

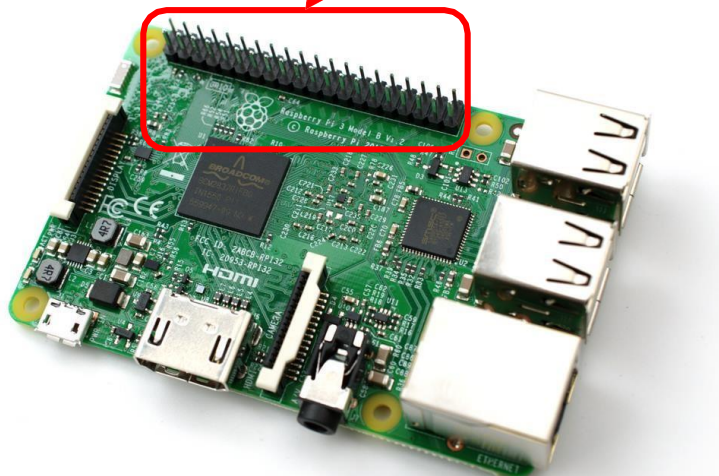
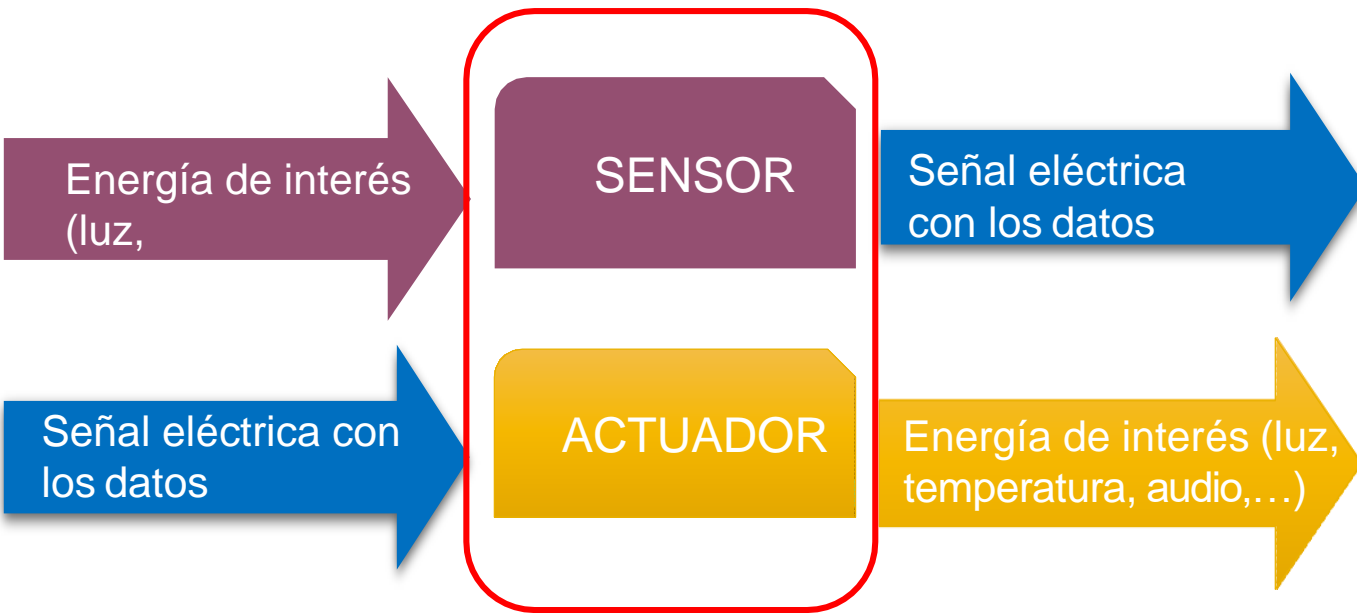
2. DISPOSITIVOS, SENSORES Y ACTUADORES IoT

