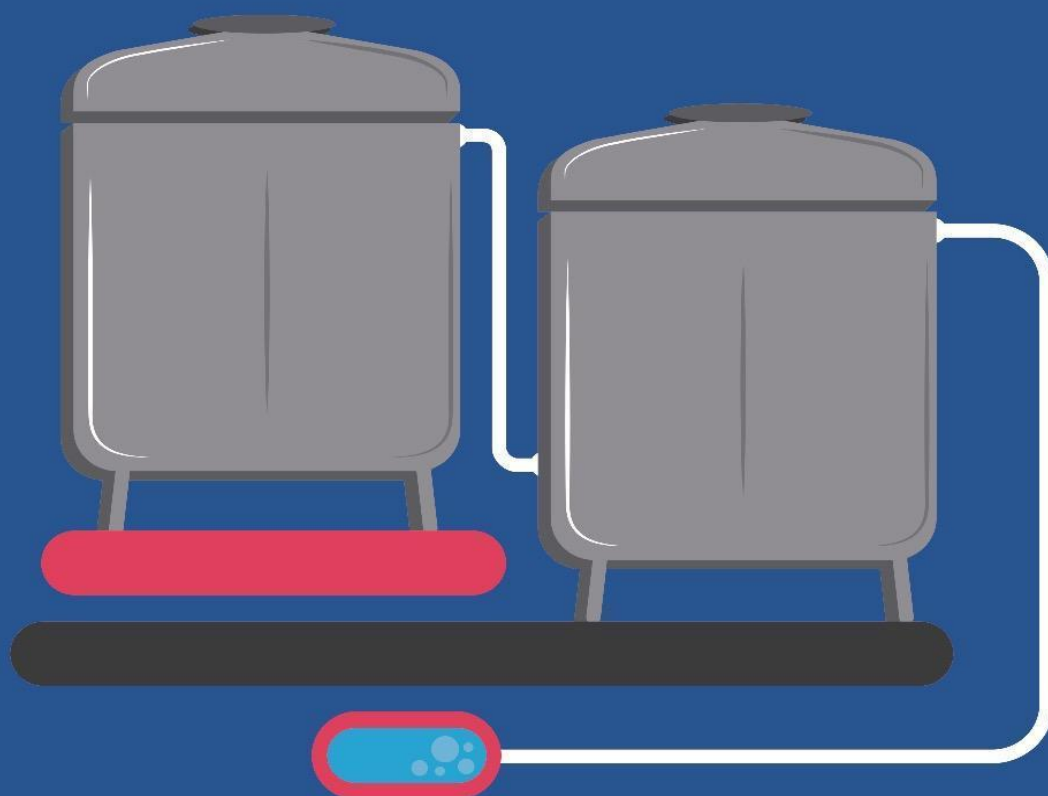


APRENDER
CONECTADOS

SABERES
DIGITALES



SISTEMA DE TANQUES

AUTORIDADES

AUTORIDADES

Presidente de la Nación

Mauricio Macri

Vicepresidenta de la Nación

Marta Gabriela Michetti

Jefe de Gabinete de Ministros

Marcos Peña

Ministro de Educación, Cultura, Ciencia y Tecnología

Alejandro Finocchiaro

Titular de la Unidad de Coordinación General del Ministerio de Educación, Cultura, Ciencia y Tecnología

Manuel Vidal

Subsecretario de Coordinación Administrativa

Javier Mezzamico

Director Ejecutivo INET

Leandro Goroyesky

Gerenta General de EDUCAR Sociedad del Estado

Liliana Casaleggio

Directora Nacional de Asuntos Federales

María José Licio Rinaldi

Director Nacional de Educación Técnico - Profesional

Fabián Prieto

Coordinador de Secundaria Técnica

Alejandro Anchava

Responsable de Formación Docente Inicial y Continua INET

Judit Schneider

Coordinador General En FoCo

Pablo Trangone

SISTEMA DE TANQUES

Ficha técnica	3
Presentación	4
Desarrollo	5
Nivel inicial	5
Paso 1: Preparar el tanque	5
Paso 2: Instalar el sensor ultrasónico y conectar la electroválvula	5
Paso 3: Establecer la altura de corte	7
Paso 4: Conectar sensor, válvula y relé a la placa Arduino.	7
Paso 5: Programar el funcionamiento del dispositivo	8
Nivel intermedio	12
Paso 1: Preparar la instalación del segundo tanque	12
Paso 2: Realizar las conexiones a la placa Arduino	13
Nivel avanzado	16
Paso 1 - Introducción a Internet de las Cosas (IoT)	16
Paso 2 - Crear un Panel de Control	17
Paso 3 - Agregar Módulo OBLOQ	25
Paso 4 - Arduino IDE	25
Paso 5 - Programar sin código bloqueante	27
Paso 6 - Programar IoT	29
Cierre	36
Glosario	37
Reconocimientos	¡Error! Marcador no definido.

SISTEMA DE TANQUES

Ficha técnica

Nivel educativo	Secundario. Ciclo Básico.
Descripción general	Diseño y construcción de una maqueta/prototipo de un sistema de abastecimiento de agua domiciliario.
Niveles de complejidad	<p>Nivel inicial: controlar el ingreso del agua al tanque a través una electroválvula (que regule el ingreso) y un sensor ultrasónico (que mida el nivel de llenado) conectados a una placa Arduino.</p> <p>Nivel intermedio: agregar un segundo tanque y una bomba sumergible que envíe agua del primer tanque al segundo.</p> <p>Nivel avanzado: informar los datos obtenidos a un dispositivo móvil a través de internet de las cosas (IoT).</p>
Insumos	<p>2 x Módulo relé de 5V</p> <p>1 x Bomba de agua sumergible</p> <p>2 x Sensores de distancia ultrasónicos</p> <p>1 x Arduino UNO R3</p> <p>1 x OBLOQ Módulo IoT</p> <p>1 x Fuente de 9V 1A (plug centro positivo, 5.5 x 2.1mm)</p> <p>1 x Cable USB tipo B</p> <p>20 x Cables dupont macho hembra</p> <p>Cables eléctricos.</p> <p>2 x Recipientes para agua con canillas de desagote</p> <p>Mangueras</p>
Equipamiento	<p>Computadora</p> <p>Soldador</p> <p>Estaño</p> <p>Alicate</p> <p>Pinza de punta</p>

	Brusela
Otros requisitos	Conexión a internet Descargar el programa "mblock3" http://www.mblock.cc/software-1/mblock/mblock3/

Presentación

Descripción ampliada del proyecto

Se propone armar una maqueta/prototipo a escala de un sistema de tanques interconectados, que simule el sistema de suministro de agua que existe en las casas y los edificios. El llenado de los tanques y las bombas se controlará midiendo el nivel del agua presente en cada tanque con sensores de ultrasonido. En el nivel más avanzado se propone incorporar Internet de la Cosas (IoT) para generar un registro sobre el llenado de los tanques.

Al final de esta guía se puede encontrar un glosario donde se provee la información técnica necesaria para poder poner el proyecto en funcionamiento. El mismo cuenta con aclaraciones sobre los diversos elementos electrónicos involucrados así como también conceptos claves.

Objetivos

Aproximarse al conocimiento y al manejo de distintos componentes electrónicos, mediante la construcción de una maqueta de un sistema de tanques en el que se regulen el ingreso y el egreso de agua.

Introducirse en el armado de circuitos utilizando placas Arduino, sensores de ultrasonido y actuadores.

Analizar la programación de estructura secuencial de un programa que permita el ingreso de agua y el pasaje de agua de un tanque a otro (nivel intermedio) cuando sea necesario.

Utilizar Internet de la Cosas (**IoT**) para registrar los datos de llenado de los tanques (nivel avanzado).

Desarrollo

Nivel Inicial

Camilo está construyendo una casa para su familia que tendrá una sola planta con una terraza. Para hacerlo tiene que, entre muchas otras cosas, realizar la conexión de agua. El agua llega a su casa por el sistema corriente que la distribuye a toda la zona desde la planta potabilizadora local.

Todas las edificaciones deben contar con uno o más tanques de almacenamiento, que se llenan con el agua que proviene de la red y a partir de allí suministran el consumo local. Estos se ubican en el nivel más alto de cada construcción y se utilizan para garantizar la disponibilidad de agua en el edificio o casa en que se encuentren.

Entonces, Camilo debe instalar un tanque en su terraza, que recibirá directamente el agua que viene desde la calle y que proveerá desde allí agua para toda la casa. También debe instalar un sistema de válvulas que controle el ingreso del agua, para evitar que se rebalse el tanque. Para esto, normalmente se usa una válvula o interruptor de flotador. Camilo, en este caso, decidió controlar el ingreso de agua utilizando un sensor de distancia ultrasónico que mide el nivel de llenado del tanque.

En esta instancia del proyecto se construirá la maqueta de un tanque de agua en la que se regule el ingreso de la misma al tanque a través de un sensor ultrasónico que mida el nivel de llenado del tanque y una electroválvula.

Paso 1 - Preparar el tanque

Para comenzar, es necesario conseguir un recipiente que cumpla la función del tanque. Este debe tener una tapa, un agujero en la parte superior para el ingreso del agua, y otro en la parte inferior donde se colocará la llave de paso que controlará la distribución del agua.

Paso 2 - Instalar el sensor ultrasónico y conectar la electroválvula

En este proyecto la ubicación de los distintos elementos en el prototipo es fundamental para el buen funcionamiento del mismo. Por lo tanto, organizamos la disposición de todos los componentes antes de realizar la conexión a la placa Arduino. Esta placa puede recibir datos de un sensor, así como enviar datos a un actuador, por medio de sus pines analógicos y digitales.

Para controlar el ingreso de agua y evitar que el tanque rebalse, colocamos en su entrada una electroválvula (válvula controlada electrónicamente, del tipo que suelen encontrarse en

los lavarropas). Esta se abrirá cuando queramos llenar el tanque y se cerrará cuando el agua haya alcanzado un nivel determinado por nosotros.



