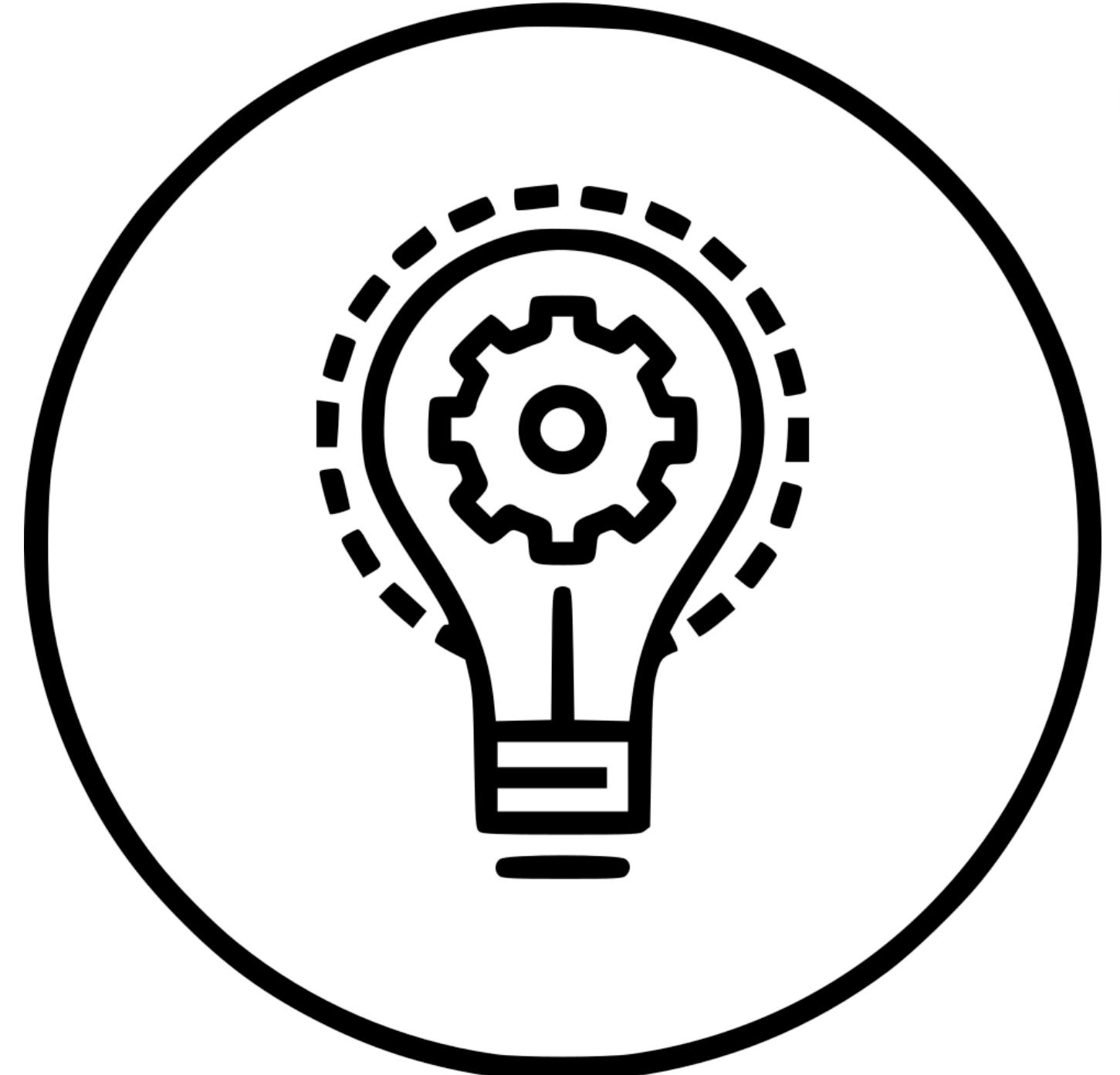