Eduard Lara

2.1 Componentes involucrados en las “Things” - Sensores y Actuadores



2.1 Componentes involucrados en las “Things” - Sensores y Actuadores

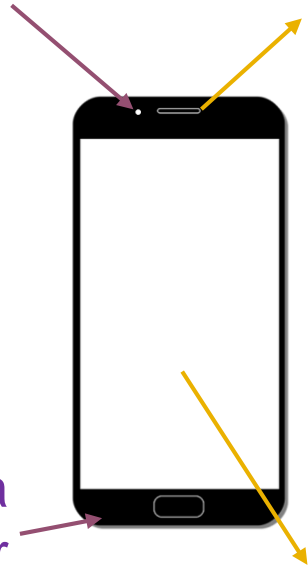


Luz entra en el sensor de la cámara
Y se convierte en señal eléctrica

La señal eléctrica entra en Altavoz y se convierte en sonido

Sonido entra en el sensor micrófono Y se convierte en señal eléctrica

La señal eléctrica entra en La pantalla y se convierte en luz



2.1 Componentes involucrados en una thing

- Los sensores y actuadores son los componentes involucrados en las things.
- **Un sensor** es un dispositivo que captura la energía de interés (luz, sonido, presión o cualquier magnitud) y lo transforma en una señal eléctrica con los datos. Si esta señal eléctrica fuera analógica tendría un valor en función de la cantidad de energía que entra. Si fuera una señal digital, tendrá una serie de unos y ceros que representarían la magnitud que está sensorizando
- **Un actuador** es el caso opuesto a un sensor. En un actuador entra la señal eléctrica con los datos concretos y en base a esa señal eléctrica, se proporciona la energía de interés. Por ejemplo un led que se ilumine a una cierta temperatura o el sonido que produce un altavoz

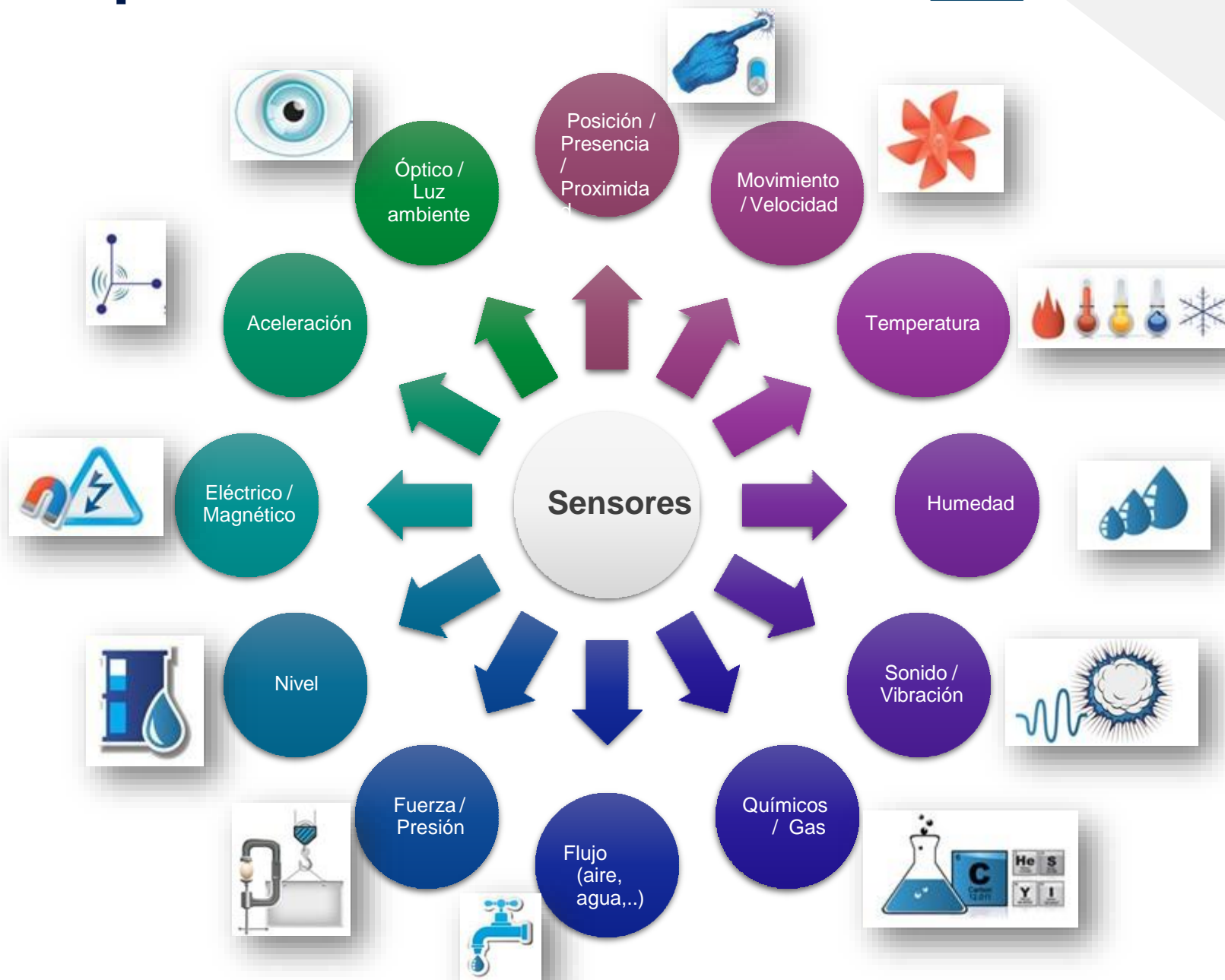
Todos estos sensores/actuadores estarían dentro de una plataforma embebida (por ejemplo una placa Raspberry) con los sensores y/o actuadores conectados a unos determinados pines, su módulo de comunicación, CPU, E/S y HDMI.

