



UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE ELECTROTECNIA Y COMPUTACIÓN

Trabajo monográfico para optar a título de Ingeniero Electrónico

“Sistema de control automatización mediante red de comunicación inalámbrica para llenado de tanque de agua potable en el barrio anexo primero de Mayo”

Autor (es):

Br. Carlos Aly Jarquín Cárcamo	Carnet - 2005-20810
Br. Eddy Lucio Gaitán Centeno	Carnet - 2001-10481
Br. Jorge Yali Díaz Murillo	Carnet - 2005-20659

Tutor (a):

MSc. Dora Inés Reyes Chávez

Noviembre 2023

Managua, Nicaragua

Dedicatoria

Primeramente, a Dios por cada día de vida que nos ha regalado, por la oportunidad de culminar una etapa de nuestras vidas y seguir adelante.

A nuestras familias, Padres, madres, esposas, hijos quienes nos aconsejan y son nuestros soportes como por igual somos un ejemplo de superación sin importar la adversidad y malos momentos.

A nuestros profesores quienes compartieron su tiempo, sabiduría y conocimiento lo que es el camino para nuestra logro personal y profesional.

A todos quienes nos han aportado para hacernos mejores personas y profesionales.

Índice de contenido

I: Introducción	2
II: Objetivos	3
2.1: Objetivo General	3
2.2: Objetivos Especifico	3
III: Marco teorico	6
3.1: Sistema de control.....	6
3.1.1: Sistema de control lazo abierto	7
3.1.2: Sistema de control lazo cerrado	8
3.2: Elementos de control para llenado de tanque en lazo cerrado	9
3.2.1: Sensor	9
3.2.2: Actuador	10
3.2.3: Arduino	11
3.2.4: Electrovalvula	12
3.2.5: Almacenamiento de agua potable	14
IV: Diseño Metodológico	15
4.1: Analisis comparativo PLC/Placa Arduino	17
4.2: Variable del sistema	18
4.3: Diagrama de flujo	20
V: Sistema de control mediante placa Arduino	22
5.1: Programa del sistema de control - Placa arduino estación	23
5.2: Esquema del sistema de control - Placa arduino estación	24
5.3: Programa del sistema de control - Placa arduino cliente	22
5.4: Esquema del sistema de control - Placa arduino cliente	28
VI: Presupuesto para sistema de control	32
VII: Conclusiones	34
VIII: Recomendaciones	34
IX: Bibliografía	35

I. Introducción

El presente trabajo consiste en la propuesta de un sistema que permite ofrecer opción para controlar de manera automatizada el proceso de llenado de recipientes de media o alto almacenamiento de agua potable en área de 50 viviendas aproximadas del barrio **anexo primero de mayo** del departamento Managua. Un aspecto fundamental es que los usuarios del agua tengan suficiente recurso en cantidad y calidad que les permita realizar sus actividades domésticas cotidianas para su bienestar a escala humana, pero también que el recurso hídrico sea aprovechado de manera racional.

Las *Normas de Diseño de Sistemas de Abastecimiento y Potabilización del Agua* Nicaragüense en su capítulo II, establece valores de dotación doméstica para la ciudad de Managua (ver tabla 2-1) en la que se clasifican los barrios en función de algunos factores socioeconómicos y el área total para obtener un estimado de consumo diario.

Para el desarrollo de este proyecto se dividirá en etapa de investigación, diseño del sistema en donde se considera una placa electrónica con un microcontrolador que posee recursos de programación para diversos ámbitos de la electrónica moderna y finalmente la búsqueda de recursos (sensores, actuadores, etc.) en conjunto al estudio de gasto para su correcta implementación.

Esta propuesta en sí pretende controlar el nivel en donde se almacena el agua potable mediante sensor indicando dos niveles diferentes para activar o desactivar los actuadores.



II. Objetivos

2.1. Objetivo General

Diseñar sistema de control automatizado mediante red de comunicación inalámbrica para llenado de tanques de almacenamiento de agua potable en barrio anexo primero de mayo.

2.2 Objetivos Específicos

1. Recopilar información sobre condiciones en suministro de agua potable en barrio anexo primero de mayo.
2. Definir una red de distribución para el llenado de tanques de almacenamiento del suministro de agua potable en los hogares del barrio primero de mayo.
3. Implementación sistema de control automatizado a través de sensor en tanques, actuadores y la utilización de un controlador (ARD-ESP8266)



III. Marco teórico

3.1 Sistema de control

Los sistemas de control automáticos son capaces de iniciar y detener procesos sin la intervención manual del usuario. para ello necesitaran recibir información de exterior, procesarla y emitir una respuesta. A esto se le llama un sistema de control.

El objeto de un sistema de control automático es mantener bajo control (de allí que se denominan variables controladas) una o más salidas del proceso. Se utiliza la palabra proceso en un sentido muy general, entendiendo que el mismo es el conjunto de fenómenos físicos que determinan la producción de las variables controladas.

Desde el punto de vista matemático, el proceso quedará representado por un conjunto de relaciones fundamentales, a través de las cuales las variables controladas quedan puestas en función de dos tipos de variables independientes:

- Variables aleatorias

Las variables aleatorias son aquellas variables que escapan a cualquier control o posibilidad de manipulación, es decir, que adoptan valores que pueden variar al azar dentro de ciertos límites prácticos, constituyen perturbaciones, pues una vez obtenidos los valores deseados en las variables controladas, se tiende a apartarlas de los mismos.

- Variables manipuladas

Una variable manipulada es una variable que cambiamos o manipulamos para comprobar o experimentar las afectaciones en alguna otra variable.

Para poder introducir cualquier grado de control, se deberá disponer de variables sobre cuyos valores sea posible operar; de allí que se denominen variables manipuladas. Son precisamente estas variables las que permiten gobernar el sistema, y su característica esencial es que pueden ser manejadas a voluntad dentro de ciertos límites.

El problema de controlar el proceso consiste en eliminar los efectos de las perturbaciones producidas por el cambio de las variables aleatorias, mediante la introducción de variaciones compensatorias en las variables manipuladas.

3.1.1 Sistema de control lazo abierto

Los sistemas de control de lazo abierto no utilizan una señal de retroalimentación y, por lo tanto, no son tan precisos ni estables como los sistemas de control de lazo cerrado. Sin embargo, aún tienen su uso en ciertas aplicaciones donde la precisión no es crítica.

En un sistema de lazo abierto el controlador es colocado en serie con el proceso, con el objetivo de poder manipularlo y, sobre todo, intentarlo llevar a la zona de operación deseada, pero sin medición o senso del estado actual de las variables del proceso (temperatura, velocidad, humedad, concentración, etc)

Es decir, que el sistema de control de lazo abierto interviene sobre el proceso únicamente por medio del conocimiento previo que se tiene del sistema.

Los sistemas de control en lazo abierto son simples y muy baratos de implementar, pero tienen la desventaja que no compensan las posibles variaciones que puede tener la planta, ni las posibles perturbaciones externas. Control Automático Educación C, S. (2023).

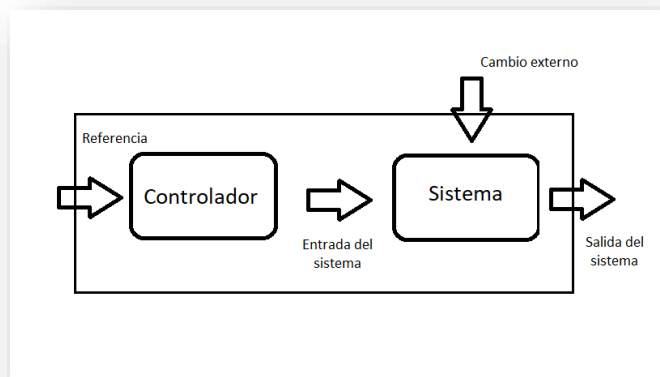


Figura 1. Sistema de control lazo abierto

Fuente: Propia Autoría

Un ejemplo de sistema de lazo abierto es el semáforo. La señal de entrada es el tiempo asignado a cada luz (roja, amarilla, verde) de cada una de las calles. El sistema cambia las luces según el tiempo indicado, sin importar que la cantidad de tránsito varíe en las calles.

3.1.2 Sistema de control lazo cerrado

Los sistemas de control de lazo cerrado son ampliamente utilizados en una variedad de industrias para regular y controlar procesos automatizados. Un sistema de control de lazo cerrado, también conocido como sistema de control de realimentación, utiliza una señal de retroalimentación para comparar el resultado deseado con el resultado actual y ajustar el proceso en consecuencia. Esto permite una mayor precisión y estabilidad en comparación con los sistemas de control de lazo abierto.

En un sistema en lazo cerrado se puede colocarse un medidor y transmisor de temperatura que realimente el sistema hacia el controlador y mantenga de esa forma la temperatura en su lugar deseado.

Este valor deseado se conocerá como el Setpoint y será el único valor que será modificado por el operario. Estos sistemas de control son clasificados como sistemas con retroalimentación o feedback.

Un ejemplo típico de un sistema de control de lazo cerrado es un controlador PID, que es utilizado para controlar la temperatura en un horno o la velocidad de un motor eléctrico. El controlador PID utiliza un algoritmo matemático para calcular la diferencia entre la temperatura deseada y la temperatura actual, y ajusta el proceso en consecuencia para lograr una mayor precisión y estabilidad. Control Automático Educación C, S. (2023).

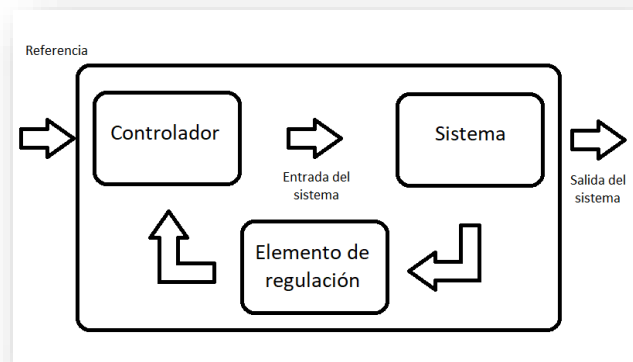


Figura 2. Sistema de control en lazo cerrado

Fuente: Propia Autoría

3.2 Elementos para control de llenado de tanque en lazo cerrado

3.2.1 Sensor

Componente que detecta indicadores externos o internos, esto pueden ser intensidad de luz, sonido, nivel de líquidos, etc. En términos estrictos, un sensor es un instrumento que no altera la propiedad censada.

Por ejemplo, un sensor de temperatura sería un instrumento tal que no agrega ni cede calor a la masa censada, es decir, en concreto, sería un instrumento de masa cero o que no contacta la masa a la que se debe medir la temperatura (un termómetro de radiación infrarroja). Omega (s. f.)

