicas del sensor DHT11

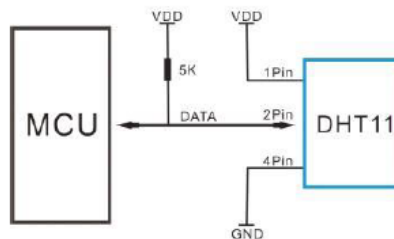


Fig. 2.13 Diagrama del sensor DHT11

2.7.1 Proceso de Comunicación general

Cuando MCU envía una señal de arranque, DHT11 cambia del modo de bajo consumo de energía para el modo de funcionamiento, a la espera de MCU completar la señal de inicio. Una vez que se ha completado, DHT11 envía una señal de respuesta de los datos de 40 bits que incluyen la humedad relativa y la información de la temperatura a la MCU. Los usuarios pueden optar por cobrar (leer) algunos datos. Sin la señal de arranque del MCU, DHT11 no dará la señal de respuesta a la MCU. Una vez recolectados los datos, DHT11 cambiará al modo de consumo de energía de baja hasta que reciba una señal de arranque de MCU de nuevo.

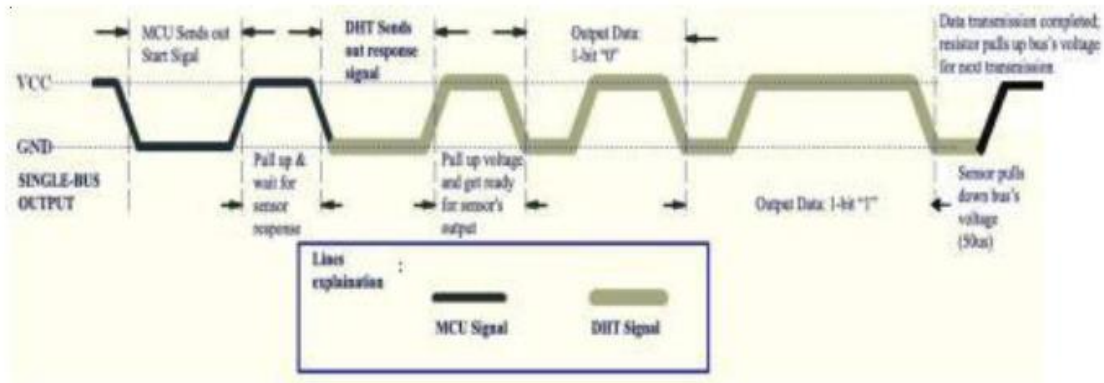


Fig. 2.14 Señal de comportamiento del sensor DHT11

2.7.2 Señal de inicio de MCU a DHT

Los datos de estado libre solo bus están en alto nivel de tensión. Cuando empieza la comunicación entre MCU y DHT11, el programa de MCU fijará el nivel de tensión de un solo bus de datos de alta a baja y este proceso debe tener al menos 18 ms para asegurar la detección de DHT de la señal de MCU, entonces MCU se tire hacia arriba de tensión y esperar 20 -40us para la respuesta de DHT.

2.8 Sensor ultrasónico HC-SR04

El HC-SR04 es un sensor ultrasónico de bajo costo que no sólo puede detectar si un objeto se presenta, como un sensor PIR (Passive Infrared Sensor), sino que también puede sentir y transmitir la distancia al objeto.

Tienen dos transductores, básicamente, un altavoz y un micrófono.

Ofrece una excelente detección sin contacto (remoto) con elevada precisión y lecturas estables en un formato fácil de usar.

El funcionamiento no se ve afectado por la luz solar o el material negro como telémetros ópticos (aunque acústicamente materiales suaves como telas pueden ser difíciles de detectar).

La velocidad del sonido en el aire (a una temperatura de 20 °C) es de 343 m/s. (por cada grado centígrado que sube la temperatura, la velocidad del sonido aumenta en 0,6 m/s)

Características

Envía un Pulso "1" de al menos de 10uS por el Pin Trigger (Disparador).

El sensor enviará 8 Pulsos de 40KHz (Ultrasonido) y coloca su salida Echo a alto (seteo), se debe detectar este evento e iniciar un conteo de tiempo.

La salida Echo se mantendrá en alto hasta recibir el eco reflejado por el obstáculo a lo cual el sensor pondrá su pin Echo a bajo, es decir, terminar de contar el tiempo.

Se recomienda dar un tiempo de aproximadamente 50ms de espera después de terminar la cuenta.

La distancia es proporcional a la duración del pulso y puedes calcularla con las siguientes formula (Utilizando la velocidad del sonido = 340m/s):

Distancia en cm (centímetros) = Tiempo medido en us x 0.017 (microsegundo)

Parámetros eléctricos	HC-SR04 Modulo ultrasónico
Rango de operación	DC-5V
Corriente de operación	15mA
Frecuencia de operación	40Khz
Rango máximo de medida	4m
Rango mínimo de medida	2cm
Angulo de medición	15 Grados
Entrada-Disparo de señal	10uS TTL pulse
Salida-Reflejo de señal	Salida TTL nivel de señal proporcional con el rango
Dimensiones	45*20*15mm

Tabla 2.4 Datos del sensor Ultrasónico

2.9 Optoacopladores

La manipulación de altas corrientes, de hasta varios centenares de amperios, implica el tener consideraciones de seguridad eléctrica para los operarios y de protección para el sistema digital.

Es deseable que la interconexión entre ambas etapas (la digital y la de potencia) se haga por un medio de acoplamiento que permita aislar eléctricamente los dos sistemas. Esto se puede lograr con los dispositivos llamados optoacopladores,

mediante los cuales se obtiene un acoplamiento óptico y, al mismo tiempo, un aislamiento eléctrico. Por ello también se les conoce como optoaisladores. El acoplamiento se efectúa en el rango del espectro infra-rojo a partir de dispositivos emisores de luz, usualmente IRED (infra-rojo) LEDs (diodos emisores de luz), actuando como emisores y utilizando dispositivos detectores de luz (optodetectores), actuando como receptores

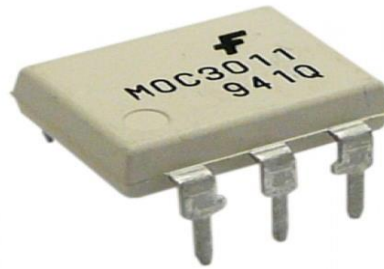


Fig. 2.15 Optoacoplador

CAPÍTULO 3. DESARROLLO E IMPLEMENTACIÓN DEL PROYECTO

3.1 Diseño del sistema de adquisición de datos

3.1.1 Diagrama a bloques

En el siguiente diagrama veremos el funcionamiento del medidor de nivel, con todos los componentes que lo integran, así como la interfaz gráfica visualizada en la computadora, de la misma manera podremos visualizar los datos leídos por el Arduino y enviado vía bluetooth a la aplicación Android.