2.1 Componentes involucrados en una thing

Usaremos nuestro Smartphone para mostrar el concepto de sensores y actuadores. Nuestro smartphone tiene una CPU, una RAM, una serie de sensores y actuadores, el mismo estilo de una plataforma embebida IoT:

- **Ejemplos de sensores en el smartphone:**
 - 1) la luz que entra en el sensor de la cámara, el Smartphone lo va a convertir en una señal eléctrica.
 - 2) El sonido que entra en nuestro micrófono, va a convertir las ondas de presión en una corriente eléctrica que representará el valor por una señal digital de 1's y 0's
- **Ejemplos de actuadores en el smartphone:**
 - 1) El altavoz convierte la señal eléctrica que le envía la placa base de nuestro Smartphone y lo transforma en sonido. Por tanto es el extremo opuesto de un sensor
 - 2) La señal eléctrica que se envía a la pantalla de nuestro smartphone, va a hacer que se convierta en luz visible.

2.2 ¿Qué podemos sensorizar?



2.2 ¿Qué podemos sensorizar?

Vamos a ver cuáles son las magnitudes que podemos sensorizar con nuestros sensores

- 1) Un sensor de posición/presencia o proximidad: Si se detecta presencia de un objeto o persona en una determinada ubicación o se está acercando al lugar, el sensor va a capturar esa información y va a enviar una señal eléctrica indicando este hecho
- 2) Un sensor de movimiento y velocidad. No solamente que haya presencia sino que detecte el movimiento concreto y la velocidad a la que va ese objeto.
- 3) Un sensor de temperatura. Controlar la temperatura ambiente para que pueda activar o no el aire acondicionado o la calefacción.
- 4) Un sensor de humedad. En base a la humedad podríamos quizás activar o desactivar un sistema de aspersión.
- 5) Sensor de sonido o vibración. En base a las ondas acústicas que llegan a ese sensor se va a producir una corriente eléctrica proporcional a esa presión sonora
- 6) Sensor de químicos o gases, para cualquier entorno de laboratorio.

2.2 ¿Qué podemos sensorizar?

- 7) Sensor de flujo de aire o de agua, para medir por ejemplo el caudal por segundo que está pasando por una tubería.
- 8) Sensor de fuerza o presión que se ejerce sobre un objeto, para aplicaciones industriales.
- 9) Sensor de nivel. Por ejemplo el nivel de una piscina cuando llegue a un nivel concreto de agua , se para de seguir llenándola
- 10) Sensorizar un campo eléctrico o un campo magnético. Si se produce una corriente eléctrica alrededor nuestro, que un sensor nos avise de este hecho
- 11) Sensor de aceleración de un dispositivo o un vehículo. Es utilizado para la eficiencia energética en caso de la aceleración eólica.
- 12) Sensor de la parte óptica, para que en función de la luz ambiente se encienda una farola o se apague.

