2020

Principios para IoT

TICS - 4”B” - EQUIPO 1

- Andrés Salvador Rodríguez Mejía
- Amaury Rodríguez Ruiz
- Dana Stephanie López López
- Francisco Escobedo Salazar
- Lucero Alhely Barraza Cedillo
- Luis Fernando Martínez Moya

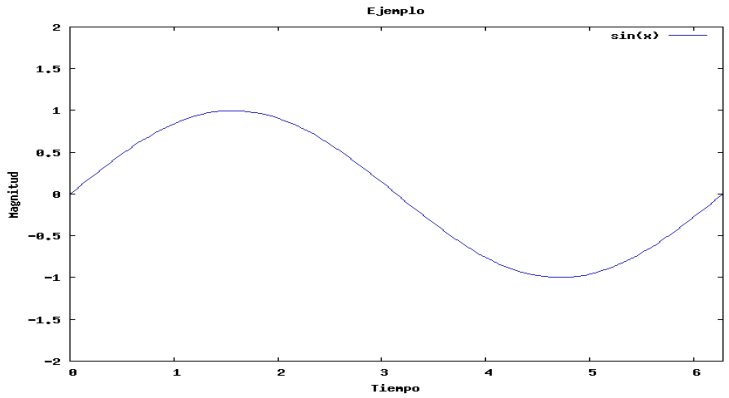
Contenido

Unidad 1 - Conceptos de Electrónica

Señales analógicas y Digitales

Señales analógicas

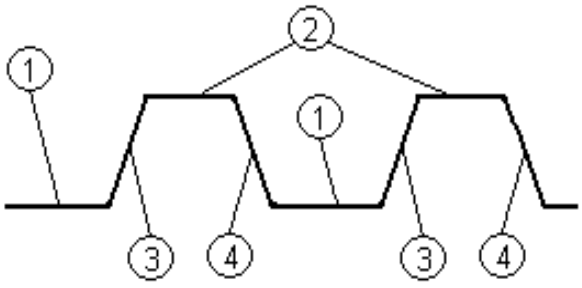
Son variables eléctricas que evolucionan en el tiempo en forma análoga a alguna variable física. Estas variables pueden presentarse en la forma de una corriente, una tensión o una carga eléctrica. Varían en forma continua entre un límite inferior y un límite superior. Cuando estos límites coinciden con los límites que admite un determinado dispositivo, se dice que la señal está normalizada. La ventaja de trabajar con señales normalizadas es que se aprovecha mejor la relación señal/ruido del dispositivo.



Señales digitales

Son variables eléctricas con dos niveles bien diferenciados que se alternan en el tiempo transmitiendo información según un código previamente acordado. Cada nivel eléctrico representa uno de dos símbolos: 0 ó 1, V o F, etc. Los niveles específicos dependen del tipo de dispositivos utilizado

Las señales digitales descriptas tienen la particularidad de tener sólo dos estados y por lo tanto permiten representar, transmitir o almacenar información binaria. Para transmitir más información se requiere mayor cantidad de estados, que pueden lograrse combinando varias señales en paralelo (simultáneas), cada una de las cuales transmite una información binaria. Si hay n señales binarias, el resultado es que pueden representarse 2n estados.



Distinguir las diferencias en el uso de señales analógicas y digitales

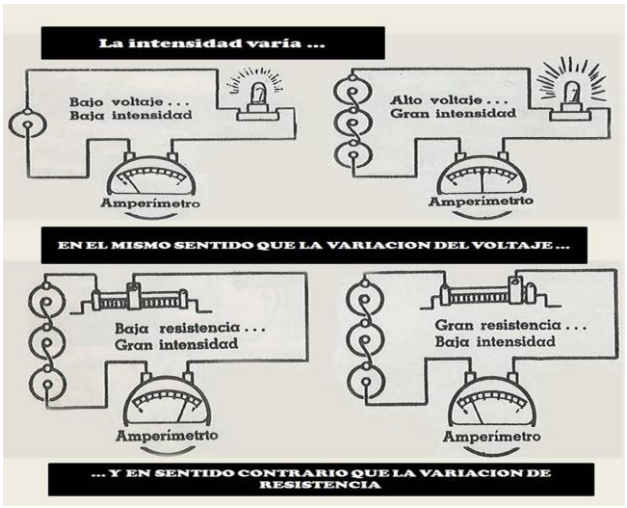
Señal analógica

Las medidas físicas se utilizan cuando hablamos de utilizar señales analógicas, que son especialmente usadas **para llevar a cabo la transmisión de elementos de vídeo o sonido**. Aunque son señales de tipo continuo hay que decir que su expansión se produce por la entrada en escena de las ondas de tipo senoidal.

Pero también tiene sus desventajas. La principal es lo complicado que resulta solucionar una transmisión fallida en comparación a si estuviéramos usando una señal digital. Sin llegar a uno de estos fallos trabajando con señales analógicas también se corre el riesgo de ver cómo el contenido en cuestión se degrada a medida que realizamos copias. Esto no ocurre en una señal digital, donde no importa el número de veces que la repliquemos, dado que nunca hay bajada de calidad

Señal digital

En el otro lado de la balanza tenemos las señales digitales, que se usan de **una forma más frecuente debido a su flexibilidad y polivalencia**. La información no se transmite de la misma forma, sino que en este caso se utiliza un sistema de códigos binarios (los números 0 y 1) con los que se lleva a cabo la transmisión bajo una pareja de amplitudes que proporciona grandes posibilidades. **Proporcionan una mayor capacidad para transmitir información** de una manera fiel. Estas señales no producen deterioro en la información ni en la calidad de los datos, lo que ayuda a que el resultado sea más adecuado.



Ley de Ohm

La **ley de Ohm** se usa para determinar la relación entre tensión, corriente y resistencia en un circuito eléctrico.