Electroválvula, es un actuador que permite o interrumpe el paso de un fluido a través del mismo. Existen versiones que trabajan con 12 V así como también los que trabajan directamente con 220 V, permitiendo trabajar en un sistema de mayor nivel de caudal

Sujetamos el sensor ultrasónico en la parte interior de la tapa del tanque. Es importante que este apunte de manera perpendicular al fondo del tanque para que las paredes no interfieran en la medición. Además, hay que asegurarnos de que el sensor no se moje cuando ingresa el agua. El sensor medirá la distancia a la que se encuentra el agua y en función de eso activará o no la electroválvula.



Sensor ultrasónico, es un tipo de sensor que se utiliza para medir distancias. El principio de funcionamiento del mismo se basa en emitir un pulso de sonido ultrasónico y medir el tiempo que pasa hasta registrar que dicho pulso regresó a la fuente tras rebotar en un obstáculo.

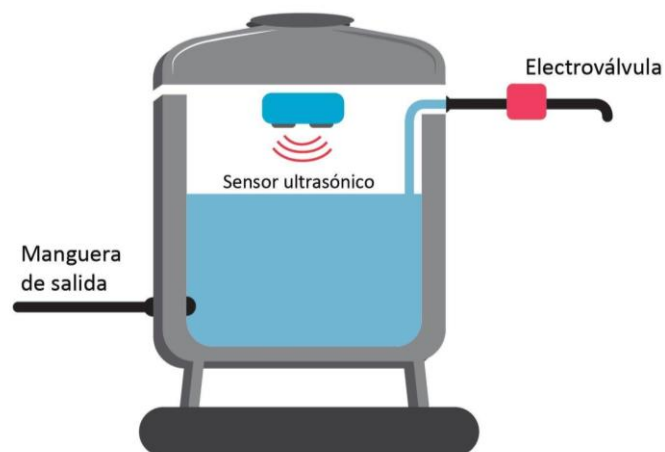


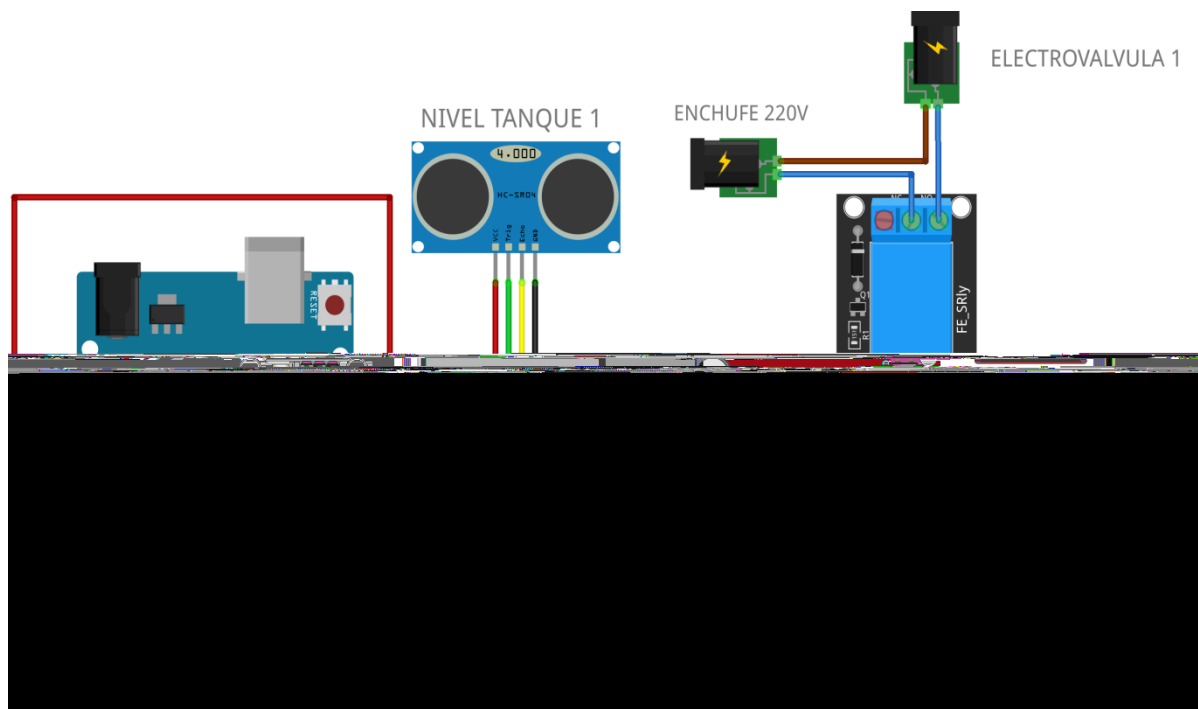
Fig. 1

Paso 3 - Establecer la altura de corte

Para evitar que el tanque rebalse, el agua no puede superar la altura de ingreso. Por lo tanto, debemos detener la entrada de agua antes de que se alcance dicho nivel. Para asegurarnos de que eso suceda, es recomendable que el tanque se deje de llenar cuando el agua esté a unos 5 o 10 cm del ingreso. Una vez determinado este nivel hay que medir la distancia desde este punto a la tapa donde se encuentra el sensor y establecer este valor como la distancia mínima de desactivación de la válvula.

Paso 4 - Conectar sensor, válvula y relé a la placa Arduino.

Una vez que tengamos ubicados los distintos elementos del sistema, conectaremos el sensor ultrasónico, la electroválvula y el relé a una placa Arduino como indica la figura.



Esquema 1

¡Atención! Para construir este dispositivo trabajaremos con una tensión de 220V. En caso de utilizar protoboard, se recomienda no incluir en el mismo las conexiones a relé y 220V.

La electroválvula funciona con una tensión de 220V. Por este motivo, para realizar la conexión necesitamos usar dos cables de 2,5 mm de diámetro y una ficha de enchufe para conectar a la red. Debemos soldar uno de los cables a un *borne* de la electroválvula y el otro conectarlo a la *bornera* "COM" del relé. Usar un tercer cable para unir el otro *borne* de la electroválvula con la *bornera* "NO" del relé.

Ahora solo queda conectar el relé a la placa Arduino. Para esto usaremos cables dupont macho hembra ya que estaremos trabajando con 5V. Conectar el pin “+” a 5V, el pin “-” a GND y el pin “S” al pin digital “4” de la placa Arduino.

Para conectar el sensor de posición ultrasónico necesitamos cuatro cables dupont hembra-macho. Conectar los pines “Vcc”, “GND”, “Trig” y “Echo” del sensor a los pines “5V”, “GND”, “8” y “9” del Arduino respectivamente.

Paso 5 - Programar el funcionamiento del dispositivo

La programación la realizaremos con mBlock3, un entorno de programación basado en Scratch2, que permite programar proyectos de Arduino utilizando bloques. Se puede descargar siguiendo este enlace: <http://www.mblock.cc/software-1/mblock/mblock3/>

Cuando abrimos mBlock3, veremos una pantalla como la siguiente:

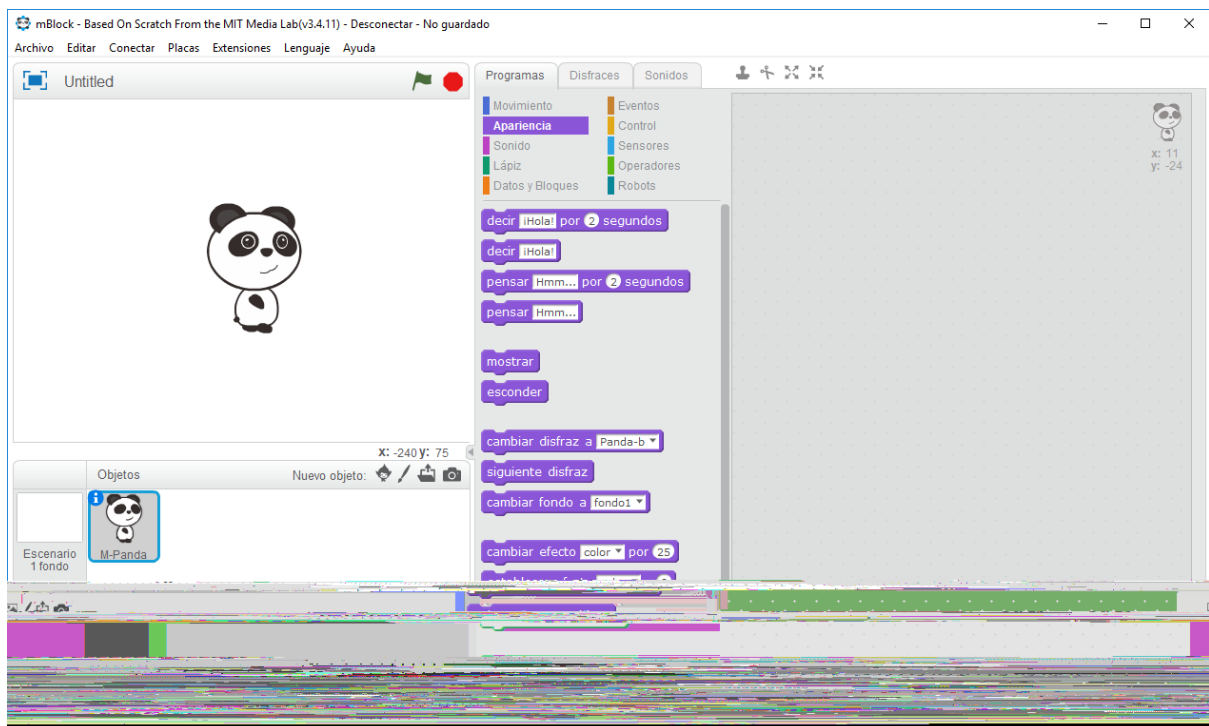


Fig. 2

Para programar un proyecto de Arduino con mBlock3 debemos seleccionar el “Modo Arduino” desde el menú.

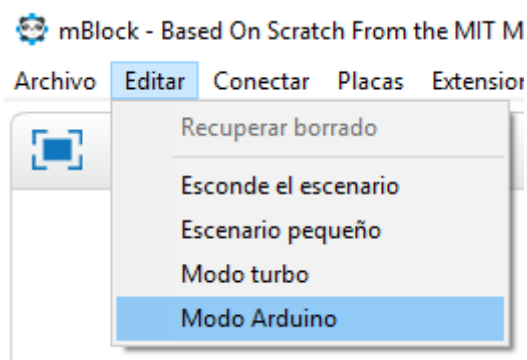


Fig. 3

Al seleccionar este modo, el programa cambiará de aspecto. Se verá un área en el centro que es la que utilizaremos para programar con bloques. A la derecha se verá un campo donde aparecerá el código escrito que le corresponde a los bloques que están en el centro. Este código se irá escribiendo automáticamente a medida que se vaya armando el programa con los bloques.