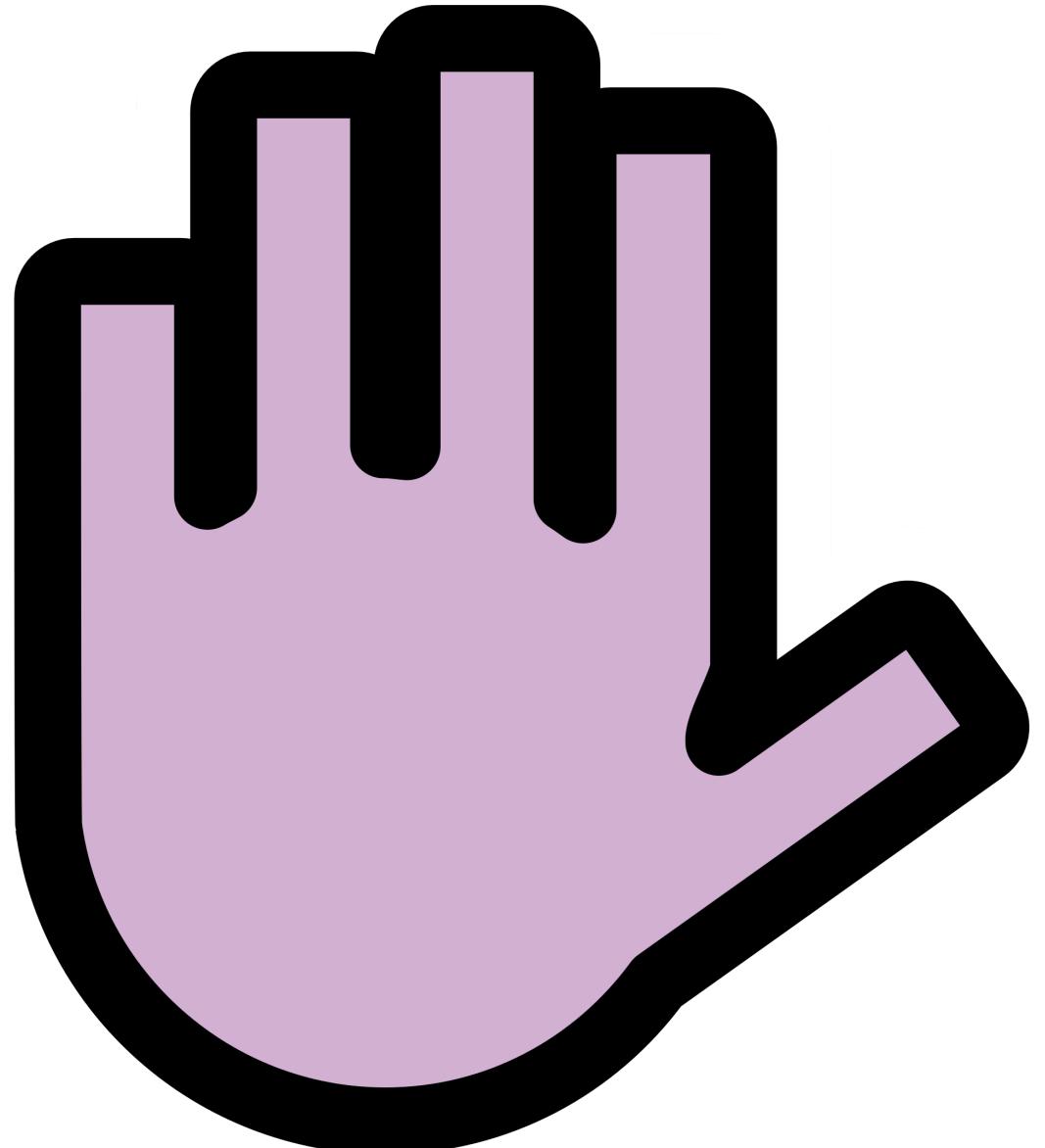