Para los estudiantes de electrónica, la ley de Ohm (E = IR) es tan fundamental como lo es la ecuación de la relatividad de Einstein ($E = mc^2$) para los físicos.

E = I x R

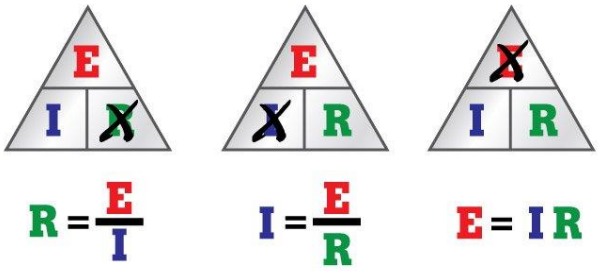
Cuando se enuncia en forma explícita, significa que **tensión = corriente x resistencia, o voltios = amperios x ohmios, o V = A x Ω**.

La ley de Ohm recibió su nombre en honor al físico alemán Georg Ohm (1789-1854) y aborda las cantidades clave en funcionamiento en los circuitos:

Cantidad	Símbolo de ley de Ohm	Unidad de medida (abreviatura)	Rol en los circuitos	En caso de que se esté preguntando:
Tensión	E	Voltio (V)	Presión que desencadena el flujo del electrones	E = fuerza electromotriz (término de la antigua escuela)
Corriente	I	Amperio (A)	Caudal de electrones	I = intensidad
Resistencia	R	Ohmio (Ω)	Inhibidor de flujo	Ω = Letra griega omega

Explicar la ley de ohm.

Si se conocen dos de estos valores, los técnicos pueden reconfigurar la ley de Ohm para calcular el tercero. Simplemente, se debe modificar la pirámide de la siguiente manera:



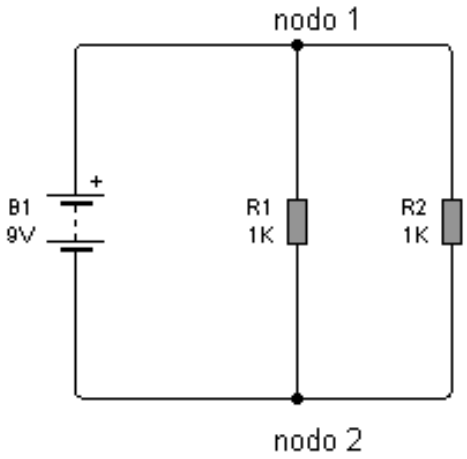
Si conoce el voltaje (E) y la corriente (I) y quiere conocer la resistencia (R), suprima la R en la pirámide y calcule la ecuación restante (véase la pirámide primera o izquierda de arriba).

Ahora, si usted conoce el voltaje (E) y la resistencia (R) y quiere conocer la **corriente** (I), suprima la I y calcule con los dos símbolos restantes (véase la pirámide media anterior).

Y si conoce la corriente (I) y la resistencia (R) y quiere saber el **voltaje** (E), multiplique las mitades de la parte inferior de la pirámide (véase la tercera pirámide o la ubicada en el extremo derecho arriba).

Ley de Kirchhoff

En un circuito eléctrico, es común que se generen nodos de corriente. Un nodo es el punto del circuito donde se unen más de un terminal de un componente eléctrico. Si se desea pronuncie «nodo» y piense en «nudo» porque esa es precisamente la realidad: dos o más componentes se unen anudados entre sí (en realidad soldados entre sí). En la figura a continuación se puede observar el más básico de los circuitos de CC (corriente continua) que contiene dos nodos.



I = V/R = 9 V/[[1000 ohm = 0,009 A = 9 mA]]

Explica las leyes de Kirchhoff.

Enunciado de la primera ley de Kirchhoff, o ley de los nodos, o ley de las corrientes

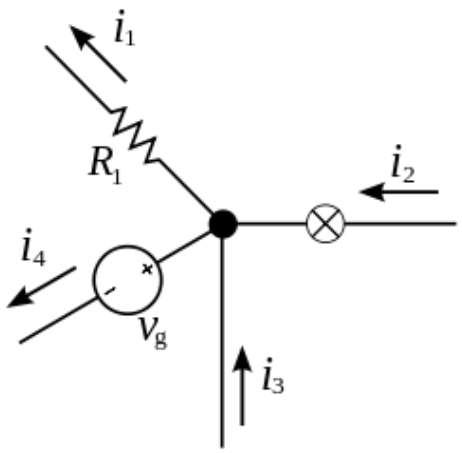
«La corriente entrante a un nodo es igual a la suma de las corrientes salientes».

La razón por la cual se cumple esta ley se entiende perfectamente en forma intuitiva si uno considera que la corriente eléctrica es debida a la circulación de electrones de un punto a otro del circuito.

Los electrones están guiados por el conductor de cobre que los lleva hacia el nodo Llegados a ese punto los electrones se dan cuenta que la resistencia eléctrica hacia ambos resistores es la misma y entonces se dividen circulando 5 por un resistor y otros 5 por el otro.

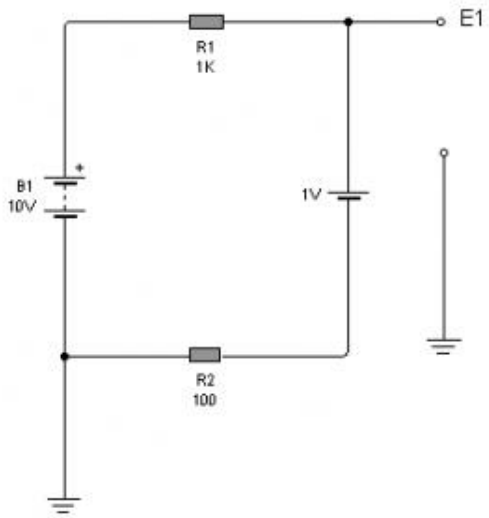
El nodo no puede generar electrones ni retirarlos del circuito solo puede distribuirlos y lo hace en función de la resistencia de cada derivación. En nuestro caso las resistencias son iguales y entonces envía la misma cantidad de electrones para

cada lado. Si las resistencias fueran diferentes, podrían circular tal vez 1 electrón hacia una y nueve hacia la otra de acuerdo a la aplicación de la ley de Ohm.