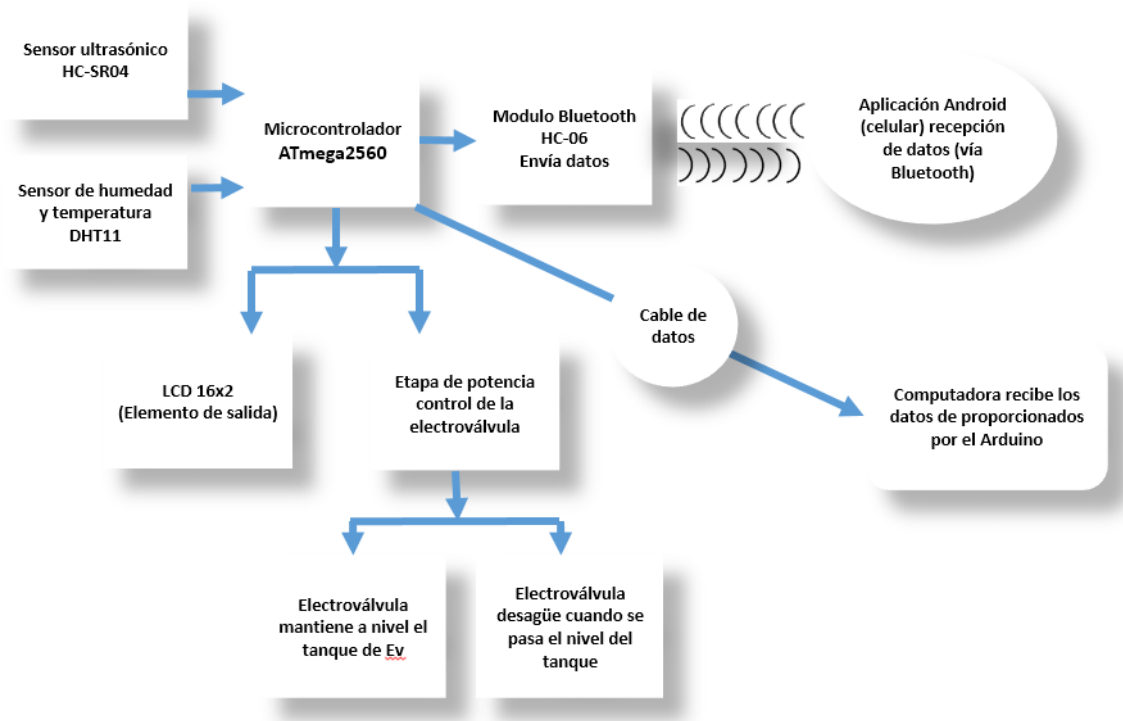


Fig. 3.3 Diagrama a bloques de funcionamiento

3.2 Construcción de la etapa de potencia

3.2.1 Electroválvula 12 VCD y Electroválvula 120 VCA

El control de la bomba se utilizó el circuito de la Fig. 3.4, para poder controlarlo con el Arduino mediante una señal de 5 volt. La programación de esta, se basa mediante una comparación de medición, las distancias que se utilizaron son la de 10 y 17 cm, cuando el nivel del agua llegara a 10 cm la bomba de agua (electroválvula) se activa hasta los 17 cm, para cuando el sistema esté lleno la electroválvula se activa empezando a vaciar el sistema. El circuito del control de potencia para la Electroválvula es la Fig. 3.4.

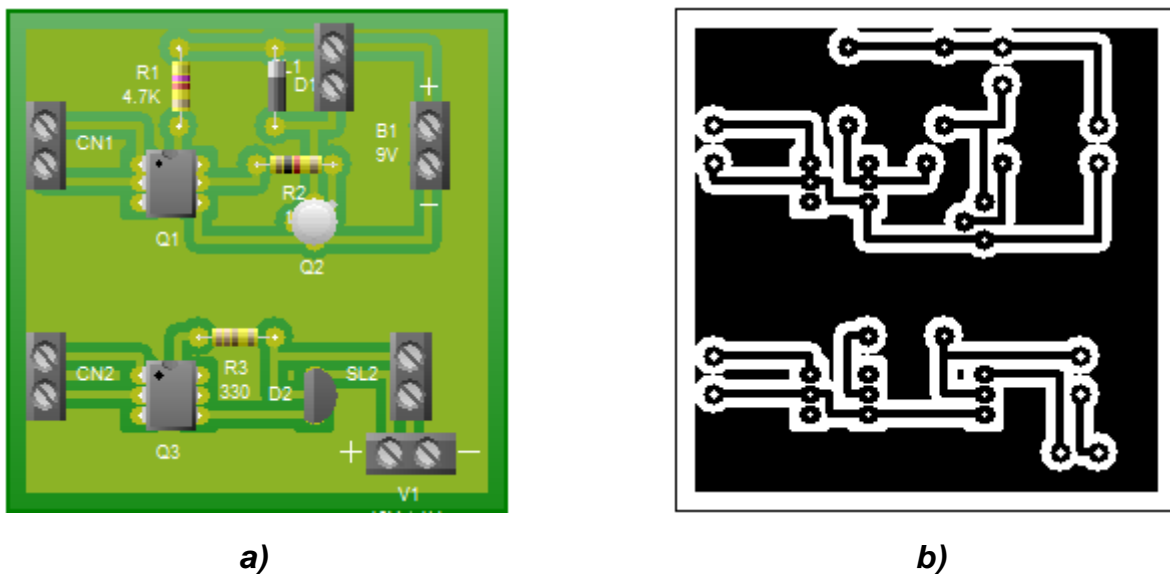
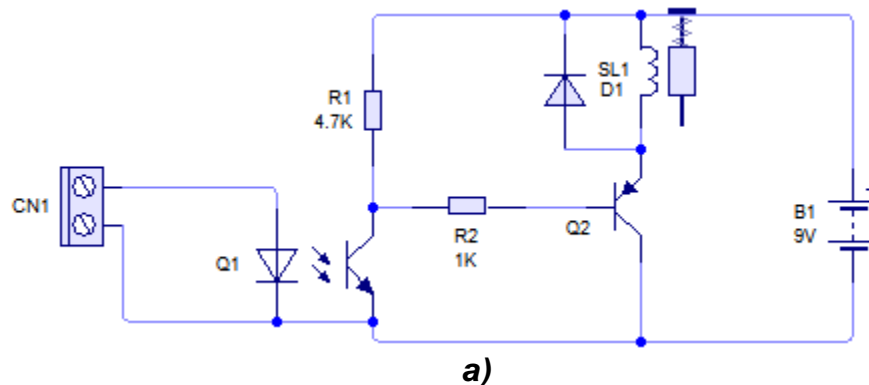
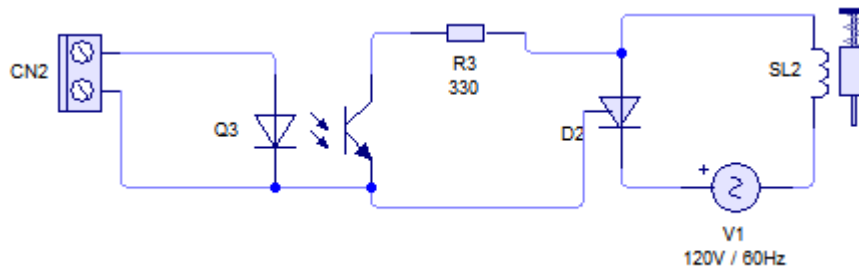


Fig. 3.4 Circuito de potencia para controlar las electroválvulas, a) Prototipo; b) PCB, diseñado en LiveWire





b)

**Fig. 3.5 Diagrama eléctrico del circuito de potencia, a) Electroválvula VCD;
b) Electroválvula VCA**

3.3 Simulación del sistema de adquisición de datos

En el siguiente diagrama se puede observar la simulación del sistema de adquisición de datos que se realiza, tal como se ve en el diagrama, fue como se plasmó ya en circuito impreso en el modelo físico.

En la simulación que se realizó se colocaron todos los elementos utilizados en el desarrollo del proyecto. Los pines que se pueden observar como que si estuvieran flotando *Trigger* y *Echo*, son los pines que fueron utilizados para la conexión del sensor ultrasónico hacia el Arduino, los siguientes pines *Rx* y *Tx*, son los pines utilizados para la conexión del módulo bluetooth, y por ultimo tenemos el pin *SenHumTem*, en este pin está colocado el sensor de humedad y temperatura.

