

TP DE DESARROLLO INTEGRAL



FACULTAD DE TECNOLOGÍA INFORMÁTICA

CARRERA: **Ingeniería en Sistemas Informáticos**

ALUMNO: **Perez Demonty, Julian; Destito Altieri, Juan Cruz; Lastra, Julian**

MATERIA: **Sistemas de Hardware para la administración.**

Trabajo Práctico: **PLC**

DOCENTE: **Kamlofsky, Jorge**

AÑO: **2022**

Arduino Cloud IoT con ESP8266

Estructura del proyecto:

Abstracto	3
ESP8266	
<i>Utilización en el trabajo</i>	
<i>Código utilizado</i>	
<i>Diagrama de conexión</i>	4
Arduino IoT	
<i>Conexión Wi-Fi</i>	
<i>Código en el IDE</i>	6
Arduino Nano	
<i>Osciloscopio</i>	
<i>Generador de señales.</i>	10
PowerBank	13
WIFI 802.11x	
<i>Conectividad.</i>	13
C++	14
GitHub	
Alcance, Objetivos y Motivación	14
Bibliografía, enlaces y referencias.	16
Repositorios de los proyectos.	17

Arduino Cloud IoT con ESP8266

Destito, Lastra, Perez Demonty.

Kamlofsky, Jorge.

Sistemas de Hardware para la Administración, Ing. en Sistemas Informáticos, Universidad Abierta Interamericana, Lomas de Zamora, Argentina.

Presentado el 22 de Julio, 2022

Abstracto. Actualmente con el avance de la tecnología electrónica e informática, contamos con controladores y computadoras cada vez más avanzadas. Estos controladores se pueden comunicar con diferentes dispositivos utilizando redes wifi (802.11x) a través de lo que hoy conocemos como Plataformas Cloud. Los mismos se programan utilizando el lenguaje C++ con la interfaz de Arduino IDE. Para este proyecto vamos a demostrar la conexión/conectividad completa entre diferentes dispositivos y su interacción con la nube utilizando un Osciloscopio para visualizar las señales y un teclado con pulsos de onda cuadrada además de diferentes señales de entrada.

1. ESP8266

[1] El ESP8266 es una solución que brinda un SoC (System on a Chip) Wi-Fi (Capítulo 6) para la continua demanda de los usuarios, con un consumo eficiente de corriente, con un diseño compacto y una performance con gran estabilidad en el campo del Internet of Things. El mismo también cuenta con un procesador de 32-bits y un integrado SRAM para el procesamiento de datos. Este mismo puede ser utilizado y se comunica mediante los GPIOs (Input Output Pins).

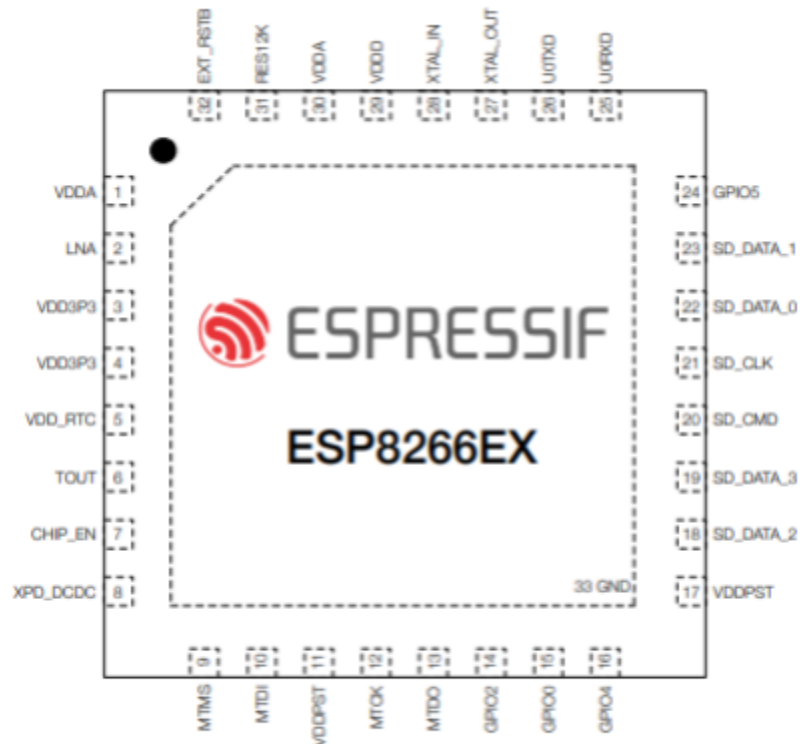


Diagrama de GPIOs del micro-controlador

Utilización en el trabajo. Para este trabajo en concreto vamos a utilizar el pin A0 (Analog 0) el cual está programado como entrada de datos. Por el mismo van a ingresar los datos de los