Enunciado de la segunda ley de Kirchhoff, o ley de los voltajes

En un circuito cerrado, la suma de las tensiones de batería que se encuentran al recorrerlo siempre será iguales a la suma de las caídas de tensión existente sobre los resistores.



Obsérvese que el circuito posee dos baterías y dos resistores y se desea saber cuál es la tensión de cada punto (o el potencial), con referencia al terminal negativo de B1 al que le colocamos un símbolo que representa a una conexión al planeta y al que se llama tierra o masa. Se debe considerar al planeta tierra como un inmenso conductor de la electricidad.

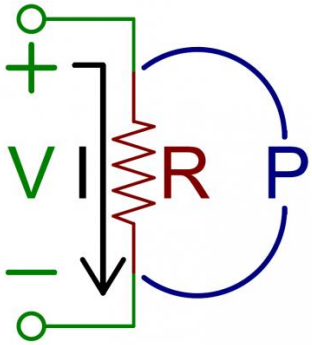
Las tensiones de fuente, simplemente son las indicadas en el circuito, pero si pretendemos aplicar las caídas de potencial en los resistores, debemos determinar primero cual es la corriente que circula por aquel.

Potencia Eléctrica

Para entender qué es la potencia eléctrica es necesario conocer primeramente el concepto de “energía”, no es más que la capacidad que tiene un mecanismo o dispositivo cualquiera para realizar un trabajo.

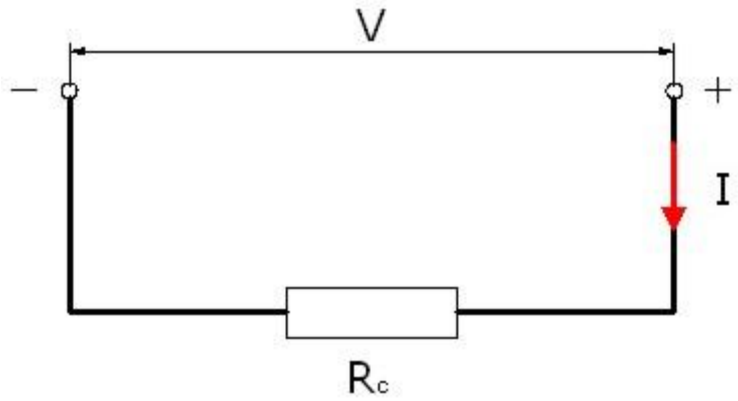
Cuando se conecta un equipo o consumidor eléctrico a un circuito alimentado por una fuente de fuerza electromotriz (F.E.M), como puede ser una batería, la energía eléctrica que suministra fluye por el conductor, permitiendo que, por ejemplo, una bombilla de alumbrado, transforme esa energía en luz y calor, o un motor pueda mover una maquinaria, esta energía consumida se mide kWh

De acuerdo con la definición de la física, “la energía ni se crea ni se destruye, se transforma”. En el caso de la energía eléctrica esa transformación se manifiesta en la obtención de luz, calor, frío, movimiento, o en otro trabajo útil que realice cualquier dispositivo conectado a un circuito eléctrico cerrado.



Potencia en corriente continua

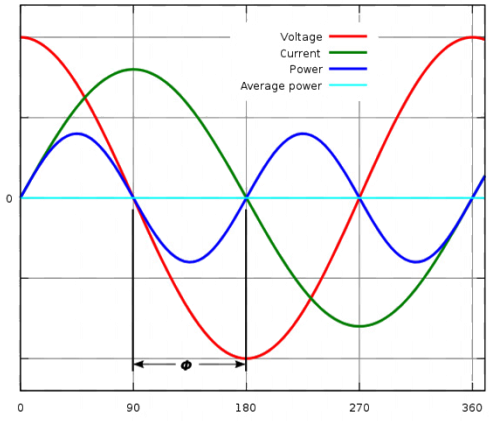
Cuando se trata de corriente continua (CC) la potencia eléctrica desarrollada en un cierto instante por un dispositivo de dos terminales, es el producto de la diferencia de potencial entre dichos terminales y la intensidad de corriente que pasa a través del dispositivo. Por esta razón la potencia es proporcional a la corriente y a la tensión.



Potencia en corriente alterna

El cálculo de la potencia eléctrica en circuito de corriente alterna se hace más complejo debido al desfase que provocan ciertos consumidores entre la corriente y la tensión.

Por esto cuando se trata de corriente alterna (AC) sinusoidal, el promedio de potencia eléctrica desarrollada por un dispositivo de dos terminales es una función de los valores eficaces o valores cuadráticos medios, de la diferencia de potencial entre los terminales y de la intensidad de corriente que pasa a través del dispositivo.



Explicar la fórmula de la potencia eléctrica.

La fórmula que más se utiliza para calcular la potencia eléctrica en electricidad es:

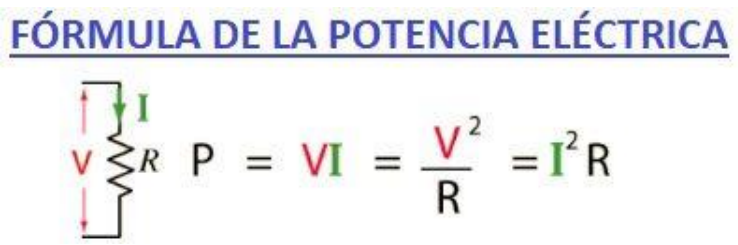
P = V x I;

Que quiere decir, que cuando conectamos un aparato eléctrico a una tensión V, si multiplicamos esta tensión por la intensidad de corriente que lo atraviesa, el resultado de la multiplicación es la potencia eléctrica del aparato.