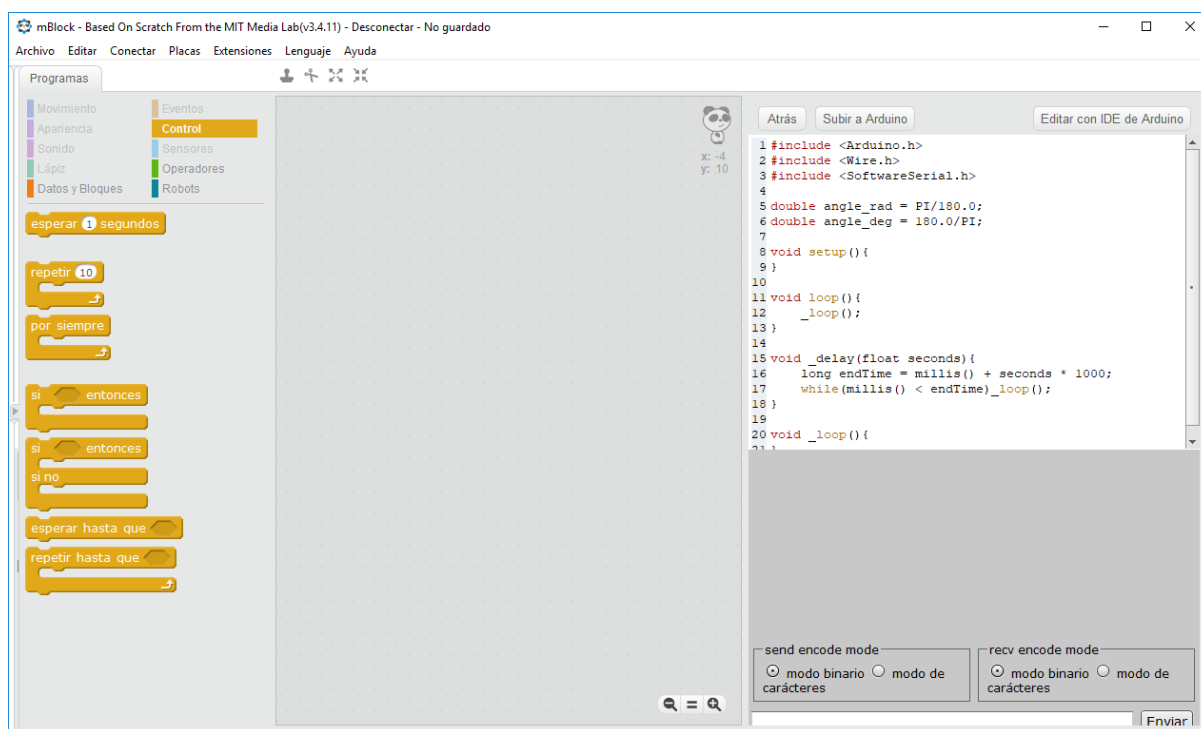


Fig. 4

Los bloques están agrupados por categorías. En este caso, se usarán las siguientes: **“Robots”**, **“Control”**, **“Operadores”** y **“Datos y Bloques”**. Cuando seleccionamos una de estas categorías, se pueden visualizar todos los bloques que pertenecen a ese grupo.



Fig. 5

Escribimos el programa que controlará la apertura de la válvula. Lo haremos de modo que permita que se cargue el tanque siempre que la distancia entre la superficie del agua y el sensor sea menor que la distancia mínima medida anteriormente. Tomando como ejemplo que la distancia mínima es de 40 cm, el código en bloques nos queda como sigue:

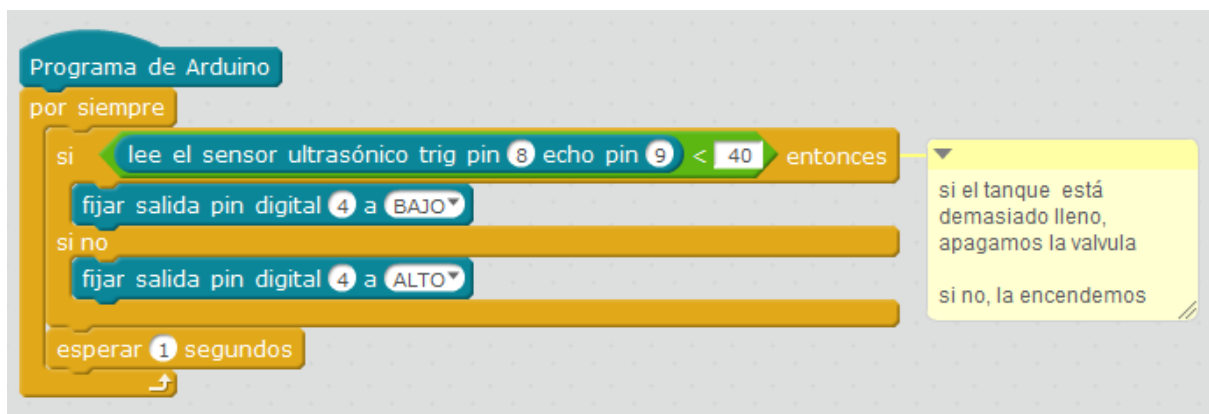


Fig. 6

Veremos que a continuación se muestra el código escrito generado por mBlock, que corresponde a este programa:

```
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>

double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;
float getDistance(int trig,int echo){
    pinMode(trig,OUTPUT);
    digitalWrite(trig,LOW);
    delayMicroseconds(2);
    digitalWrite(trig,HIGH);
    delayMicroseconds(10);
    digitalWrite(trig,LOW);
```

```

    pinMode(echo, INPUT);
    return pulseIn(echo,HIGH,30000)/58.0;
}

void setup(){
    pinMode(4,OUTPUT);
}

void loop(){
    if((getDistance(8,9)) < ( 40 )){
        digitalWrite(4,0);
    }else{
        digitalWrite(4,1);
    }
    _delay(1);
    _loop();
}

void _delay(float seconds){
    long endTime = millis() + seconds * 1000;
    while(millis() < endTime)_loop();
}

void _loop(){
}

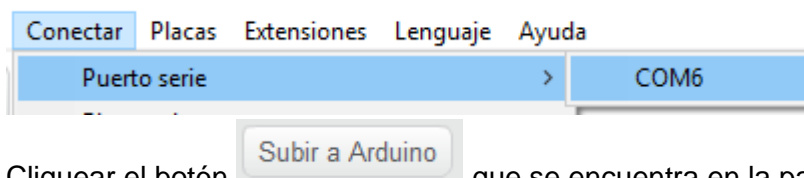
```

Fragmento de código 1

Paso 6 - Subir el código a la placa Arduino

Para subir el código de nuestro programa a la placa Arduino, necesitamos:

1. Conectar la placa Arduino a la entrada USB.
2. Chequear que en el **menú** “Placas” esté seleccionado “Arduino Uno”.
3. Seleccionar el puerto serie al que está conectada la placa.



4. Clickear el botón **Subir a Arduino** que se encuentra en la parte derecha superior de la interfaz.

Al terminar de subir nuestro código, veremos este mensaje:

Fig. 7

Nivel Intermedio

Luego, conectamos el relé a la Placa Arduino. Para esto, usamos cables dupont macho hembra ya que estamos trabajando con 5V. Conectar el pin “+” a 5V, el pin “-” a GND y el pin “S” al pin digital “5” del Arduino.

El armado de este circuito tiene como función conseguir que el sensor de posición del tanque elevado (A) envíe una señal al Arduino para activar la bomba sumergible del tanque cisterna cuando el nivel de llenado sea inferior al máximo. Si en el tanque cisterna hay agua suficiente, se activará la bomba para enviar agua al tanque elevado. La activación se mantendrá hasta que se llene el tanque elevado o hasta que el nivel de agua del tanque cisterna sea inferior al mínimo.

Paso 3 - Programar el funcionamiento del dispositivo

Editamos nuestro programa para que contemple este tanque extra, así como su llenado, siempre que el primer tanque esté en condiciones de activar la bomba. Supongamos que la distancia mínima de ambos tanques es de 40 cm, y la máxima del primero, de 140 cm. El programa debe ser similar al siguiente modelo:

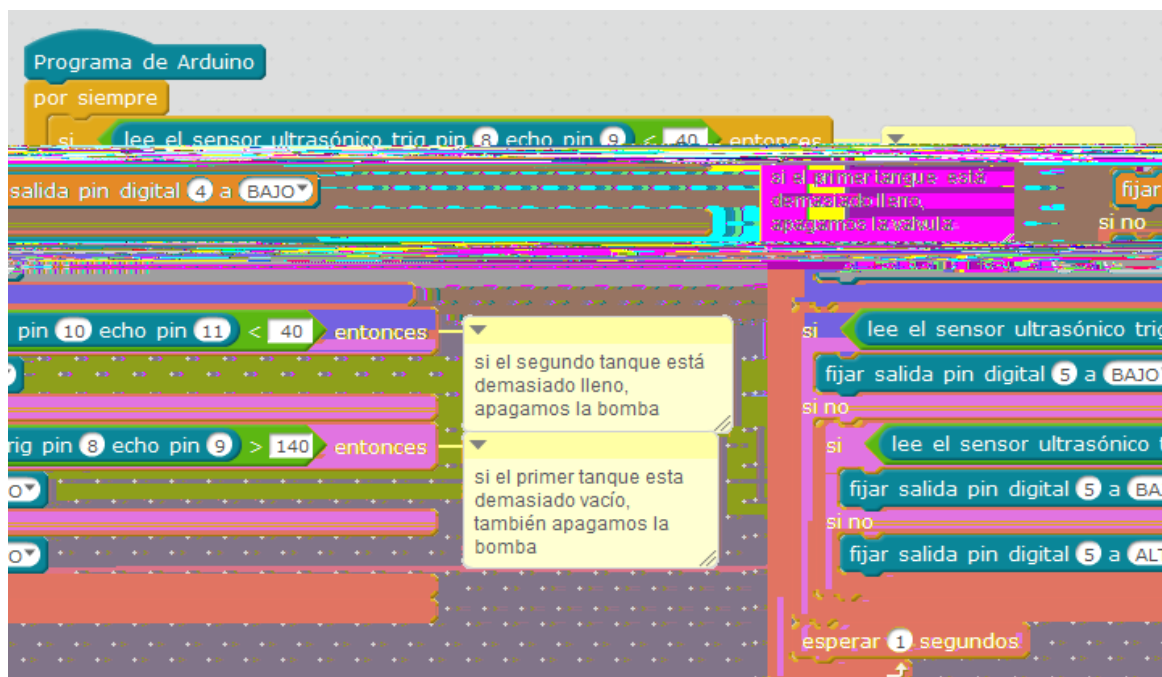


Fig. 9

Veremos que a continuación se muestra el código escrito generado por mBlock, que corresponde a este programa:

```
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>

double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;
```

```

float getDistance(int trig,int echo){
    pinMode(trig,OUTPUT);
    digitalWrite(trig,LOW);
    delayMicroseconds(2);
    digitalWrite(trig,HIGH);
    delayMicroseconds(10);
    digitalWrite(trig,LOW);
    pinMode(echo, INPUT);
    return pulseIn(echo,HIGH,30000)/58.0;
}

void setup(){
    pinMode(4,OUTPUT);
    pinMode(5,OUTPUT);
}

void loop(){
    if((getDistance(8,9)) < ( 40 )){
        digitalWrite(4,0);
    }else{
        digitalWrite(4,1);
    }
    if((getDistance(10,11)) < ( 40 )){
        digitalWrite(5,0);
    }else{
        if((getDistance(8,9)) > (140)){
            digitalWrite(5,0);
        }else{
            digitalWrite(5,1);
        }
    }
    _delay(1);
    _loop();
}

void _delay(float seconds){
    long endTime = millis() + seconds * 1000;
    while(millis() < endTime)_loop();
}

void _loop(){
}

```

Fragmento de código 2

Nivel Avanzado

Luego de haber terminado la instalación del tanque y la bomba, los vecinos de Melina le plantearon que sería muy útil poder tener acceso a la información que toma el sensor respecto al nivel del agua. Hoy en día, si quisieran saber cuánta agua hay en el tanque, deberían subir al techo, acceder al tanque y sacarle la tapa, lo cual resulta muy complicado.

Si pudieran registrar los datos que toma el sensor y monitorearlos desde sus casas, podrían saber cuánta agua hay disponible en el tanque para cuidar este recurso entre todo el consorcio.

En esta tercera instancia se programará el envío de datos obtenidos a un dispositivo móvil a través de internet (IoT).

Paso 1 - Introducción a Internet de las Cosas (IoT)

Internet de las Cosas (en inglés *Internet of Things*, abreviado IoT) es un concepto que refiere a la interconexión digital de objetos cotidianos con internet. Esta interconexión puede tener diversas funciones. Por ejemplo, puede utilizarse para monitorear la temperatura de un ambiente, enviando los datos obtenidos por un sensor a una central donde se recopile la información. De esta manera podría visualizarse en un dispositivo móvil la temperatura de un laboratorio, de un invernadero o de una sala de un hospital.

Para poder incorporar IoT a nuestro proyecto es necesario:

1. Un dispositivo capaz de conectarse a internet.
2. Un servidor que reciba y aloje los datos.

Existen diversas formas de lograr el cometido de registrar y almacenar los datos del sistema de tanques construido. En este caso, habrá que detallar cómo hacerlo con un módulo OBLOQ de DFRobot, y con los servidores de Adafruit.

El módulo UART OBLOQ es un dispositivo WiFi a serie pensado para desarrolladores no profesionales. Permite enviar y recibir datos mediante los protocolos HTTP y MQTT.

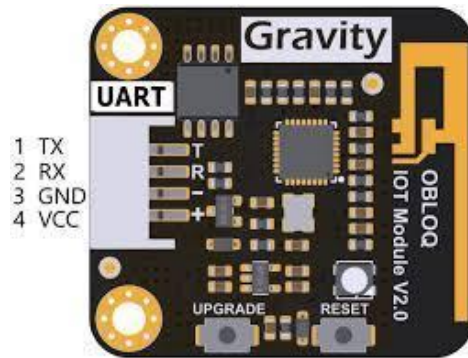


Fig. 10

Paso 2 - Crear un Panel de Control

En primer lugar, se explicará cómo crear un Panel de Control en Adafruit. Luego, se verá cómo vincular los controles del Panel con los datos que se intercambian con el dispositivo. Primero debemos crear una cuenta de usuario en io.adafruit.com.

Una vez que ingresamos con nuestro usuario, creamos un nuevo panel haciendo click en "Create a New Dashboard".

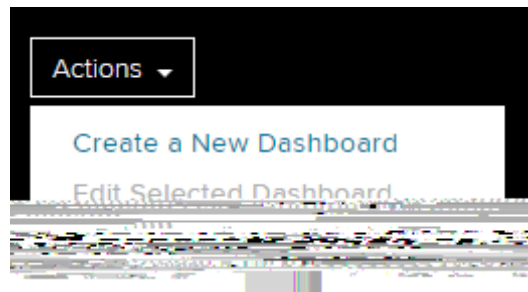


Fig. 11

Luego designamos un nombre y una descripción. Y al finalizar presionamos el botón "Create".

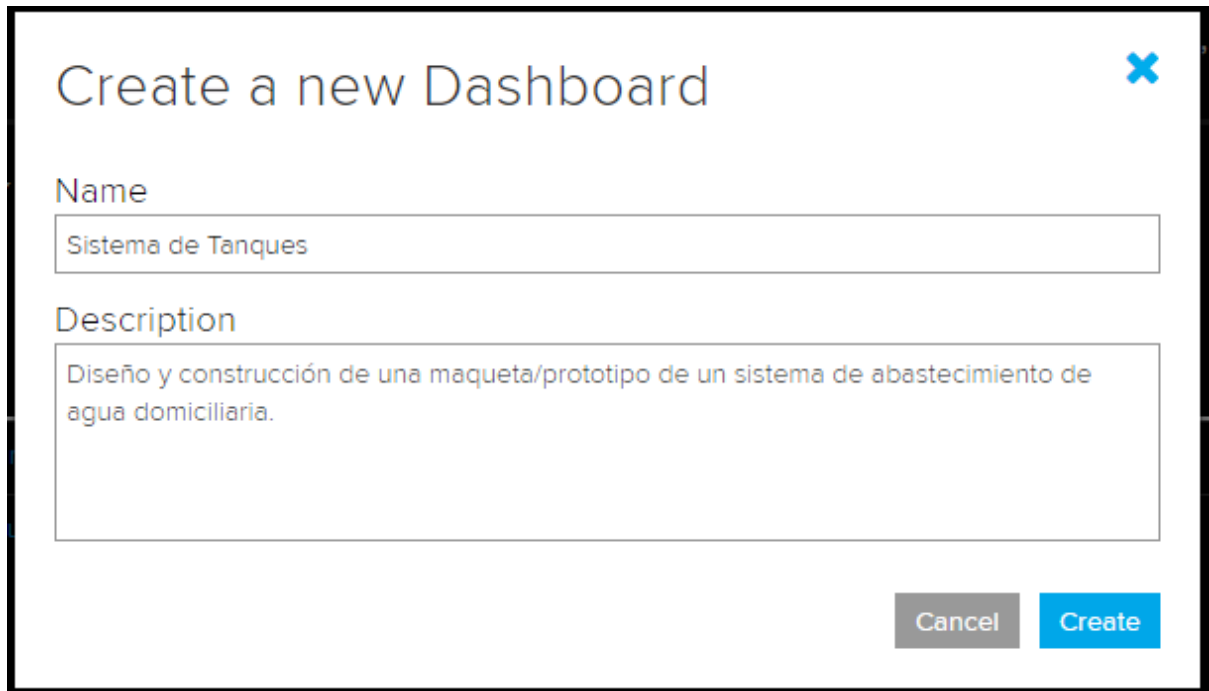
A screenshot of a web application dialog box titled "Create a new Dashboard" with a blue close button (X) in the top right corner. The dialog contains two input fields: "Name" with the text "Sistema de Tanques" and "Description" with the text "Diseño y construcción de una maqueta/prototipo de un sistema de abastecimiento de agua domiciliaria." At the bottom right, there are two buttons: "Cancel" (grey) and "Create" (blue).

Fig. 12

Seguidamente hacemos click en el nuevo panel creado y veremos una pantalla vacía.


Podemos comenzar a agregar bloques haciendo click en 



Fig. 15

Veremos una serie de controles posibles como en la siguiente imagen:

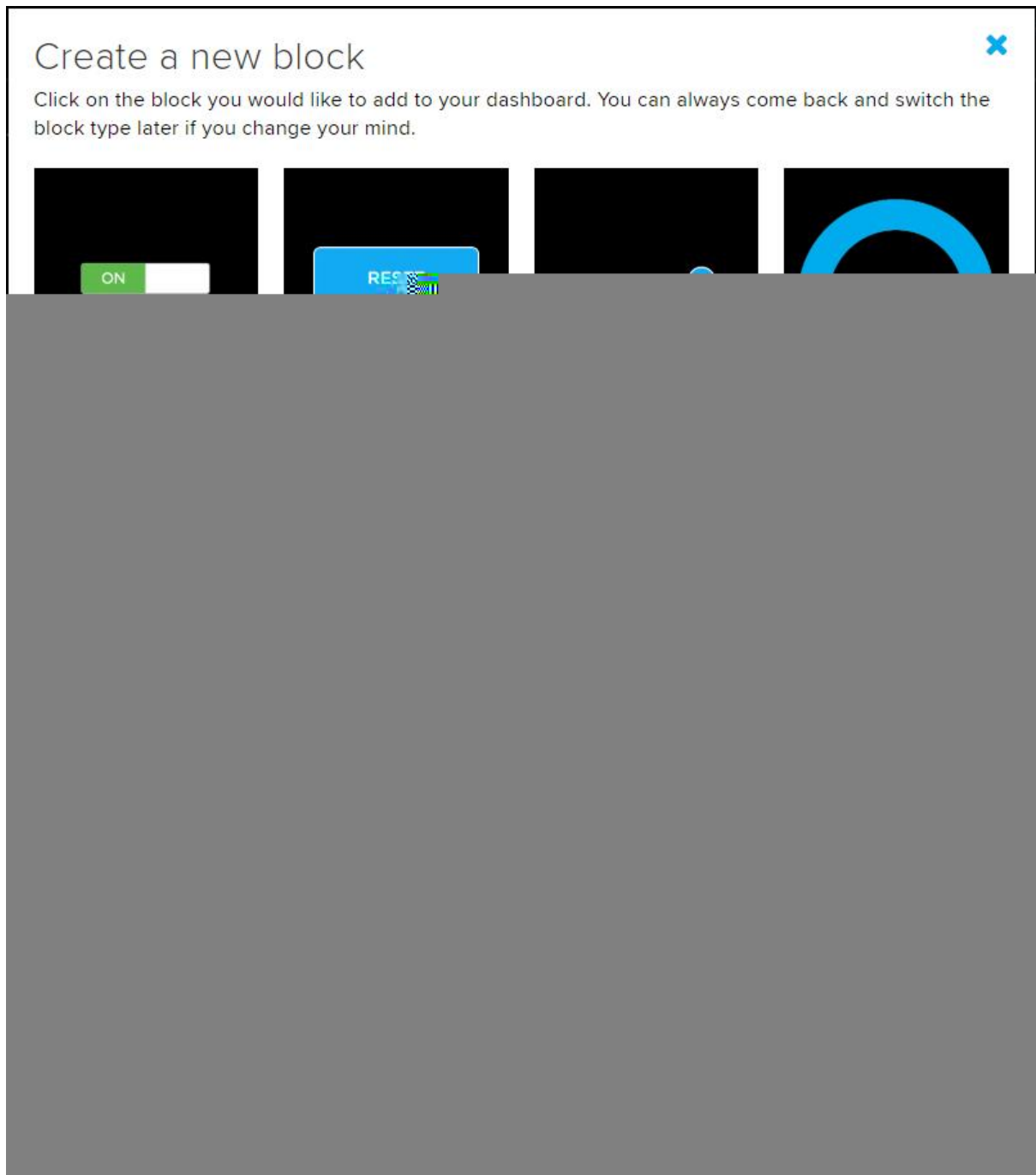


Fig. 16

Para nuestro sistema de tanques podríamos ubicar dos *Gauge* para mostrar el nivel de los tanques y dos *Indicator* para indicar el estado de la válvula y el relé.



Gauge
Fig. 17



Indicator
Fig. 18

Cuando agregamos un control al panel debemos asociarlo a un “*feed*”.

Un ***feed*** es una fuente de datos en la que uno puede publicar así como también se puede suscribir para recibir los datos de cierto feed.

Llamamos al primer feed “tanque1” y le damos al botón “Create”. En él publicaremos, desde nuestro dispositivo, la información sobre el nivel de llenado del primer tanque.

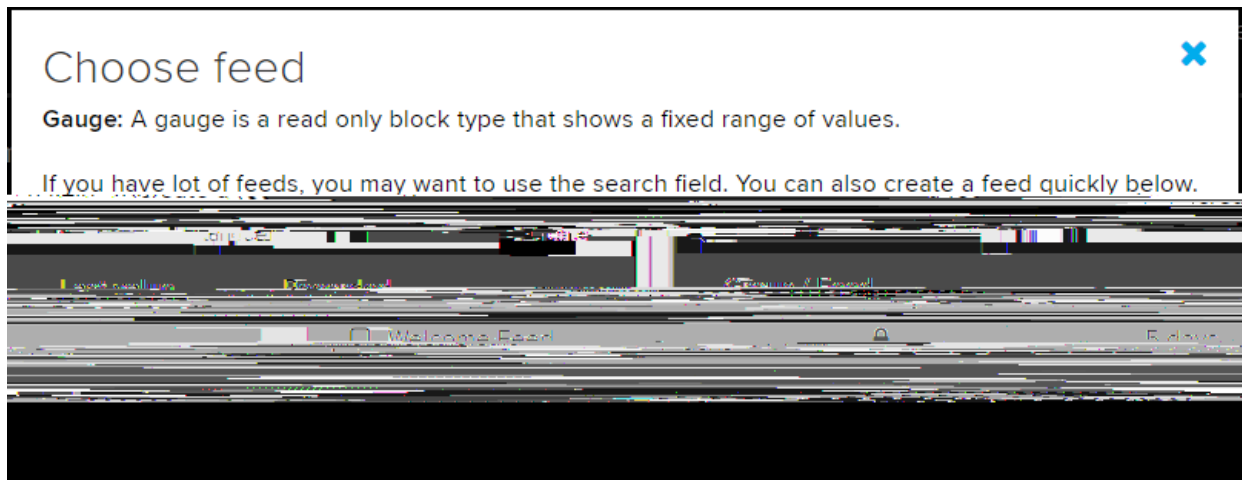


Fig. 19

Luego de crearlo, seleccionamos el feed creado y hacemos click en “*Next step*” (paso siguiente) para configurar nuestro control.

Fig. 20

Ajustamos los parámetros del bloque como se ve en la imagen.

Block settings

In this final step, you can give your block a title and see a preview of how it will look. Customize the look and feel of your block with the remaining settings. When you are ready, click the "Create Block" button to send it to your dashboard.

Block Title (optional)
Nivel Tanque 1

Gauge Min Value
0

Gauge Max Value
120

Gauge Width
25px

Block Preview
Nivel Tanque 1
45 cm

Test Value
45

Published Value
0 bytes

Previous step Update block

Fig. 21

Luego, hacemos click en "Create block" (crear bloque) para completar la operación.

Si se quiere, podemos modificar el tamaño y ubicación de los bloques haciendo click en "la rueda de configuración".

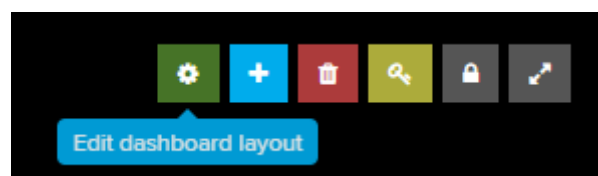


Fig. 22

Repetimos este procedimiento para “tanque2”, quedando algo similar a lo siguiente:

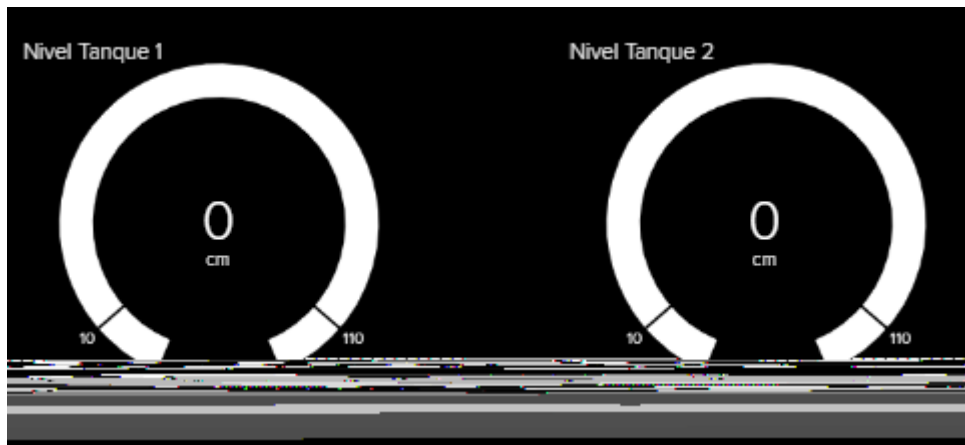


Fig. 22

Ahora creamos “valvula” (sin tilde) y ajustamos los parámetros del bloque como se ve en la imagen:

Fig. 23

Si se quiere, podemos modificar el tamaño y ubicación de los bloques haciendo click en “la rueda de configuración”.

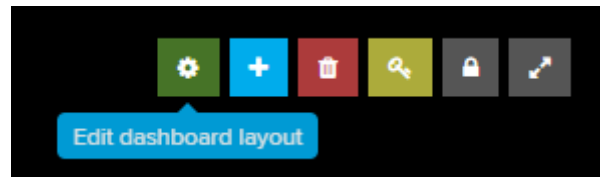


Fig. 24

Repetimos este procedimiento para “bomba”, quedando algo similar a lo siguiente:

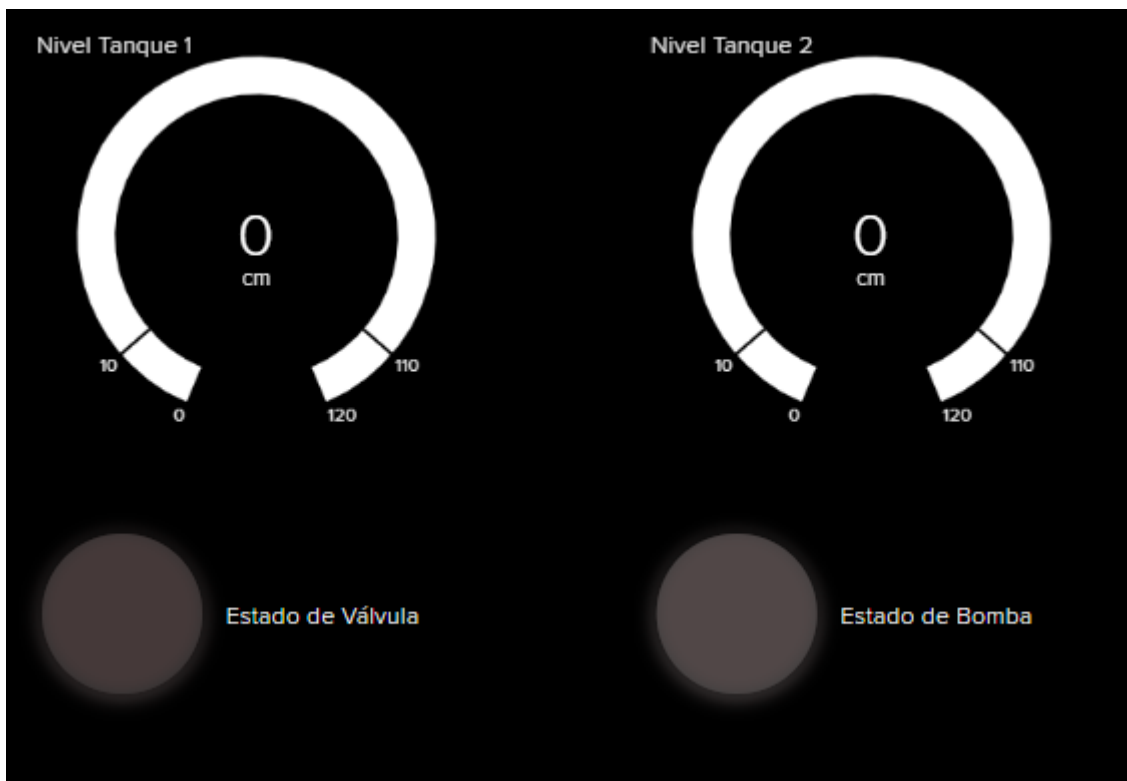
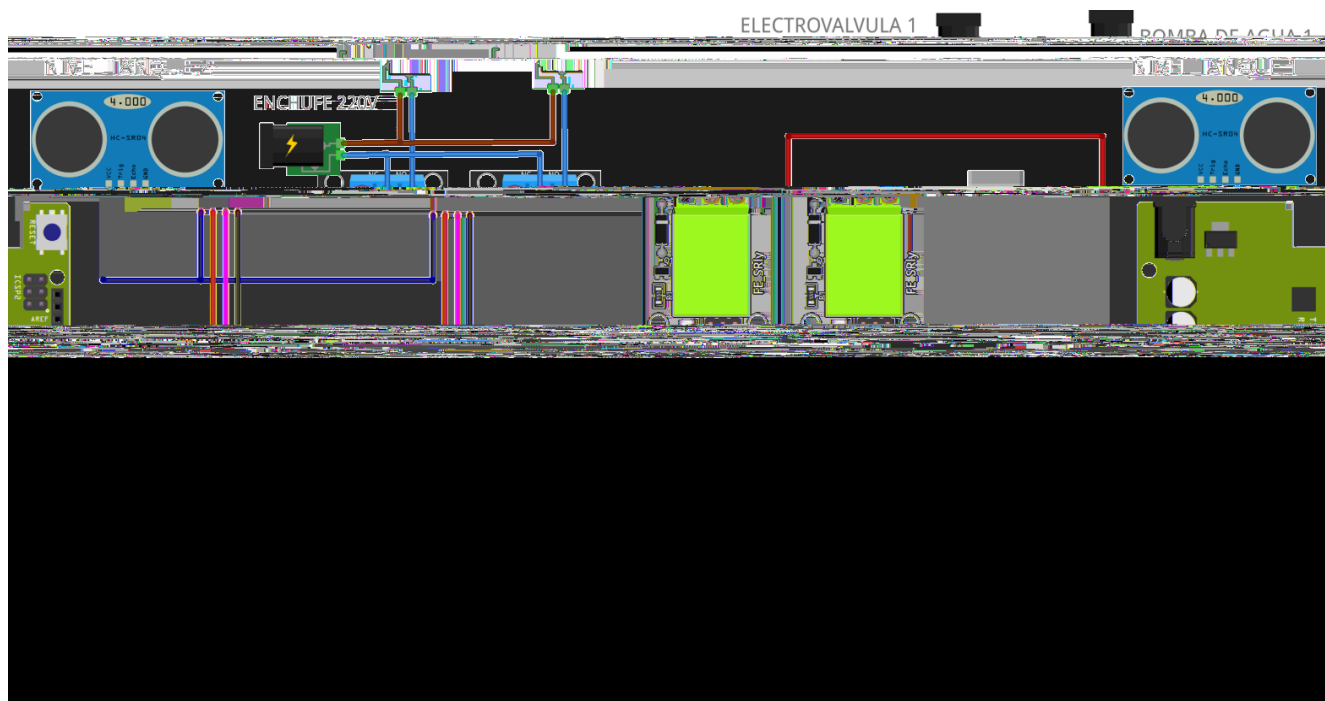


Fig. 25

Una vez realizado el panel, publicaremos en nuestros *feeds* los estados de nuestros tanques para poder monitorearlos de manera remota.

Paso 3 - Agregar módulo OBLOQ

Conectamos el módulo OBLOQ como se muestra en el esquema siguiente:



Esquema 3

Paso 4 - Arduino IDE

La programación por bloques tiene sus ventajas desde un punto de vista didáctico, pero tiene sus límites. Cuando el programa comienza a ser más complejo, hay ciertas operaciones que no se pueden resolver con bloques, o bien que intentar hacerlo resulta algo mucho más engorroso y difícil de leer que el código escrito.

Hasta ahora hemos visto cómo al realizar nuestra programación en bloques se generaba simultáneamente un código escrito en el área lateral derecha. Para esta sección de la actividad se propone trabajar directamente sobre el código, para ello vamos a recurrir a el entorno nativo de Arduino que llamamos Arduino IDE (entorno de desarrollo integrado).

Para ello descarga el Arduino IDE desde el siguiente enlace y luego procede con la instalación del mismo: www.enfoco.net.ar/sd

Veremos que se nos presenta la siguiente interfaz:

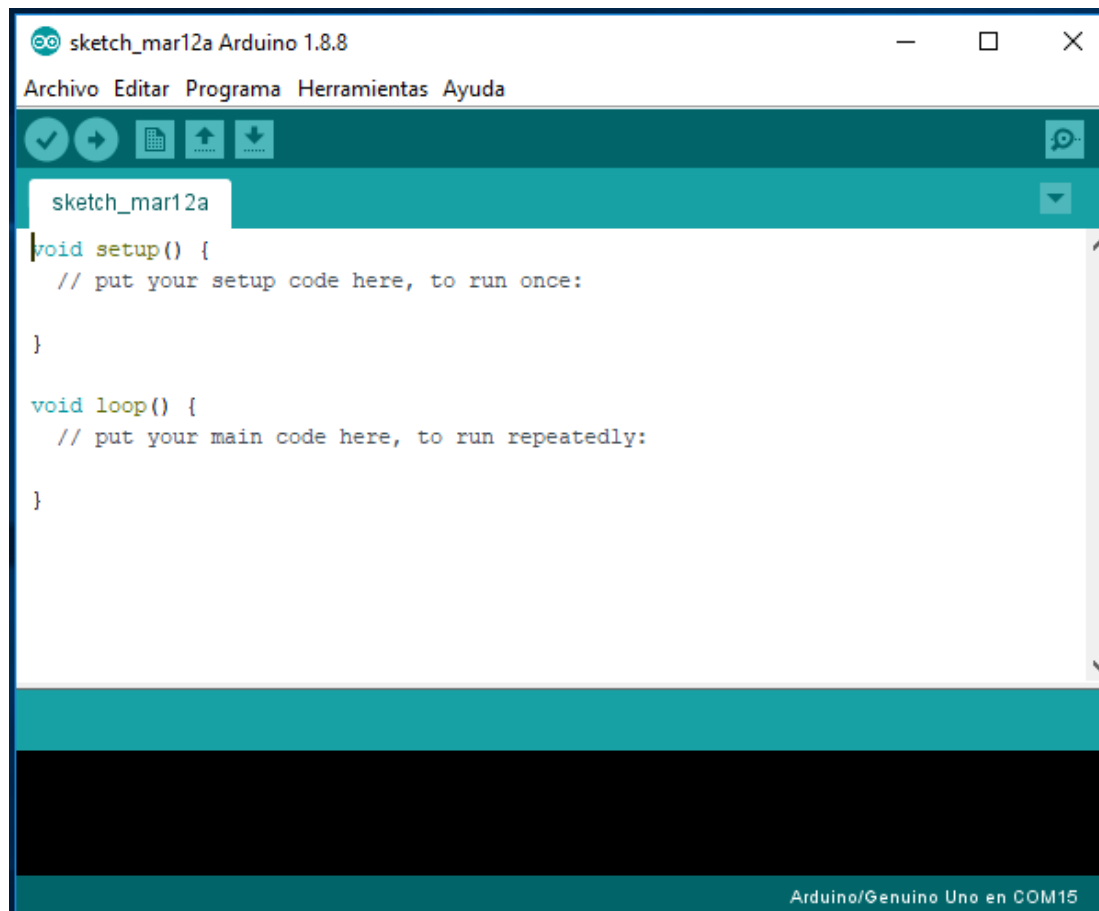


Fig. 26

A continuación, se presenta una estructura mínima de un *sketch* (un programa) de Arduino:

```
void setup() {  
  // Código de inicialización. Se ejecuta una sola vez.  
}  
  
void loop() {  
  // Código principal. Se ejecuta repetidamente.  
}
```

En líneas generales, un programa de Arduino es:

1. Un bloque de código que se ejecuta por única vez al inicializarse el dispositivo. Este bloque de código está contenido dentro de la función “setup” (se coloca dentro de `void setup() { y }`).

2. Un bloque de código que se ejecuta repetidamente luego de la función “setup”. Este bloque de código está contenido dentro de la función “loop” (se coloca dentro de `void loop() { y }`).

Después de `//` se incluyen comentarios para el lector que no tienen ningún efecto en el programa. Estos comentarios sirven para clarificar el código y que sea más fácil de interpretar para otras personas.

Los pasos para subir el código a través del Arduino IDE son simi

todo el procesamiento hasta que se cumpla el tiempo indicado. En otras palabras, cuando el programa entra al *delay* queda “colgado” por el período de tiempo establecido.

Al utilizar IoT, es conflictivo utilizar código “bloqueante” ya que, al detenerse el procesamiento, se impide también que el sistema realice otras operaciones que funcionan en simultáneo. Por ejemplo, las tareas de publicación y el mantenimiento constante de la conexión a internet.

Para evitar estos problemas, se puede utilizar una alternativa de código “no bloqueante”, como la función `millis()`. Esta función arroja un valor sobre un conteo de tiempo, que se realiza desde el momento en que se inicia el sistema. Es decir, funciona como un cronómetro (en milisegundos) que, cada vez que es consultada desde el código, “devuelve” el valor en el que se encuentra. De esta manera podemos pedirle al sistema que informe cuánto tiempo transcurrió desde el inicio de las operaciones para dar indicaciones temporales sobre una tarea, sin detener todas las demás.