diferentes dispositivos o sensores que se requieran, los cuales van a ser publicados en la nube de Arduino utilizando la conexión Wi-Fi integrada.

Código utilizado. Para programar este dispositivo, utilizamos el lenguaje C++ (Capítulo 8) con las librerías integradas y creadas por Arduino y la comunidad. Para los diferentes cambios que se fueron realizando utilizamos el repositorio de GitHub (Capítulo 9).

```
Arduino IoT Cloud Thing "ESP8266"

https://create.arduino.cc/cloud/things/a684c126-e108-4fb5-a93f-5b85d5eb051a

#include "thingProperties.h"

//Vars float fc = 1;
//Inputs const int voltageInput = A0;

void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without
  blocking if none is found
  delay(1500);

  // Defined in thingProperties.h
  initProperties();
  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
}

void loop() {
  ArduinoCloud.update();

  onSignalVoltageChange();
}
```

```

/* Since SignalVoltage is READ_WRITE variable, onSignalVoltageChange() is
executed every time a new value is received from IoT Cloud. */

void onSignalVoltageChange() {

    // Add your code here to act upon SignalVoltage change

    vg = analogRead(voltageInput);

    signalVoltage = vg;

}

```

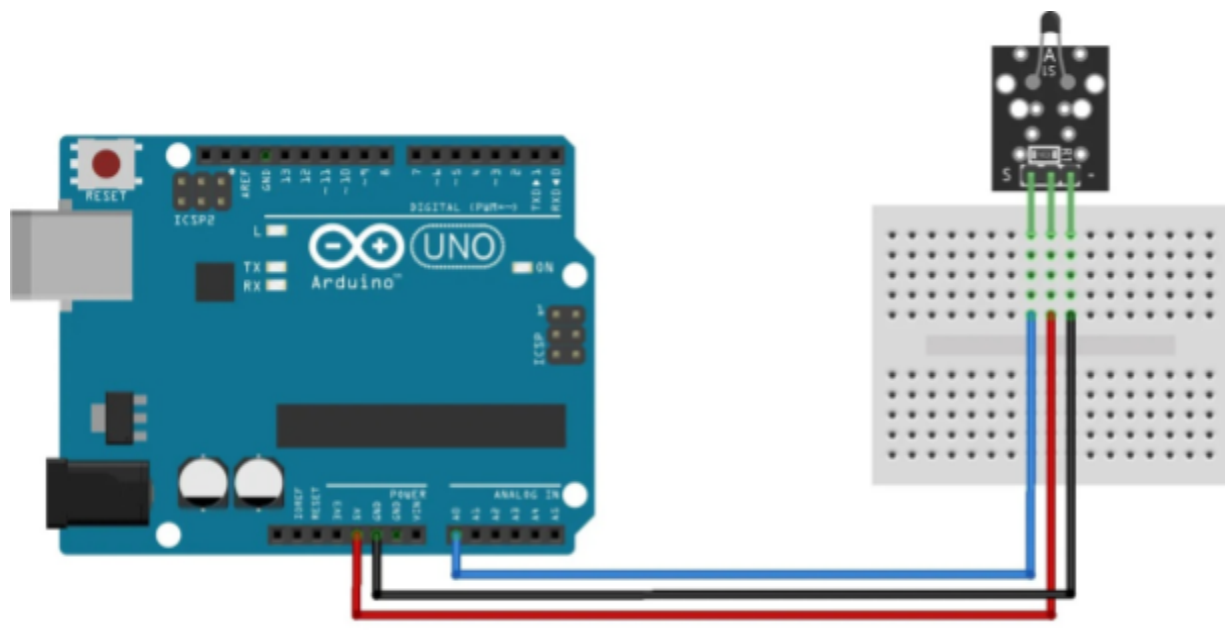
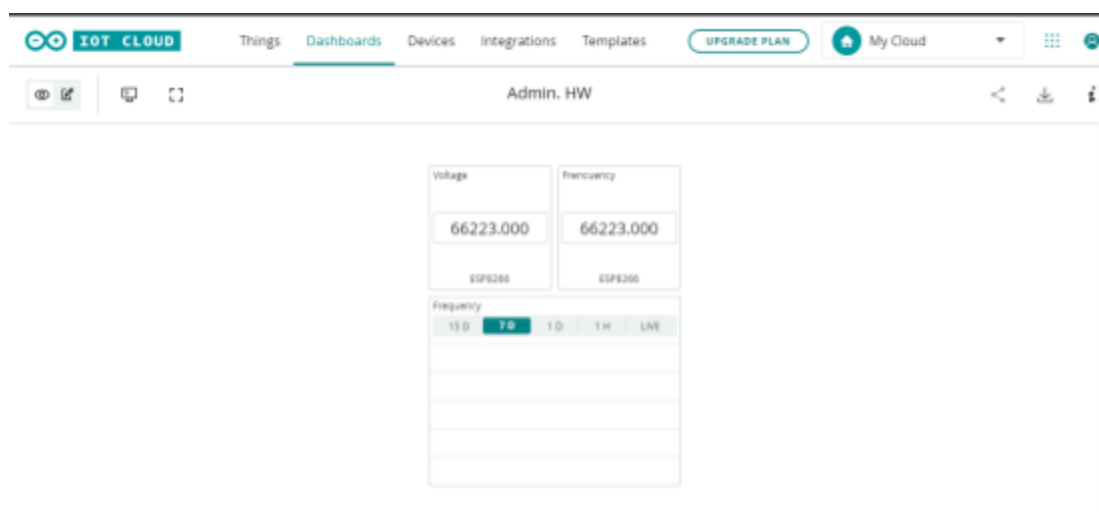