La potencia eléctrica **se mide en vatios (w)** aunque es muy común verla en Kilovatios (Kw). 1.000w es 1Kw de potencia. Para pasar de w a kw solo tendremos que dividir entre 1.000.

Para obtener la potencia en vatios en la fórmula anterior, la tensión se debe de poner en Voltios y la Intensidad en Amperios.

Ahora fíjate que pasa si sustituimos la V de la fórmula o la I, por su valor según la ley de ohm:

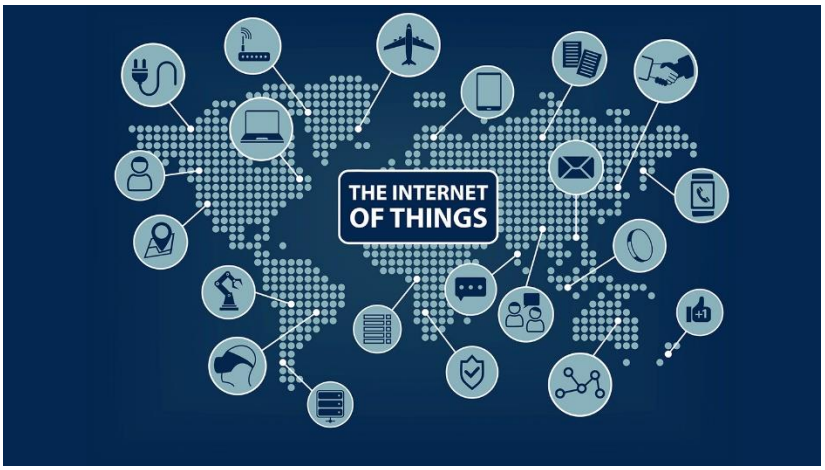


Unidad 2 - Introducción al IoT

Conceptos de IoT

La definición de IoT podría ser la agrupación e interconexión de dispositivos y objetos a través de una red (bien sea privada o Internet, la red de redes), dónde todos ellos podrían ser visibles e interaccionar. Respecto al tipo de objetos o dispositivos podrían ser cualquiera, desde sensores y dispositivos mecánicos hasta objetos cotidianos como pueden ser el frigorífico, el calzado o la ropa.

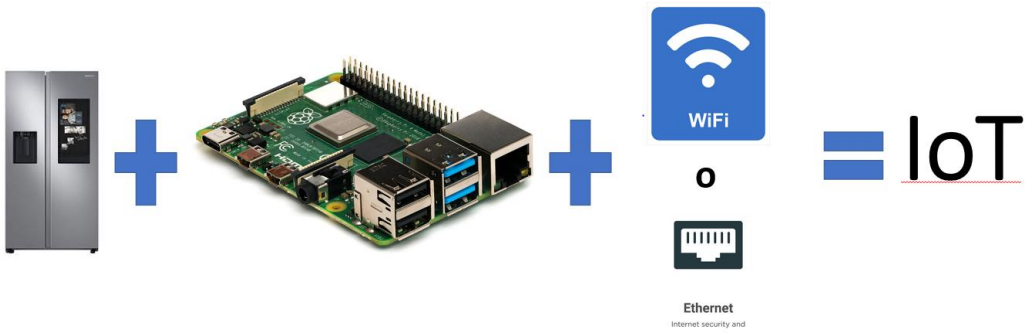
Cualquier cosa que se pueda imaginar podría ser conectada a internet e interaccionar sin necesidad de la intervención humana, el objetivo por tanto es una interacción de máquina a máquina, o lo que se conoce como una interacción M2M (machine to machine) o dispositivos M2M.



Definir los conceptos de IoT, Sistemas embebidos y Hardware abierto.

Sistema Embebido

Un Sistema Embebido es un sistema electrónico diseñado para realizar pocas funciones en tiempo real, según sea el caso. Al contrario de lo que ocurre con las computadoras, las cuales tienen un propósito general, ya que están diseñadas para cubrir un amplio rango de necesidades y los Sistemas Embebidos se diseñan para cubrir necesidades específicas.



Los Sistemas Embebidos a pesar de no ser muy nombrados están en muchas partes, en realidad, es difícil encontrar algún dispositivo cuyo funcionamiento no esté basado en algún sistema embebido, desde automóviles hasta teléfonos celulares e incluso en algunos electrodomésticos comunes como refrigeradores y hornos de microondas.

Hardware abierto

Hardware libre es aquel cuyas especificaciones y diagramas esquemáticos son de acceso público, ya sea bajo algún tipo de pago o de forma gratuita". Tradicionalmente estos proyectos se han asociado a perfiles muy técnicos, aunque algunos avances de los últimos años y fenómenos como la fabricación digital o el movimiento 'maker' están ayudando a popularizarlo.

Estas son algunas de las iniciativas más interesantes en torno a este concepto y que promueven desde la creación de ordenadores y teléfonos modulares hasta la fabricación de automóviles, pasando por el impulso de la robótica en el aula.

- **RepRap Project**
- **Raspberry Pi**
- **Arduino**
- **e-puck**
- **Tabby EVO**
- **Uzebox**

Arquitectura de sistemas IoT

La arquitectura tiene que cumplir ciertos requerimientos para que esta tecnología sea viable. Debe permitir que la tecnología sea distribuida, donde los objetos puedan interactuar entre ellos, escalable, flexible, robusta, eficiente y segura.