3.2.1.1 Sensor de nivel

El sensor de nivel es un dispositivo electrónico que mide la altura del material (líquido), dentro de un tanque u otro recipiente. Integral para control de procesos en muchas industrias, dicho sensor se divide en dos tipos principales.

- Sensor de nivel de punto: se utiliza para marcar una altura de un líquido en un determinado nivel preestablecido, generalmente funciona como alarma indicando un sobre llenado cuando el nivel determinado ha sido adquirido o contrario un bajo nivel.

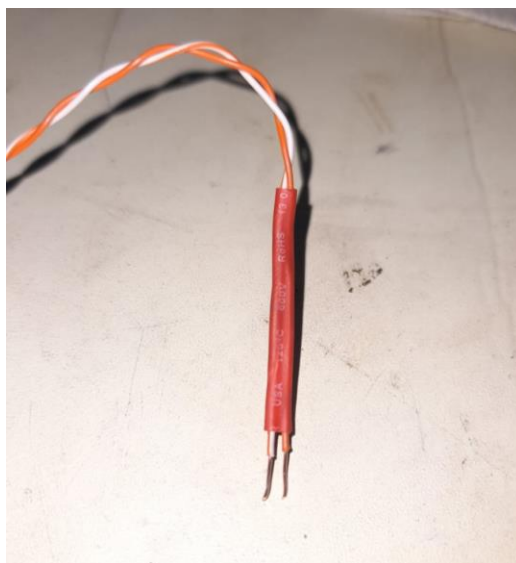


Figura 3. Sensor de nivel de un punto conductivo

Fuente: Propia Autoría

- Sensor de nivel continuo: son equipos más complejos que pueden realizar el seguimiento del nivel de todo un sistema. Estos miden el nivel de fluido dentro de un rango específico, en lugar de un único punto, produciendo una salida analógica que se correlaciona directamente con el nivel en el recipiente.



Figura 4. Sensor de nivel de un punto conductivo

Fuente: <https://www.industriasasociadas.com/producto/sensor-de-nivel-continuo-nsf-f/>

3.2.2 Actuadores

Un actuador es un dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre elemento externo. Este recibe la orden de un regulador, controlador o en nuestro caso un Arduino y en función a ella genera la orden para activar un elemento final de control como, por ejemplo, una válvula.

Existen varios tipos de actuadores como son:

- Electrónicos
- Hidráulicos
- Neumáticos
- Eléctricos
- Motores
- Bombas

Para cada actuador, necesitamos un “driver” para poder mandar órdenes desde Arduino. Jecrespom. (2021, February 16)



Figura 5. Válvulas solenoide y bomba de agua

Fuente: Propia Autoría

3.2.3 Arduino

Arduino es una plataforma de desarrollo basada en una placa electrónica de hardware libre que incorpora un microcontrolador reprogramable y una serie de pines hembra. Estos permiten establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores de una manera muy sencilla. Aguayo, P. (2023). ¿Qué es Arduino?

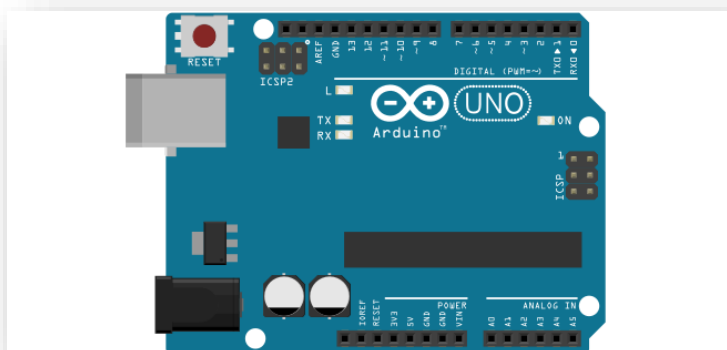


Figura 6. PCB de Arduino

Fuente: <https://arduino.cl/que-es-arduino/>

Gracias a su alcance hay una gran comunidad trabajando con esta plataforma. Así se genera una cantidad de documentación bastante extensa, la cual abarca casi cualquier necesidad.

Su lenguaje de programación basado en C++ es de fácil comprensión. C++ permite una entrada sencilla a los nuevos programadores y a la vez con una capacidad

tan grande, que los programadores más avanzados pueden exprimir todo el potencial de su lenguaje y adaptarlo a una necesidad.

Tiene gran versatilidad y reusabilidad porque una vez terminado el proyecto es muy fácil poder desmontar los componentes. Aguayo, P. (2023). ¿Qué es Arduino?

3.2.4 Electroválvula

Es un dispositivo encargado de controlar el flujo de líquido que viaja a través de una tubería o conducto.

Las electroválvulas son consideradas también como válvulas electromecánicas debido a que son controladas por una corriente eléctrica a través de una bobina solenoide. son más fáciles de controla mediante programas de software. (QimiNet, 2011)

Son utilizadas en gran número de sistemas y rubros industriales que manejan líquidos como el agua, aceites livianos, etc.

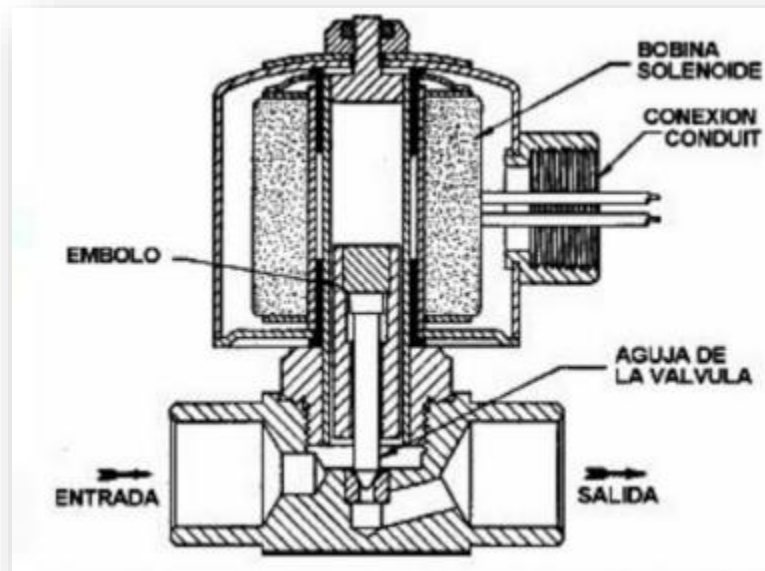


Figura 7. Elementos de electroválvulas

Fuente: <https://blog.valvulasarco.com/electrovalvulas>

3.2.4.1 Aspectos para elección de válvula

La elección adecuada de una válvula solenoide es esencial para garantizar el funcionamiento óptimo de cualquier sistema de control industrial. Hay varios aspectos importantes que debes considerar al elegir una válvula solenoide, y a continuación, te daremos algunos consejos para hacerlo correctamente. Oloarte, G. (2023, May 19)

- **Tamaño y tipo de la válvula:** Lo primero que debes considerar es el tamaño (diámetro de conexión) y el tipo de la válvula solenoide (2 vías, 3 vías, 5 vías, NC, NO) que necesitas. Esto dependerá de las condiciones de operación y del flujo requerido en el sistema. Si se trata de una instalación preexistente, el tamaño de la válvula estará ya definido, de lo contrario será necesario calcularlo mediante el método del factor de flujo. La operación está definida en una instalación preexistente; sin embargo, en una instalación nueva, existen ciertos criterios para determinar si la válvula deber ser normalmente cerrada o normalmente abierta.
- **Presión de trabajo:** La presión de trabajo es otro factor importante que debes considerar al seleccionar una válvula solenoide. Es importante asegurarse de que la válvula pueda manejar la presión del sistema sin sufrir daños o fallas en el funcionamiento por lo cual esta presión debe estar dentro del rango de presión diferencial de la válvula. El rango de presión diferencial está siempre especificado en la documentación técnica de la válvula, así que asegúrate de revisarla cuidadosamente.
- **Temperatura:** La temperatura es otro factor crítico a considerar al elegir una válvula solenoide. Algunas válvulas están diseñadas para manejar temperaturas extremadamente altas o bajas, mientras que otras son más adecuadas para temperaturas moderadas. Asegúrate de que la válvula que elijas pueda manejar la temperatura del fluido que estás controlando sin dañarse.
- **Materiales de construcción:** Los materiales de construcción son importantes cuando se trata de elegir la válvula solenoide adecuada. La selección de los materiales dependerá no solo de la temperatura y la presión del sistema sino también de la composición química del fluido que se está controlando. Los materiales comunes utilizados en los cuerpos de válvulas solenoide incluyen latón, acero inoxidable, bronce, aluminio y plástico, mientras que los sellos o asientos podrían ser de PTFE, FKM, EPDM, NBR o algunos más especializados como el Rulón, el HNBR o el Rubí.
- **Funciones especiales:** Por último, pero no menos importante, debes considerar si necesitas alguna función especial en tu válvula solenoide. Algunas válvulas solenoide tienen características adicionales, como bobinas a prueba de explosión y operadores manuales, entre otras.

3.2.5 Almacenamiento del agua potable

El almacenamiento de agua potable juega un papel básico para el diseño del sistema de control automatizado tanto desde el punto de vista económico, así como su importancia en el funcionamiento y en el mantenimiento de un servicio eficiente.

Para almacenamientos de agua potable existen diferentes opciones domesticas las cuales puede ser directamente sobre la superficie del suelo o torre (metálicos o de concreto), cuando por razones de servicio son requerido elevarlos.

Los almacenadores de agua potable movibles de plástico tienen mayor aceptación por ligero, resistente, bajo costo en comparación con otros materiales, adicional Impide que el agua vea alterada sus características físicas y químicas siendo apropiado para su almacenamiento. Aunque siempre considerando que estará expuesto a condiciones medio ambiental



Figura 8. Almacenadores en un sistema de control

Fuente: Control automático de llenado y vaciado de tanque -Equipo. # 9 control - YouTube

IV. Diseño metodológico

Tomando en consideración los siguientes elementos:

- Medio para recopilar datos
- Redacción de las preguntas
- Formato de preguntas
- Precisión de la información
- Presentación visual

([blog/es/7-puntos-importantes-que-debe-considerar-en-el-diseno-de-encuestas](https://blog.es/7-puntos-importantes-que-debe-considerar-en-el-diseno-de-encuestas))

Se elabora un cuestionario de preguntas abiertas y cerradas para obtener información de la población respecto al uso doméstico que le dan al recurso hídrico, además, si la comunidad utiliza el recurso hídrico de manera racional y/o eficiente o por el contrario las actividades domésticas muestran un desperdicio y/o un uso no racional, con las implicaciones que esto significa para el abastecimiento el recurso hídrico. (ver. Anexo)



Figura 9. Vista aérea anexo primero de mayo con abastecimiento de agua potable de 6 horas

Fuente: Google earth 2023

Estimando una vivienda con 4 personas con consumo promedio de agua potable (basado en facturas del municipio de Ciudad Sandino-Nicaragua) entre 15m³ a 30m³, dividiendo entre 28 días el consumo máximo resulta en 1m³ al día y al dividir esto entre 4 habitantes es obtenido el valor 0.25 m³/hab/día, con la conversión m³ a barril disponible en los hogares se concluye que es requerido aproximado de 1.5 barril/hab/día disponible para consumo ducha, inodoro, etc (ver tabla de conversión)

Para realizar el control de nivel de líquidos, en el presente trabajo, se ha previsto la utilización de un prototipo acoplado a una válvula que permite simular un sistema de control de lazo cerrado.

