Erick Daniel Hernández Tó, Aníbal Roberto Gómez Morales, Cesar Estuardo Tejaxun Xunic

ESTACION DE DESINFECCION 3 PASOS (ED3)

(13 noviembre 2020)

*Resumen— Se construyó una estación de desinfeccion inteligente para personas y sus pertenencias el cual su objetivo es evitar el ingresar a una habitación u otro lugar, virus y bacterias dañinas para el ser, de este modo minimizar la propagación del virus SARS COV. 2, que desarrolla la enfermedad de Covid19, se logró hacer la estación de desinfección en tres pasos que ayuda al usuario a utilizarlo con guías de voz y también teniendo el modo de desactivar el sonido para las personas que ya pueden utilizarlo y así minimizar el tiempo en que se tardan en ingresar al recinto deseado y con la ayuda de la tecnología IOT cada usuario puede ingresar con su número de usuario y desinfección por alcohol y uso de luz ultravioleta tipo C, los dispositivos Arduino, módulos de conexión a Internet y el uso de una aplicación para teléfono con el cual se pueden ver el historial de las personas que han ingresado y que temperatura tuvieron al ingresar al hacerse la prueba, esto indica que efectivamente se puede usar un módulo de este tipo para poder evitar la propagación del virus por medio de las tecnologías de IOT utilizadas.*

*Palabras clave— (Application programming interfaces) (Cleaning) (Electronic circuits) (Internet of Things) (Mobile applications) (Motors) (Open source hardware) (Sensor systems and applications) (Wireless networks)*

# Nomenclatura

Arduino, GPIO, Motor Lineal, PWM, ThingSpeak, TTL, React Native, UV.

# Introducción

Este documento proporciona la información de la fabricación, ejecución y funcionamiento de un buzón inteligente que su finalidad es evitar el contacto directo entre personas que dejan paquetes y las personas que reciben los paquetes, se utilizaron tecnologías en open source hardware como arduino unido a ello la tecnología iot necesaria para la recolección de datos por medio de un servicio de api de thingspeak para mostrar posteriormente la información en la aplicación desarrollada en react native, usando notificaciones para que el usuario pueda ver el detalle de cada paquete entregado y de este modo saber si hay un paquete en su buzón, su masa, si este está desinfectado o no y también el estado de la cantidad de líquido desinfectante del depósito en el cual se encuentra almacenado al verificar su estado con el uso de sensores por medio de lectura de valores que generan cada uno de ellos. Desarrollo del artículo

**Antecedentes:** La electrónica digital ha facilitado de manera enorme el uso de herramientas prediseñadas o programables para el desarrollo de plataformas en proyectos con diferentes funcionalidades que requieren de un usuario. El uso de electrónica ha evolucionado de forma exponencial en la cual se puede evidenciar la evolución tecnológica a través de los últimos años, con la cual se puede realizar infinidad de aplicaciones que involucren sistemas electrónicos de gran capacidad de programabilidad y también bajo consumo de energía eléctrica e incluso bajo costo de las piezas pre-ensambladas que sean de bajo costo en mantenimiento.

**Estado Actual**: En base a lo anterior expuesto surge la tecnología de electrónica programable/no programable Open Source Hardware, en la cual esta se puede utilizar en infinidad de proyectos a nivel industrial, nivel doméstico e incluso personal. En esta categoría de Hardware se puede encontrar varias plataformas electrónicas programables que se pueden utilizar para desarrollar gran cantidad de proyectos e ideas de innovación, en las cuales se encuentra Arduino la cual tiene a disposición de cualquier usuario el diseño de la mayor parte de sus circuitos electrónicos programables que pueden ser desarrollados por los usuarios incluso modificarlos para satisfacer su necesidad de aplicación.

**Arduino Mega 2560[2]:** Es una placa electrónica que se puede programar por medio de un puerto USB tipo B, el cual tiene un microcontrolador ATmega2560 de 100 pines de conexión, el cual puede soportar hasta 256KB de memoria programable, 4k de memoria EEPROM y 8KB en memoria interna de SRAM, puede soportar hasta 10000 procesos de escritura/borrar memoria y funciona con un reloj de cristal externo de hasta 16MHz con el cual puede procesar 135 instrucciones por ciclo de reloj con un voltaje de operación de entre 4.5volts y 5volts.

La cantidad de pines utilizables se puede ver en la siguiente tabla comparativa con otros modelos del ATmega2560.

Tabla 1: Resumen de Comparación

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Disp. AT-Mega | Flash (KB) | EEPROM (KB) | RAM (KB) | I/O pins | 16bits pwm pin | Serial pin | ADC pin |
| 640 | 64 | 4 | 8 | 86 | 12 | 4 | 16 |
| 1280 | 128 | 4 | 8 | 86 | 12 | 4 | 16 |
| 1281 | 128 | 4 | 8 | 54 | 6 | 2 | 8 |
| 2560 | 256 | 4 | 8 | 86 | 12 | 4 | 16 |
| 2561 | 256 | 4 | 8 | 54 | 6 | 2 | 8 |

**Sensor de Proximidad HCSR04-05[3],[4]:** Es una placa electrónica que contiene un emisor y un receptor de pulsos ultrasónicos que pueden medir un tiempo desde su emisión hasta su recepción y así medir una distancia por medio de la velocidad del sonido. La ecuación sugerida para determinar la medida en cm es la siguiente:

Ecuacion 1: distancia en cm medida por un sensor HCSR04



Donde d es la Distancia medida en cm, t el tiempo que tarda en regresar el eco del sonido emitido y Vs es la velocidad del sonido (340 m/s).