Cabe mencionar que los circuitos que se observan en la simulación para el control de las electroválvulas, es elemental para proteger a nuestro Arduino al filtrarse algún voltaje que pudiera quemar al Arduino

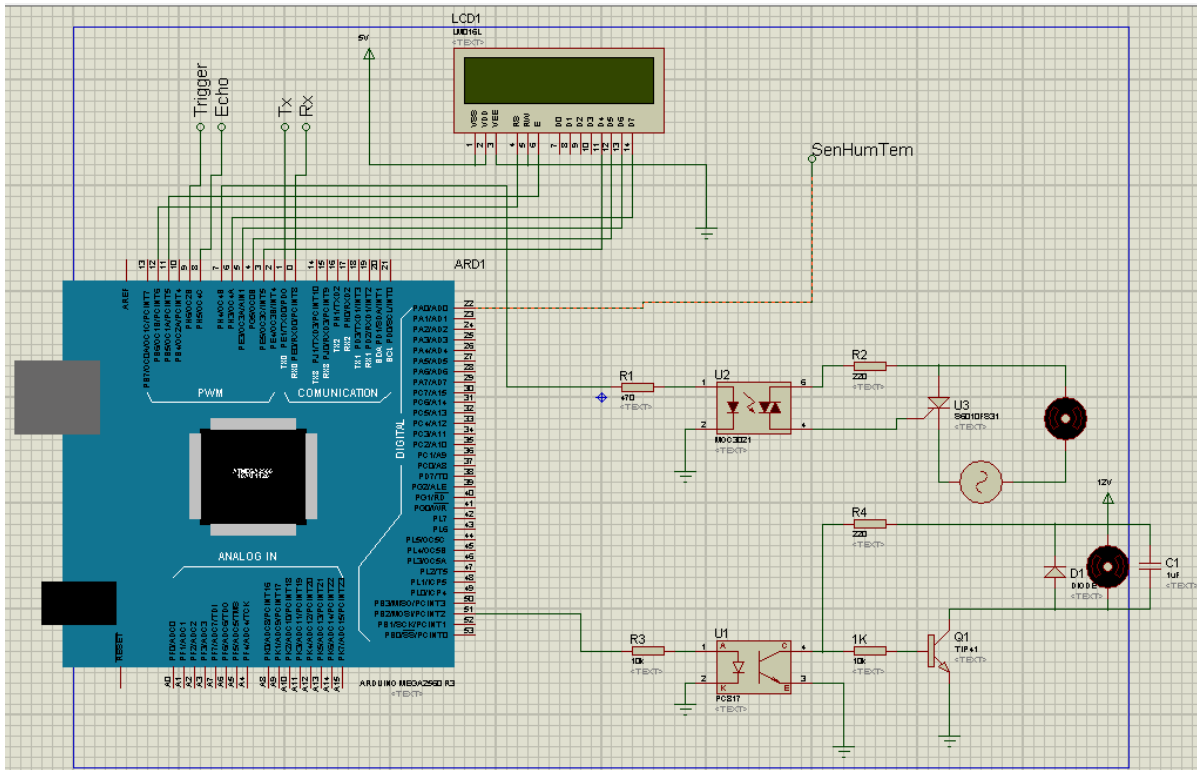
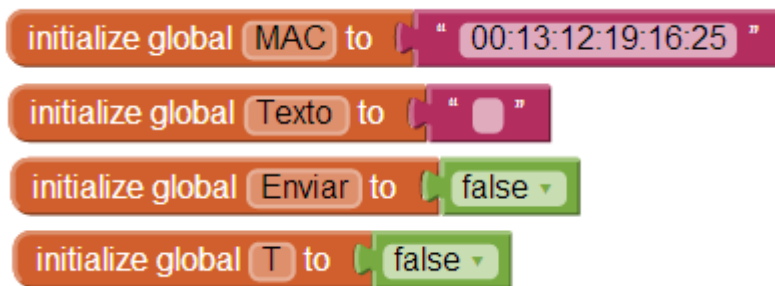


Fig. 3.6 Simulación del sistema de adquisición de datos

3.4 Implementación del sistema de adquisición de datos

En la siguiente figura se muestra diagrama de bloques de la aplicación desarrollada en App Inventor, en esta aplicación lo que se pretende, es mostrar los datos proporcionados por el Arduino utilizando la conexión inalámbrica del módulo bluetooth.



```

when Screen1.Initialize
do
  set ButtonConectar.Enabled to true
  set Clock1.TimerEnabled to false
  set LabelTemperatura.Text to ""
  set LabelHumedad.Text to ""
  set ButtonSalir.Enabled to true

```

```

to RecibirTexto
do
  set global Texto to ""
  if call BluetoothClient1.BytesAvailableToReceive > 0
  then
    set global Texto to call BluetoothClient1.ReceiveText
    numberOfBytes call BluetoothClient1.BytesAvailableToReceive

```

```

when ButtonSalir.Click
do
  close application

```

```

when ButtonConectar.Click
do
  if ButtonConectar.Text = "Conectar"
  then
    if call BluetoothClient1.Connect
    address get global MAC
    then
      set ButtonConectar.Text to "Desconectar"
      set Clock1.TimerEnabled to true
      set global Enviar to true
      set global T to true
      set ButtonSalir.Enabled to false
    else
      call DesconectarBluetooth

```

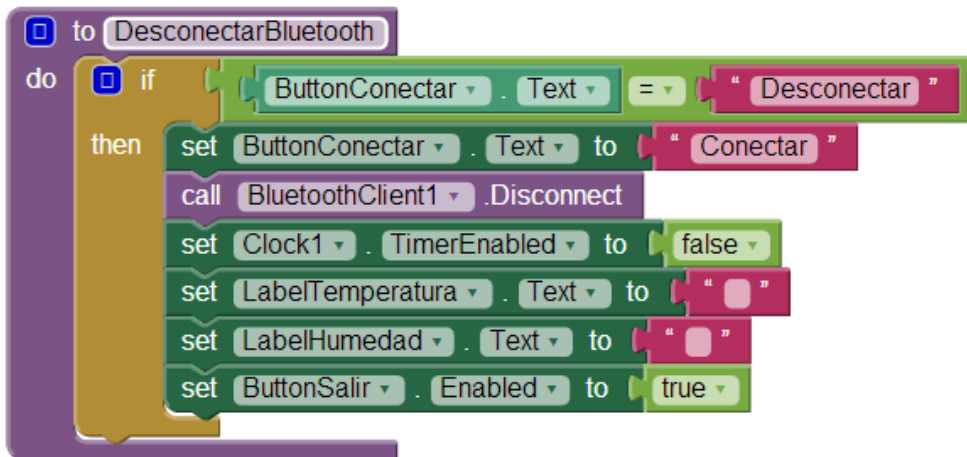
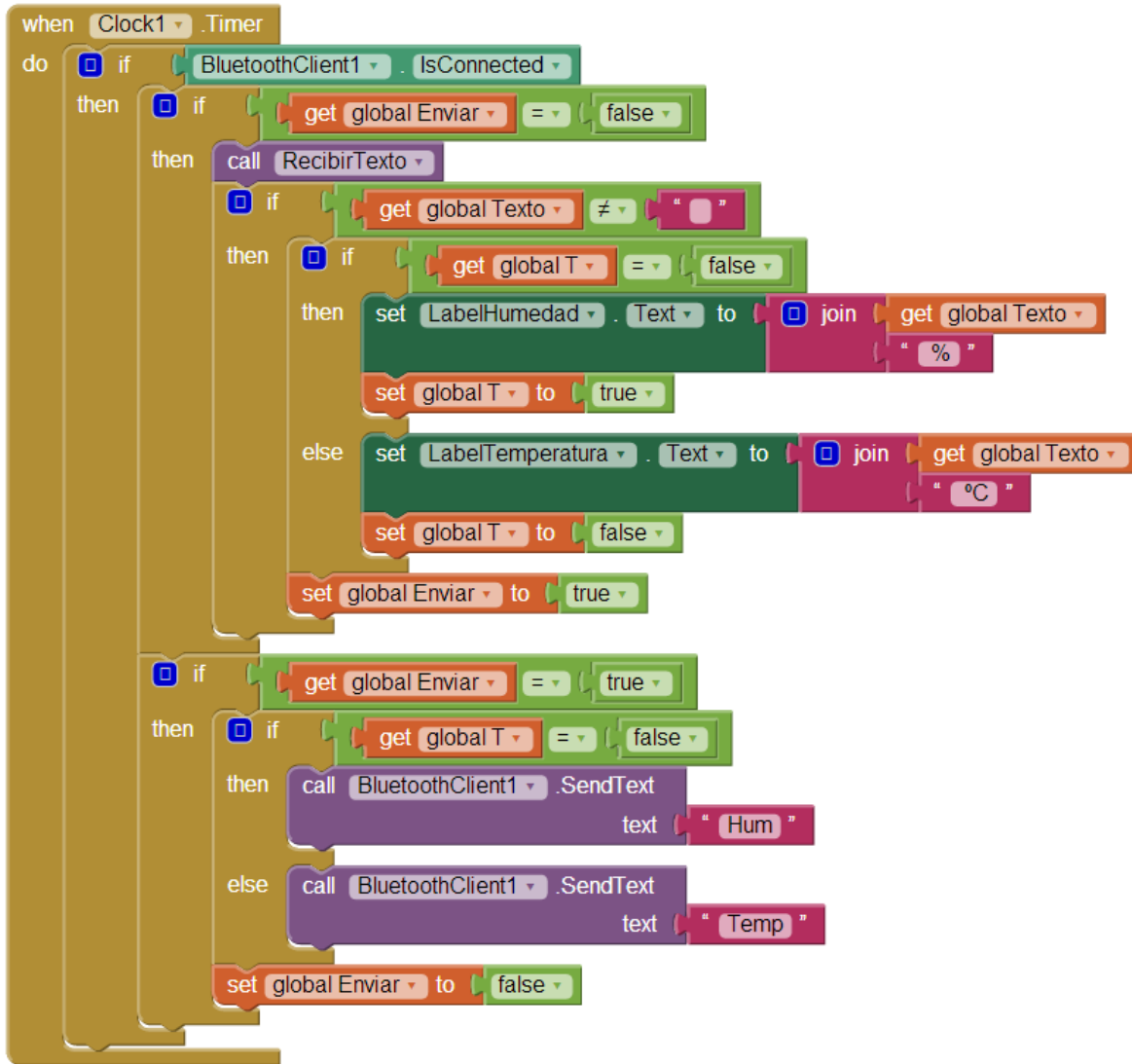


