

Proyecto 1 Buzón Inteligente

(20 Septiembre 2020)

Resumen— Se construyó un vehículo de entregas inteligente para paquetería el cual su objetivo es evitar el contacto directo entre personas que dejan paquetes y los clientes de dichas personas y de este

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

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

modo minimizar la propagación del virus SARS COV. 2, que desarrolla la enfermedad de Covid19, se logró hacer el vehículo que sigue una línea negra que le indica el camino a donde debe llevar el paquete y con la ayuda de la tecnología IOT, los dispositivos Arduino, módulos de conexión a Internet y el uso de una aplicación para teléfono que da un aviso al usuario al momento que se coloca un paquete, si hay obstáculos cuando va el paquete en camino y también cuando llega al punto de entrega, así mismo también cuando regresa al buzón, esto indica que efectivamente se puede usar un módulo de este tipo para poder evitar el contacto entre personas 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)

I. NOMENCLATURA

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

II. 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

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.	Flash	EEP	RAM	I/O	16bits	Seri	AD
AT-	(KB)	ROM	(KB)	pins	pwm	al	C
Mega		(KB)			pin	pin	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:

Ecuación 1: distancia en cm medida por un sensor HCSR04

$$d = t * \frac{V_s}{2}$$

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:

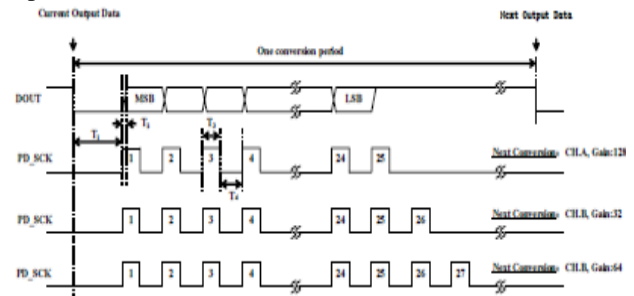
Tabla 2. Especificaciones de funcionamiento normal HCRS04

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 envío 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, ganancia y control de tiempo



Celda de Carga 5Kg: Es una barra metálica normalmente de aluminio que soportan fuerza de tensión, presión y flexión que contienen normalmente cuatro dispositivos de sensor extensiométrico capaz de medir la deformación hecha por una carga sobre el aluminio y traducirlo a un nivel eléctrico proporcional a la fuerza ejercida en la barra de aluminio que produce la deformación. Los sensores extensiométricos están en un arreglo llamado Puente Wheatstone [1], tiene un error de precisión con 0.02% correctamente calibrado.

Módulo de Puente H L298N [7]: También conocido como Dual full-Bridge Driver, Este módulo proporciona una facilidad para poder controlar la dirección de rotaje de los motores DC y también a la velocidad con la cual puede dar vuelta el rotor de un motor por medio de pulsos PWM. Según las especificaciones técnicas en la hoja de datos [7], soporta pulsos TTL, puede soportar hasta una conmutación de corriente hasta los 2 A y voltaje para motores de hasta 30 Volts. Como se aprecia en la siguiente tabla.

Tabla 4. Propiedades de Modulo L298N (N.A. significa No aplica)

Descripción	Valor
Voltaje de entrada	Hasta 50 volts Max.
Entrada Voltaje Lógica	Max 7 volts
Corriente de Puenteo	2 A en DC hasta 3 A
Temperatura Máxima de trabajo	75 °C aprox. 25 Watts
Censado de Voltaje	-1 a 2.3 volts

Modulo Sensor Infrarrojo TCRT5000 [8]: También llamado “Reflective Optical Sensor with Transistor Output”, permite medir la cantidad de luz reflejada por una superficie, utiliza luz infrarroja para su objetivo, utiliza un LED infrarrojo como emisor y un fototransistor como receptor para medir que tanta luz ha sido reflejada por una superficie perpendicular al sensor, puede medir de manera eficiente desde 0.2 mm hasta los 15 mm de distancia.