Sus especificaciones de uso para el correcto funcionamiento son las siguientes:

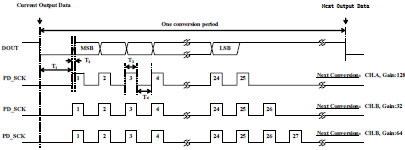
Tabla 2. Especificaciones de funcionamiento normal HCRS04

|  |  |
| --- | --- |
| Descripción | Valor |
| Voltaje de funcionamiento | 4.5 a 5.5 volts |
| Corriente de funcionamiento | 15 mA |
| Rango Máximo | 4 m |
| Rango Mínimo | 2 cm |
| Angulo de Medición | 15° |
| Señal de Entrada Trigger | 10 us TTL |
| Señal de salida Echo | TTL señal Proporcional |
| Dimensiones | 40\*20\*15 mm |
| Medición en Superficie | Rígida |

**Módulo WIRELESS ESP8266-01[5]**: Módulo que es utilizado para conexiones inalámbricas a redes de computadoras donde se provee de servicio de internet, utiliza comandos AT y tiene el microcontrolador ESP8266 que tiene la posibilidad de funcionar como un dispositivo individual para el control de dispositivos, cuenta con entradas y salidas GPIO (General purpose in/out), pero en este modelo solo cuenta con salida visible de dos conexiones GPIO siendo estas GPIO1 y GPIO2, su comunicación se basa en serial UART con otros microcontroladores, tiene soporte para generar una red propia y funcionamiento como estación de trabajo, funcionamiento de bajo consumo con un voltaje nominal de funcionamiento de 3.3volts con un máximo de hasta 3.6volts, soporta señales TTL para comunicación exterior TX/RX pero es muy sensible al ruido electrónico, su reloj interno funciona a 80 MHz, soporte para conexión a redes con seguridad WPA, WPA2, encriptación de envio y recepción por Tkip, AES, WEP y soporta protocolos de red TCIPv4, UDP, FTP, HTTP.

**Módulo HX711[6]:** Es un circuito integrado que permite convertir valores análogos a datos binarios o digitales (ADC) y es utilizado en aplicaciones de control industrial para conversión de escalas en dispositivos que proporcionan datos con valores análogos, su comunicación es serial Data/clock y su precisión es de 24 bits, su voltaje nominal de trabajo es de 5 volts y su ciclo y medición depende a la frecuencia de clock utilizado en su pin sck como se muestra en esta gráfica.

Ilustración 1: Tiempos de respuesta, ganacia y control de tiempo

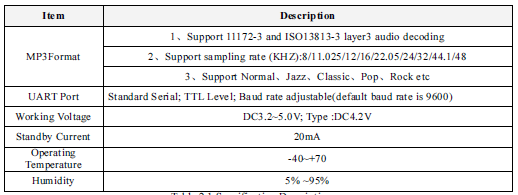


**Módulo de Relé:** Circuito electrónico que se utiliza para activar componentes eléctricos de más capacidad de voltaje o amperaje que puede proporcionar un micro controlador, puede tener configuración NO (normalmente abierto) o NC (normalmente cerrado) y se activa por pulso bajo también llamado 0 lógico, para seguridad de su funcionamiento se utilizan optoacopladores para activarlos y diodos entre sus bobinas para evitar el ruido tras su activación o desactivación.

**Bomba para Agua**: Es un dispositivo capaz de bombear líquido por medio del uso de un motor DC a 12 Volts con una velocidad normal de trabajo a 2 o 3 litros por minuto consumiendo 300 mA de corriente. Normalmente para proyectos IOT se necesita de un módulo de Relé para poder activarlo.

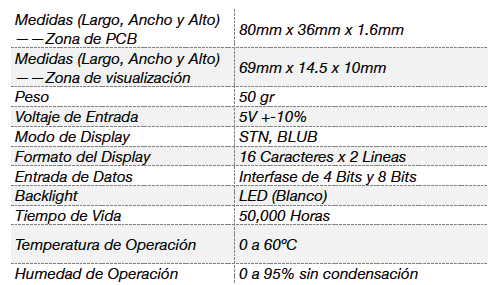
**Módulo DFPlayer:** es un dispositivo capaz de reproducir archivos de audio en formato MP3 y Wav el cual se comunica por medio de una interfaz serial, soporta particiones fat32 y fat16 hasta 32 GB, además tiene un amplificador interno de 3 Watts y 30 niveles de volumen. En la siguiente tabla se muestra las características del módulo.