Más allá de Arduino



Interrupciones



Una interrupción es una función o fragmento de código que se ejecutará al ser "disparada" por un cambio de estado en algún pin que cuente con función de interrupción. El microcontrolador, dejará en "pausa" cualquier cosa que esté haciendo para ejecutar la función asociada con la interrupción, una vez ejecutada, el microcontrolador regresará a lo que se encontraba haciendo.(1)



Interrupciones



Las interrupciones se declaran en el ciclo `void setup(){}`

`attachInterrupt(digitalPinToInterrupt(pin), ISR, mode);`

Palabra reservada que indica la declaración de una interrupción.

Función que indica que el pin indicado será usado como interrupción

Función que se va a ejecutar cuando la interrupción se disparé.

El modo de disparo, puede ser LOW, CHANGE, RISING, FALLING, HIGH.

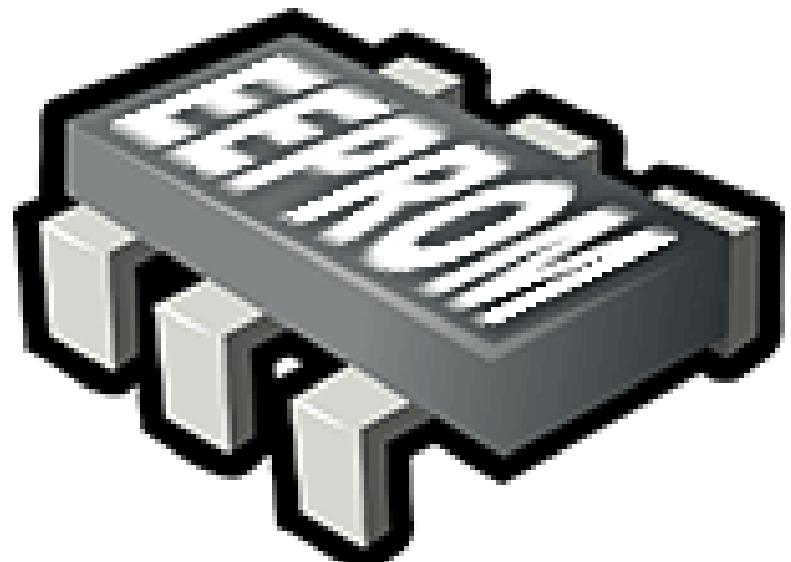
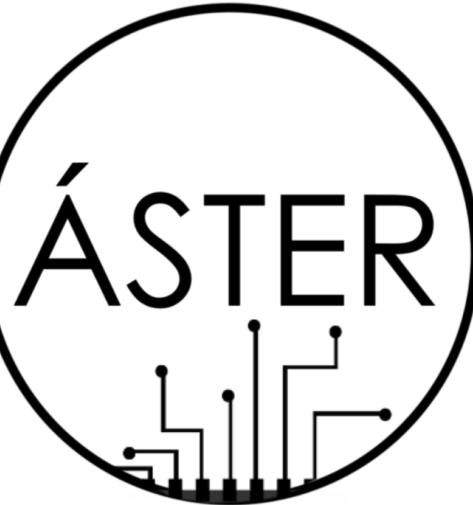
Usos de interrupciones



Las interrupciones se suelen emplear para tareas que requieren de alguna acción o atención instantánea, por ejemplo, para ejecutar alguna rutina en caso de alarma de algún sensor, para monitorear señales de entrada de alguna interfaz y no perder datos.



Memoria EEPROM



(Electrically Erasable Programmable Read-Only Memory). Esta memoria es un tipo de memoria no volátil, es decir la información que se encuentre dentro de este banco de memoria no se borrará una vez que el sistema de encuentre apagado o desenergizado. El ATmega328 tiene una memoria EEPROM de 1Kbyte.