Diagrama de conexión

2. Arduino IoT

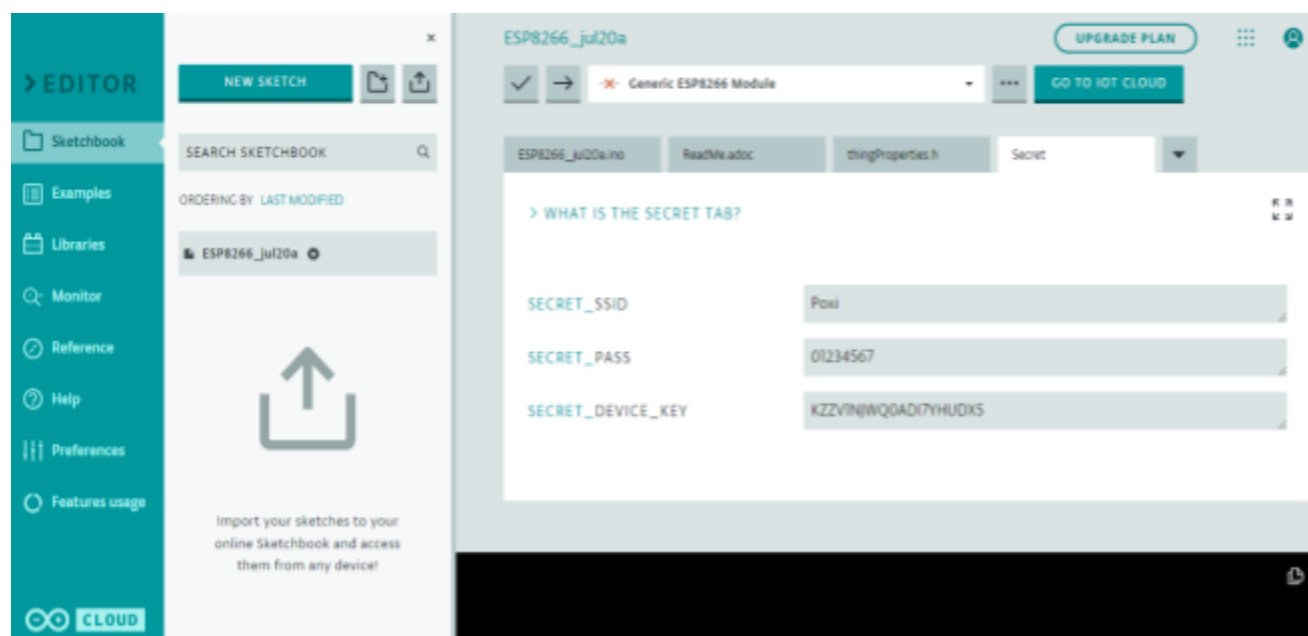
[2] La IoT Cloud de Arduino es una plataforma online que nos permite crear fácilmente, desarrollar, poner a funcionar y monitorear proyectos IoT. La misma le permite a cualquiera