Tabla 3: Características de módulo DFPlayer



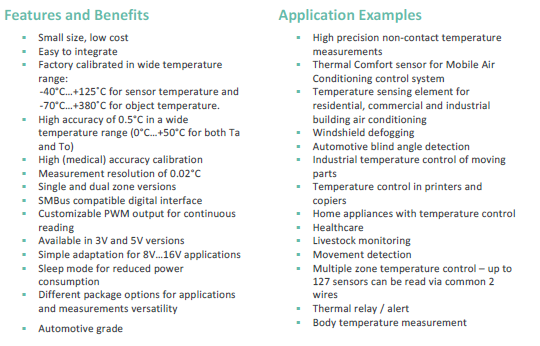
**Pantalla 16x2 [9]:** Es un display de cristal líquido de tipo alfanumérico y retroiluminación incorporado capaz de mostrar mensajes y ser controlado desde un dispositivo Arduino.

Tabla 3: Características de Pantalla 16x2



**Sensor de Temperatura MLX90614:** Es un sensor que permite medir la temperatura a una distancia máxima de 3cm de su emisor, si se desea puede agregarse un lente que permite aumentar el rango de distancia para realizar la medición. Su transmisión de datos es por medio de I2C por lo cual puede agregarse a canal i2c con otros dispositivos en paralelo.

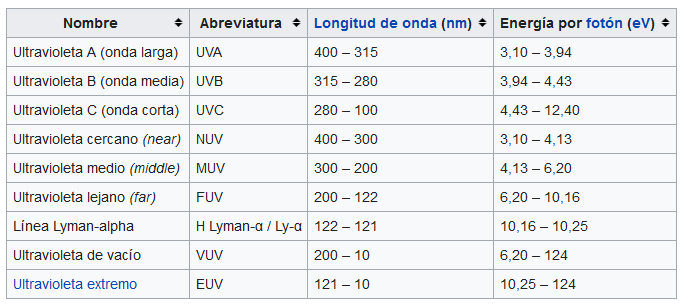
Ilustración 2: Características del módulo MLX90614 y ejemplos de aplicación (en inglés).



**Solenoide:** Es un actuador ser activado por corriente eléctrica por medio de su bobina interna atraer un metal por medio de la creación de un campo electromagnético y así abrir una puerta.

**Luz UV (LED)[11]:** Es una forma no invasiva de poder desinfectar objetos de bacterias, virus y demás elementos nocivos a la salud de los seres vivos, por lo cual es peligrosa en un rango de onda. Para este caso se utilizó luz UVA de 490 nm la cual es de menos potencia, pero se puede cambiar por una luz de tipo UVC pero tiene que tener un recipiente obscuro cerrado.

Tabla 4: Subtipos de Luz UV.



**Capa IOT Infraestructura:**

Tabla 4. Recursos usados en el producto (N.A. significa No aplica)

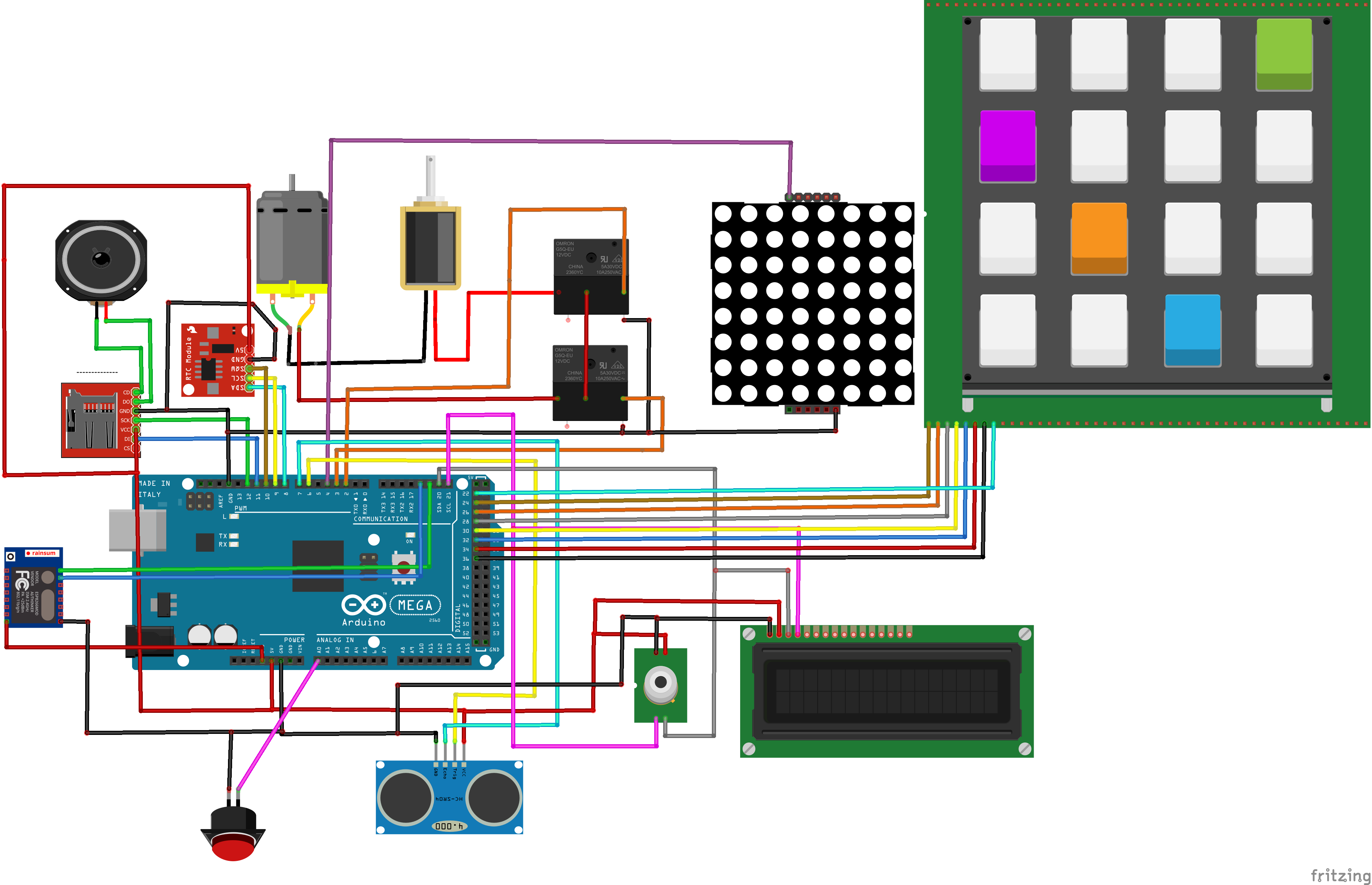
|  |  |  |
| --- | --- | --- |
| Cant. | Recurso | Medición |
| 40 | Cable de conexión | N.A. |
| 4 | Bases de madera Simulación de habitación | N.A. |
| 4 | Reglas base para caja de madera | N.A. |
| 1 | Arduino Mega 2560 | N.A. |
| 1 | Sensor HCSR04 | Distancia Cm |
| 1 | Wifi ESP8266 | N.A. |
| 1 | Pantalla 16X2 | N.A. |
| 1 | Módulo FPlayer | N.A. |
| 60 | Led UV | N.A. |
| 30 | Resistencias | N.A. |
| 1 | Bomba de Agua 12V | N.A. |
| 1 | Actuador Solenoide 12V | N.A. |
| 1 | Sensor de Temperatura MLX9006 | Temperatura °C |
| 1 | Switch para estado de Puerta | Estado de puerta |
| 1 | Solenoide | N.A. |
| 2 | Fuentes de poder 9 V y 12 V | N.A. |