La parte de control contiene placa Arduino ESP8266 que cuenta con una automatización por programación que se sirve de un sistema de adquisición y salida de datos y un modo de operación manual por protocolo de comunicación inalámbrica.

La señal de realimentación al sistema de control proviene de un sensor que registra indirectamente el nivel de líquido en el almacenador, para controlar la variable de salida del lazo de control, que en este caso la altura o nivel del líquido.

En resumen, el equipo cuenta con dos maneras eficientes de realizar el control y que ofrecen un medio didáctico a través del cual se puede analizar y experimentar con la variable a tratar dentro del proceso.

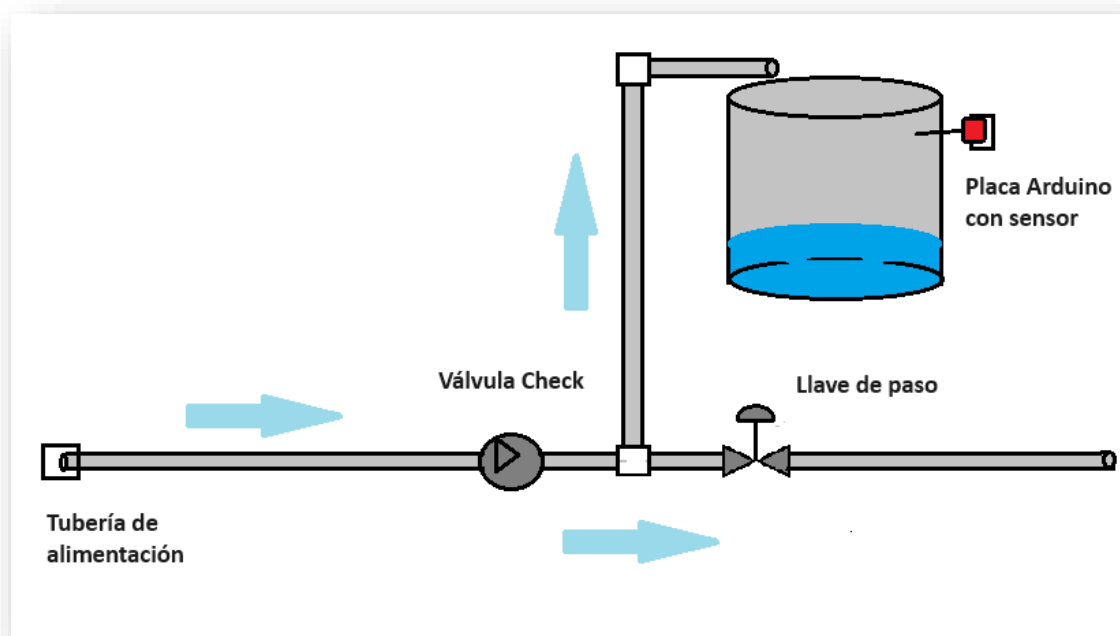


Figura 10. Esquema físico de llenado de almacén del agua potable

Fuente: Propia Autoría



4.1 Análisis comparativo PLC / Placa Arduino

PLC (Controlador Lógico Programable) y una placa Arduino son dos tecnologías utilizadas en sistemas de control, pero tienen diferencias significativas en términos de arquitectura, aplicaciones y características detalladas a continuación:

Arquitectura:

PLC: son dispositivos industriales diseñados específicamente para el control de procesos y automatización industrial. Suelen estar basados en microcontroladores especializados y tienen una construcción robusta para entornos industriales.

Placa Arduino: son dispositivos de desarrollo de código abierto que utilizan microcontroladores AVR o ARM. Según la familia incorporan módulos WiFi para conectividad a Internet.

Aplicaciones:

PLC: se utilizan principalmente en aplicaciones industriales, como líneas de producción, control de maquinaria, sistemas de control de edificios, y automatización de procesos industriales.

Placa Arduino: se utilizan en una variedad de aplicaciones, pero tienden a ser más comunes en proyectos de IoT (Internet de las cosas), domótica, robótica y proyectos de control y monitoreo a pequeña escala.

Programación y Lenguaje de Programación:

PLC: se programan generalmente utilizando lenguajes de programación específicos para PLC, como Ladder Logic o Structured Text. Estos lenguajes están diseñados para la lógica de control industrial.

Placas Arduino: se programan en lenguajes como C/C++ a través del entorno de desarrollo Arduino. Esto les da flexibilidad para implementar una amplia variedad de aplicaciones, pero no están optimizados para tareas de control industrial complejas.

Conectividad:

PLC: suelen ofrecer una variedad de interfaces de entrada/salida para conectarse a sensores, actuadores y sistemas industriales. También pueden admitir protocolos de comunicación industriales como Modbus.

Placas Arduino: están diseñadas para la conectividad a Internet y suelen tener puertos digitales y analógicos para interactuar con sensores y actuadores. Utilizan WiFi o Ethernet para la comunicación en la mayoría de los casos.

Costo:

PLC: tienden a ser más costosos debido a su construcción robusta y características industriales.

Placas Arduino: son más económicas y asequibles, lo que las hace populares para proyectos de bricolaje y educativos.

4.2 Variable del sistema

Las variables de entradas y salidas que se seleccionaron para el sistema son detalladas en las siguientes tablas. Las entradas, así como las salidas deben estar correctamente identificadas en la placa Arduino ESP8266 para poder realizar la conexión respectiva.

Valores Variables					
Entradas				Salida	
Inicio o suministro eléctrico disponible	suministro de agua potable disponible	Flujo de agua	Nivel deseado en cliente	Bomba de Estación	Electroválvula de cliente
No	No	No	No	Off	Off
No	Si	No	No	Off	Off
Si	No	No	No	Off	Off
Si	No	No	Si	Off	Off
Si	Si	Si	No	On	On
Si	Si	Si	Si	Off	Off

Tabla de valores variables del sistema de control

Fuente: Autoría propia



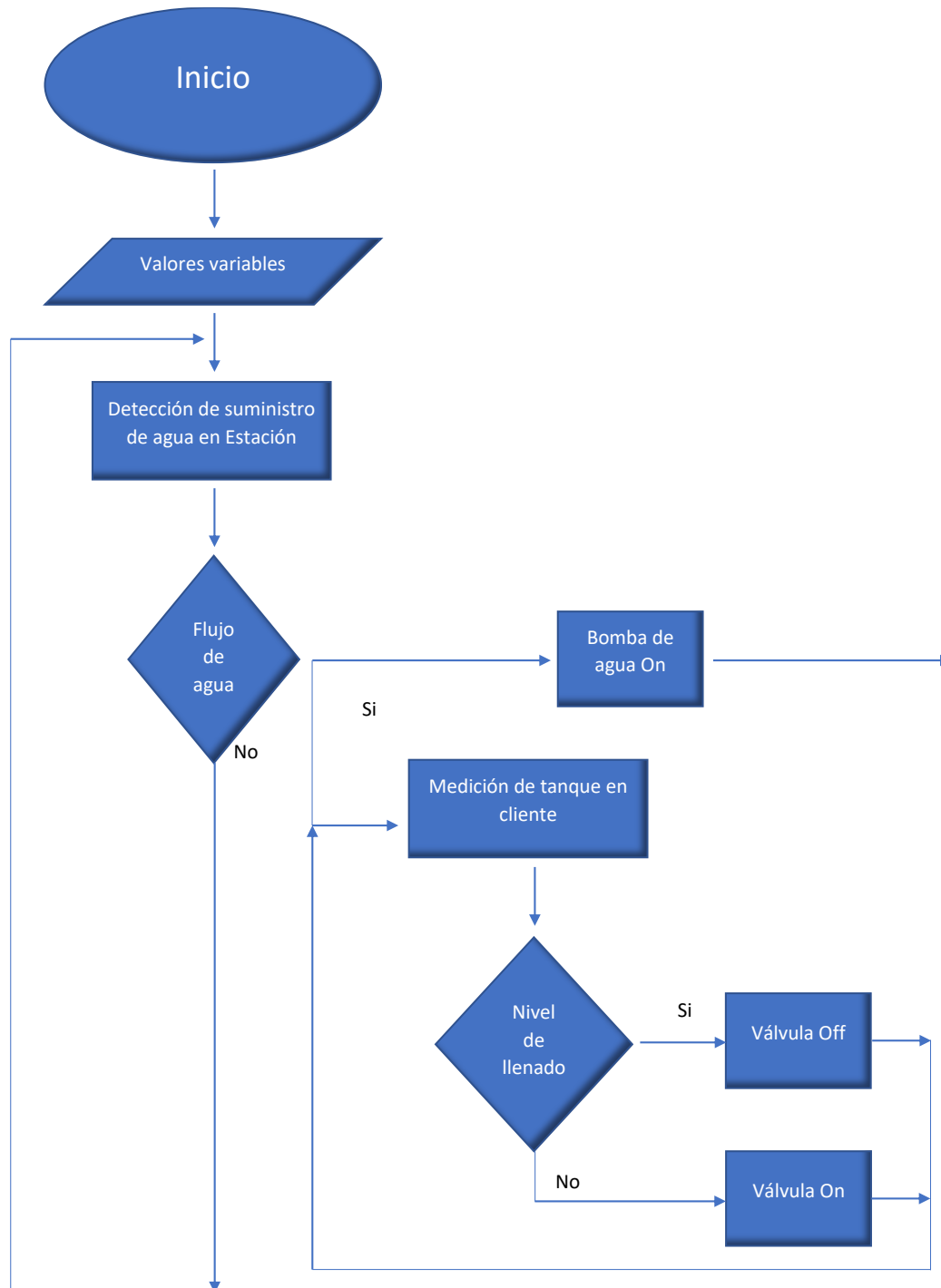
La parte más crítica de todo proceso es la medición exacta de la variable de salida.