utilizando una interfaz amigable, tener una solución todo en uno para poder configurar, escribir código, subir y visualizar los resultados al instante. Para utilizar esta plataforma se requiere una placa de desarrollo compatible, que en nuestro caso va a ser el microcontrolador ESP8266 de Nodemcu. Aquí mismo vamos a poder visualizar los datos obtenidos y requeridos.



Interfaz de Arduino IoT Cloud

Conexión Wi-Fi. Para la conexión Wi-Fi con nuestro proyecto, esta plataforma nos brinda una alternativa simple para poder guardar nuestras claves con seguridad, utilizando la encriptación SHA-256. Esta conectividad es la más utilizada y conveniente para proyectos de bajo rango con fines educativos o profesionales.



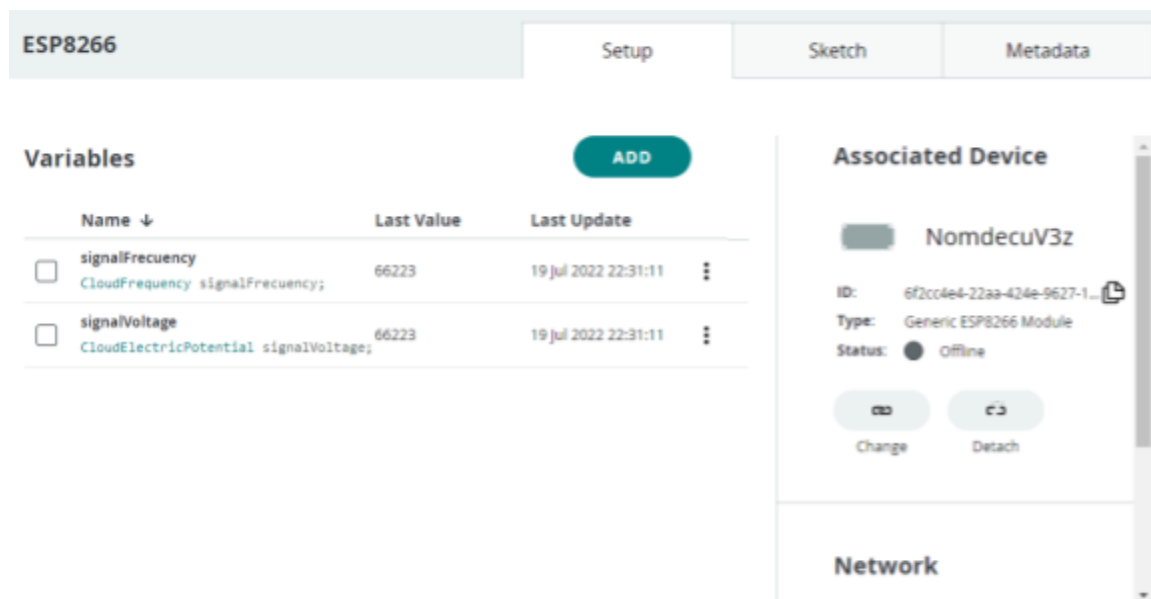
Interfaz para el almacenamiento de claves

Código en el IDE. Para el código, utilizamos el editor integrado de la plataforma, el cual nos permite “codear” en la nube, mediante nuestro navegador. A esto hay que agregarle que nos descargamos un agente llamado “Arduino IoT Agent” el cual nos va a permitir grabar los datos en nuestro controlador mediante la comunicación serial interna de la placa ESP.