**Capa IOT Conectividad:**

Tabla 4. Recursos de conectividad usados

|  |  |
| --- | --- |
| Recurso | Protocolo |
| Internet | Http |
| Network /WIFI | tcp |

## Ilustración 2: Diagrama de Conexión:



## **ThinkSpeak**

Es un servicio de plataforma de análisis de IoT que le permite agregar, visualizar y analizar flujos de datos en vivo en la nube. Proporciona visualizaciones instantáneas de los datos publicados por sus dispositivos en ThingSpeak. Con la capacidad de ejecutar código MATLAB, puede realizar análisis y procesamiento en línea de los datos a medida que ingresan. Se usa a menudo para la creación de prototipos y pruebas de sistemas de IoT de concepto que requieren análisis

Es de [código abierto](https://en.wikipedia.org/wiki/Open-source_software) para almacenar y recuperar datos de cosas mediante el [protocolo HTTP](https://en.wikipedia.org/wiki/HTTP) y MQTT a través de Internet o mediante una red de área local. Permite la creación de aplicaciones de registro, aplicaciones de rastreo de ubicación y una red social de cosas con actualizaciones de estado. Tiene soporte integrado del software de computación numérica [MATLAB](https://en.wikipedia.org/wiki/MATLAB) de [MathWorks](https://en.wikipedia.org/wiki/MathWorks) , permitiendo a los usuarios de ThingSpeak analizar y visualizar los datos cargados usando Matlab sin requerir la compra de una licencia de Matlab de Mathworks. Puede integrarse con Arduino, Raspberry Pi, ioBridge / RealTime.io, Electric Imp, aplicaciones móviles/Web, redes sociales y análisis de datos con MATLAB.

Permite agregar, visualizar y analizar flujos de datos en directo en la nube. Algunas de las capacidades clave de ThingSpeak incluyen la capacidad de:

* Configurar fácilmente los dispositivos para que envíen datos a ThingSpeak utilizando los protocolos de IO más populares.
* Visualizar los datos de los sensores en tiempo real. Agregar datos bajo demanda de fuentes de terceros.
* Utilizar la potencia de MATLAB para dar sentido a tus datos de IO.
* API abierta
* Recolección de datos en tiempo real
* Datos de geolocalización
* Procesamiento de datos
* Mensajes de estado del circuito
* Plugins

**Canal ThingSpeak**

El elemento central de ThingSpeak es un canal que almacena los datos que enviamos a ThingSpeak y se compone de los siguientes elementos:

* 8 campos para almacenar datos de cualquier tipo. Estos pueden ser usados para almacenar los datos de un sensor o de un dispositivo integrado.
* 3 campos de ubicación. Se pueden utilizar para almacenar la latitud, la longitud y la elevación. Estos son muy útiles para rastrear un dispositivo en movimiento.
* 1 campo de estado. Un mensaje corto para describir los datos almacenados en el canal. Para usar ThingSpeak, necesitamos registrarnos y crear un canal.

Una vez que tenemos un canal, podemos enviar los datos, permitir que ThingSpeak los procese y también recuperarlos.