En un principio consideramos utilizar un sensor del tipo capacitivo para medir el nivel de agua en el recipiente de almacenado, pero este tipo de sensor depende de la constante dieléctrica del líquido con el que se está trabajando. En este caso el agua tiene una constante dieléctrica que varía con la temperatura y además depende del grado de impurezas que posee el agua, por tal motivo la calibración del acondicionamiento de señal de este tipo de sensor tendría que ser permanente.

Otro tipo de sensor como opción para utilizar es del tipo flotador potenciométrico, pero fue también descartado por tener partes móviles dentro del tanque que con el tiempo se desgastan y deterioran alterando las medidas, además de ser sensible a las ondulaciones en la superficie del líquido.

Finalmente se optó por utilizar un sensor de contacto que carece de las desventajas de los sensores anteriores y siendo de bajo costo dentro de sus limitantes.

4.3 Diagrama de Flujo





Existen dos almacenadores de agua, ambos están conectados por medio de tuberías. El primero cuenta con el control de la bomba y la placa Arduino denominada “estación” en donde el sensor de flujo hace el llamado de activación al obtener 1 lógico en la entrada D1 que indica la afluencia de agua potable. En dado caso, que el sensor de flujo no detecte afluencia de agua mantendrá desactivado la bomba.

Completado el proceso de “Estación” continuamos con la llamada a la placa Arduino denominada “cliente” instalada en el almacenador de agua definido por los usuarios, los cuales mediante un sensor de contacto declaramos un 1 lógico para el nivel deseado de llenado; en dado caso no se cumpla, el 0 lógico establece que la electroválvula sea utilizada para llegar al nivel deseado de llenado.

Es requerido tener en consideración la siguiente fórmula para verificar la tasa de flujo que tendremos para llenado en cada “cliente”:

Tasa de flujo de llenado (en litros o galones por hora) = Capacidad del almacenador de agua (en litros o galones) / Tiempo (en horas)

Donde:

- Tasa de Flujo: Es la cantidad de líquido que por unidad de tiempo (generalmente se mide en litros por segundo, litros por minuto, galones por minuto, etc.).
- Volumen del almacenador de agua: Es la capacidad total, es decir, el volumen máximo que puede contener (generalmente se mide en litros, galones, metros cúbicos, etc.).
- Tiempo de Llenado: Es el tiempo necesario para llenar el almacenador de agua desde un nivel bajo hasta el nivel deseado (se mide en segundos, minutos u horas).

El tiempo activo de la bomba en placa Arduino “Estación” depende de la cantidad de almacenadores llenado al nivel deseado en cada cliente.

V. Sistema de control mediante placa Arduino

A continuación, se presentará sistema de control mediante placa Arduino desde un punto de vista Eléctrico y programación. El microcontrolador ESP8266 cuenta con 30 pines los cuales podrán ser utilizados de manera estratégica como salidas o entradas, para la programación requerida utilizando un lenguaje programación el cual es propio basado en el lenguaje de programación de alto nivel Processing que es similar a C++

La alimentación de los equipos se realizará con fuente de alimentación de conmutada marca RQG con entrada de 120/230V y salida 12V 3A, las cuales fueron adquiridas mediante plataforma de venta y compra "Ali Express" en un periodo de descuento del producto.

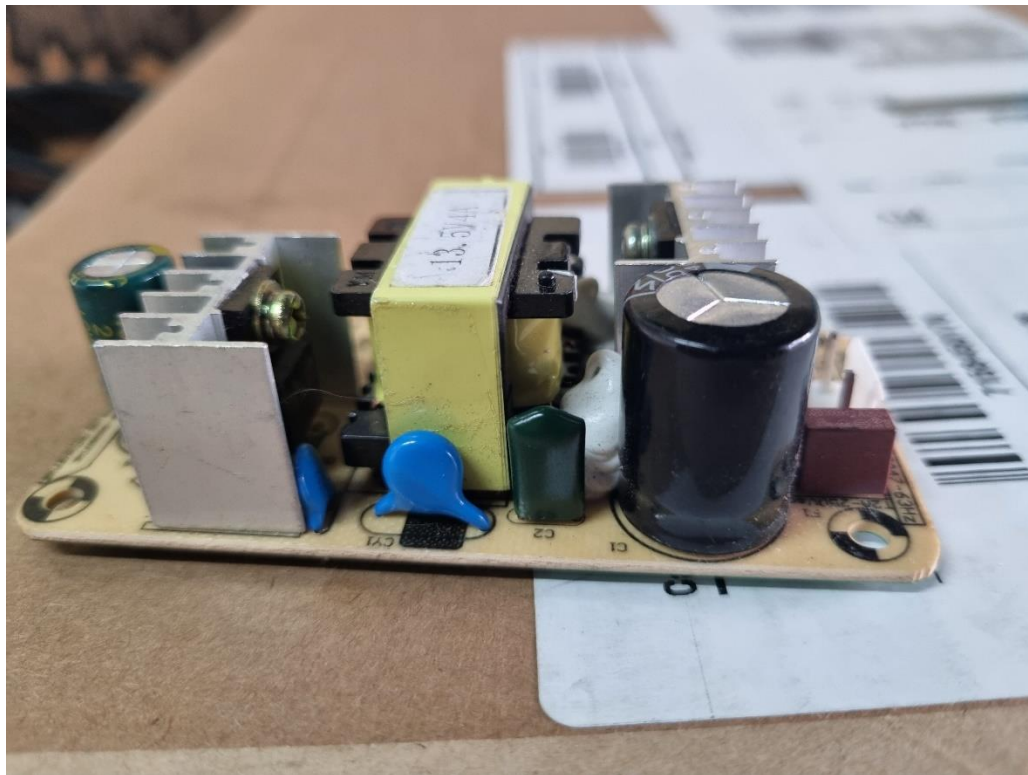


Figura 11. Fuente de alimentación conmutada marca RQG

Fuente: Autoría propia

5.1. Esquema del sistema de control - Placa Arduino “Estación”

El circuito con placa Arduino "Estación" está conformada por las siguientes etapas:

- Fuente de alimentación: se encuentra regulador de voltaje 7805 que transforma 12 V a 5V requerido por la etapa de control.
- Control: Contiene placa Arduino ESP8266 en donde el pin 29 es la entrada del sensor de flujo, el cual, indica la disponibilidad del suministro de agua potable en tubería o almacenador (según la necesidad), activa mediante pin 30 que es la salida de un acoplador la etapa de potencia.
- Potencia: se tiene transistor que genera la activación de bomba de agua.

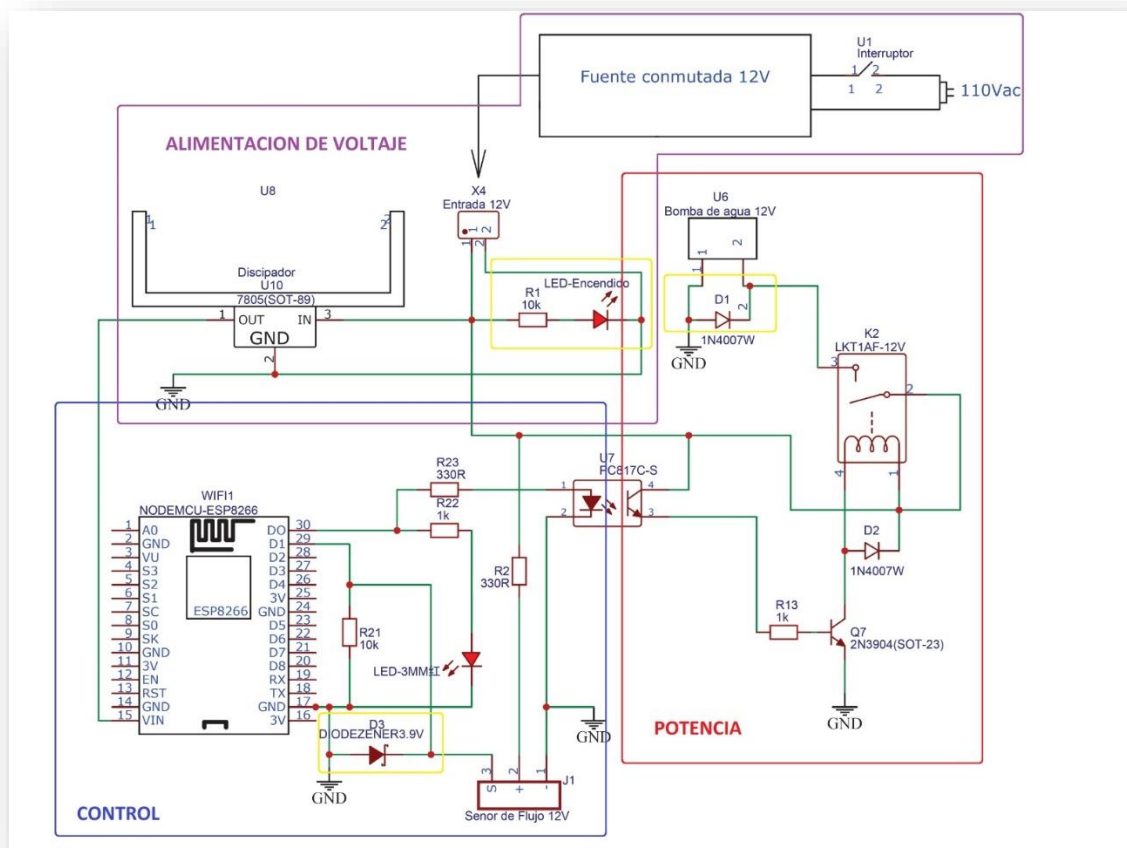


Figura 12. Diagrama de pines placa Arduino “Estación”