(2)

Memoria EEPROM



```
#include <EEPROM.h>
```

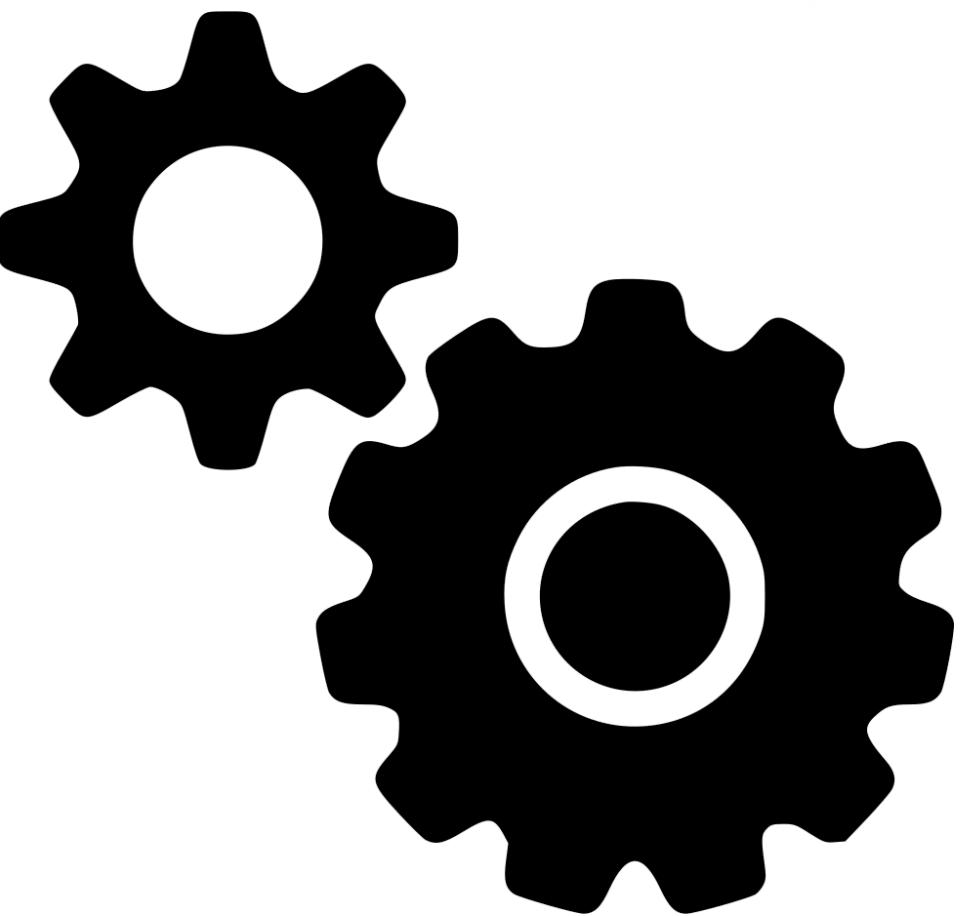
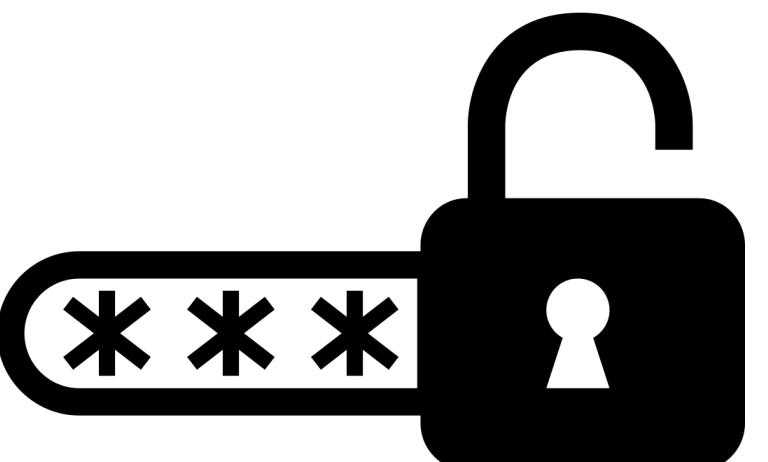
Functions

- `read()` }
 - `write()` }
 - `update()`
 - `get()` }
 - `put()` }
 - `EEPROM[]`
- Solo regresan datos tipo `byte`
- Regresan cualquier tipo de dato