2.3 Dispositivos IoT - Plataformas Prototyping

Aspectos a tener en cuenta

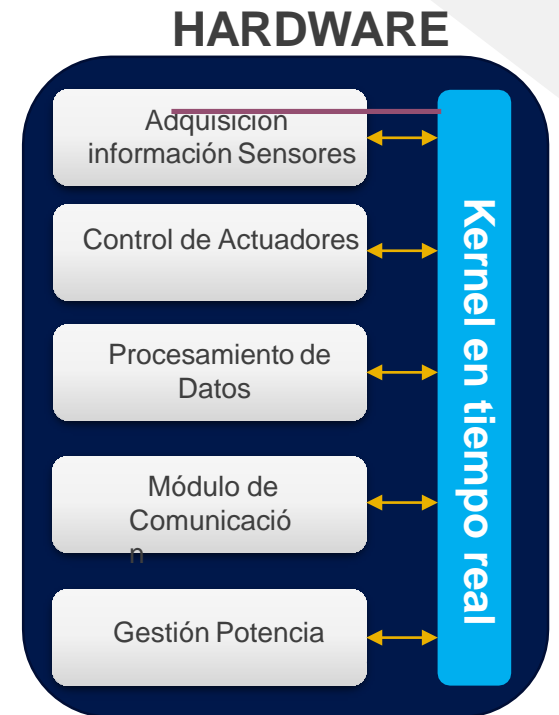
- **Restricciones**

- Tamaño
- Potencia / Consumo
- Coste
- Funcionalidades
- Memoria
- HW Especializado
- Seguridad



- **Módulos**

- Conectividad
- Procesador
- Alimentación
- Interfaz de usuario
- Sensores / Actuadores



2.3 Dispositivos IoT - Plataformas Prototyping

Vamos a ver una comparativa de las plataformas de prototyping o thing, que podríamos utilizar en nuestro diseño. Recordar que todos los sensores/actuadores al final van a estar comunicados y embebidos dentro de una plataforma que vamos a prototipar con su CPU, almacenamiento, dispositivos E/S, etc

Aspectos a tener en cuenta en la restricciones de una plataforma de prototipado.

- 1) Restricción en cuanto al tamaño, de cara a poderlo embeber en algún otro producto final
- 2) La potencia del consumo que puede tener. Hay muchas thinks que van a requerir un consumo muy bajo, puesto que van a estar ubicadas en una posición que requieren de decenas de años de vida útil.
- 3) Coste.
- 4) Funcionalidades específicas de esa plataforma, capacidad de cálculo concreta o si necesitamos una memoria determinada de cara a las operaciones que va a realizar
- 5) Necesidades de interconexión con algún HW especializado
- 6) Restricciones de seguridad mayores que las comunes

2.3 Dispositivos IoT - Plataformas Prototyping

Aspectos a tener en cuenta respecto los módulos y las plataformas de prototyping son:

- 1) Un módulo de conectividad
- 2) Un procesador
- 3) Un módulo de alimentación
- 4) Una parte de interfaz de usuario
- 5) Todos los sensores/actuadores que hemos comentado

2.3 Dispositivos IoT - Plataformas Prototyping

La parte hardware que constaría una plataforma de prototipado :

- 1) Un módulo de adquisición de información de los sensores
- 2) Un módulo de control de los actuadores
- 3) Un módulo para el procesamiento de datos. Esto lo haría la CPU de nuestra unidad
- 4) Un módulo de comunicaciones
- 5) Un módulo de gestión de potencia

Todas estas partes estarían en tiempo real contactando con el kernel de nuestro dispositivo para poder manejar toda esta información.