Fuente: Autoría propia



5.2 Programa del sistema de control - Placa Arduino “Estación”

```
v//Libraries
#include
<ESP8266WiFi.h>//https://github.com/esp8266/Arduino/blob/master/libraries/
ESP8266WiFi/src/ESP8266WiFi.h
//Constants
#define NUM_SLAVES 2
//#define BOMBAD6
int T1; // tanque 1
int T2;// tanque 2

const int BOMBA= 16;
const int Sa= 5;

int x;
int z;

//Parameters
String nom = "Master";

const char* ssid = "A01"; // COLOCAR RED DONDE SE VA USAR
const char* password = "graciela"; // PASSWORD DONDE SE VA USAR
//Variables
bool sendCmd = false;
String slaveCmd = "0";
String slaveState = "0";
//Object
WiFiServer server(80);
WiFiClient browser;
IPAddress ip(192, 168, 43, 65); // IP CONFIG - arp -a
IPAddress gateway(192, 168, 43,96) ;
IPAddress subnet(255, 255, 255, 0);
void setup() {
  pinMode (BOMBA, OUTPUT);
  pinMode (Sa, INPUT);

  Serial.begin(115200);
  Serial.println(F("Initialize System"));

  pinMode(BOMBA,OUTPUT);
  pinMode (Sa, INPUT);
```




```
}  
void clientRequest() {  
  
    WiFiClient client = server.available();  
    client.setTimeout(50);  
    if (client) {  
        if (client.connected()) {  
  
            Serial.print(" ->"); Serial.println(client.remoteIP());  
            String request = client.readStringUntil('\r'); //receives the message  
from the client  
            if ((request.indexOf("Slave0") == 1)&&(request.indexOf("Slave1") == 1)) {  
                digitalWrite(BOMBA, HIGH);  
                Serial.print("From "); Serial.println(request);  
                int index = request.indexOf(":");  
                String slaveid = request.substring(0, index);  
                slaveState = request.substring(request.indexOf("x") + 1,  
request.length());  
                Serial.print("state received: "); Serial.println(slaveState);  
                client.print(nom);  
                if (sendCmd) {  
                    sendCmd = false;  
                    client.println(": Ok " + slaveid + "! Set state to x" + String(slaveCmd)  
+ "\r");  
  
                } else {  
                    client.println(slaveid + "! \r"); // sends the answer to the client  
  
                }  
                client.stop();  
            } else {  
                Serial.println(request);  
  
                client.flush();  
                if (request=="Slave0: x0") // tanque 1 esta lleno  
                {  
                    T1=1;  
  
                }  
                if (request=="Slave1: x0")// tanque 2 esta lleno  
                {  
                    T2=1;  

```



```
}
  if (request=="Slave0: x1") // tanque 1 esta vacio
  {
    T1=0;

  }
  if (request=="Slave1: x1") // tanque 2 esta vacio
  {
    T2=0;

  }
  z=digitalRead(Sa);

  if ((T1==1)&&(T2==1)&&(z==1))
  {
    digitalWrite(BOMBA, LOW);

  }

  else if ((T1==1)&&(T2==0)&&(z==1))
  {
    digitalWrite(BOMBA, HIGH);

  }

  else if ((T1==0)&&(T2==1)&&(z==1))
  {
    digitalWrite(BOMBA, HIGH);

  }

  else if ((T1==0)&&(T2==0)&&(z==1))
  {
    digitalWrite(BOMBA, HIGH);

  }

  else if ((T1==0)&&(T2==0)&&(z==0))
  {
    digitalWrite(BOMBA, LOW);

  }
  else if ((T1==0)&&(T2==1)&&(z==0))
  {
```



```
    digitalWrite(BOMBA, LOW);

}
else if ((T1==1)&&(T2==0)&&(z==0))
{
    digitalWrite(BOMBA, LOW);

}
else if ((T1==1)&&(T2==1)&&(z==0))
{
    digitalWrite(BOMBA, LOW);

}
}
}
}

void loop(){
if (digitalRead(Sa) == LOW){ // Si sensor no detecta agua
    digitalWrite(BOMBA, LOW); // apaga la Bomba
    delay (2000);
}
else
{
WiFi.config(ip, gateway, subnet); // forces to use the fix IP
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(F("."));
    server.begin();
    Serial.print(nom);
    Serial.print(F(" connected to Wifi! IP address : http://"));
    Serial.println(WiFi.localIP()); // Print the IP address
}
clientRequest();
}
}
```

5.3 Esquema del sistema de control - Placa Arduino “Cliente”

Los circuitos con placa Arduino "Cliente" está conformada por las siguientes etapas:

- **Fuente de alimentación:** se encuentra regulador de voltaje 7805 que transforma 12 V a 5V requerido por la etapa de control.
- **Control:** Contiene placa Arduino ESP8266 en donde el pin 29 es la entrada del sensor de flujo, el cual, indica la disponibilidad del suministro de agua potable en tubería o almacenador (según la necesidad), activa mediante pin 30 que es la salida de un acoplador la etapa de potencia. Adicional tiene indicador visual de 7 segmentos que refleja el estatus del nivel deseado de agua potable en el almacenador del usuario.
- **Potencia:** se tiene transistor Q1 que mediante su polarización genera la activación de una electroválvula dando inicio al llenado del almacenador de agua potable.
- **Niveles:** es la parte final del sistema de control el cual lo encontraremos en el almacenador de agua potable del usuario para detectar el estatus

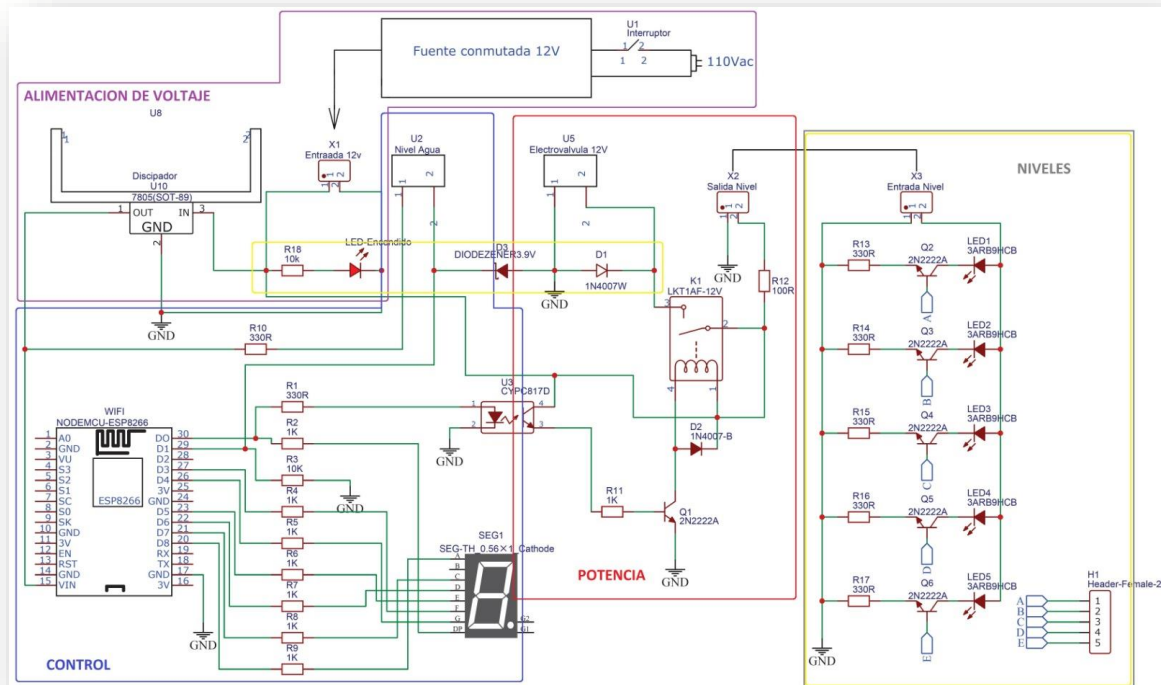


Figura 13. Diagrama de placa Arduino “Cliente”

Fuente: Autoría propia



5.4 Programa del sistema de control - Placa Arduino “Cliente”

```
v//Libraries
#include
<ESP8266WiFi.h>//https://github.com/esp8266/Arduino/blob/master/libraries/
ESP8266WiFi/src/ESP8266WiFi.h
//Constants
const int output = 16;
const int SegF = 0;
const int SegE = 14;
const int SegG = 2;
const int SegB = 15;
const int SegC = 13;
const int SegD = 12;
#define UPDATE_TIME 500
const int tanque=5;

int Y;
//Parameters
String nom = "Slave1";
const char* ssid = "A01"; // nombre de la red
const char* password = "graciela"; // contraseña de la red
//Variables
String command;
unsigned long previousRequest = 0;
//Objects
WiFiClient master;
IPAddress server(192, 168, 43, 65); // identificar la ip fija
void setup() {
    pinMode(tanque, OUTPUT);
    pinMode(SegF ,OUTPUT);
    pinMode(SegE ,OUTPUT);
    pinMode(SegG , OUTPUT);
    pinMode(SegB , OUTPUT );
    pinMode(SegC , OUTPUT);
    pinMode(SegD , OUTPUT);
    pinMode(output , OUTPUT);
    Serial.begin(115200);

    Serial.println(F("Initialize System"));

    WiFi.begin(ssid, password); // conectar
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(F("."));
    }
```



```
}
Serial.print(nom);
Serial.print(F(" connected to Wifi! IP address : "));
Serial.println(WiFi.localIP()); // Print the IP address
pinMode(tanque, INPUT);
}
void loop() {

    requestMaster();

}
void requestMaster( ) {

    if ((millis() - previousRequest) > UPDATE_TIME) { // client connect to
server every 500ms
        previousRequest = millis();
        if (master.connect(server, 80)) { // Connection to the server
master.println(nom + ": x" + String(!digitalRead(tanque)) + "\r");

        String answer = master.readStringUntil('\r');
        master.flush();
        Serial.println(answer);
        Serial.print(answer);

Y=digitalRead(tanque);

        if(Y==HIGH)
        {
            alto();
        }
        else
        {
            bajo();
        }

        }
        else
        {
            {
                digitalWrite(output, LOW);
            }
        }
    }

}
void alto(){
```



```
// digitalWrite(VA, LOW);
digitalWrite(output, LOW);
digitalWrite(SegF, HIGH);
digitalWrite(SegE, HIGH);
digitalWrite(SegG, HIGH);
digitalWrite(SegB, HIGH);
digitalWrite(SegC, HIGH);
digitalWrite(SegD, LOW);

delay(10);

}
void bajo(){

    digitalWrite(output, HIGH);
    digitalWrite(SegF, HIGH);
    digitalWrite(SegE, HIGH);
    digitalWrite(SegG, LOW);
    digitalWrite(SegB, LOW);
    digitalWrite(SegC, LOW);
    digitalWrite(SegD ,HIGH);
    delay(10);
}
```

VI. Presupuesto para sistema de control

Es muy importante la elaboración del presupuesto para ofrecer una idea hacia dónde se dirige el proyecto en términos económicos, en muchos de los proyectos esta información es la parte fundamental del mismo, puesto que tienen como objetivo el ahorro económico, pero no en todos los proyectos sucede esto, aunque en el sistema desarrollado el costo económico fue el esperado.