Fig. 3.7 Diagrama a bloques de la aplicación App Inventor

3.5 Materialización del circuito medidor de nivel

En las siguientes figuras podemos ver las placas que se realizaron para el control de potencia de nuestras electroválvulas.

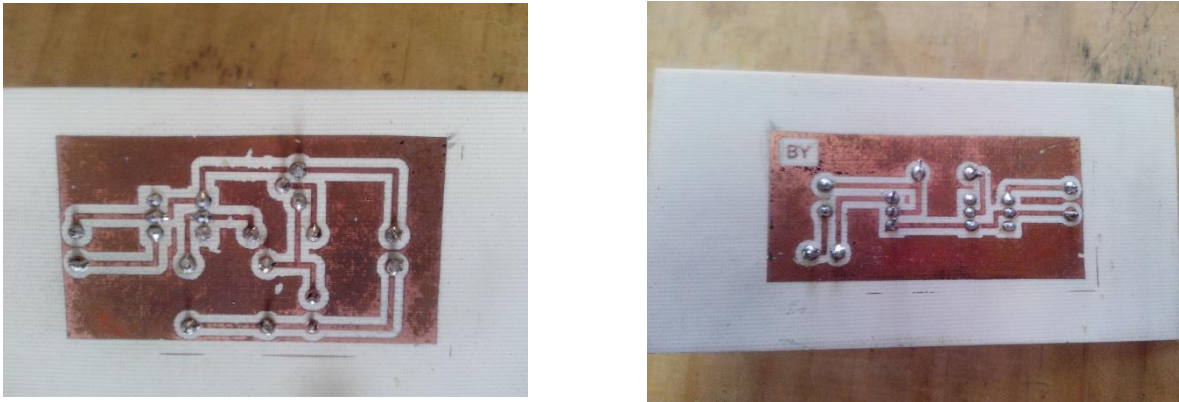


Fig. 3.8 Parte trasera de la placa impresa

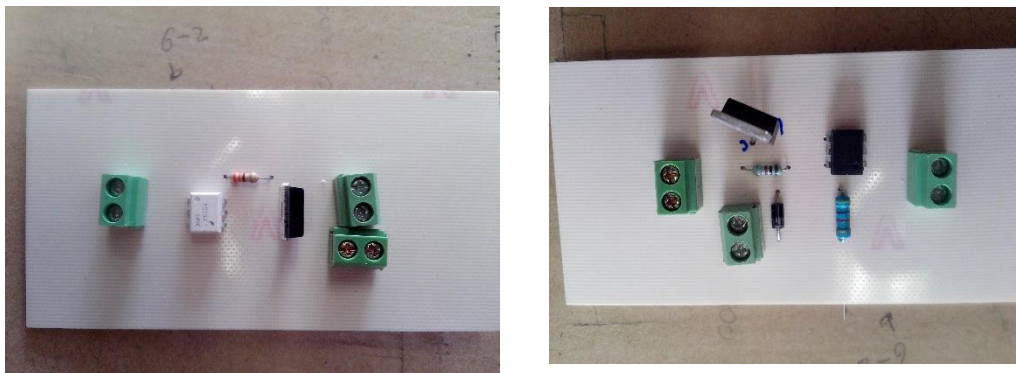


Fig. 3.9 Parte frontal de la placa impresa

A continuación podemos ver los componentes con los que cuenta el medidor de niveles e indicador de nivel que fue la pantalla LCD.

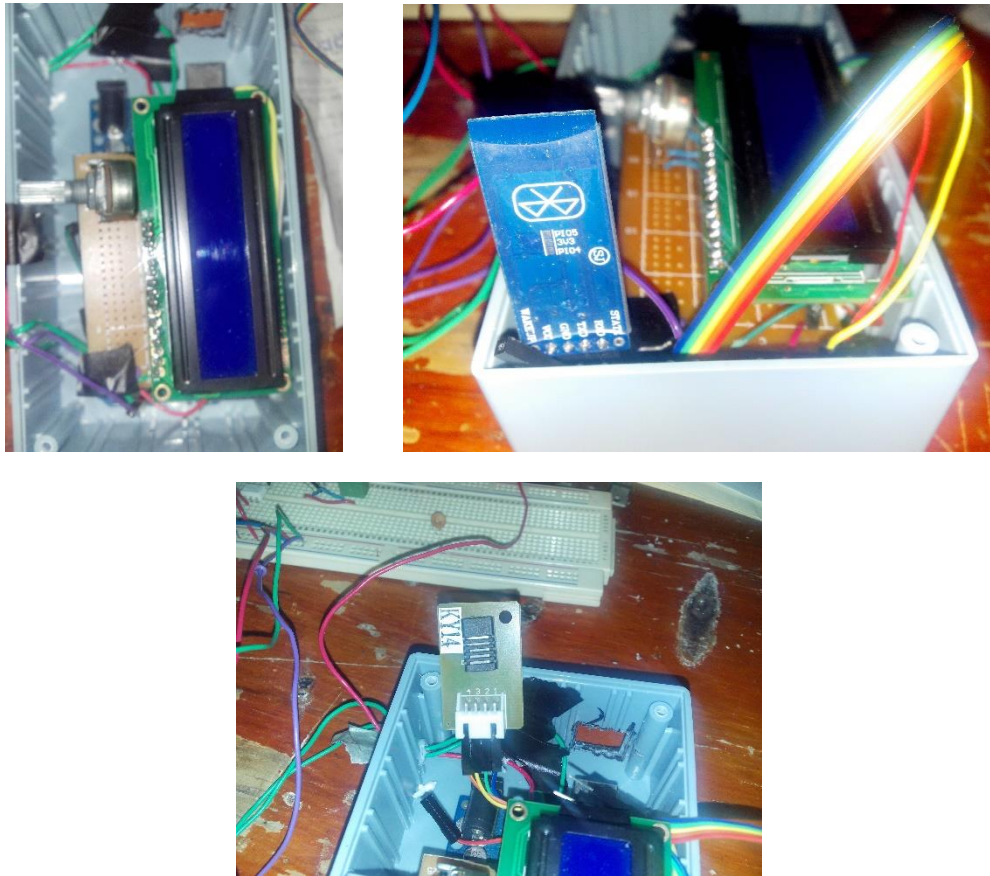


Fig. 3.10 Elementos con el que cuenta en medidor de niveles, dentro de una caja de plástico

3.6 Interfaz gráfica de LabVIEW

A continuación veremos la interfaz gráfica de usuario de LabVIEW, en el cual se podrá ver los valores obtenidos de los sensores en tiempo real y la estimación de la evaporación (Ev)