2.3 Dispositivos IoT - Plataformas Prototyping

La parte de software de nuestra unidad thing, constaría de:

- 1) Los protocolos de comunicaciones que se van a utilizar para que el dispositivo esté codificado y programado para poder comunicarse en base a esos protocolos.
- 2) Una capa de abstracción hardware para que no tengamos que trabajar directamente con el nivel físico (a nivel de voltajes) y podamos ir a una capacitación mayor para poder programar nuestras unidades.
- 3) También se incluiría en determinados casos un sistema operativo. No todas las plataformas de prototipado tiene S.O.: Arduino no lo tiene, pero Raspberry sí.

Dentro de la parte software vamos a tener siempre una parte de gestión remota. Veremos que las plataformas de IoT van a tener siempre una gestión remota para todas las things de nuestra solución

2.3. Comparativa Plataformas Prototyping

	 Arduino	 Raspberry Pi	 ARM Cortex M Series	 ASICS
Conectividad	Ethernet RJ45 / Wi-Fi (Arduino Yun)	Ethernet RJ45 / Wi-Fi / USB / HDMI	No es una plataforma embebida, pero tiene capacidad de utilizar conectividad WiFi, 2G/3G/4G,...	Dispositivo a medida para cada aplicación
Procesamiento / Memoria	1 core 8 bit 16 MHz/ 2kB SRAM	4 cores 16 bits (Pi3-B) – 1.2 GHz / 1GB RAM	CPU 32 bits Amplia gama para adaptarse a las necesidades y compensar rendimiento y consumo energético (M4 – 180 MHz)	Dispositivo a medida para cada aplicación
Sistema Operativo	Unidad embebida sin S.O.	Unidad embebida con S.O.Raspbian	Utilizado con más de 40 S.O. diferentes	Se puede fabricar a medida para optimizar el uso de un S.O. concreto
Ventajas	Fácil de usar, comunidad extensa para soporte Mayor compatibilidad con sensores y actuadores	Conexión a internet automática con DHCP. Comunidad extensa Capacidad de almacenamiento y procesamiento elevada	Es una CPU que no depende de los ciclos de producción de las plataformas embebidas como Arduino / Raspberry, por tanto acelera el despliegue de soluciones IoT	Permite diferenciación de la solución IoT concreta al fabricarse a medida para la aplicación Muy bajo coste en casos de un volumen de necesidad muy elevado
Desventajas	Bajo rendimiento	No es fácil integrar en un producto	No tiene la facilidad de uso de las plataformas embebidas para iniciarse en aplicaciones IoT	Coste muy elevado si el número de unidades es bajo
Escenario donde usarlo	Aplicaciones sencillas y que requieren de captar información de sensores y controlar actuadores desde un punto de vista de cliente. No es apto para aplicaciones Real-time	Aplicaciones IoT server-based debido a sus altas capacidades de almacenamiento y procesamiento y su soporte para múltiples lenguajes de programación Apto para aplicaciones Real-time	SOC muy utilizado en dispositivos conectados que requieren de altas capacidades, reducido tamaño y no depender de ciclos de producción como wearables Aplicación en escenarios real-time	Necesidad de aplicaciones a medida con un volumen muy elevado de unidades Escenarios con necesidades de alta eficiencia energética (posibilidad de uso de CPU en 7nm) Utilizado en aplicaciones que requieren de un tamaño muy pequeño al poder integrar sensores dentro de la plataforma a medida ASIC
Precio	20€	35€ (Raspberry Pi-3 B)	3€ (Cortex M0) – 7€ (Cortex M4)	Altamente influenciado por el volumen de unidades

2.3. Comparativa Plataformas Prototyping

Vamos a evaluar las plataformas de prototipos, desde el punto de vista de conectividad, procesamiento, S.O., ventajas y desventajas, escenario de uso y precio