En las siguientes tablas se detalla los materiales utilizados para la elaboración del proyecto con la placa Arduino “estación” y “cliente”:

Descripción	Cantidad	Precio C\$ / Unidad	Precio total C\$
tarjeta perforada 76mm X 125mm (HS-06)	1	50.00	50.00
ESP8266MOD	1	700.0	700.0
Regulador L7805C	1	25.00	25.00
Relay 12V	1	45.00	45.00
Resistencias	8	5.00	40.00
optoacoplador pc817	1	15.00	15.00
Disipador aluminio 22*15*10 mm para TO-220	1	25.00	25.00
Bornera de 2 pines verde	1	15.00	15.00
Bornera de 3 pines verde	1	20.00	20.00
conector 2 pines blanco estándar	1	10.00	10.00
Transistor C3203	1	6.00	6.00
LEDS azules 3mm	1	10.00	10.00
Diodo Rectificador 1N4007	2	5.00	10.00
Fuente 13.5V 2A (Conmutada)	1	150.0	150.0
cable AC con terminal doble	1	85.00	85.00
Terminales de 8 líneas	1	55.00	55.00
Caja PLEXO (155mmX110mmX74mm)	1	419.0	419.0
Bomba 1/2" 12V	1	547.0	547.0
Total, general C\$			2227.00

Tabla de costos del sistema de control – placa Arduino “estación”

Fuente: Autoría propia

Descripción	Cantidad	Precio C\$ / Unidad	Precio total C\$
tarjeta perforada 90mm X 145mm (HS-03)	1	60.00	60.00
ESP8266MOD	1	700.0	700.0
Regulador L7805C	1	25.00	25.00
Relay 12V	1	45.00	45.00
Resistencias	21	5.00	105.0
optoacoplador pc817	1	15.00	15.00
display 8 segmentos	1	50.00	50.00
Disipador aluminio 22*15*10 mm para TO-220	1	25.00	25.00
Bornera de 2 pines verde	1	15.00	15.00
Bornera de 2 pines azul	1	15.00	15.00
conector 2 pines blanco estándar	2	10.00	20.00
Transistor C3203	6	6.00	36.00
Diodo Rectificador 1N4007	2	5.00	10.00
LEDS azules 3mm	5	10.00	50.00
Fuente 13.5V 2A (Conmutada)	1	150.0	150.0
cable AC con terminal doble	1	85.00	85.00
Terminales de 8 líneas	1	55.00	55.00
Caja PLEXO (180mmX140mmX86mm)	1	459.0	459.0
Válvula 1/2" 12V	1	255.0	255.0
Total, general C\$			2175.00

Tabla de costos del sistema de control – placa Arduino “cliente”

Fuente: Autoría propia

Descripción	Cantidad	Precio C\$ / Unidad	Precio total C\$
Unión 1/2" (codo liso PVC)	5	10.00	50.00
Unión 1/2" (hembra PVC)	5	10.00	50.00
Unión 1/2" (T lisa PVC)	1	10.00	10.00
Unión 1/2" (lisa PVC)	1	10.00	10.00
pegamento PVC	1	45.00	45.00
cinta teflón	1	25.00	25.00
estaño 8mm 60/40	1	300.0	300.0
Manos de Obra			2201.0
Total, general C\$			2691.0

Tabla de costos adicional al sistema de control

Fuente: Autoría propia

VII. Conclusiones

La recopilación de información sobre las condiciones del suministro de agua potable proporciona la base necesaria para abordar la problemática principal y seleccionar entre un PLC o una placa Arduino dependiendo de la complejidad del sistema de control y monitoreo requerido para solventar a las familias el barrio anexo primero de mayo con expectativa a largo plazo.

Se requiere en los hogares del barrio anexo primero de mayo una mejora en la infraestructura y la identificación de los puntos de distribución estratégicos para garantizar la cantidad de agua requerida para el uso diario en un hogar de 4 o más personas. Aunque puede variar significativamente según varios factores, como los hábitos de consumo de agua, la eficiencia de los electrodomésticos y accesorios, el clima y las condiciones de suministro de agua.

Se concluye que el sistema de control automatizado implementado con sensores, actuadores y placa Arduino (ESP8266) permite el monitoreo de manera eficiente el suministro de agua en tiempo real. Esto mejorará la gestión de los recursos hídricos, reducirá el desperdicio de agua y brindará a la comunidad acceso confiables para la recolección en horarios nocturnos o en caso no esté presente alguna persona.

VIII. Recomendaciones

Como mejora del sistema de control se recomienda cambiar los sensores de contacto por flotador siendo simples y confiables. Utiliza un flotador que se eleva y cae con el nivel del líquido permitiendo una respuesta rápida, pero es importante tener en cuenta la calidad del agua, ya que esto puede acortar la vida útil.

Arduino tiene las herramientas necesarias para elaborar una interfaz web, para monitorear y ajustar el sistema por lo tanto se recomienda diseñar una aplicación móvil o un panel de control en una computadora, esto tomando la consideración de la demanda del ancho de banda como limitante.

Realizar supervisión técnica del sistema y mantenimiento del mismo cada 3 meses o cuando se estime conveniente y mantener aislado de humedad o exposición a agentes externos que interfieran con su buen funcionamiento.

Se sugiere elaborar un manual de usuario, ya que el sistema diseñado es amigable y da la facilidad al cliente a brindar mantenimiento al equipo.



IX. Bibliografía

1. Mejia, H. M., & Espinazo, C. E. (2016, noviembre). "Sistema de automatización para el llenado de un tanque de agua por bombas con la ayuda de sensores". <https://core.ac.uk/>. Recuperado 11 de octubre de 2023, de <https://core.ac.uk/download/pdf/250143154.pdf>
2. Espinoza Medina, J. B., Pérez Rodríguez, D. J., & González Mendoza, M. I. (2006). Evaluación y mejoramiento del sistema de abastecimiento de agua potable en la localidad de El Sauce, Departamento de León. Obtenido de <https://repositorio.cnu.edu.ni/Record/RepoUNANM4921>
3. Control Automático Educación C, S. (2023). *Lazo Abierto y Lazo Cerrado*. Obtenido de <https://controlautomaticoeducacion.com/control-realimentado/lazo-abierto-y-lazo-cerrado/>
4. QuimiNet (2023, octubre 11). *El uso de las electroválvulas en la industria*. Obtenido de <http://www.quiminet.com/articulos/el-uso-de-las-electrovalvulas-en-la-industria-2648462.html>
5. Oloarte, G. (2023, May 19). ¿Cómo Elegir la Válvula Solenoide Correcta? FVK Válvulas. Obtenido de <https://www.fvkvalvulaseinstrumentos.com/post/como-elegir-la-valvula-solenoide-correcta>
6. Omega s. f. (2023, octubre). *Sensores de nivel*. Obtenido de <https://es.omega.com/prodinfo/sondas-de-nivel-medicion.html#>
7. Jecrespom. (2021, February 16). Actuadores y periféricos de salida. Aprendiendo Arduino. Obtenido de <https://aprendiendoarduino.wordpress.com/2015/03/31/actuadores-y-perifericos-de-salida/>
8. Aguayo, P. (2023). ¿Qué es Arduino? | Arduino.cl - Compra tu Arduino en Línea. Arduino.cl - Compra Tu Arduino En Línea. Obtenido de <https://arduino.cl/que-es-arduino/>.

ANEXO

Encuesta sobre la Duración del Suministro de Agua

Por favor, tómese un momento para responder a las siguientes preguntas relacionadas con la duración del suministro de agua en su área. Sus respuestas son importantes para nosotros.

1. ¿Cuánto tiempo dura generalmente el suministro de agua en su hogar o área de residencia?
 - ☐ Menos de 12 horas al día
 - ☐ 12 horas al día
 - ☐ Más de 12 horas al día
2. ¿Con qué frecuencia experimenta interrupciones en el suministro de agua en su área?
 - ☐ Nunca
 - ☐ Rara vez (menos de una vez al mes)
 - ☐ Ocasionalmente (una vez al mes)
 - ☐ Frecuentemente (más de una vez al mes)
3. Si ha experimentado interrupciones en el suministro de agua, ¿cuánto tiempo suelen durar estas interrupciones?
 - ☐ Menos de 1 hora
 - ☐ 1-3 horas
 - ☐ 3-6 horas
 - ☐ Más de 6 horas
4. ¿Ha tenido que tomar medidas especiales, como almacenar agua, debido a interrupciones en el suministro?
 - ☐ Sí
 - ☐ No
5. En su opinión, ¿cómo podría mejorar la duración del suministro de agua en su área? (opcional)

Gracias por participar en nuestra encuesta. Sus respuestas son valiosas y nos ayudarán a comprender mejor la situación del suministro de agua en su área.

Programa de interfaz usuario Web.

```
// Respuesta del servidor web
    client.println("HTTP/1.1 200 OK"); // La respuesta empieza con una
    linea de estado
    client.println("Content-Type: text/html"); //Empieza el cuerpo de la
    respuesta indicando que el contenido será un documento html
    client.println(""); // Ponemos un espacio
    client.println("<!DOCTYPE HTML>"); //Indicamos el inicio del Documento
    HTML
    client.println("<html lang=\"en\">");
    client.println("<head>");
    client.println("<meta charset=\"UTF-8\">");
    client.println("<meta name=\"viewport\" content=\"width=device-width,
    initial-scale=1.0\">"); //Para que se adapte en móviles
    client.println("<title>Servidor Web EDDY</title>");
    client.println("</head>");
    client.println("<body>");
    client.println("<br><br>");

    client.println("<h1 style=\"text-align: center;\">Servidor Web
    EDDY</h1>");

    client.println("<p style=\"text-align: center;\">");
    client.println("Click <a target=\"_blank\" class=\"fcc-
    btn\" href=\"/LED=ON\" </a> para encender LED en el pin 2 ON<br>");
    //Oración que contiene un hipervínculo
    client.println("Click <a href=\"/LED=OFF\">Aqui</a> para apagar el led
    en el pin 2 OFF<br>");
    client.println("Click <a href=\"/LED=BLINK\">Aqui</a> para parpadar el
    led en el pin 2<br> <br>");

    client.println("<button onclick=location.href=\"/LED=ON\"> Encender
    LED</button> <br> <br>"); // Botón sencillo que contiene hipervínculo
    client.println("<button onclick=location.href=\"/LED=OFF\" >Apagar LED
    </button> <br> <br>");
    client.println("<button onclick=location.href=\"/LED=BLINK\">Parpadear
    LED </button> <br> <br>");

    client.println("</p>");
    client.println("</body>");

    client.println("</html>"); //Terminamos el HTML
```



Interfaz usuario Web desde computadora

Fuente: Propia Autoría

TABLA. 2-1

Clasificación de Barrios	Dotación	
	gl/hab/día	lt/hab/día
- Asentamientos progresivos	10	38
-Zonas de máxima densidad y de actividades mixtas.	45	170
- Zonas de alta densidad	40	150
- Zonas de media densidad	100	378
- Zonas de baja densidad	150	568

Tabla de valores de dotación doméstica Managua

Fuente: <http://www.inaa.gob.ni/taxonomy/term/525>

Clasificación de los Barrios:

Asentamientos progresivos

Son unidades de viviendas construidas con madera y láminas, frecuentemente sobre un basamento de concreto. Estos barrios no tienen conexiones privadas en la red de agua potable, pero se abastecen mediante puestos públicos.