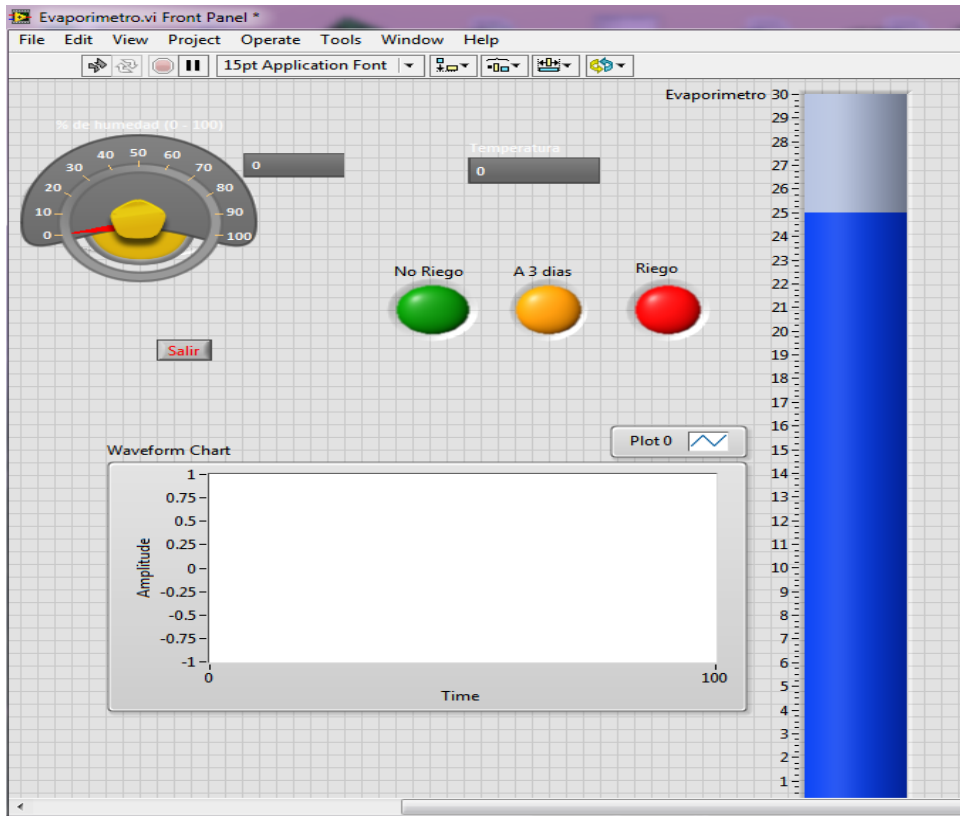


Fig. 3.11 Interfaz de usuario LabVIEW.

CAPÍTULO 4. PRUEBAS Y RESULTADOS

4.1 Pruebas realizadas

En las siguientes imágenes veremos las lecturas que tomamos en nuestro medidor de nivel, en pantalla LCD.

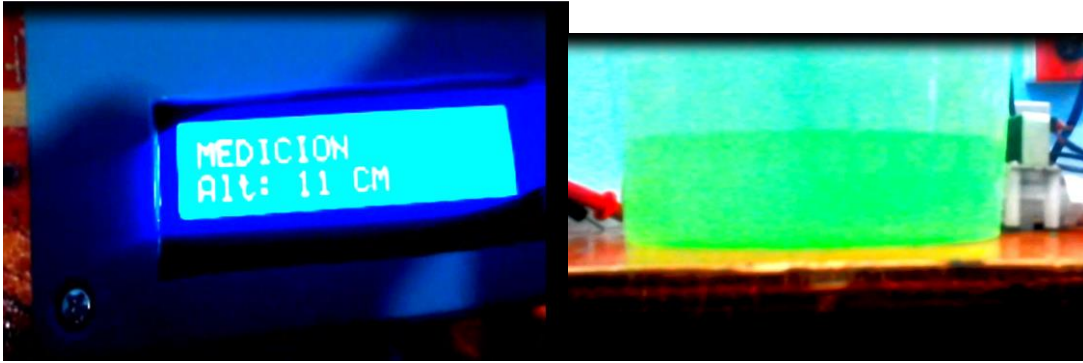
Como podemos ver, cuando el recipiente está vacío la distancia que toma el sensor es mayor, entre menos distancia o altura del agua muestra el sensor, nos indica que el recipiente está llenando hasta llegar a un nivel determinado la electroválvula se apagan y el nivel se mantiene.

Cuando llega nuevamente a un nivel de vaciado del recipiente, es decir, ya se evaporó el agua, las electroválvulas se activan para mantenerlo en un nivel establecido, en este caso se colocó a un nivel de 10 cm que es donde se mantiene el nivel.

En el caso que hubiera lluvias y se rebasara el nivel máximo de agua en el recipiente, colocamos otra electroválvula para el desagüe y así el nivel siempre este en un nivel determinado, es decir en 10 cm.



4.1 Nivel máximo de vaciado del recipiente



4.2 Nivel máximo de llenado del recipiente (11cm)

4.2 Resultados obtenidos

En la siguiente tabla se muestra las lecturas que tomamos de nuestro sensor, para después calcular la evaporación por día. Del día 14 de mayo al 1 de junio del 2015.

DATOS RECIBIDOS			
DIAS	TEMPERATURA (c°)	HUMEDAD (%)	DISTANCIA cm
DIA 1	38	54	12
DIA 2	37	53	13
DIA 3	37	52	11
DIA 4	36	52	11
DIA 5	34	53	12
DIA 6	33	54	12
DIA 7	35	52	11
DIA 8	36	54	11
DIA 9	35	51	10
DIA 10	33	50	13
DIA 11	34	51	10
DIA 12	33	54	12
DIA 13	34	56	11
DIA 14	34	51	10
DIA 15	34	55	11
DIA 16	34	52	12
DIA 17	33	53	11
DIA 18	29	58	13
DIA 19	30	54	12

Tabla 4.1 Datos obtenidos de las mediciones de los sensores

MES DE MAYO	EVAPORACION (mm)
DIA 1 (14/MAY/2015)	81.35 mm
DIA 2 (15/MAY/2015)	81.34 mm
DIA 3 (16/MAY/2015)	81.34 mm
DIA 4 (17/MAY/2015)	81.32 mm
DIA 5 (18/MAY/2015)	81.24 mm
DIA 6 (19/MAY/2015)	81.31 mm
DIA 7 (20/MAY/2015)	81.31 mm
DIA 8 (21/MAY/2015)	81.32 mm
DIA 9 (22/MAY/2015)	81.31 mm
DIA 10 (23/MAY/2015)	81.32 mm
DIA 11 (24/MAY/2015)	81.32 mm
DIA 12 (25/MAY/2015)	81.34 mm
DIA 13 (26/MAY/2015)	81.31 mm
DIA 14 (27/MAY/2015)	81.31 mm
DIA 15 (28/MAY/2015)	81.24 mm
DIA 16(29/MAY/2015)	81.24 mm
DIA 17(30/MAY/2015)	81.32 mm
DIA 18(31/MAY/2015)	81.34 mm
DIA 19 (1/JUN/2015)	81.35 mm

Tabla 4.2 En base a la formula TURC de Ev obtuvimos estos resultados

$$ETR = \frac{P}{\sqrt{0.9 + \frac{P^2}{L^2}}} \dots\dots\dots Ec.2$$

Donde:

ETR= EVAPOTRANSPIRACION EN mm

PRECIPITACION EN mm/MES

L=300+25 T+0.05T³

T=TEMPERATURA



Grafica 4.1 Comportamiento de los datos obtenidos de Evaporación

Conclusiones

En conclusión podemos decir que el proyecto de logro terminar a tiempo, cumpliendo los objetivos que se habían marcado en un comienzo, aunque, cabe mencionar que se tuvieron algunos inconvenientes al estar utilizando los dispositivos que conformarían el proyecto.

Una de los problemas que tuvimos fue en la utilización del sensor ultrasónico, ya que por el margen de error que tiene era complicado estabilizarlo para tener una mejor medición del nivel del agua en el tanque. Con respecto a los demás dispositivos no se encontraron problemas ya que solo era cuestión de configurar las terminales al Arduino.

Con respecto a la comunicación de datos que se realizó podemos decir que fue muy buena, tanto la inalámbrica donde se utilizó los módulos Bluetooth haciendo utilización de una aplicación desarrollada en Android como interfaz de usuario, y en la recepción de datos directamente de Arduino a nuestra PC, teniendo como interfaz de usuario LabVIEW.

Referencias

[1] <http://www.arduino.cc/en/Main/ArduinoBoardMega2560>

[2] <https://www.imta.gob.mx/gaceta/anteriores/g03-07-2007/eficiencias-uso-agua.html>

1.- Comisión Nacional del Agua, “Compendio Básico del Agua en México”, *Programa Nacional Hidráulico 2001-2006*, México, 2002.