La arquitectura debe:

- Permitir escalabilidad, ampliación de capacidades y soporte de nuevos estándares.
- Servir de modelo para la creación de arquitecturas más específicas.
- Ser horizontal, para permitir la integración de diferentes soluciones IoT.
- Se necesita un correcto control de los numerosos elementos involucrados en cualquier solución IoT.
- La gestión necesita ser garantizada desde sus cinco áreas funcionales.
- Debe incluir la posibilidad de programar aplicaciones para los sistemas IoT, lo cual permite mayores beneficios

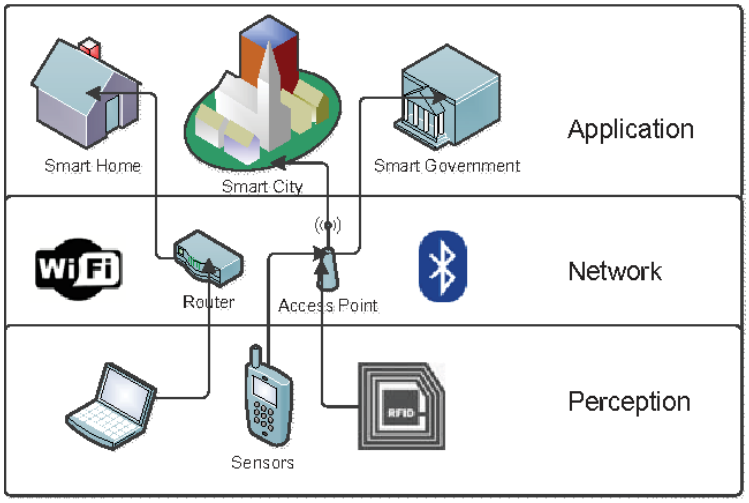


Figure 1. Three-layer IoT architecture.

Identificar los elementos de sistemas IoT.

Actuadores: controlan el estado físico o lógico de un elemento a través de señales, como encender/apagar un dispositivo.

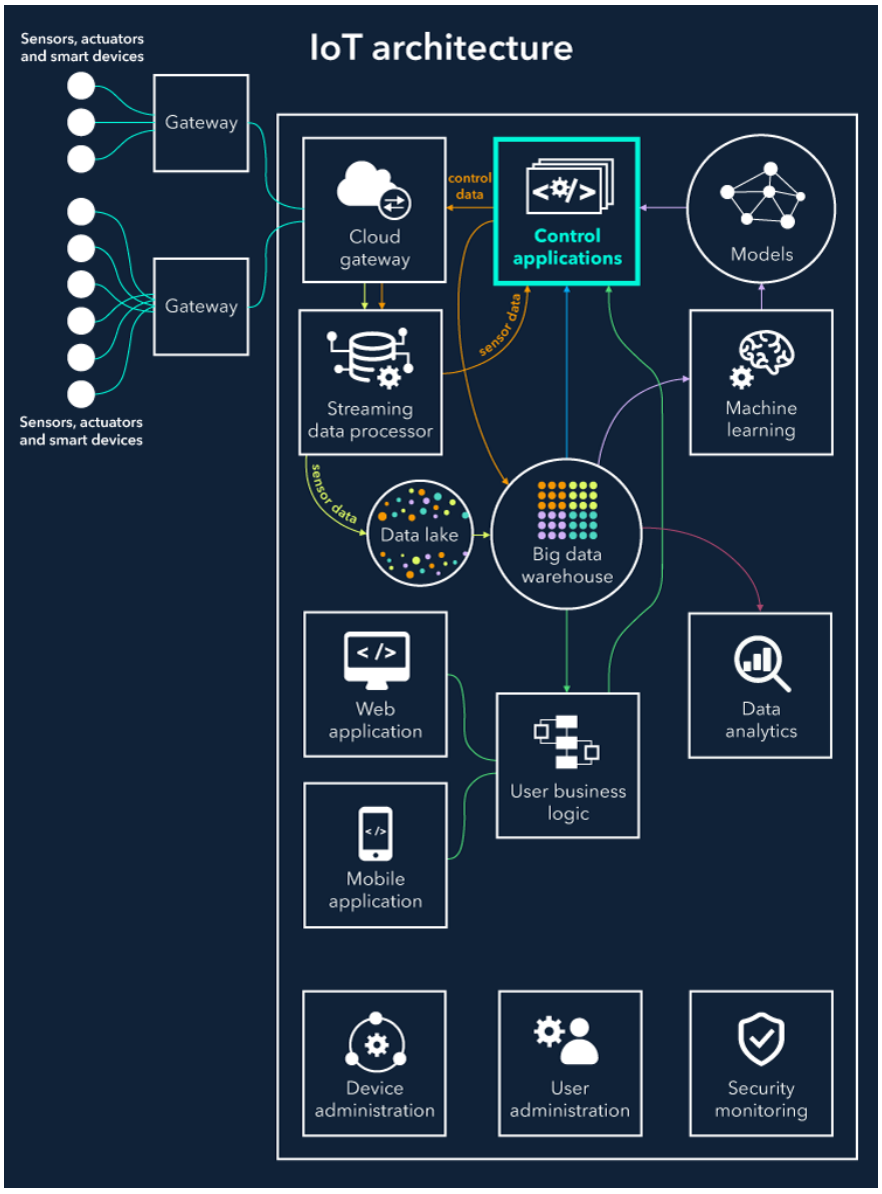
Agentes: actúan de intermediarios entre un actuador y la nube, eligiendo qué datos enviar y a quién.

Comunicación: simplemente es la capa física de comunicación, aunque la elección de la misma condiciona el protocolo de comunicación.

Dispositivo de cálculo en campo: debido al gran volumen de información que se puede recopilar, en ocasiones conviene tratar la información antes de enviarla a la red.

Sensores: recogen y envían información del estado actual de los elementos a los que están conectados. Podemos medir temperatura, presión, luz, movimiento, posicionamiento, etc.