Interfaz de edición de código

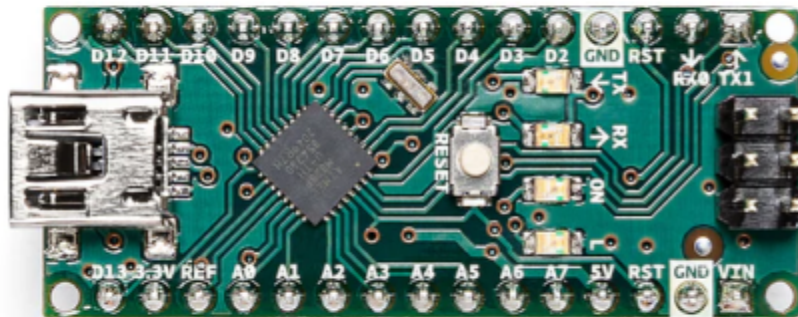
Datos a la nube. Para poder conectar nuestra placa a la nube, tenemos que utilizar la plataforma de “Things” la cual nos permite crear variables, las cuales van a recibir y almacenar los datos que son enviados por el controlador 8266 utilizando la conexión Wi-Fi definida en la sección de “Network”.



Interfaz de Things

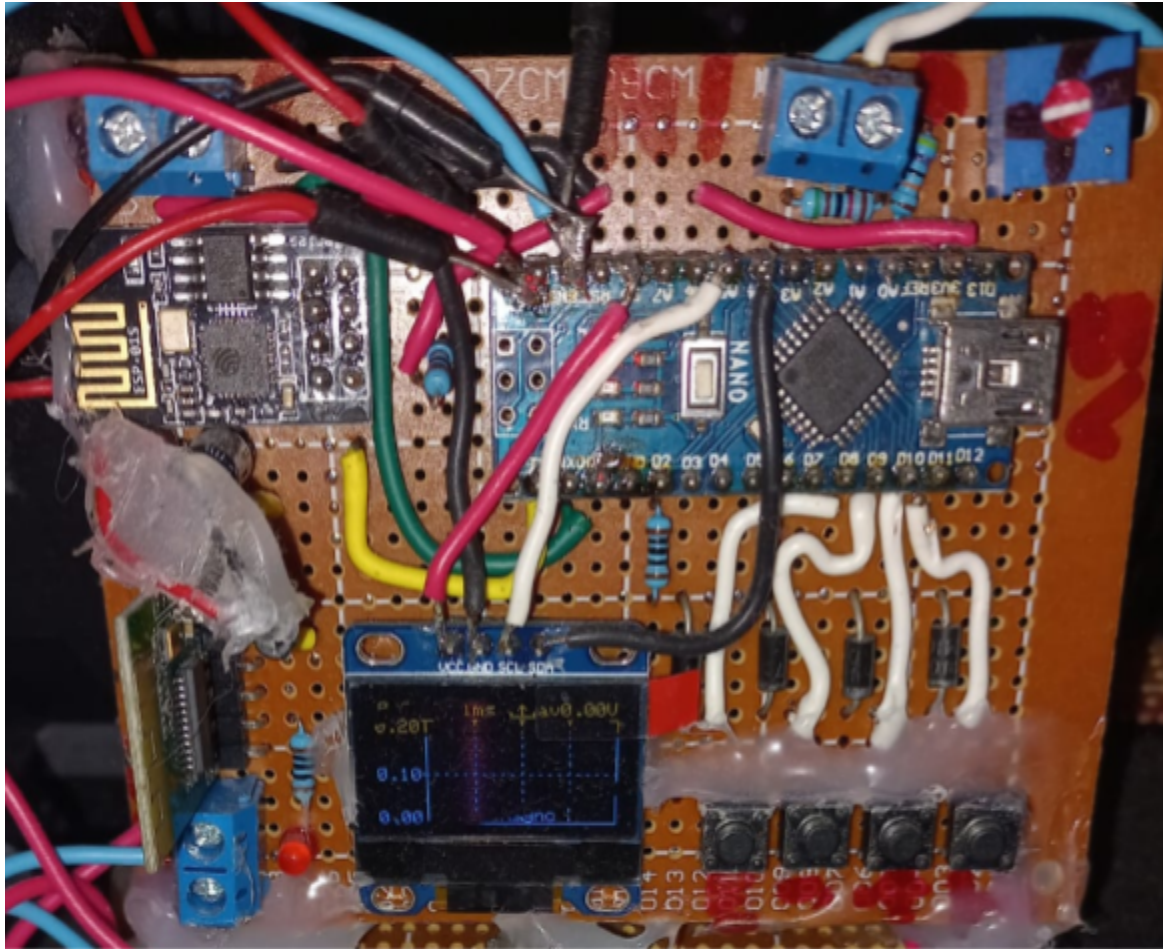
3. Arduino Nano

[3] El Arduino Nano es una pequeña, completa y protoboard-friendly (amigable), basada en el circuito integrado ATmega328. Para su alimentación se utiliza la interfaz mini-usb, tomando como entrada una tensión entre 3 y 5 Volt. La placa cuenta con 32 KB de memoria de procesamiento, una SRAM de 2 KB y una EEPROM de 1 KB donde se almacena el código generado. En nuestro proyecto lo vamos a utilizar con dos propósitos, el primero para el generador de señales, el cual mediante una librería de tonos, vamos a generar frecuencias de ondas cuadradas que tienen referencia a las notas musicales (Do, Re, Mi, Fa, Sol y Si), y como segundo el manejo del osciloscopio para poder visualizar las ondas que son procesadas en tiempo real.



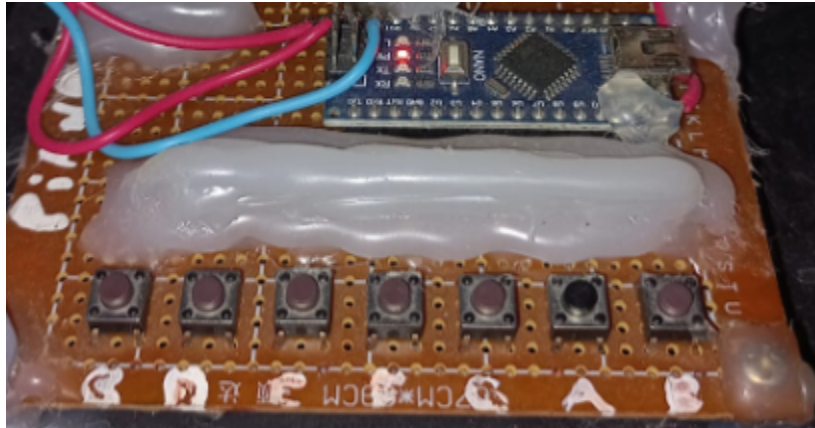
Arduino Nano

Osciloscopio. Para poder realizar este proyecto, utilizamos un código de uso libre descargado desde github, el cual modificamos para nuestro propio uso. El mismo se encarga de la lectura de señales de entrada mediante el pin 3 que se utiliza como entrada analógica. Además utilizamos el protocolo de comunicación I2C para poder enviar la información que llega por dicha entrada, a la pantalla OLED.