Tabla 4. Características básicas del sensor infrarrojo TCRT5000

BASIC CHARACTERISTICS ⁽¹⁾						
PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT
INPUT (EMITTER)						
Forward voltage	$I_F = 60 \text{ mA}$	V_F		1.25	1.5	V
Junction capacitance	$V_{BI} = 0 \text{ V}, f = 1 \text{ MHz}$	C_j		17		pF
Radiant intensity	$I_F = 60 \text{ mA}, t_p = 20 \text{ ms}$	I_e			21	mW/sr
Peak wavelength	$I_F = 100 \text{ mA}$	λ_p	940			nm
Virtual source diameter	Method: 63 % encircled energy	d		2.1		mm
OUTPUT (DETECTOR)						
Collector emitter voltage	$I_C = 1 \text{ mA}$	V_{CEO}	70			V
Emitter collector voltage	$I_e = 100 \mu\text{A}$	V_{ECO}	7			V
Collector dark current	$V_{CE} = 20 \text{ V}, I_F = 0 \text{ A}, E = 0 \text{ lx}$	I_{CEO}		10	200	nA
SENSOR						
Collector current	$V_{CE} = 5 \text{ V}, I_F = 10 \text{ mA}, D = 12 \text{ mm}$	$I_C \text{ (2) (3)}$	0.5	1	2.1	mA
Collector emitter saturation voltage	$I_F = 10 \text{ mA}, I_C = 0.1 \text{ mA}, D = 12 \text{ mm}$	$V_{CEsat} \text{ (2) (3)}$			0.4	V

Capa IOT Infraestructura:

Tabla 5. Recursos usados en el producto (N.A. significa No aplica)

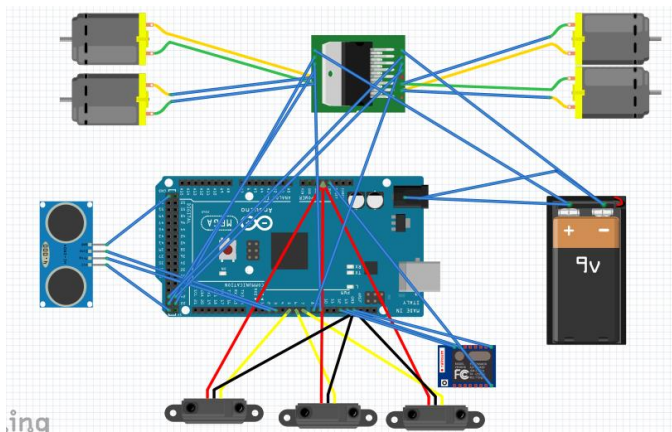
Cant.	Recurso	Medición
35	Cable de conexión	N.A.
2	Bases para vehículo de plástico	N.A.
4	Ruedas con motor DC y caja reductora	N.A.
1	Arduino Mega 2560	N.A.
1	Sensor HCSR04	Distancia
1	Wifi ESP8266	N.A.
1	ADC HX711	Análogo-Digital
1	Celda de carga 5Kg	Masa
3	Sensor Infrarrojo	Línea Negra
1	Puente dual L298N	Estado Puerta
2	Baterías 18650 5 A, a 8 volts.	N.A.
2	Porta Baterías 18650	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 para almacenar y recuperar datos de cosas mediante el protocolo 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 de 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.

Channel Info
Name: DatosArduino Channel ID: 1117472 Access: Public Read API Key: IREIF7337EH8HWYL Write API Key: 8NCUJ70GZ0KS1Q5F Fields: 1: ubicacion 2: estado_vehiculo 3: paquetes 4: obstaculos 5: peso 6: t_entrega 7: t_retorno
Name: Promedios Channel ID: 1135979 Access: Public Read API Key: AG9TYKD91EIYP061 Write API Key: AQ0XIG183C029RDF Fields: 1: promedio_D 2: promedio_T 3: cant_paquetes 4: cant_obstaculos
Name: Peticiones Channel ID: 1135982 Access: Public Read API Key: XED0552B491ZSGXY Write API Key: 3ACLO7AQWY0ZJMOX Fields: 1: hacer_promedios

Imagen no. 8
Información de los canales

Código en Canal Promedios

```
//Ids del canal ArduinoApp para lectura de datos.
readChannelID = 1117472;
writeChannelID = 1135979;
writeAPIKey = 'AQ0XIG183C029RDF';

//Variable que almacena la suma de
paquetes. El 9997200 hace alusión al tiempo
que se determinan los datos a procesar.
sumapaquetes =
thingSpeakRead(readChannelID, 'Fields', 3, 'Nu
mminutes', 9997200);

//Variable que almacena la suma de
obstaculos.
sumaObstaculos =
thingSpeakRead(readChannelID, 'Fields', 4, 'Nu
mminutes', 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, 'Nu
mminutes', 9997200);
array2 =
thingSpeakRead(readChannelID, 'Fields', 6, 'Nu
mminutes', 9997200);
array3 =
thingSpeakRead(readChannelID, 'Fields', 7, 'Nu
mminutes', 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,p
romedio2,promedioPaquetes,promedioObstaculo
s], 'Fields',[1,2,3,4], 'Timestamps',tStamp, '
Writekey',writeAPIKey);

```

Canales Implementados

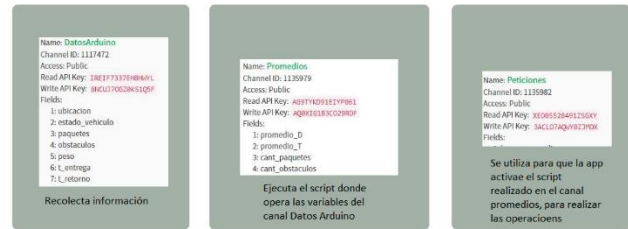


Imagen no. 9
Esquema General de la funcionalidad.