Zonas de máxima densidad y actividades mixtas.

Las viviendas avicinan talleres y pequeñas industrias en un tejido urbano heterogéneo. En términos de superficie, las viviendas ocupan un promedio del 65% del área total del terreno y todas están conectadas a la red de agua potable.

Zonas de alta densidad

En los núcleos de viviendas de estas zonas se encuentran construcciones de todo tipo, desde la más sencilla hasta casas de alto costo pero en lotes con dimensiones y áreas homogéneas (150 m² a 250 m²). Casi todas las viviendas están conectadas a la red de agua potable.

Zonas de media densidad

Se trata de viviendas de buen nivel de vida con áreas de lotes que varían entre los 500 m² y 700 m². Todas están conectadas a la red de agua potable.

Zonas de baja densidad

Son áreas de desarrollo con viviendas de alto costo y de alto nivel de vida construidas en lotes con área mínimas de 1.000 m². Todos conectados a la red de agua potable.

			Pie cúbico	Barril	Litro	Metro cúbico
VOLUMEN	pie cúbico	ft ³	1	0,1781	28,32	0,0283
	barril	bbl	5.615	1	158,984	0,1590
	litro	l	0,0353	0,0063	1	0,001
	metro cúbico	m ³	35,3107	6,2898	1.000	1

Tabla de Conversión

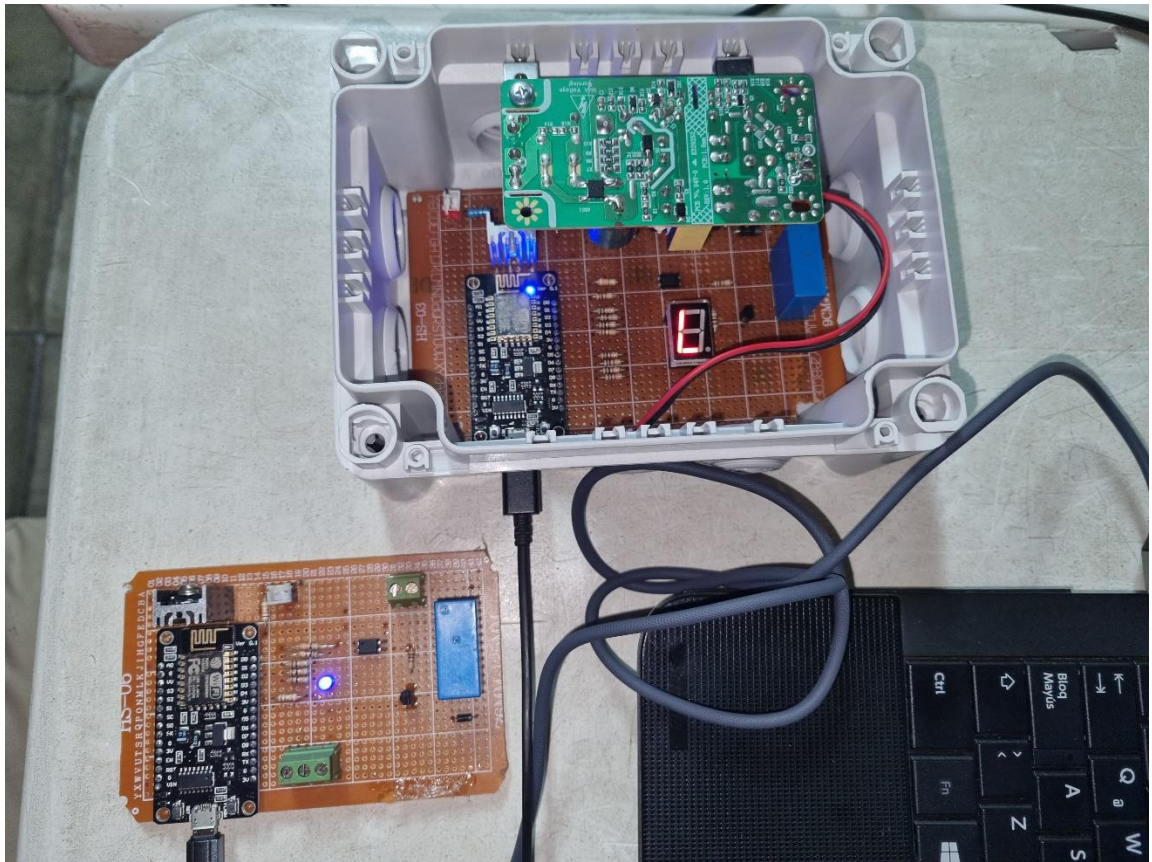
Fuente: <https://www.patricioruiz.es/wp-content/uploads/tabla-de-conversiones-1.pdf>



Almacén de agua potable



Tubería para red de distribución



Sistema de control con placa Arduino – (Estación – Cliente)

Especificaciones:

Nombre: bomba de agua

Material: plástico

Color: Negro

Entrada/Salida: cabezal macho
de 1/2"

Voltaje: 12V DC/24V DC

Corriente nominal máxima:
1000ma

Potencia: 22W

Caudal máximo: 800 L/H

Cabezal máximo de agua: 5m

Temperatura máxima del agua
circulante: 100 °C

Tamaño: aproximadamente 6,5 cm(D) x 8cm(H)



Nombre del producto: válvula solenoide
de 220V CA/12V 24V CC

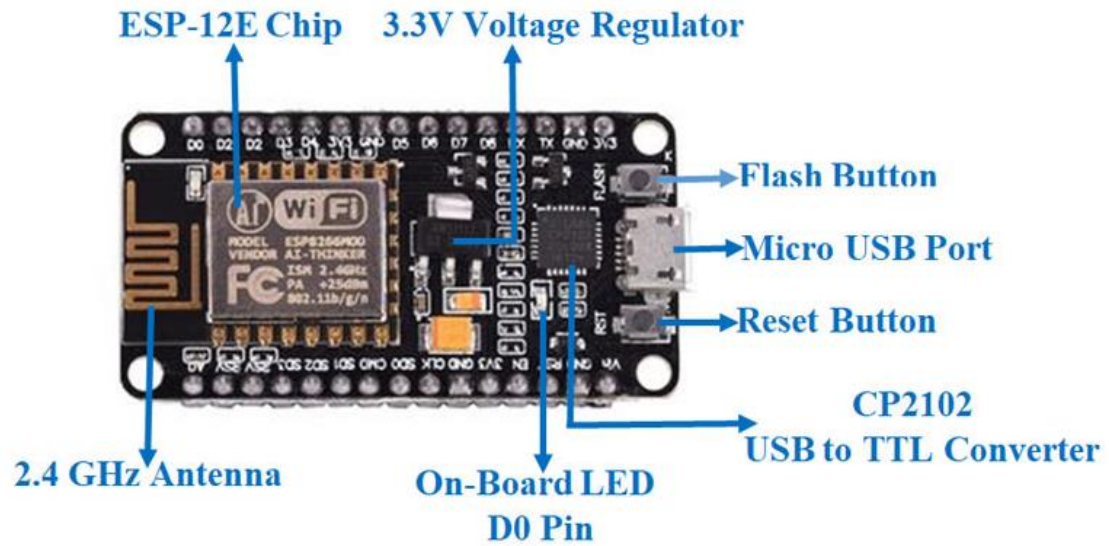
Voltaje: 12V 24V DC /220V AC

Presión: 0,02-0.8Mpa

Tamaño de la conexión: rosca macho
de 1/2 ",3/4"