Arduino

- Plataforma con conectividad a través de Ethernet RJ45 y Wifi (Arduino Yun)
- La capacidad de procesamiento y memoria es muy baja. Tiene un núcleo de 8 bits a 16 MHz y la memoria RAM es de 2 KB.
- Es una plataforma embebida pero sin sistema operativo. Vamos a poder tocar a muy bajo nivel pero algunas aplicaciones van a ser muy complicado programarlas por no tener S.O.
- Arduino es muy fácil de usar y hay una comunidad muy extensa para el soporte de dudas. Tiene la mayor compatibilidad con sensores y actuadores (trabaja a muy bajo nivel)
- Desventaja en base a la capacidad de procesamiento → tiene un bajo rendimiento.
- Se puede usar Arduino en aplicaciones sencillas y que requieren de captar información de sensores y controlar actuadores desde un punto de vista de cliente,
- No es apto para aplicaciones realtime por la poca capacidad de procesamiento
- El precio es estimado está en torno a los 20 euros por unidad de Arduino Uno

2.3. Comparativa Plataformas Prototyping

Raspberry

- Ofrece conectividad con Ethernet RJ45 y WiFi. También tiene puertos USB, HDMI para conectar a cualquier pantalla (mayor conectividad externa que Arduino).
- Dispone de una alta capacidad de procesamiento con 4 cores de 16 bit en la Raspberry Pi3B que funcionan con un reloj de 1,2 GHz y con una RAM de 1 GB.
- La Raspberry es una unidad embebida que dispone de S.O. Raspbian
- Ofrece conexión a Internet automática con DHCP activado en el router. Existe una comunidad muy extensa para soporte. Su capacidad de almacenamiento y procesamiento elevada habilita a muchos proyectos que requieren de esa capacidad.
- La Raspberry no es fácil integrar en un producto final porque tiene un tamaño elevado.
- El escenario de uso de la Raspberry Pi es en aplicaciones de IoT basadas en servidor ya que tiene altas capacidades de almacenamiento y procesamiento de la información y un S.O. basado en múltiples lenguajes de programación
- Es apto para aplicaciones Real-Time por su alta capacidad de procesamiento.
- Ronda los 35 euros para el modelo de Raspberry Pi3

2.3. Comparativa Plataformas Prototyping

ARM Cortex M series

- No es una plataforma embebida de prototipado como tal, sino que es una CPU muy utilizada en entornos IoT.
- En cuanto a la conectividad, nos permite conectarnos a través de WiFi, 2G, 3G, 4G
- Tiene una CPU de 32 bits. Dispone de una amplia gama para adaptarse a las necesidades y compensar rendimiento y consumo energético. La CPU va desde el modelo M0 (menor capacidad de procesamiento y menor consumo energético) hasta el M4 -180MHz (mayor rendimiento y mayor consumo energético). La CPU ha sido utilizada en más de 40 S.O.
- Es una CPU que no depende de los ciclos de producción de las plataformas embebidas como Arduino / Raspberry, por tanto acelera el despliegue de soluciones IoT
- No tiene la facilidad de uso, como lo tienen plataformas embebidas Arduino y Raspberry
- Es usado en dispositivos conectados que requieren de altas capacidades, reducido tamaño y no depender de ciclos de producción como los wearables
- Debido a que tiene un rendimiento medio elevado es apto para aplicaciones realtime.
- El precio de la CPU del cortex M0 rondaría los 3 euros y el corte M4 los 7 euros.