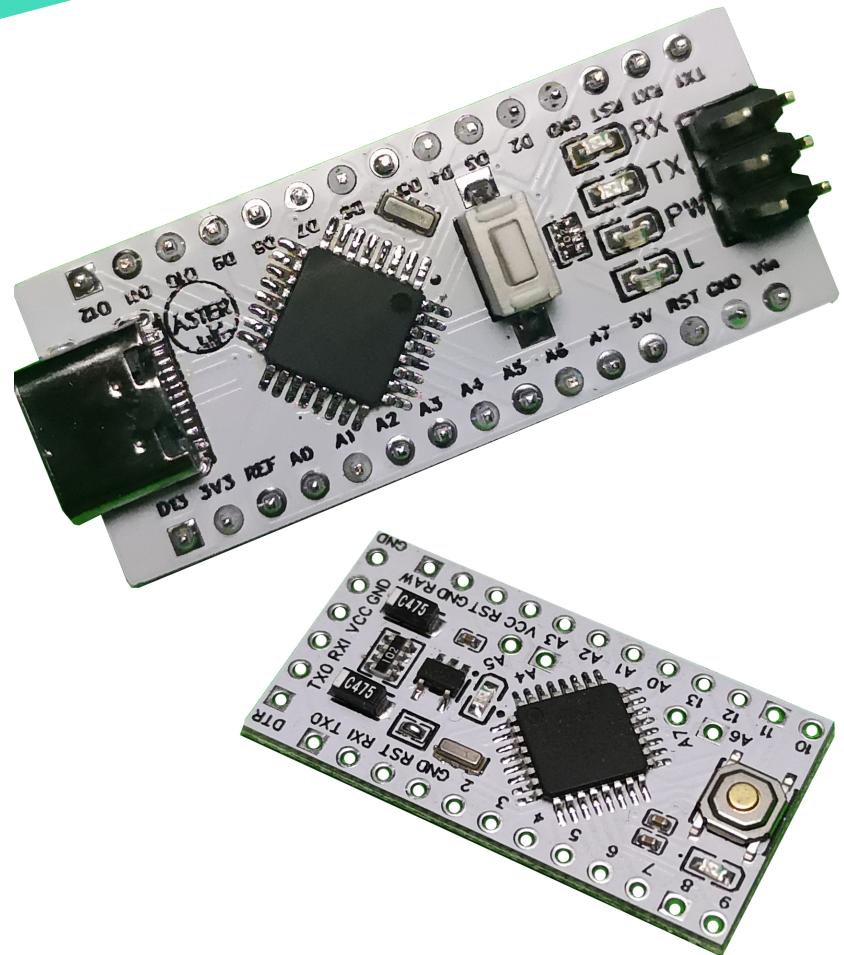
Uso de memoria EEPROM



Aprovechando la no volatilidad de la memoria EEPROM suele ser usada para almacenar datos o variables para la configuración establecida por un usuario en el sistema, o bien, para almacenar claves y contraseñas.

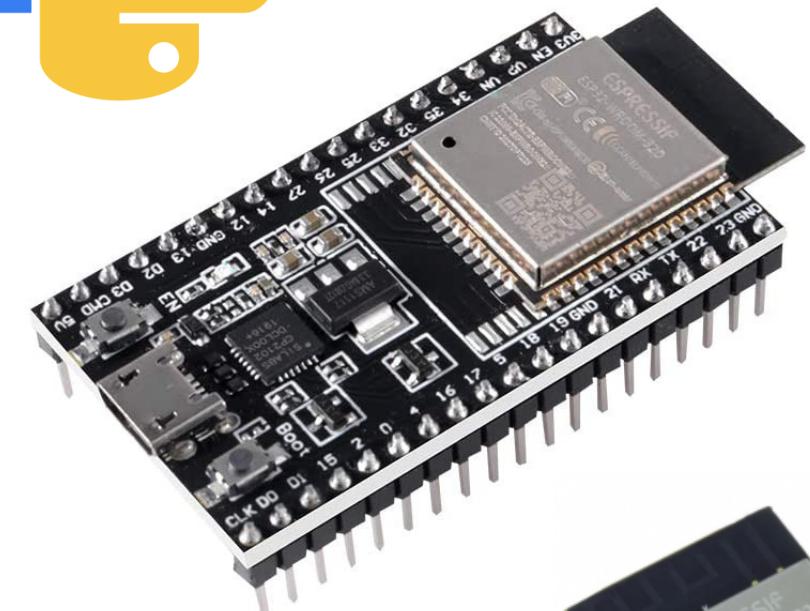
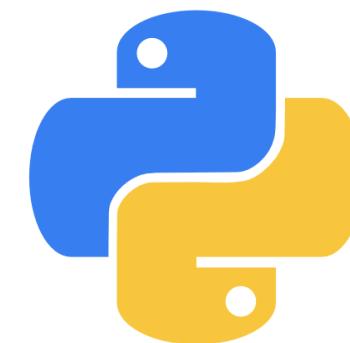