2.- Paz Soldán, Gustavo, “El valor Económico del agua”, Conferencia “Francisco Torres H. 2006”, *Tlaloc* Núm. 38, Asociación Mexicana de Hidráulica, México, 2006.

[3] <http://www.ni.com/getting-started/labview-basics/esa/environment#frontPanel>

[4] [http://www.ni.com/labview/esa/ Introduction to LabVIEW](http://www.ni.com/labview/esa/Introduction%20to%20LabVIEW)

[5] Gobierno regional la libertad proyecto especial chavimochic uso del tanque evaporímetro para la agricultura

[6] División de medio ambiente sub gerencia de desarrollo agrícola, febrero 2014 revista informática.

[7] http://www.evolta.cl/product_info.php?products_id=3970&osCsid=16bf5183d20735c6525fedc97fd97055

[8] <http://educalab.es/recursos> INTEF

[9] <http://www.conagua.gob.mx/OCFS07/Contenido/Documentos/mediasgrfs.htm>
Tabla de precipitaciones anual, estado de Chiapas

[10] <http://fjferreer.webs.ull.es/Bibliog/Biblio/Evapotranspiracion.pdf>

Anexos

A1. DataSheet, MOC3011

MOC3009X, MOC3010X, MOC3011X, MOC3012X
MOC3009, MOC3010, MOC3011, MOC3012



OPTICALLY COUPLED BILATERAL SWITCH NON-ZERO CROSSING TRIAC

APPROVALS

- UL recognised, File No. E91231

'X' SPECIFICATION APPROVALS

- VDE 0884 in 3 available lead form :-
- STD
- G form
- SMD approved to CECC 00802

DESCRIPTION

The MOC3009,301_ series are optically coupled isolators consisting of a Gallium Arsenide infrared emitting diode coupled with a light activated silicon bilateral switch performing the functions of a triac mounted in a standard 6 pin dual-in-line package.

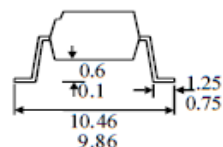
FEATURE

- Options :-
10mm lead spread - add G after part no.
Surface mount - add SM after part no.
Tape&reel - add SMT&R after part no.
- High Isolation Voltage (5.3kV_{RMS}, 7.5kV_{PK})
- 250V Peak Blocking Voltage
- All electrical parameters 100% tested
- Custom electrical selections available

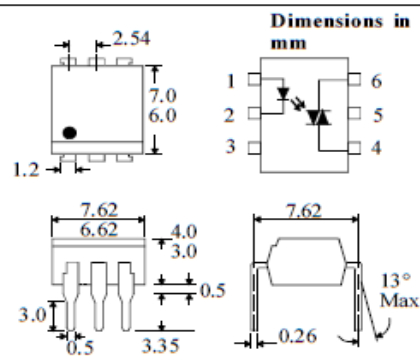
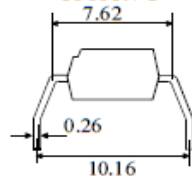
APPLICATIONS

- CRTs
- Power Triac Driver
- Motors
- Consumer appliances
- Printers

OPTION SM SURFACE MOUNT



OPTION G



ABSOLUTE MAXIMUM RATINGS (25 °C unless otherwise noted)

Storage Temperature _____ -55°C-+150°C
Operating Temperature _____ -40°C-+100°C
Lead Soldering Temperature _____ 260°C
(1.6mm from case for 10 seconds)

INPUT DIODE

Forward Current _____ 50mA
Reverse Voltage _____ 6V
Power Dissipation _____ 70mW
(derate linearly 0.93mW/°C above 25°C)

OUTPUT PHOTO TRIAC

Off-State Output Terminal Voltage ____ 250V
Forward Current (Peak) _____ 1A
Power Dissipation _____ 300mW
(derate linearly 4.0mW/°C above 25°C)

POWER DISSIPATION

Total Power Dissipation _____ 330mW
(derate linearly 4.4mW/°C above 25°C)

ELECTRICAL CHARACTERISTICS ($T_A = 25^\circ\text{C}$ Unless otherwise noted)

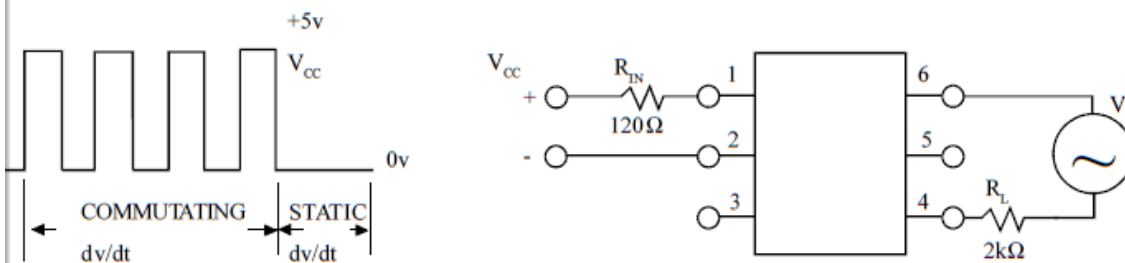
PARAMETER		MIN	TYP	MAX	UNITS	TEST CONDITION
Input	Forward Voltage (V_F)		1.2	1.5	V	$I_F = 10\text{mA}$ $V_R = 6\text{V}$
	Reverse Current (I_R)			100	μA	
Output	Peak Off-state Current (I_{DRM})	250		100	nA	$V_{\text{DRM}} = 250\text{V}$ (note 1) $I_{\text{DRM}} = 100\text{nA}$
	Peak Blocking Voltage (V_{DRM})				V	
	On-state Voltage (V_{TM})		1.5	3.0	V	$I_{\text{TM}} = 100\text{mA}$ (peak) $I_{\text{load}} = 15\text{mA}$, $V_{\text{IN}} = 30\text{V}$ (fig 1.)
	Critical rate of rise of off-state Voltage (dv/dt) (note 1)		10		V/ μs	
	Critical rate of rise of commutating Voltage (dv/dt) (note 1)	0.1	0.2		V/ μs	
Coupled	Input Current to Trigger (I_{PT}) (note 2)					$V_D = 3\text{V}$ (note 2)
	MOC3009			30	mA	
	MOC3010			15	mA	
	MOC3011			10	mA	
	MOC3012			5	mA	
	Holding Current , either direction (I_H)		100		μA	
	Input to Output Isolation Voltage V_{ISO}	5300			V_{RMS} V_{PK}	See note 3 See note 3
		7500				