**Primeros pasos en ThingSpeak usando la plataforma Arduino**

**Paso 1: Crea tu cuenta**. Para poder cargar datos en ThingSpeak para su análisis y procesamiento, debes crear una cuenta.

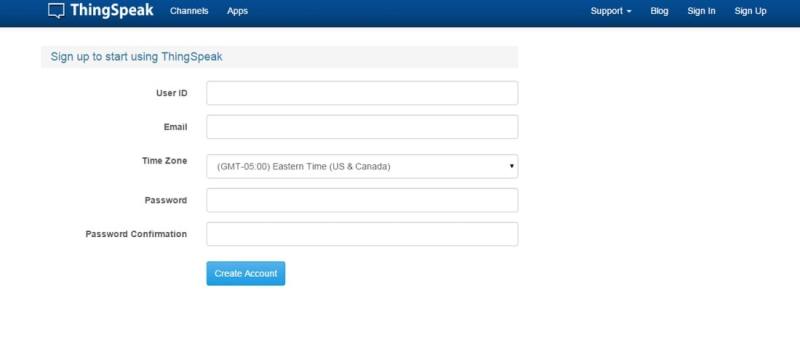


Imagen no. 2

Registro de usuario en ThinkSpeak.

**Paso 2: Crear un nuevo canal**.

Se introduce los detalles del nuevo canal creado. Por ejemplo, puedes utilizar el nombre del proyecto en el que está trabajando como nombre del canal. En caso de duda, puedes dejar algunas opciones en blanco (descripción, metadatos y etiquetas). Los campos reflejan los datos que está a punto de descargar

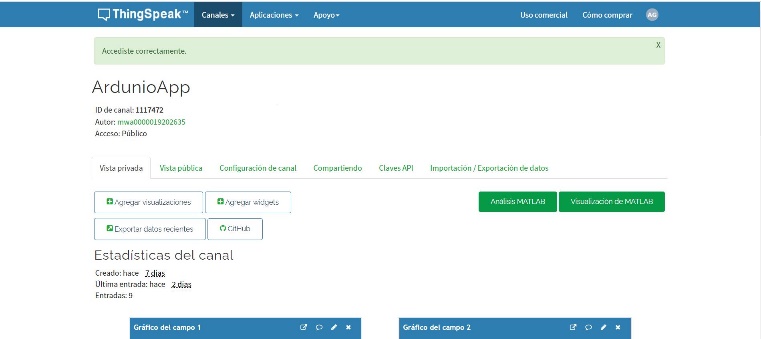


Imagen no. 3

Creación de un Canal con ThingSpeak.

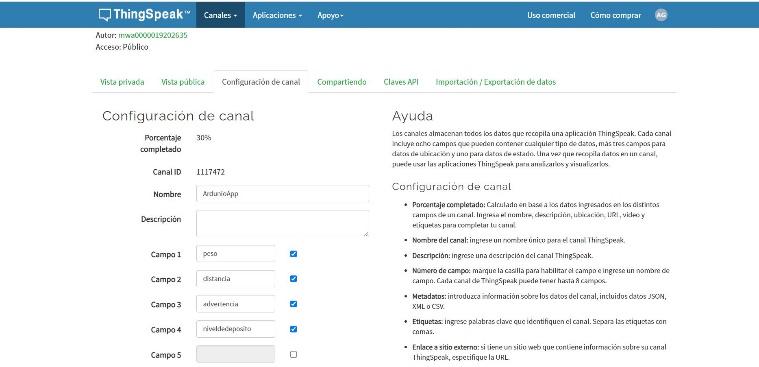


Imagen no. 4

Configuración de un Canal

**Paso 3: Servicios REST**

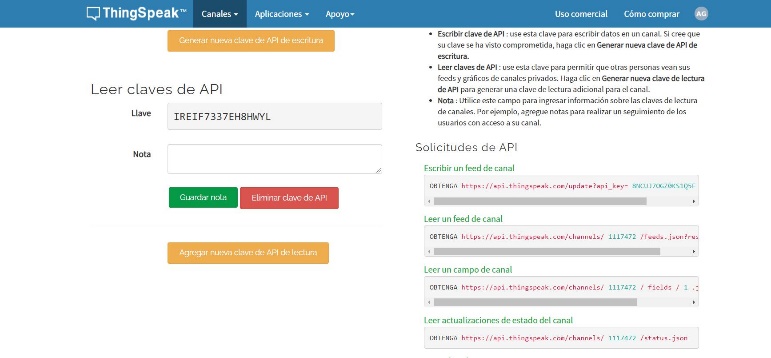


Imagen no. 5

Servicios de un Canal

**MATLAB ANALISIS**

Las app de MATHLAB permite realizar tareas iterativas de forma interactiva, como entrenar modelos de machine learning o etiquetar datos. Estas app generan el código de Matlab necesario para reproducir de forma programática el trabajo realizado de forma iteractiva, permitiendo programar acciones para entregar información a travez de los canales según estén en cuestión.