Tarjetas compatibles



Fiuino NANO y Pro Mini, son la versión "compacta de la versión UNO.

Raspberry Pico, el primer microcontrolador de la empresa. Tiene una velocidad de reloj de 133MHz y se programa nativamente con Python.



ESP32, este microcontrolador es ideal para domótica y IoT ya que incorpora funcionalidades como WiFi y BT sin necesidad de módulos externos.



Recomendaciones



bitwiseAr

<https://www.youtube.com/c/BitwiseAr>



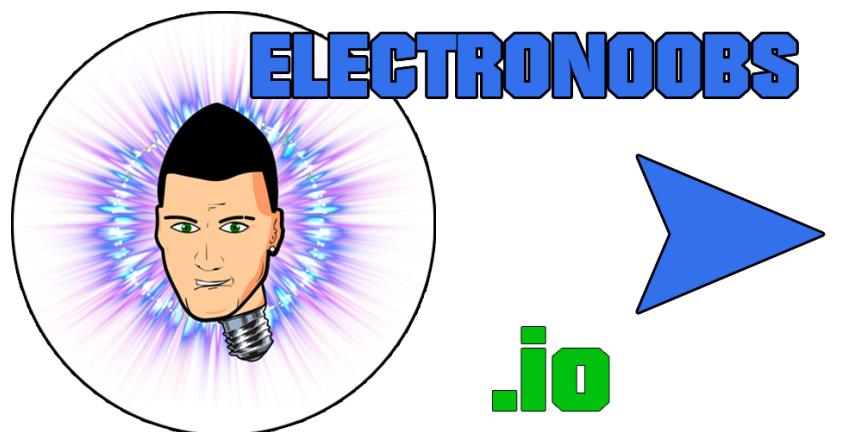
GreatScott!

<https://www.youtube.com/c/greatscottlab>



ELECTRONOOBS

<https://www.youtube.com/c/ELECTRONOOBSEnEspa%C3%B1ol>

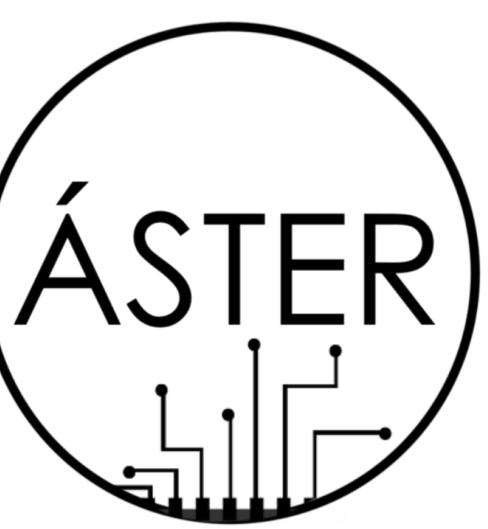


Felicidades!



Te deseamos el mejor de
los éxitos en todos tus
proyectos.(:

Fuentes



1. <https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/>
2. <https://www.arduino.cc/en/Reference/EEPROM>