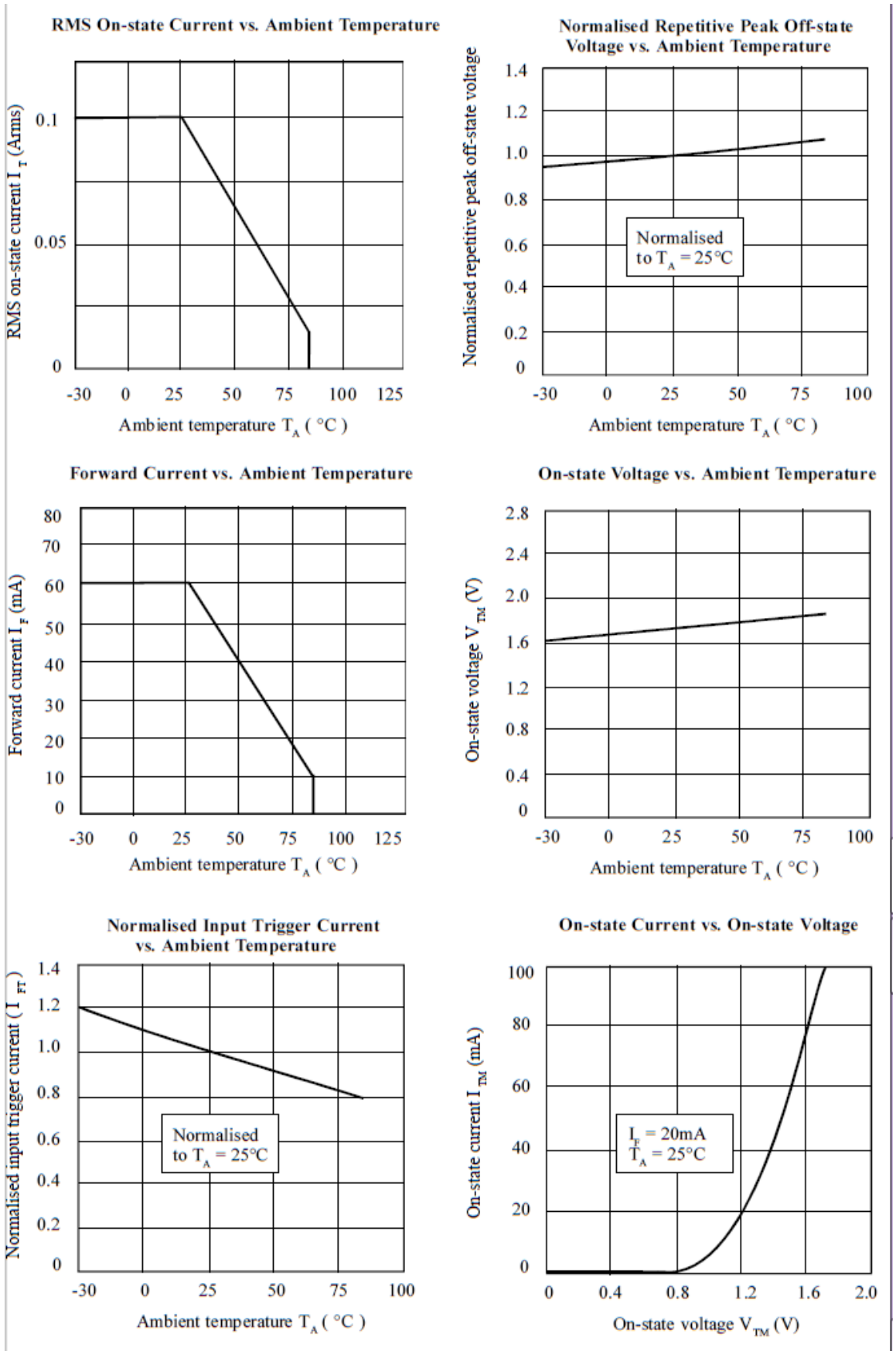
Note 1. Test voltage must be applied within dv/dt rating.

Note 2. Guaranteed to trigger at an I_F value less than or equal to max. I_{PT} , recommended I_F lies between Rated I_{PT} and absolute max. I_{PT} .

Note 3. Measured with input leads shorted together and output leads shorted together.