2.3. Comparativa Plataformas Prototyping

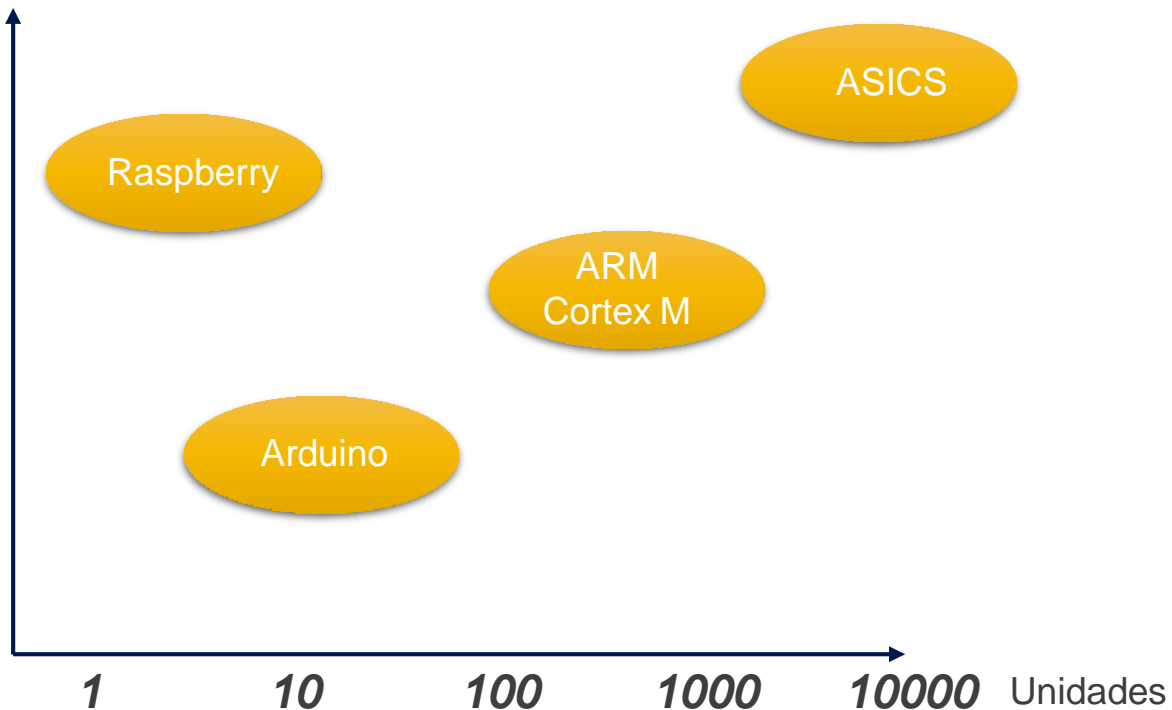
ASICS

- Lo anterior serían plataformas relativamente genéricas donde después podemos incorporar los diferentes elementos. En ASICS se hacen los dispositivos a medida para un escenario y una aplicación concreta.
- Tanto la conectividad como el procesamiento dependen justo de esa aplicación y de cómo solicitemos la creación y fabricación de esos ASICS
- Se puede fabricar a medida para optimizar la utilización con un S.O. concreto.
- Como estamos fabricando algo muy a medida, tendrá un coste muy bajo en caso de un volumen de necesidad muy elevado. Para pocas unidades el coste va a ser muy elevado.
- Escenarios con necesidad de aplicaciones a medida con un volumen muy elevado de unidades y en escenarios con necesidades de una alta eficiencia energética
- Muy usada en aplicaciones que requieren de un tamaño muy pequeño ya que como lo hacemos a medida, lo vamos a integrar junto con otros sensores, dentro de una thing
- No se puede dar una referencia respecto el precio puesto que va a estar muy influenciado tanto por la capacidad de este modelo así como el volumen de unidades que solicitemos.

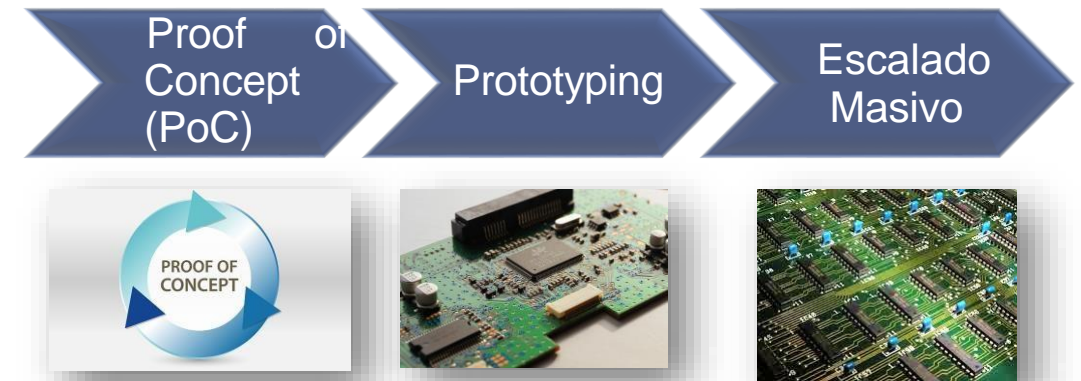
2.4 Selección Solución y Ciclo de Producción

¿Qué solución escoger para nuestro escenario?