Cuando se tiene una secuencia de comandos que se debe ejecutar de forma repetida o que se desea guardar para referencia futura, estos comandos se deben almacenar en un archivo de programa. El tipo mas simple de programa de Matlab es un script, que contiene un conjunto de comandos que están escritos exactamente como deberían introducirse en la línea de comandos. Para una mayor flexibilidad de programación, permite crear funciones que acepten entradas y devuelva salidas. Cuando tenga estructuras de datos especializados o necesite muchas funciones para interactuar con tipos de datos especiales, se crean clases mediante técnicas de programación orientada a objetos.

Matlab permite evaluar expresiones o funciones de forma indirecta, establecer temporizadores y gestionar excepciones.

Los Canales Implementados son:

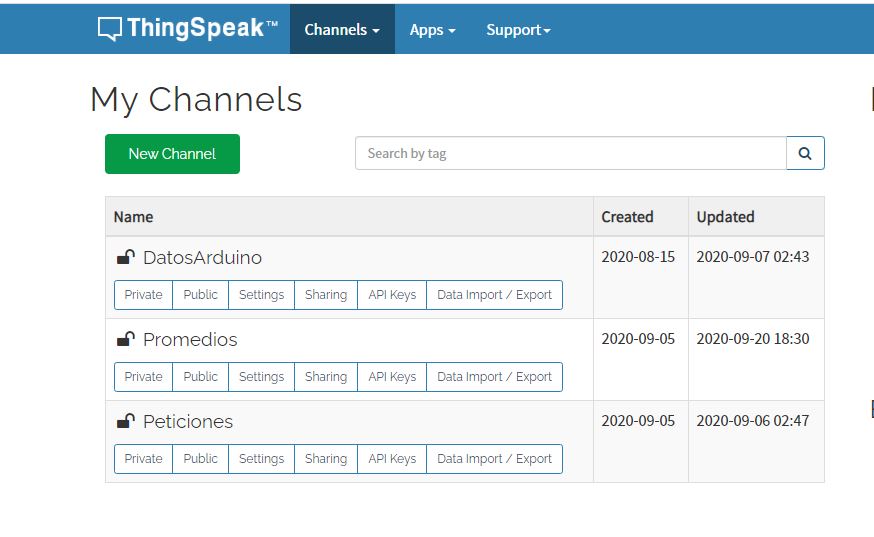


Imagen no. 7

Canales en Cuestión

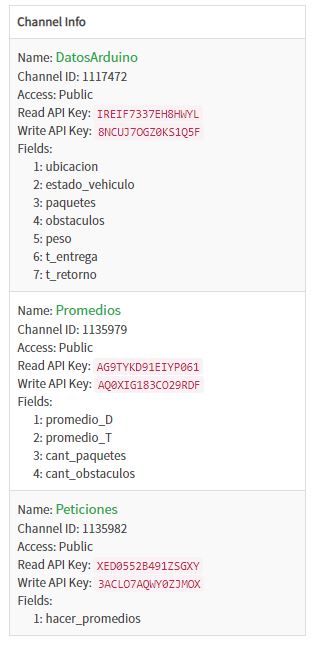


Imagen no. 8

Información de los canales

**Codigo en Canal Promedios**

**//**Ids del canal ArduinoApp para lectura de datos.

readChannelID = 1117472;

writeChannelID = 1135979;

writeAPIKey = 'AQ0XIG183CO29RDF';

//Variable que alamacena la suma de paquetes. El 9997200 hace alusión al tiempo que se determinan los datos a procesar.

sumapaquetes = thingSpeakRead(readChannelID,'Fields',3,'Numminutes',9997200);

//Variable que alamacena la suma de obstaculos.

sumaObstaculos = thingSpeakRead(readChannelID,'Fields',4,'Numminutes',9997200);

indicePaquetes=length(sumapaquetes);

conteoPaquetes =0;

sumadorpaquetes=0;

for i=1:indicePaquetes

if sumapaquetes(i)<0

disp('There is at least one value null');

else

disp('element is OK.');

disp(sumapaquetes(i));

conteoPaquetes=conteoPaquetes + 1;

sumadorpaquetes=sumadorpaquetes + sumapaquetes(i);

disp('valor del sumador:');

disp(sumadorpaquetes);

end

disp(i);

end

promedioPaquetes=sumadorpaquetes;

disp(promedioPaquetes);

indiceObstaculos=length(sumaObstaculos);

conteoObstaculos =0;

sumadorObstaculos=0;

for i=1:indiceObstaculos

if sumaObstaculos(i)<0

disp('There is at least one value null');

else

disp('element is OK.');

disp(sumaObstaculos(i));

conteoObstaculos=conteoObstaculos + 1;

sumadorObstaculos=sumadorObstaculos + sumaObstaculos(i);

disp(sumadorObstaculos);

end

disp(i);

end

promedioObstaculos=sumadorObstaculos;

disp('El promedio es:');

disp(promedioObstaculos);