Cosas: El objetivo real del IoT es obviamente conectar objetos, aunque se puede aplicar tanto a productos físicos como coches u otros elementos como cultivos o ganado, por ejemplo.

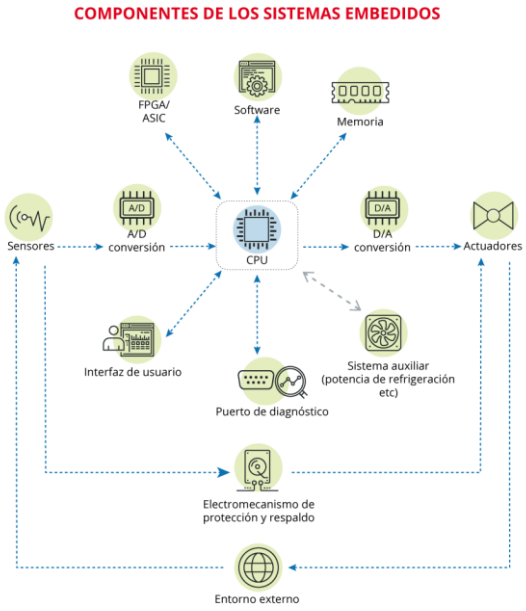


Medios de comunicación de sistemas embebidos

En la parte central se encuentra el microprocesador, microcontrolador, DSP, etc. Es decir, la CPU o unidad que aporta capacidad de cómputo al sistema, pudiendo incluir memoria interna o externa, un micro con arquitectura específica según requisitos.

La comunicación adquiere gran importancia en los sistemas embebidos. Lo normal es que el sistema pueda comunicarse mediante interfaces estándar de cable o inalámbricas. Así un SI normalmente incorporará puertos de comunicaciones del tipo RS-232, RS-485, SPI, I²C, CAN, USB, IP, Wi-Fi, GSM, GPRS, DSRC, etc.

Se denominan actuadores a los posibles elementos electrónicos que el sistema se encarga de controlar. Puede ser un motor eléctrico, un conmutador tipo relé etc. El más habitual puede ser una salida de señal PWM para control de la velocidad en motores de corriente



Describir los medios de comunicación de datos y señales:

Un sistema embebido, es un sistema computacional que se ha diseñado para realizar funciones dedicadas, estos son programables y tienen microcontroladores y microprocesadores incorporados sobre el mismo, crear un dispositivo de IoT con un sistema embebido, designa a un Endpoint que puede realizar tareas de procesamiento en tiempo real.

Red de datos.

Red de datos: Se denomina red de datos a aquellas infraestructuras o redes de comunicación que se ha diseñado específicamente a la Transmisión de información mediante el intercambio de datos.

Bluetooth.

Bluetooth: Bluetooth es una especificación industrial para redes inalámbricas de área personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2.4 GHz.

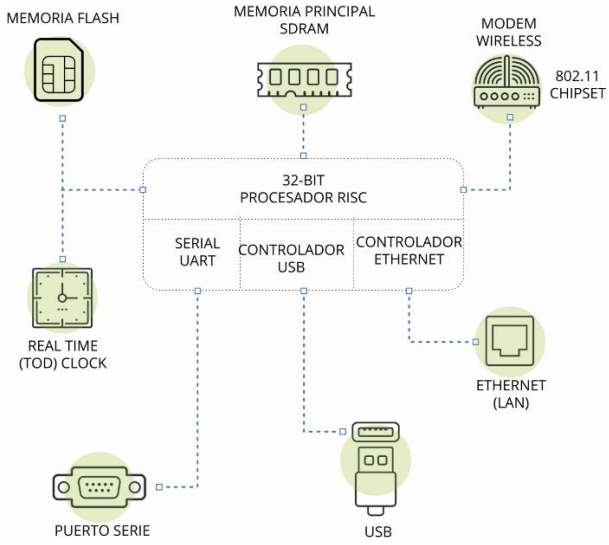
Serial.

Serial: Comunicación serial Sigue siendo un medio importante de comunicación en muchas aplicaciones informáticas y de red. Se usa ampliamente en sistemas integrados debido a que la mayoría de las computadoras pueden manejar conexiones en serie de forma nativa o con la ayuda de emuladores o conversores.

GSM.

GSM: es un sistema estándar, libre de regalías, de telefonía móvil digital.

Se considera, por su velocidad de transmisión y otras características, un estándar de segunda generación (2G). Su extensión a 3G se denomina UMTS y difiere en su mayor velocidad de transmisión, el uso de una arquitectura de red ligeramente distinta y sobre todo en el empleo de diferentes protocolos de radio (W-CDMA).



Sensores y actuadores

Son artefactos que permiten determinar valores de una magnitud determinada, es decir que detectan indicadores externos e internos o también variables de instrumentación, ya sea intensidad de la luz, sonido, temperatura del ambiente, presencia de personas, nivel de agua, fuerza , torsión, pH, etc.

Un sensor es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. Las variables de instrumentación pueden ser por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, movimiento, pH, etc. Una magnitud eléctrica puede ser una resistencia eléctrica (como en una RTD), una capacidad eléctrica (como en un sensor de humedad o un sensor capacitivo), una tensión eléctrica (como en un termopar), una corriente eléctrica (como en un fototransistor), etc.

Un actuador es un dispositivo capaz de transformar una energía en activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado. Recibe la orden de un regulador o controlador y en función a ello genera la orden para activar un elemento final de control como por ejemplo, una válvula. Algunos de ellos son:

- Electrónicos, también son muy utilizados en los aparatos mecatrónicos, como por ejemplo, en los robots. Los servomotores CA sin escobillas se utilizan en el futuro como actuadores de posicionamiento preciso debido a la demanda de funcionamiento sin tantas horas de mantenimiento.
- Hidráulicos, que son los de mayor antigüedad, pueden ser clasificados de acuerdo con la forma de operación funcionan en base a fluidos a presión.
- Neumáticos, actuadores mecánicos que convierten la energía del aire comprimido en trabajo mecánico. El rango de compresión es mayor que en los actuadores

Identificar los tipos de sensores y actuadores utilizados en sistemas embebidos.