III. BOCETO

FIGURA 6.



IV. PANTALLAS DE APLICACIÓN MÓVIL

NO DATA. EL APP ESTÁ EN ESPERA DE RECIBIR DATOS.

Figura 7



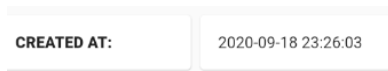
Recepción del mensaje. Se obtuvo información del estado del api que monitorea el buzón.

Figura 8



Se muestra la fecha correspondiente a la última actualización

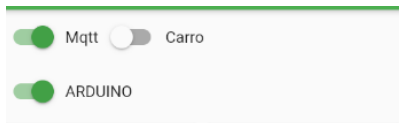
Figura 9



La app consiste de 2 partes la cuales son:

- Control de conectividad en el servicio mqtt y el dispositivo arduino.
 - Switch Mqtt.
 - Switch carro. Encargado para mandar mensaje al arduino para apagar o encender 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

UBICACIÓN:	1
ESTADO:	null
PAQUETES ENTREGADOS:	0
OBSTACULOS:	0
PESO:	null
TIEMPO DE ENTREGA:	null
TIEMPO DE RETORNO:	null

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

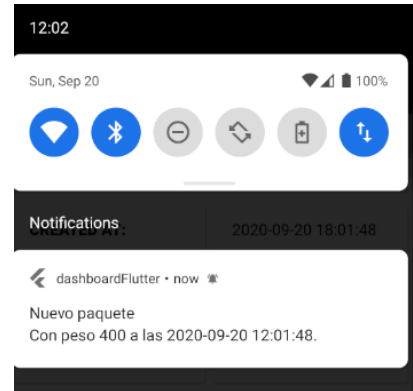
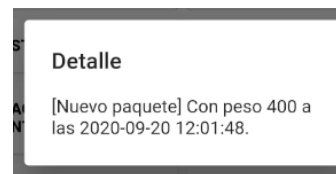


Figura 14



Paquete entregado

Figura 15

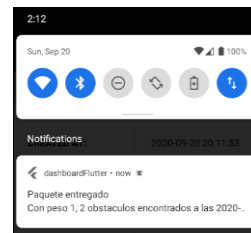
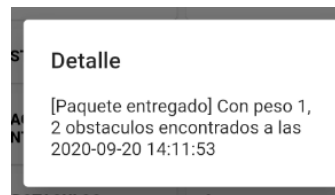


Figura 16



Carro retorno al punto de inicio

Figura 17

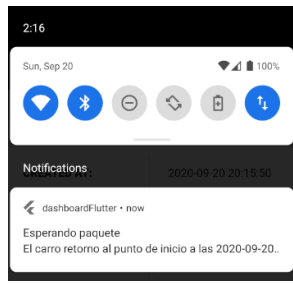
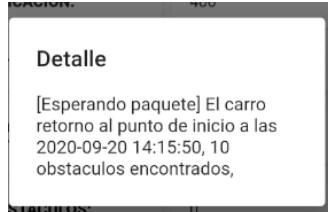


Figura 18



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

V. CONCLUSIONES

El uso de tecnologías programables amplía 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)		Componente causante
Nombre (entre paréntesis otros nombres usuales)	Símbolo	Nombre	Símbolo	
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 (CONTINUACIÓN)

Cantidad de electricidad (carga eléctrica)	q	culomb	C	
Densidad de corriente	j	ampere/metro ²	A/m ²	
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)	ϕ	weber	Wb	
Luminancia	L	lambert	L	
Flujo luminoso	Φ	lumen	lm	
Iluminación (iluminancia)	E	lux	lx	
Ángulo plano	$\alpha, \beta, \gamma, \theta$	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):

- [2] 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
- [3] Datasheet HCSR04 disponible en <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>
- [4] User manual HCSR04 disponible en https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL_pfa39RsB-x2qR4vP8saG73rE/edit#
- [5] Datasheet Wireless ESP8266-01 disponible en https://components101.com/sites/default/files/component_datasheet/ESP8266%20Datasheet.pdf.
- [6] Datasheet HX711 disponible en https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf

[7] <https://www.st.com/resource/en/datasheet/l298.pdf>

[8] <https://www.vishay.com/docs/83760/tcrt5000.pdf>

Autores:



Daniel Hernández Tó Carnet: 201114438



Anibal Robeto Gómez Morales Carnet:
201025198



Cesar Estuardo Tejaxun Xunic

200819197