Generador de señales. Para poder realizar este proyecto, utilizamos la librería “Tone” que nos permite generar ondas cuadradas de una determinada frecuencia y duración de pulso, para así poder lograr las notas deseadas. En nuestro caso contamos con las 7 notas de la escala de Do en su 5ta Octava. Para poder generar las frecuencias específicas se debe pulsar el botón

correspondiente a la nota deseada. En la imagen de referencia las notas están en su escala correspondiente con la nomenclatura Inglesa. (C = Do, D = Re y así sucesivamente).



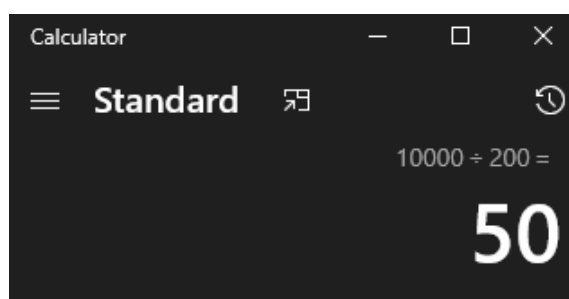
Generador de señales

4. PowerBank

La powerbank es un dispositivo portátil que nos permite almacenar energía eléctrica tal como lo hace una pila. La ventaja de esto es que la tensión que la misma entrega a su salida es 5v lo que la hace perfecta para nuestro proyecto, ya que todas las tensiones de alimentación de nuestros dispositivos es la tal mencionada. La misma proporciona una vida útil de aproximadamente 50 horas. Tomando en cuenta el siguiente cálculo:

Capacidad: 10.000mAh

Consumo del proyecto total: 200mAh



5. WIFI 802.11x

[5] El propósito de este estándar es proveer conectividad inalámbrica, de manera portable, contando con estaciones móviles sin importar un área en particular. También el mismo ofrece cuerpos reguladores los cuales estandarizan una o más bandas de frecuencia para el propósito de la comunicación. En nuestro proyecto vamos a estar utilizando la banda de 2.4 Ghz.