A la hora de elegir un sensor para Arduino debemos tener en cuenta los valores que puede leer las entradas analógicas o digitales de la placa para poder conectarlo o sino adaptar la señal del sensor a los valores que acepta Arduino.

Características de los sensores

- Rango de medida: dominio en la magnitud medida en el que puede aplicarse el sensor.
- Precisión: es el error de medida máximo esperado.
- Linealidad o correlación lineal.
- Sensibilidad de un sensor: suponiendo que es de entrada a salida y la variación de la magnitud de entrada.
- Resolución: mínima variación de la magnitud de entrada que puede detectarse a la salida.
- Rapidez de respuesta: puede ser un tiempo fijo o depender de cuánto varíe la magnitud a medir. Depende de la capacidad del sistema para seguir las variaciones de la magnitud de entrada.
- Repetitividad: error esperado al repetir varias veces la misma medida.

Ejemplos Sensores para Arduino

- ACS714
- Sensor 4 en 1
- DHT22
- DS18B20
- Sensor ultrasonico de distancia
- Sensor de presion barometrica y altura (I2C)
- Sensor laser distancia
- Sensor temperatura kit
- Caudalimetro

Actuadores y Periféricos

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

Ejemplos de Actuadores y Periféricos para Arduino

- Sacar por TV datos de Arduino, librería TV OUT
- Relés
- Actuadores lineales
- Displays
- Dimmer AC
- Motores

Unidad 3 - Programación de sistemas embebidos

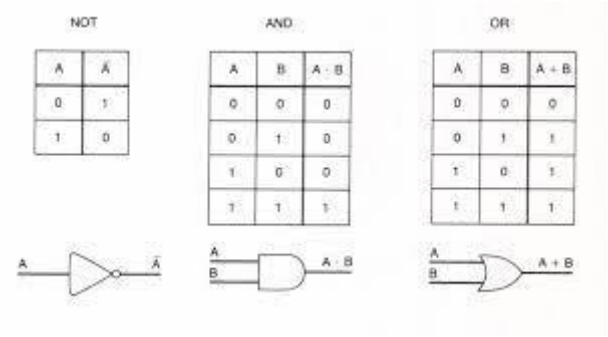
Configuración del hardware abierto

El hardware de código abierto (OSH) consiste en artefactos físicos de tecnología diseñados y ofrecidos por el movimiento de diseño abierto. El término generalmente significa que la información sobre el hardware se discierne fácilmente para que otros puedan hacerlo, acoplándola estrechamente al movimiento del fabricante. El diseño de hardware (es decir, dibujos mecánicos, esquemas, listas de materiales, datos de diseño de PCB, código fuente HDL y datos de diseño de circuito integrado), además del software que maneja el hardware, se publican bajo términos libres.

Su objetivo es crear diseños de aparatos informáticos de forma abierta, de manera que todas las personas puedan acceder, como mínimo, a los planos de construcción de los dispositivos.

Operadores Lógicos o Booleanos:

- NOT = Negación
- AND = Conjunción
- OR = Disyunción



El operador **NOT**, también conocido como INVERSOR. ¿Por qué inversor? Porque a toda entrada A que puede tener sólo los valores "0" (FALSO) o "1" (VERDADERO), corresponde como salida el valor contrario: 1 o 0 (VERDADERO o FALSO) respectivamente.

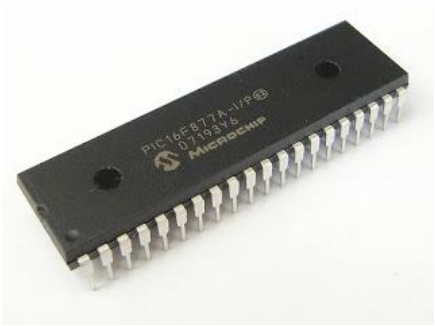
En el caso de la Conjunción o **AND**, aquí necesitamos al menos DOS entradas: "A" y "B". Observe que en la Tabla de Verdad, la única manera en la que el resultado es "1" o "VERDADERO" es cuando el valor de ambas variables "A" y "B" es "1" o "VERDADERO". De otra manera el resultado es "0" o "FALSO".

Por último en el caso de la Disyunción u **OR**, siempre que exista una entrada en "1" o "VERDADERO", el resultado será "1" o "VERDADERO". De otra forma será "0" o "FALSO".

Describir el funcionamiento del hardware abierto.

Microcontroladores

Un microcontrolador (abreviado µC, UC o MCU) es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida.



Microprocesadores

El microprocesador (o simplemente procesador) es el circuito integrado central más complejo de un sistema informático; a modo de ilustración, se le suele llamar por analogía el «cerebro» de un ordenador.

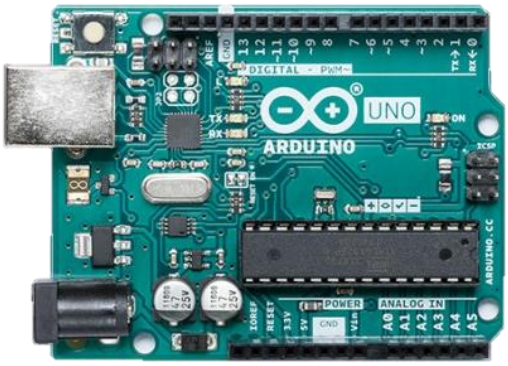
FPGA. Matrices de Compuertas Programables "In Situ"

Una FPGA o matriz de puertas programables (del inglés field-programmable gate array) es un dispositivo programable que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada en el momento mediante un lenguaje de descripción especializado. La lógica programable puede reproducir desde funciones tan sencillas como las llevadas a cabo por una puerta lógica o un sistema combinacional hasta complejos sistemas en un chip.