A continuación se presenta un ejemplo de cómo se puede programar la intermitencia de un LED que se prenda y apague cada un segundo (expresado en 1000 milisegundos) sin utilizar código bloqueante. Lo haremos utilizando la función *millis* para consultar cuánto tiempo pasó.

```
int estado = LOW;
// Se declara "millisAnterior" con valor inicial igual a cero.
long millisAnterior = 0;

void setup() {
  // Inicializa el pin digital 13 como una salida.
  pinMode(13, OUTPUT);
}

void loop() {

  long millisActual = millis();

  if (millisActual - millisAnterior >= 1000) {
    // Conmuta el estado del led.
    if (estado == LOW) {
      estado = HIGH;
    } else {
      estado = LOW;
    }

    // Setea el estado del led.
    digitalWrite(13, estado);
  }
}
```

```

    // Guarda la última vez que conmutamos el led.
    millisAnterior = millisActual;
}

// Y en este punto nuestro procesador queda libre
// para realizar otras tareas.

}

```

Fragmento de código 3

En el ejemplo se puede observar que para tomar el valor de *millis* se define un valor inicial, al que llamamos “millisAnterior”, que es igual a cero. Luego, para consultar el tiempo transcurrido desde el inicio del sistema, se calcula la diferencia entre el valor de “millisActual” y el de “millisAnterior”. En el caso de nuestro ejemplo, cómo queremos generar una intermitencia de 1 segundo, necesitamos evaluar si esta diferencia es mayor o igual a 1000. En caso de que haya transcurrido más de un segundo, el sistema modifica el estado de la luz. Si ha transcurrido menos tiempo, el estado se mantiene estable.

En última instancia, se establece que, si esta diferencia es mayor o igual a 1000, se le asigne a “millisAnterior” el valor de “millisActual”. De esta manera, la diferencia entre “millisActual” y “millisAnterior” vuelve a ser cero hasta que vuelva a transcurrir otro segundo.

Paso 6 - Programar IoT

Utilizaremos la librería ObloqAdafruit para informar a Adafruit los niveles de los tanques. Podremos monitorear estos niveles desde el Panel de Control que hemos creado.

En primer lugar debemos instalar la librería en el Arduino IDE. Para esto debemos ingresar al menú Programa > Incluir Librería > Gestionar Librerías.

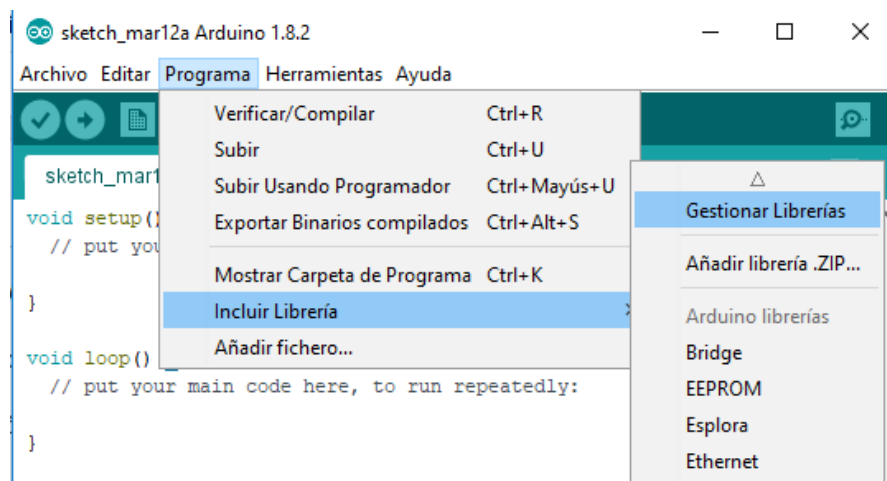


Fig. 28

Se abrirá una ventana con un buscador en margen superior. Debemos escribir Obloq, seleccionar la librería ObloqAdafruit y apretar el botón Instalar.

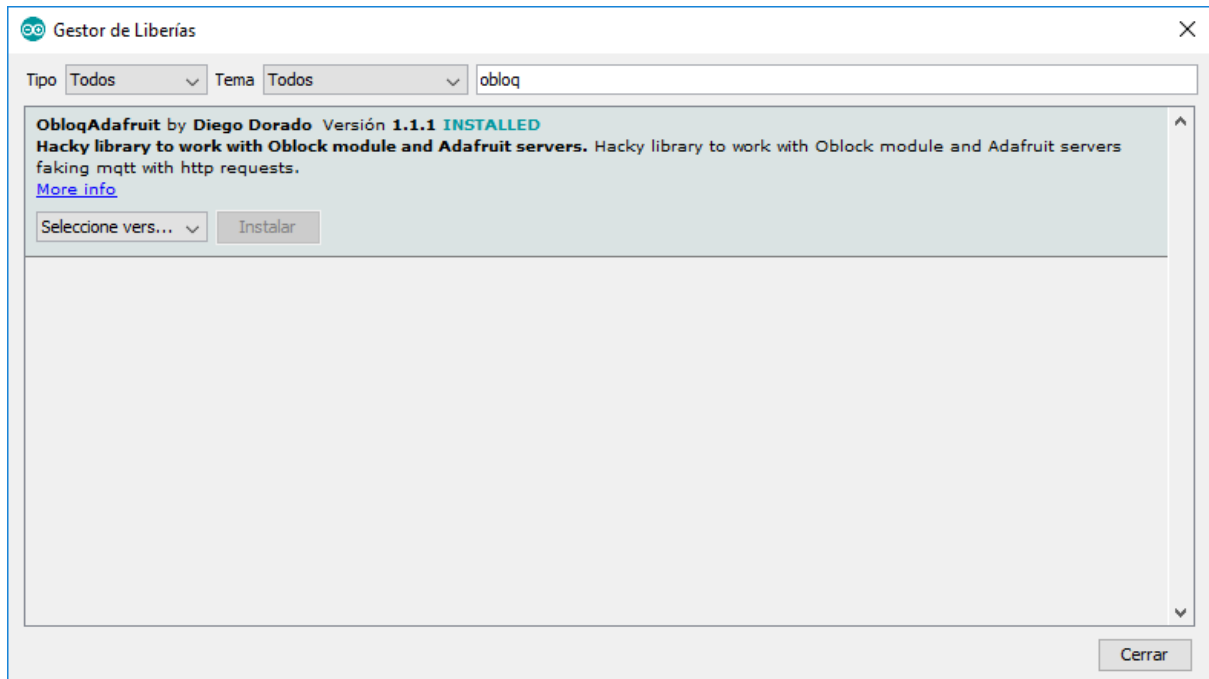


Fig. 29

En general, las librerías traen códigos de ejemplo como referencia. Abrimos el ejemplo “Publicar” ubicado en Archivo -> Ejemplo -> OBLOQADAFRUIT-master.

Debemos reemplazar el SSID de la WiFi, su password, el IO_USERNAME e IO_KEY por los que copiamos de Adafruit. Para ello maximizamos la pantalla de Adafruit y hacemos click en el ícono de la “llave”.

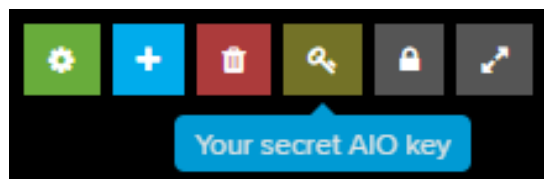


Fig. 30

Copiamos el código que nos ofrece para Arduino, con nuestro usuario y key.

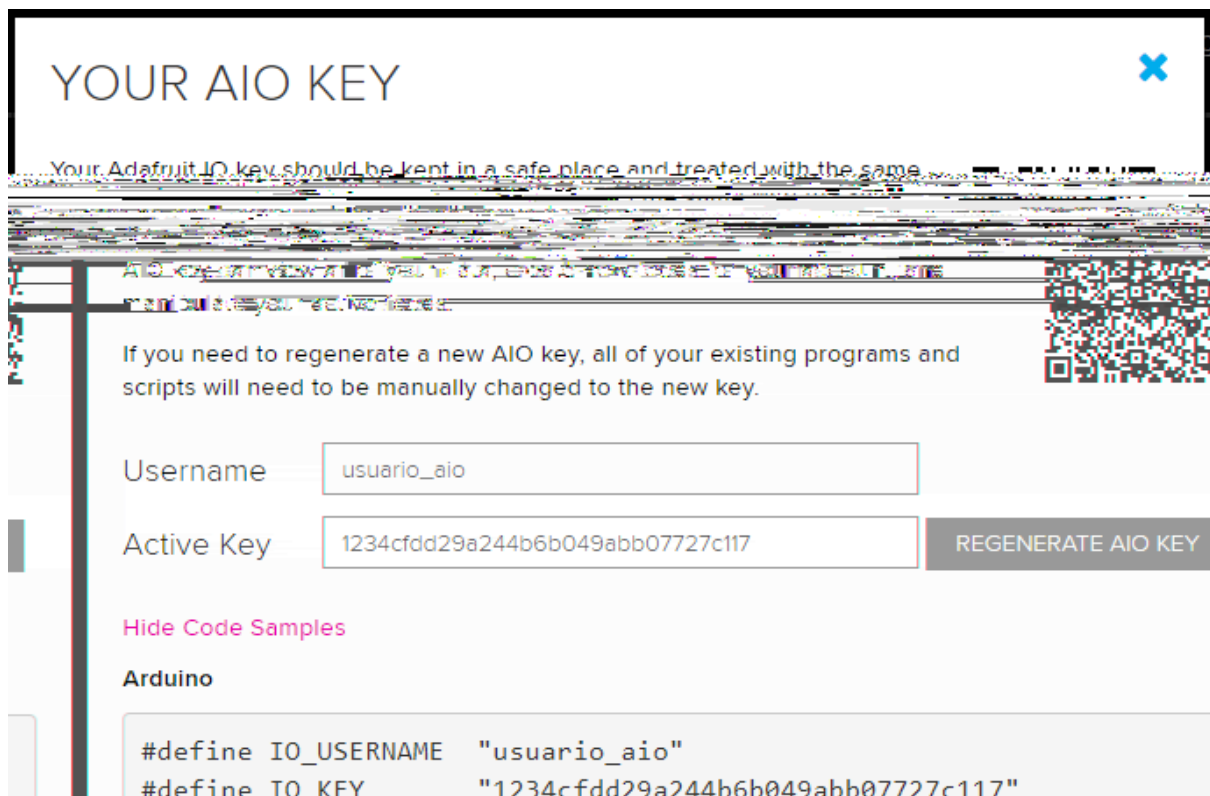


Fig. 31

Estos datos aparecen en el código de la siguiente manera:

```
#define IO_USERNAME "usuario_adafruit"
#define IO_KEY "key_adafruit"
```