Conectividad. Para poder conectar nuestro microcontrolador con la IoT de Arduino utilizamos un celular con conexión 4G el cual nos provee conectividad en cualquier sitio utilizando la plataforma de “Mobile Hotspot”. El mismo tiene una red predefinida, la cual esta cargada previamente en el ESP.

6. C++

[6] C + + es un lenguaje de propósitos generales orientado a objetos. Fue creado por Bjarne Stroustrup en los laboratorios Bell en el año 1980. C + + es compatible con C, lo cual lo hace apropiado para proyectos que requieran un lenguaje más cercano al bajo nivel. En nuestro caso requerimos del mismo dado que los microcontroladores que utilizamos para nuestro proyecto requieren de este lenguaje y de su compilado final. La tarea de compilación se realiza mediante la interfaz de Arduino IDE citada previamente.

7. GitHub

[7] GitHub es una plataforma que nos permite almacenar código para su colaboración y control de versionado. Esta misma permite que tanto nosotros como otras personas puedan

trabajar juntas en un mismo proyecto. En nuestro caso la utilizamos para poder compartir nuestros códigos e ir trabajando a lo largo del desarrollo del proyecto de una manera organizada.

Objetivos del trabajo, motivación y alcance. El propósito de este proyecto es aplicar los conocimientos adquiridos durante la cursada en un sistema propio. El profesor nos guió durante la cursada para que podamos tanto adquirir los conocimientos, como para aplicarlos. Tal como menciona el nombre de la materia, la motivación principal fue poder realizar un proyecto en donde podríamos utilizar nuestros conocimientos previos en sistemas de información, tales como fueron la programación, y ciertos conceptos de electrónica aprendidos en la carrera, con las tecnologías nuevas aprendidas durante el cuatrimestre, los cual nos llevó a pensar este proyecto en conjunto. Agregando a esto, el proyecto abarca dentro de su alcance las tecnologías de plataformas IoT actuales (utilizando Arduino IoT), hardware ESP8266 y sensores de diferentes tipos que nos permiten tomar parametros en tiempo real y realizar diferentes tareas, como en nuestro caso es el monitoreo mediante un dashboard interactivo que puede ser visualizado desde un celular o un PC.

Conclusión. Este proyecto nos sirvió para poder demostrar cómo se realiza la comunicación y la utilización de una plataforma cloud de IoT con microcontroladores y así poder manejar o controlar dispositivos a nuestro deseo y/o requisito. En el mismo aplicamos diversas técnicas, tecnologías y plataformas que fuimos adquiriendo mediante la cursada a lo largo del cuatrimestre. La idea general era demostrar la conectividad de ambas plataformas mencionadas, utilizando el ESP8266 aplicando las tecnologías actuales de administración de hardware.

Bibliografía, enlaces y referencias.

[1] https://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf,

Espressif Systems. DataSheet, 2022. Espressif IoT Team.

[2] <https://docs.arduino.cc/arduino-cloud/>,

Arduino, 2022. Arduino Dev Team.

[3] <https://store.arduino.cc/products/arduino-nano>,

Arduino, 2021 Arduino S.R.L.

[5] <https://www.ieee802.org/11/>,

Institute of Electrical and Electronics Engineers, Inc. (IEEE), 26 de Febrero, 2021.

[6] https://sceweb.sce.uhcl.edu/helm/WEBPAGE-Cpp/my_files/Quick%20Guide/cpp_overview,

Barbara Thompson, SCE, 19 de Julio, 2022.

[7] https://www.w3schools.com/whatis/whatis_github.asp,

w3schools, Refsnes Data, 2022.

Repositorios de los proyectos.

<https://github.com/poxx1/UAI-Hardware-ProyectoFinal>

Proyecto final de hardware. Licencia libre para su uso. 2022.

<https://github.com/Picaio/osciloscopio>

Osciloscopio. Licencia libre para su uso. 2022.

<https://github.com/poxx1/ArduiunoNano-PolifonicRhytm>

Generador de señales. Licencia libre para su uso. 2022.