Medios: agua o fluidos de baja
viscosidad

Modelo de operación: normalmente
cerrado (N/C) o normalmente abierto



Name: NodeMCU ESP8266

Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106

Operating Voltage: 3.3V

Input Voltage: 7-12V

Digital I/O Pins (DIO): 16

Analog Input Pins (ADC): 1

UARTs: 1

SPIs: 1

I2Cs: 1

Flash Memory: 4 MB

SRAM: 64 KB

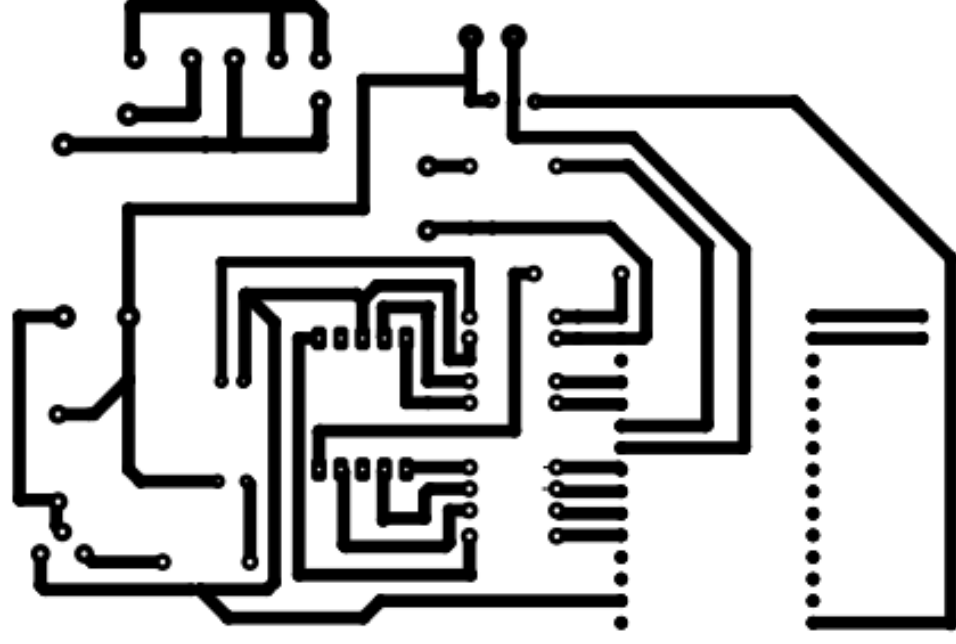
Clock Speed: 80 MHz

USB-TTL based on CP2102 is included onboard, Enabling Plug n Play

PCB Antenna

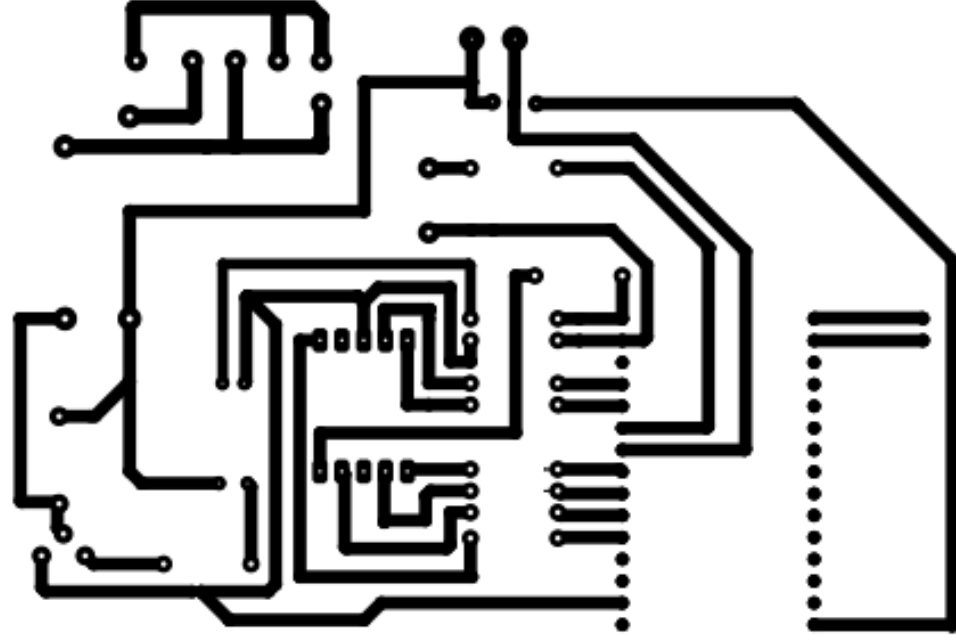
Small Sized module to fit smartly inside your IoT projects

CONTROL GLEIADO DE TANGNE



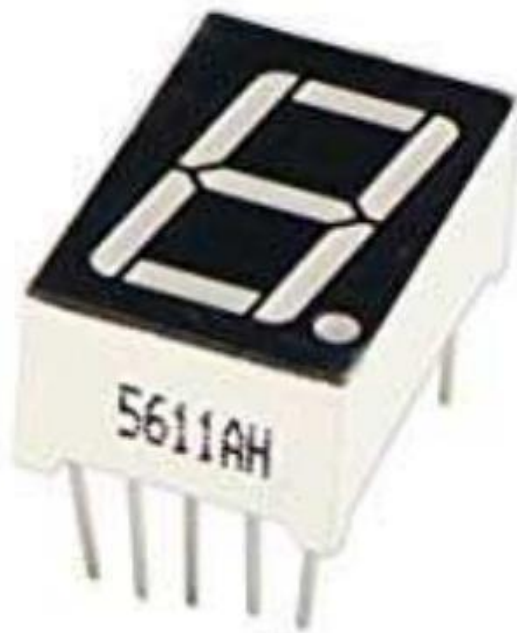
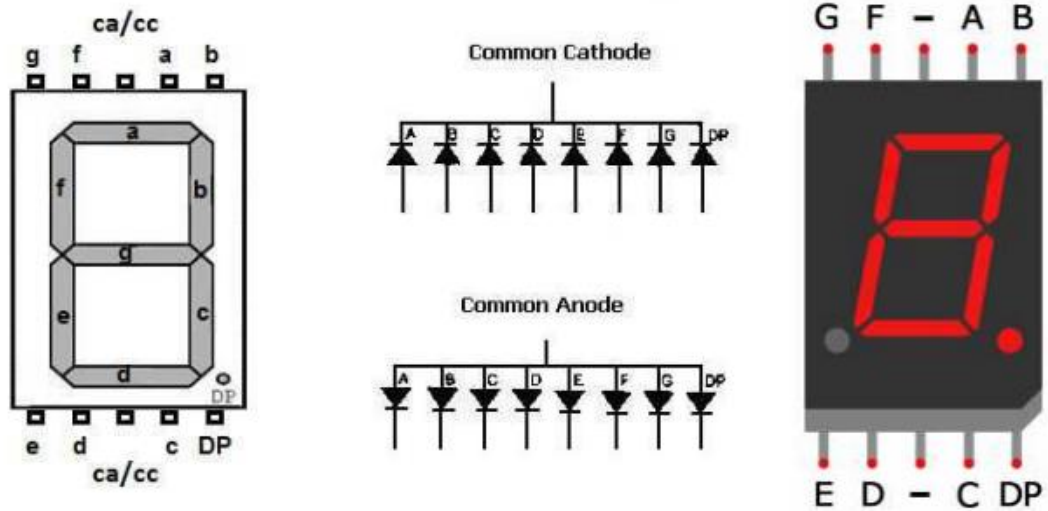
ЕКНО 3

CONTROL GLEIADO DE TANGNE

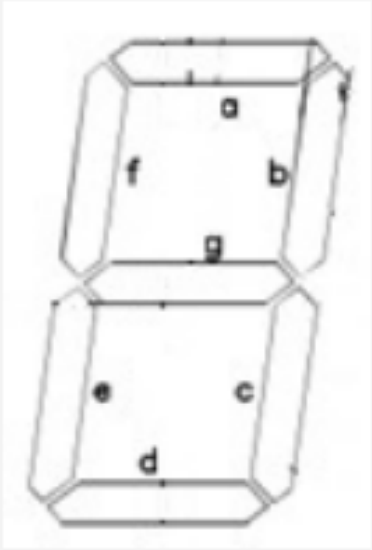


ЕКНО 3

El display “7 segmentos” que recomendado en la placa Arduino es un 5161AH con los pinout:



Programa para encendido de segmentos.

	DIGITO	A	B	C	D	E	F	G	PUNTO
	0	1	1	1	1	1	1	0	0
	1	0	1	1	0	0	0	0	0
	2	1	1	0	1	1	0	1	0
	3	1	1	1	1	0	0	1	0
	4	0	1	1	0	0	1	1	0
	5	1	0	1	1	0	1	1	0
	6	1	0	1	1	1	1	1	0
	7	1	1	1	0	0	0	0	0
	8	1	1	1	1	1	1	1	0
	9	1	1	1	0	0	1	1	0

```
//Enciende barra a barra una a la vez
int tiempo = 150; //definimos una variable de tipo entera para fijar
retardo
void setup()
{

    // Coloque el codigo de configuracion (setup) aqui - se ejecutara solo
una vez:
pinMode(2, OUTPUT); // configura 'pin' como salida
pinMode(3, OUTPUT);
pinMode(4, OUTPUT);
pinMode(5, OUTPUT);
pinMode(6, OUTPUT);
pinMode(7, OUTPUT);
pinMode(8, OUTPUT);
```

```

}
void loop()
{

    // Coloque el codigo principal (main) aqui - se ejecutara repetidamente
    (loop):
    digitalWrite(2,HIGH );// PIN es colocado en estado alto (HIGH)
    delay(tiempo); // retardo
    digitalWrite(2,LOW ); // PIN es colocado en estado bajo (LOW)
    delay(tiempo); // otro retardo de 1000 mseg o 1 seg
    digitalWrite(3,HIGH );// PIN es colocado en estado alto (HIGH)
    delay(tiempo); // retardo
    digitalWrite(3,LOW ); // PIN es colocado en estado bajo (LOW)
    delay(tiempo); // otro retardo de 1000 mseg o 1 seg
    digitalWrite(4,HIGH );// PIN es colocado en estado alto (HIGH)
    delay(tiempo); // retardo
    digitalWrite(4,LOW ); // PIN es colocado en estado bajo (LOW)
    delay(tiempo); // otro retardo de 1000 mseg o 1 seg
    digitalWrite(5,HIGH );// PIN es colocado en estado alto (HIGH)
    delay(tiempo); // retardo
    digitalWrite(5,LOW ); // PIN es colocado en estado bajo (LOW)
    delay(tiempo); // otro retardo de 1000 mseg o 1 seg
    digitalWrite(6,HIGH );// PIN es colocado en estado alto (HIGH)
    delay(tiempo); // retardo
    digitalWrite(6,LOW ); // PIN es colocado en estado bajo (LOW)
    delay(tiempo); // otro retardo de 1000 mseg o 1 seg
    digitalWrite(7,HIGH );// PIN es colocado en estado alto (HIGH)
    delay(tiempo); // retardo
    digitalWrite(7,LOW ); // PIN es colocado en estado bajo (LOW)
    delay(tiempo); // otro retardo de 1000 mseg o 1 seg
    digitalWrite(8,HIGH );// PIN es colocado en estado alto (HIGH)
    delay(tiempo); // retardo
    digitalWrite(8,LOW ); // PIN es colocado en estado bajo (LOW)
    delay(tiempo); // otro retardo de 1000 mseg o 1 seg
}

```

Programa para encendido de dígitos

```

//Enciende de un dígito a la vez
int tiempo = 500;//definimos una variable de tipo entera para fijar
retardo
void setup()
{

```



```
// Coloque el codigo de configuracion (setup) aqui - se ejecutara solo una vez:
```

```
pinMode(2, OUTPUT); // configura 'pin' como salida
pinMode(3, OUTPUT);
pinMode(4, OUTPUT);
pinMode(5, OUTPUT);
pinMode(6, OUTPUT);
pinMode(7, OUTPUT);
pinMode(8, OUTPUT);
}
void loop()
{
```

```
// Coloque el codigo principal (main) aqui - se ejecutara repetidamente (loop):
```

```
//Muestra el cero
```

```
digitalWrite(2,HIGH );// PIN es colocado en estado alto (HIGH)
digitalWrite(3,HIGH );// PIN es colocado en estado alto (HIGH)
digitalWrite(4,HIGH );// PIN es colocado en estado alto (HIGH)
digitalWrite(5,HIGH );// PIN es colocado en estado alto (HIGH)
digitalWrite(6,HIGH );// PIN es colocado en estado alto (HIGH)
digitalWrite(7,HIGH );// PIN es colocado en estado alto (HIGH)
digitalWrite(8,LOW ); // PIN es colocado en estado bajo (LOW)
delay(tiempo); // retardo
```

```
//Muestra el uno
```

```
digitalWrite(2,LOW );
digitalWrite(3,HIGH );
digitalWrite(4,HIGH );
digitalWrite(5,LOW );
digitalWrite(6,LOW );
digitalWrite(7,LOW );
digitalWrite(8,LOW );
delay(tiempo); // retardo
```

```
//Muestra el dos
```

```
digitalWrite(2,HIGH );
digitalWrite(3,HIGH );
digitalWrite(4,LOW );
digitalWrite(5,HIGH );
digitalWrite(6,HIGH );
digitalWrite(7,LOW );
digitalWrite(8,HIGH );
delay(tiempo); // retard
```