Se deberán reemplazar en esas dos líneas el usuario y key que se hayan obtenido en Adafruit. Por ejemplo:

```
#define IO_USERNAME "usuario_aio"
#define IO_KEY "1234cfdd29a244b6b049abb07727c117"
```

También modificaremos `softSerial(10,11)` por `softSerial(6,7)` ya que así es como lo conectamos en nuestra placa, quedándonos el siguiente código:


```
#include "SoftwareSerial.h"
#include "ObloqAdafruit.h"

// Indicamos conexión de wifi.
#define WIFI_SSID      "SSID_de_Wifi"
#define WIFI_PASSWORD  "PWD_de_WIFI"

// Copiamos las credenciales obtenidas anteriormente en Adafruit.
#define IO_USERNAME    "usuario_adafruit"
#define IO_KEY         "key_adafruit"

SoftwareSerial softSerial(6,7);
ObloqAdafruit olq(&softSerial,WIFI_SSID,WIFI_PASSWORD,IO_USERNAME,IO_KEY);
```

El *setup* debe incluir la línea de inicialización del softwareSerial:

```
void setup()
{
    softSerial.begin(9600);
}
```

Se debe agregar también la función de actualización o “update”: `olq.update()`. Por esto es importante que nuestro código no sea bloqueante.

```
void loop()
{
    olq.update();
    // ..
    // ..
}
```

Para publicar un *feed*, utilizaremos la función `publish` del objeto `olq` :

```
// informar que el nivel del tanque1 es de 92 cm
olq.publish("tanque1", 92);
```

Finalmente, nuestro programa de tanques con IoT queda como sigue:

```

// Incluimos las librerías que usaremos.
#include "SoftwareSerial.h"
#include "ObloqAdafruit.h"

// Indicamos conexión de wifi.
#define WIFI_SSID      "SSID_de_Wifi"
#define WIFI_PASSWORD  "PWD_de_WIFI"

// Copiamos las credenciales obtenidas anteriormente en Adafruit.
#define IO_USERNAME    "usuario_adafruit"
#define IO_KEY         "key_adafruit"

// Indicamos a qué distancia debe cortar la válvula.
#define DISTANCIA_MINIMA1 40
#define DISTANCIA_MAXIMA1 140
#define DISTANCIA_MINIMA2 40
#define VALVULA 4
#define BOMBA 5

// Definimos la función que calcula la distancia
// del sensor ultrasónico (en cm).
float getDistance(int trig,int echo){
    pinMode(trig,OUTPUT);
    digitalWrite(trig,LOW);
    delayMicroseconds(2);
    digitalWrite(trig,HIGH);
    delayMicroseconds(10);
    digitalWrite(trig,LOW);
    pinMode(echo, INPUT);
    return pulseIn(echo,HIGH,30000)/58.0;
}

// Indicamos a qué pines se conecta el módulo OBloq.
SoftwareSerial softSerial(6,7);
// Declaramos nuestro objeto IoT.
ObloqAdafruit
olq(&softSerial,WIFI_SSID,WIFI_PASSWORD,IO_USERNAME,IO_KEY);

long millisAnterior = 0;

void setup() {
    pinMode(VALVULA, OUTPUT); // Seteamos VALVULA como salida.
    pinMode(BOMBA , OUTPUT); // Seteamos BOMBA como salida.

```

```

// Inicializamos la comunicación con el módulo OBloq.
softSerial.begin(9600);
}

void loop() {

    long millisActual = millis();

    //Ejecutamos solo cada 3 segundos.
    if (millisActual - millisAnterior >= 3000) {

        // Leemos la distancia del agua en cm en el primer tanque.
        int distancia1 = getDistance(8,9);
        if(distancia1 < DISTANCIA_MINIMA1){
            // Si la distancia medida es menor que
            // la distancia mínima, apagamos la válvula.
            digitalWrite(VALVULA, LOW);
            olq.publish("valvula", 0); // Publicar en Adafruit.
        }else{
            // Si no, la encendemos para que cargue el tanque.
            digitalWrite(VALVULA, HIGH);
            olq.publish("valvula", 1); // Publicar en Adafruit.
        }

        // Leemos la distancia del agua en cm en el segundo tanque.
        int distancia2 = getDistance(10,11);
        if(distancia2 < DISTANCIA_MINIMA2){
            // Si la distancia medida es menor que
            // la distancia mínima, apagamos la bomba.
            digitalWrite(BOMBA, LOW);
            olq.publish("bomba", 0); // publicar en adafruit
        }else{
            // Si no, primero debemos verificar que el primer
            // tanque no esté demasiado vacío, ya que la
            // bomba se arruina si se enciende y no está
            // completamente sumergida.

            if(distancia1 > DISTANCIA_MAXIMA1){
                // El nivel de agua está muy bajo.
                // Apago la bomba.
                digitalWrite(BOMBA, LOW);
                olq.publish("bomba", 0); // Publicar en Adafruit.
            }
        }
    }
}

```


Cierre

Una vez finalizado este proyecto,

encuentra el objeto, esto se debe a que la velocidad de propagación del sonido en el aire es lineal. Este tipo de sensor tiene cuatro pines de conexión, de estos se utilizan dos para la alimentación eléctrica del mismo (VCC y GND). Los dos pines restantes se utilizan para generar el pulso (Trigger) y para detectar la llegada del mismo (Echo).

La utilización del mismo requiere programar el mecanismo de disparo, medición de tiempo y sensado de la llegada del rebote. En general la mayoría de los entornos de programación facilitan alguna librería que resuelva de manera simple la gestión de estos procesos, pudiendo acceder como usuario directamente a la información en Centímetros de la medición realizada.

Electroválvula: Se trata de un actuador que permite o interrumpe el paso de un fluido a través del mismo. Existen versiones que trabajan con 12 V así como también los que trabajan directamente con 220 V, permitiendo trabajar en un sistema de mayor nivel de caudal. El accionamiento del mismo se realiza directamente a través de la alimentación de la misma, por lo que tiene únicamente 2 pines para la conexión. Los estados posibles son activado o no, por lo que se puede controlar mediante una salida digital y un relé.

Relé: Dispositivo electromagnético que funciona como un interruptor controlado por un circuito eléctrico. Por medio de un electroimán se acciona uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes. Se utiliza comúnmente para aislar eléctricamente dos circuitos así como también permitir controlar con bajo voltaje a dispositivos que utilizan mayor voltaje. Arduino trabaja con señales de 5 V que mediante un relé permiten controlar la activación o desactivación de dispositivos de 220 V.

Internet de las cosas

Panel de Control Adafruit: Los sistemas IoT trabajan apoyándose en un servidor que se encarga de gestionar la información que reportan los diversos sensores así como responder a las consultas de los dispositivos que buscan acceder a esta información (mostrarla en pantalla, tomar decisiones, etc). Adafruit es una plataforma online con posibilidad de uso gratuito que ofrece el servicio de gestión de esta información. La misma ofrece un alto grado de compatibilidad con diversos estándares de trabajo IoT, principalmente orientada al uso educativo.

Feed: Fuente de datos en la que uno puede publicar y a la que puede suscribirse. Es decir, permite enviar datos, para que estos sean almacenados en el tiempo así como también leerlos, recibiendo las actualizaciones de quienes estén publicando allí. Es una forma de almacenar información en una gran base de datos de forma ordenada, utilizando el concepto de etiquetas tanto al momento de escribirla como el de leerla.

Reconocimientos

Este trabajo es fruto del esfuerzo creativo de un enorme equipo de entusiastas y visionarios de la pedagogía de la innovación, la formación docente, la robótica, la programación, el diseño y la impresión 3D. Les agradecemos por el trabajo en equipo inspirador para traer a la realidad la obra que, en forma conjunta, realizamos INET y EDUCAR del Ministerio de Educación, Cultura, Ciencia y Tecnología de la Nación Argentina.

Contenidos

Equipo INET

Alejandro Anchava
Joreliz Andreyana Aguilera Barragán
Omar Leandro Bobrow
Alejandro Cesar Cáceres
Ezequiel Luberto
Gustavo Roberto Mesiti
Alejandro Palestrini
Judit Schneider
Pablo Trangone

Equipo Educar:

Pablo Aristide
Mayra Botta
Anabela Cathcarth
Eduardo Chiarella
María Laura Costilla
Diego Dorado
Facundo Dyszel
Federico Frydman
Matías Rinaldi
Uriel Rubilar
Camila Stecher
Carolina Sokolowicz
Nicolás Uccello

Para la confección de esta obra se contó con el apoyo de la Universidad Pedagógica Nacional "UNIPE". En particular en el desarrollo de los capítulos 1 y 2, los cuales estuvieron a cargo de los profesores Fernando Raúl Alfredo Bordinon y Alejandro Adrián Iglesias.

Producción y comunicación

Juliana Zugasti

Diseño y edición

Leonardo Frino
Mario Marrazzo

Corrección de estilo

María Cecilia Alegre

Agradecimientos especiales

Mariano Consalvo. Equipo ABP

Damián Olive. Equipo de ABP

María José Licio Rinaldi, Directora Nacional de Asuntos Federales INET, quien siempre acompañó a este equipo en todas las gestiones para su implementación

Estamos comprometidos en instalar la innovación en la escuela secundaria técnica: la robótica, la programación, el pensamiento computacional, los proyectos tecnológicos, el ABP, la impresión 3D, de manera más accesible para todos.

Agradecemos enormemente, docente, tu continua dedicación y compromiso con el futuro de tus estudiantes.

¡Estamos ansiosos por saber qué es lo que vamos a crear juntos!