// Arreglo que almacena las cantidades y las promedia

array1 = thingSpeakRead(readChannelID,'Fields',5,'Numminutes',9997200);

array2 = thingSpeakRead(readChannelID,'Fields',6,'Numminutes',9997200);

array3 = thingSpeakRead(readChannelID,'Fields',7,'Numminutes',9997200);

indice=length(array1);

conteo1 =0;

sumador1=0;

for i=1:indice

if array1(i)<0

else

disp('element is OK.');

disp(array1(i));

conteo1=conteo1 + 1;

sumador1=sumador1 + array1(i);

disp('valor del sumador:');

disp(sumador1);

end

disp(i);

end

promedio1=0;

if conteo1~=0

promedio1=sumador1/conteo1;

end

disp(promedio1);

indice2=length(array2);

conteo2 =0;

sumador2=0;

for j=1:indice2

if array1(j)<0

disp('There is at least one value null');

else

disp('element is OK.');

disp(array2(j));

conteo2=conteo2 + 1;

sumador2=sumador2 + array1(j);

disp('valor del sumador:');

disp(sumador2);

end

disp(j);

end

promedio2=0;

if conteo2~=0

promedio2=sumador2/conteo2;

end

disp('El promedio es:');

disp(promedio2);

tStamp = datetime('now')

thingSpeakWrite(writeChannelID,[promedio1,promedio2,promedioPaquetes,promedioObstaculos],'Fields',[1,2,3,4],'TimeStamps',tStamp,'Writekey',writeAPIKey);



Imagen no. 9

Esquema General de la funcionalidad.

# Boceto

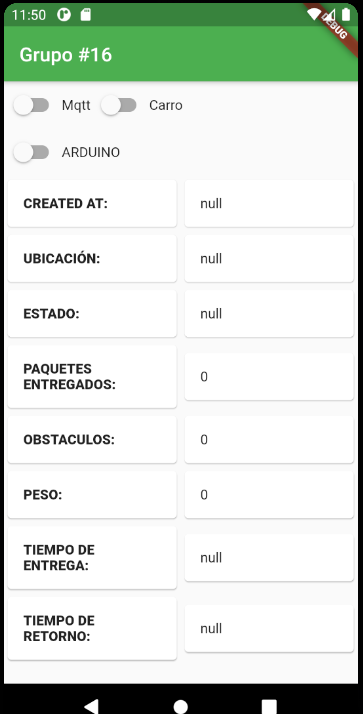
Figura 6.



# Pantallas de aplicación móvil

No data. El app está en espera de recibir datos.

Figura 7



Recepción del mensaje. Se obtuvo información del estado del api que monitorea el buzón.

Figura 8



Se muestra la fecha correspondiente a la última actualización

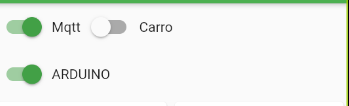
Figura 9



La app consiste de 2 partes la cuales son:

* Control de conectividad en el servicio mqtt y el despositivo arduino.
* Switch Mqtt.
* Switch carro. Encargado para mandar mensaje al arduino para apagar o encerder carro.
* Switch arduino

Figura 10



* Estadísticas del los estados del carro; como:
* Ubicación
* Estado
* Paquetes entregados
* Obstáculos
* Peso paquete
* Tiempo de entrega
* Tiempo de retorno

Figura 11



**Alertas y notificaciones**

La app cuenta con una serie de notificaciones y alertas, las cuales son las siguientes:

El carro empieza, recibe paquete y empieza moverse hacia al punto de entrega

Figura 13

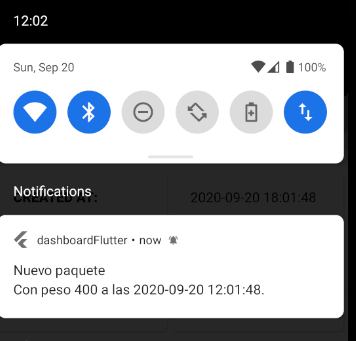
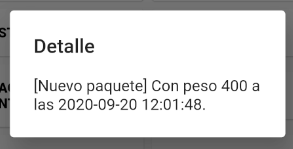


Figura 14



Paquete entregado

Figura 15

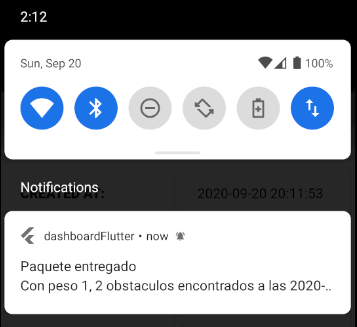
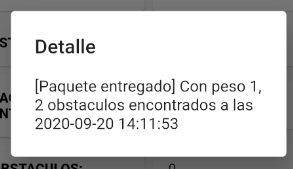


Figura 16



Carro retorno al punto de inicio

Figura 17

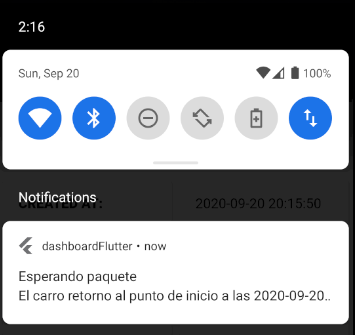
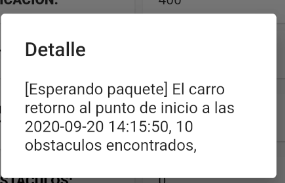


Figura 18



**Capa IoT - Smart Apps:**

Funcionalidad

* Visualización de la información procesada mostrada de una forma fácil de digerir.
* Alertas en los siguientes casos.

Cuando se recibe un paquete, y esta en camino al punto de entrega

Cuando llega al punto de entrega.