Capacidades



¿Cuál es el Ciclo de Producción?



2.4 Selección Solución y Ciclo de Producción

Hemos visto las diferentes plataformas de prototipado y en qué consisten los sensores/actuadores. Vamos a ver ahora para una idea de negocio con IoT que tengamos cómo poderla enfocar y qué elementos deberíamos seleccionar.

¿Qué solución escogeremos para nuestro escenario en función de las capacidades y el número de unidades? Tenemos una grafica donde en el eje x tenemos las unidades de dispositivos IoT y en el eje y las capacidades que se requieren.

- ❖ **Raspberry** → Mejor solución cuando se requiera un nº bajo de unidades de dispositivos pero las capacidades requeridas en cuanto a procesamiento sean elevadas. Proyecto que está relativamente acotado con un número bajo de unidades, que podríamos hacer nosotros mismos, porque hay una una gran comunidad que nos podría dar soporte
- ❖ **Arduino** → Podríamos utilizarlo a partir de un nº relativamente superior. Recordad que las capacidades de procesamiento que tiene Arduino son muy bajas y es un punto clave a la hora de seleccionar entre la plataforma Raspberry la plataforma Arduino

2.4 Selección Solución y Ciclo de Producción

- ❖ **CPU Series M** → Si ya estamos en un número de unidades relativamente elevado en torno a 1000 y queremos unas capacidades relativamente elevadas, probablemente no hace falta tener ya las capacidades de la Raspberry pero sí que sean muy superiores a Arduino, pues podríamos optar por esta solución. Ya no depende de los ciclos de producción que tiene Raspberry-Arduino para acelerar así nuestras soluciones.
- ❖ **ASICS** → Solución si lo que queremos es conseguir economías de escala y además hacer un producto totalmente a medida. Opción más barata en cuanto a coste y además podríamos hacer nuestra solución IoT a medida..

2.4 Selección Solución y Ciclo de Producción

¿Cuál sería el ciclo de producción de nuestra solución o cuál sería el enfoque de cara a empezar a toda la fabricación y prototipado?

- 1) Lo recomendable primero es hacer una prueba de concepto (PoC), decir una vez intentar hacer alguna pequeña simulación para ver que es una buena solución aquello que estamos aportando.
- 2) Una vez que tengamos la prueba de concepto, pasaríamos al prototipado. En base a nuestros requisitos crearíamos la primera plataforma/dispositivos/things con todos los sensores/actuadores que hayamos seleccionado previamente. Haríamos un primer escenario con things reales
- 3) Cuando realmente hayamos hecho un testeo exhaustivo de esa solución, podremos pasar un escalado masivo donde solicitaremos la fabricación de un bloque muy elevado de ASICs, de cara a poder desplegar a gran escala nuestra solución de IoT

2.5 Cuestionario

Pregunta 1:

¿Cuál es la diferencia entre un sensor y un actuador?

- ☐ Un sensor capta la energía de interés y la convierte en energía eléctrica mientras que el actuador realiza el proceso inverso, en base a la energía eléctrica de entrada emite una energía de interés como luz, sonido,..
- ☐ Un sensor en base a la energía eléctrica de entrada emite una energía de interés como luz, sonido mientras que el actuador realiza el proceso inverso, capta la energía de interés y la convierte en energía eléctrica
- ☐ Son lo mismo, el nombre depende de la plataforma de prototipado que se utilice

2.5 Cuestionario

Pregunta 2:

¿Cuáles son las restricciones a tener en cuenta cuando elegimos una plataforma de prototipado?

- ☐ **Tamaño, consumo, funcionalidades, memoria, coste y seguridad**
- ☐ **Consumo, color, memoria, sensores integrados**
- ☐ **Tamaño, potencia, latencia, capacidad de la conectividad y consumo.**

2.5 Cuestionario

Pregunta 3:

¿Qué solución de prototipado seleccionarías si la solución de IoT utilizará más de 10000 dispositivos?

☐ **Raspberry**

☐ **ASICS**

☐ **Arduino**