Programación de hardware abierto

Arduino es una plataforma de hardware abierto que facilita la programación de un microcontrolador, además simplifica el trabajo con microcontroladores y ofrece las siguientes ventajas: barato, multiplataforma, entorno de programación sencillo, software libre y extensible mediante librerías en C++ y dar el salto a AVR-C, hardware libre y extensible

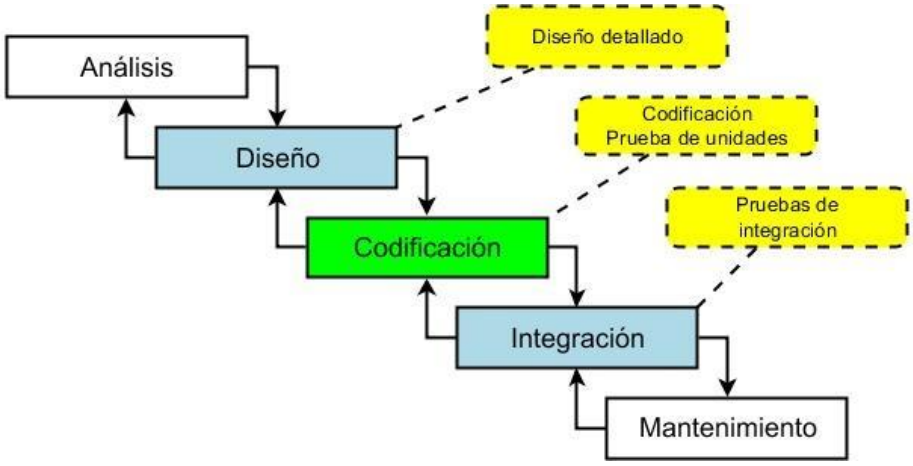
Por otro lado Arduino nos proporciona un software consistente en un entorno de desarrollo (IDE) que implementa el lenguaje de programación de arduino y el bootloader ejecutado en la placa. La principal característica del software de programación y del lenguaje de programación es su sencillez y facilidad de uso.



Identificar el entorno de programación de hardware abierto.

Un entorno de desarrollo de software es una combinación de herramientas que automatiza o soporta al menos una gran parte de las tareas (o fases) del desarrollo: análisis de requisitos, diseño de arquitectura, diseño detallado, codificación, pruebas de unidades, pruebas de integración y validación, gestión de configuración, mantenimiento, etc. Las herramientas deben estar bien integradas, pudendo interoperar unas con otras.

Están formados por el conjunto de instrumentos (hardware, software, procedimientos) que facilitan o automatizan las actividades de desarrollo. En el contexto de esta asignatura se consideran básicamente los instrumentos software.



Entornos de Desarrollo para El Lenguaje C++

GNAT GPS (C++): es entorno libre multi-idioma de desarrollo integrado (IDE) por AdaCore. GPS utiliza los compiladores de la colección de compiladores de GNU, tomando su nombre de GNAT, el compilador de GNU para el lenguaje de programación Ada. GPS es multi-plataforma, que se ejecuta en Linux, Microsoft Windows y Solaris.

Eclipse CDT (C++): La CDT (C / C + + Herramientas de Desarrollo) es Proyecto que ofrece un completo y funcional Entorno de desarrollo integrado (IDE) de C y C + + para la plataforma Eclipse. Web

Borland C++ Builder (C++): Varios productos de Borland están también disponibles para GNU/Linux, entre ellos Interbase, JBuilder y Kylix que integraba Delphi y C++Builder (aunque Kylix fue abandonado tras la versión 3.0). C++Builder es un entorno de desarrollo rápido de aplicaciones en lenguaje C++ para Windows

Dev-C++: es un entorno de desarrollo integrado (IDE) para programar en lenguaje C/C++. Usa MinGW que es una versión de GCC (GNU Compiler Collection) como su compilador. Dev-C++ puede además ser usado en combinación con Cygwin y cualquier compilador basado en GCC.

Visual C++ (Visual Studio Microsoft): Visual C++ 2010 proporciona un entorno de desarrollo eficaz y flexible para crear aplicaciones basadas en Microsoft Windows y en Microsoft .NET. Puede utilizarlo en un sistema de desarrollo integrado o puede utilizar herramientas individuales. Web

Code::Blocks (C++) : es un libre de C + +, IDE diseñado para satisfacer las necesidades más exigentes de sus usuarios. Se ha diseñado para ser muy extensible y totalmente configurable

Identificar la sintaxis del lenguaje de programación de hardware abierto.

Entornos como Eclipse o NetBeans, están basados en Java; o MonoDevelop, basado en C#. También puede incorporarse la funcionalidad para lenguajes alternativos mediante el uso de plugins. Por ejemplo, Eclipse y NetBeans tienen plugins para C, C++, Ada, Perl, Python, Ruby y PHP, entre otros.

Como primera instancia c++ es un lenguaje complejo que requiere de concentración y exigencia ya que un solo pequeño error no servirá el programa, teniendo en cuenta que todas los códigos se cierran con punto y coma (;). Aquí se presenta los comandos más básicos.

- INT: Este comando sirve para declarar variables, se utiliza así: **int num1, num2**;
- PRINTF: Este comando sirve para (escribir lo que desea el usuario), se maneja así: **printf ("digite un número");**
- SCANF: Este comando sirve para guardar las variables declaradas con int se tendrá en cuenta poner el ampersan y el símbolo de porcentaje, se utiliza así: **scanf ("%d", &num1);**
- RESULTADO: Este comando sirve para definir el resultado de la variable, se utiliza así: **resultado=num1+num2;**