FIGURE 1

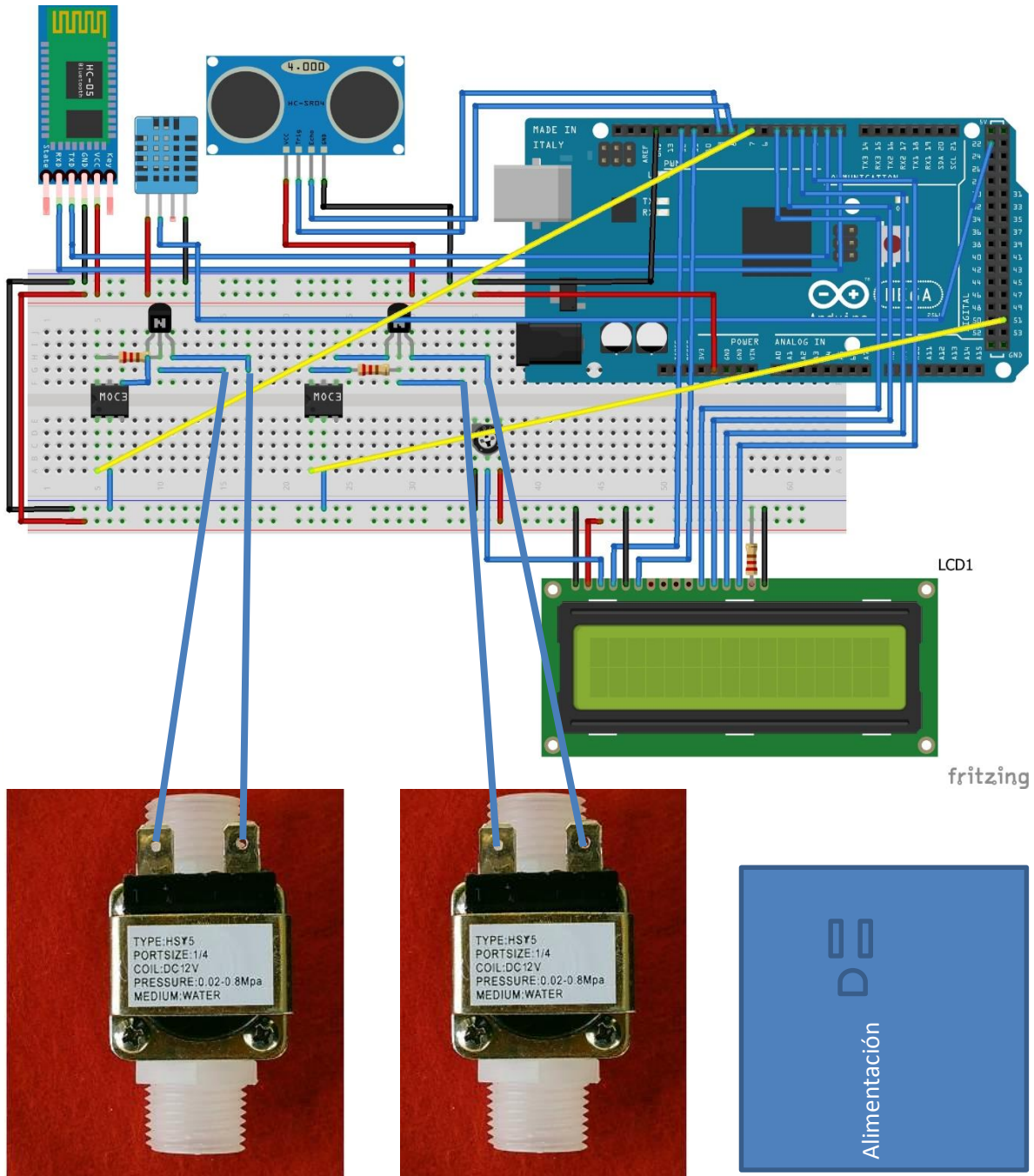




A2. Tabla de radiación solar del hemisferio norte a 10° latitud

Northern Hemisphere												Lat.
Jan	Feb	Mar	Apr	May	Jun	July	Aug	Sep	Oct	Nov	Dec	deg.
0.0	2.6	10.4	23.0	35.2	42.5	39.4	28.0	14.9	4.9	0.1	0.0	70
0.1	3.7	11.7	23.9	35.3	42.0	38.9	28.6	16.1	6.0	0.7	0.0	68
0.6	4.8	12.9	24.8	35.6	41.4	38.8	29.3	17.3	7.2	1.5	0.1	66
1.4	5.9	14.1	25.8	35.9	41.2	38.8	30.0	18.4	8.5	2.4	0.6	64
2.3	7.1	15.4	26.6	36.3	41.2	39.0	30.6	19.5	9.7	3.4	1.3	62
3.3	8.3	16.6	27.5	36.6	41.2	39.2	31.3	20.6	10.9	4.4	2.2	60
4.3	9.6	17.7	28.4	37.0	41.3	39.4	32.0	21.7	12.1	5.5	3.1	58
5.4	10.8	18.9	29.2	37.4	41.4	39.6	32.6	22.7	13.3	6.7	4.2	56
6.5	12.0	20.0	30.0	37.8	41.5	39.8	33.2	23.7	14.5	7.8	5.2	54
7.7	13.2	21.1	30.8	38.2	41.6	40.1	33.8	24.7	15.7	9.0	6.4	52
8.9	14.4	22.2	31.5	38.5	41.7	40.2	34.4	25.7	16.9	10.2	7.5	50
10.1	15.7	23.3	32.2	38.8	41.8	40.4	34.9	26.6	18.1	11.4	8.7	48
11.3	16.9	24.3	32.9	39.1	41.9	40.6	35.4	27.5	19.2	12.6	9.9	46
12.5	18.0	25.3	33.5	39.3	41.9	40.7	35.9	28.4	20.3	13.9	11.1	44
13.8	19.2	26.3	34.1	39.5	41.9	40.8	36.3	29.2	21.4	15.1	12.4	42
15.0	20.4	27.2	34.7	39.7	41.9	40.8	36.7	30.0	22.5	16.3	13.6	40
16.2	21.5	28.1	35.2	39.9	41.8	40.8	37.0	30.7	23.6	17.5	14.8	38
17.5	22.6	29.0	35.7	40.0	41.7	40.8	37.4	31.5	24.6	18.7	16.1	36
18.7	23.7	29.9	36.1	40.0	41.6	40.8	37.6	32.1	25.6	19.9	17.3	34
19.9	24.8	30.7	36.5	40.0	41.4	40.7	37.9	32.8	26.6	21.1	18.5	32
21.1	25.8	31.4	36.8	40.0	41.2	40.6	38.0	33.4	27.6	22.2	19.8	30
22.3	26.8	32.2	37.1	40.0	40.9	40.4	38.2	33.9	28.5	23.3	21.0	28
23.4	27.8	32.8	37.4	39.9	40.6	40.2	38.3	34.5	29.3	24.5	22.2	26
24.6	28.8	33.5	37.6	39.7	40.3	39.9	38.3	34.9	30.2	25.5	23.3	24
25.7	29.7	34.1	37.8	39.5	40.0	39.6	38.4	35.4	31.0	26.6	24.5	22
26.8	30.6	34.7	37.9	39.3	39.5	39.3	38.3	35.8	31.8	27.7	25.6	20
27.9	31.5	35.2	38.0	39.0	39.1	38.9	38.2	36.1	32.5	28.7	26.8	18
28.9	32.3	35.7	38.1	38.7	38.6	38.5	38.1	36.4	33.2	29.6	27.9	16
29.9	33.1	36.1	38.1	38.4	38.1	38.1	38.0	36.7	33.9	30.6	28.9	14
30.9	33.8	36.5	38.0	38.0	37.6	37.6	37.8	36.9	34.5	31.5	30.0	12
31.9	34.5	36.9	37.9	37.6	37.0	37.1	37.5	37.1	35.1	32.4	31.0	10
32.8	35.2	37.2	37.8	37.1	36.3	36.5	37.2	37.2	35.6	33.3	32.0	8
33.7	35.8	37.4	37.6	36.6	35.7	35.9	36.9	37.3	36.1	34.1	32.9	6
34.6	36.4	37.6	37.4	36.0	35.0	35.3	36.5	37.3	36.6	34.9	33.9	4
35.4	37.0	37.8	37.1	35.4	34.2	34.6	36.1	37.3	37.0	35.6	34.8	2
36.2	37.5	37.9	36.8	34.8	33.4	33.9	35.7	37.2	37.4	36.3	35.6	0

A3. Circuito virtual del medidor de niveles



A4. Programa grabado en ATmega2560

```
#include <Ultrasonic.h>
#include <LiquidCrystal.h>
#include <DHT.h>
#define DHTPIN 22
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // Librería del LCD
Ultrasonic ultrasonic(9,8); // Pines del sensor (Trig PIN,Echo PIN)
void setup()
{
  lcd.begin(16, 2);
  pinMode(7,OUTPUT);
  pinMode(51,OUTPUT);
  dht.begin();
  int h = dht.readHumidity();// Lee la humedad
  int t= dht.readTemperature();//Lee la temperatura

}
void loop()
{
  //////////////////////////////////////Control de la electroválvula
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("MED=");
  lcd.setCursor(5,0);
  lcd.print("Niv:");
  lcd.setCursor(10,0);
  lcd.print(ultrasonic.Ranging(CM)); // Medida en centímetros
  lcd.print(" CM");
  delay(500);
  if (ultrasonic.Ranging(CM)<=10) // Condición de apagado de la bomba
  {
    digitalWrite(7, LOW); // Bomba apagada
  }
  if (ultrasonic.Ranging(CM)>17) // Condición de encendido de la bomba
  {
    digitalWrite(7, HIGH); //activar la bomba o electroválvula
  }
  //////////////////////////////////////Control de la electroválvula y/o bomba
```

```

lcd.clear();
lcd.setCursor(0,0);
lcd.print("MED=");
lcd.setCursor(5,0);
lcd.print("Niv:");
lcd.setCursor(10,0);
lcd.print(ultrasonic.Ranging(CM)); // Medición de la altura en centímetros
lcd.print(" CM");
delay(500);
if(ultrasonic.Ranging(CM)<10) // Condición de encendido de la electroválvula
{
digitalWrite(51, HIGH); // Electroválvula encendido
}
else
{
digitalWrite(51, LOW); // electroválvula apagada
}
}

//////////Humedad
lcd.setCursor(1,1);
lcd.print("H: ");
lcd.setCursor(4,1);
lcd.print(h);//Escribe la humedad
lcd.setCursor(6,1);
lcd.println("%");
//////////Temperatura
lcd.setCursor(8,1);
lcd.print("T: ");
lcd.setCursor(11, 1);
lcd.print(t);//Escribe la temperatura
lcd.print(13,1);
lcd.println("°C"); //delay (150);
}
} // tll es el pin 7 va a r2 de diagrama, 5v va a positivo

```


A5. Programación (configuración) del módulo Bluetooth

/*Cambio de configuración del módulo Bluetooth mediante comandos AT

Para ello el modulo no debe estar vinculado con ningún dispositivo

Bluetooth, en mi caso el led debe estar en modo intermitente, si esta

Vinculado aparece encendido de forma permanente*/

```
int contador=1;
```

```
void setup()
```

```
{
```

```
  //Led 13 para indicar final de operación de configuración AT
```

```
  pinMode(13,OUTPUT);
```

```
  //Velocidad del módulo Bluetooth, 9600 por defecto
```

```
  Serial.begin(9600);
```

```
  //Apagamos el led 13
```

```
  digitalWrite(13,LOW);
```

```
}
```

```
void loop()
```

```
{
```

```
  //Es para realizar los cambios una sola vez
```

```
  while (contador==1)
```

```
  {
```

```
    //Indicación de tiempo de espera iniciado
```

```
    digitalWrite(13,HIGH);
```

```
    /*Tiempo de espera de 15 segundos (prudencial, se puede cambiar, depende de lo que tardes
```

```
    En volver a conectarlos) para reconectar cables RX y TX del módulo Bluetooth
```

A la placa Arduino ya que para programar esta deben estar desconectados*/

```
delay(15000);  
//Indicación de tiempo de espera finalizado  
digitalWrite(13,LOW);  
//Iniciamos comunicación con módulo Bluetooth mediante comandos AT  
Serial.print("AT");  
//Espera de 1 segundo según datasheet entre envió de comandos AT  
delay(1000);  
//Cambio de nombre donde se envía AT+NAME y seguido el nombre que deseemos  
Serial.print("AT+NAMEpatagoniatec");  
//Espera de 1 segundo según datasheet entre envió de comandos AT  
delay(1000);  
/*Cambio de la velocidad del módulo en baudios  
Se envía AT+BAUD y seguido el número correspondiente:
```

1 --> 1200 baudios

2 --> 2400 baudios

3 --> 4800 baudios

4 --> 9600 baudios (por defecto)

5 --> 19200 baudios

6 --> 38400 baudios

7 --> 57600 baudios

8 --> 115200 baudios

```
*/  
Serial.print("AT+BAUD4");  
//Espera de 1 segundo según datasheet entre envió de comandos AT  
delay(1000);  
//Configuración Password, se envia AT+PIN y seguido password que queremos  
Serial.print("AT+PIN1234");  
//Espera de 1 segundo según datasheet entre envió de comandos AT  
delay(1000);  
//Mostramos tanto por puerto serial y por led la finalización de la  
//configuración AT del módulo Bluetooth  
Serial.print("OK Cambios Realizados correctamente");  
digitalWrite(13,HIGH);  
//Al tener contador=2 ya no se vuelve a repetir el while, a no ser que  
//se produzca un reset, por tanto comenzaría un nuevo cambio de configuración  
contador=2;  
}  
}
```