Cuando retorna al punto inicio

# Conclusiones

El uso de tecnologías programables amplia las formas de aplicaciones en la vida real

La tecnología IOT amplía las aplicaciones de sistemas inteligentes

Los sensores son una forma de capturar datos del entorno donde se aplica la tecnología IOT.

# Apéndice: Unidades del sistema MKSA

TABLA 6

Magnitudes y unidades principales del sistema MKSA. Nombres de algunos componentes causantes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Magnitud física** | | **Unidad MKSA**  (entre paréntesis, unidades usuales de otros sistemas y equivalencia) | | **Compo-nente**  **causante** |
| **Nombre** (entre paréntesis otros nombres usuales) | **Sím-**  **bolo** | **Nombre** | **Sím-**  **bolo** |  |
| **Unidades fundamentales MKSA** | | | | |
| Longitud | *l* | metro | m |  |
| Masa | *m* | kilogramo | kg |  |
| Tiempo | *t* | segundo | s |  |
| Intensidad eléctrica (corriente) | *I*, *i* | ampere | A |  |
| Cantidad de materia | *n* | mol | mol |  |
| **Unidades auxiliares MKSA** | | | | |
| Intensidad luminosa | *I* | candela | cd |  |
| Temperatura | *T* | Kelvin | K |  |
| **Unidades derivadas MKSA** | | | | |
| Frecuencia | *f* | hertz | Hz |  |
| Fuerza | *F* | newton | N |  |
| Presión (tensión) | *p* | pascal | Pa |  |
| Energía (trabajo) | *T, W* | joule | J |  |
| Potencia | *p* | watt | W |  |

TABLA II (contiuación)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cantidad de electricidad  (carga eléctrica) | | *q* | culomb | C |  |
| Densidad de corriente | | *j* | ampere/metro2 | A/m2 |  |
| Tensión eléctrica  (potencial eléctrico,  fuerza electromotriz) | | *u* | volt |  |  |
| Campo eléctrico | | *ε* | volt/metro | V/m |  |
| Resistencia eléctrica | | *R* | ohm | Ω | resistencia  resistor |
| Conductancia | | *G* | siemens | S |  |
| Inductancia | | *L* | henry | H | bobina  inductor |
| Capacidad eléctrica | | *C* | farad | F | condensador  capacitor |
| Campo magnetizante  (magnetización) | | *H* | ampere/metro | A/m |  |
| **Campo magnético**  **(inducción magnética,**  **densidad de flujo magnético)** | | ***B*** | **tesla**  **(gauss=10-4 T)** | **T**  **(G)** |  |
| **Flujo magnético**  **(flujo de inducción magnética)** | | ****** | **weber** | **Wb** |  |
| **Luminancia** | | **L** | **lambert** | **L** |  |
| **Flujo luminoso** | | ***Φ*** | **lumen** | **lm** |  |
| **Iluminación (iluminancia)** | | **E** | **lux** | **lx** |  |
| **Ángulo plano** | **α, β, γ, θ** | | **radián** | **rad** |  |
| **Ángulo sólido** | **α, β, γ** | | **estereorradián** | **sr** |  |

# **Referencias**

***Periodicals (Artículos de revista):***

1. **Miguel Angel Rodríguez Pozueta, “Puente de Wheatstone”, 23./ago. 2020, disponible en https://personales.unican.es/rodrigma/PDFs/Puente%20de%20Wheatstone.pdf.**

***Books (Libros):***

***Technical Reports (Informes técnicos):***

1. Datasheet AT mega 2560 microcontroller microchip disponible en http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561\_datasheet.pdf
2. Datasheet HCSR04 disponible en https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf
3. User manual HCSR04 disponible en https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL\_pfa39RsB-x2qR4vP8saG73rE/edit#!:
4. Datasheet Wireless ESP8266-01 disponible en [**https://components101.com/sites/default/files/component\_datasheet/ESP8266%20Datasheet.pdf**](https://components101.com/sites/default/files/component_datasheet/ESP8266%20Datasheet.pdf).
5. Datasheet HX711 disponible en https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711\_english.pdf
6. https://www.st.com/resource/en/datasheet/l298.pdf
7. https://www.vishay.com/docs/83760/tcrt5000.pdf
8. https://wassermatic.com.mx/assets/img/productos/8be10-lcd-16x2.pdf
9. https://www.melexis.com/en/documents/documentation/datasheets/datasheet-mlx90614
10. https://es.wikipedia.org/wiki/Radiaci%C3%B3n\_ultravioleta

***Autores:***

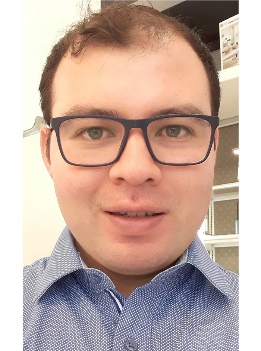
**Repositorio GitHub:**

**https://github.com/nibalmorales/ACE2\_2S20\_G-16**

**Video del Proyecto:**

**https://www.youtube.com/watch?v=rzHBjSjwejU**

**Erick Daniel Hernández Tó Carnet: 201114438**

**Anibal Robeto Gómez Morales** Carnet: 201025198



**Cesar Estuardo Tejaxun Xunic**

**200819197**