

# Taller: Interrupciones

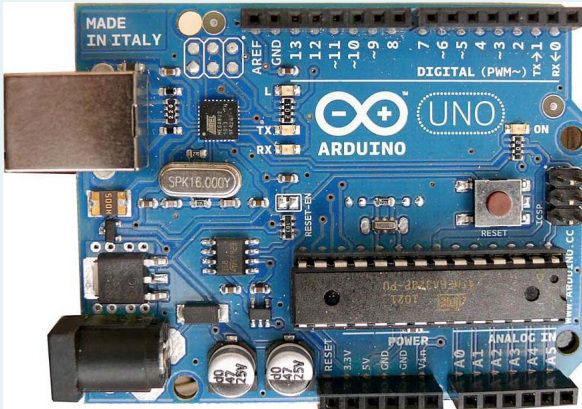
## Organización del Computador I

Organización del Computador I  
DC - UBA

1er. Cuatrimestre 2018

# Arduino UNO

## Esquema

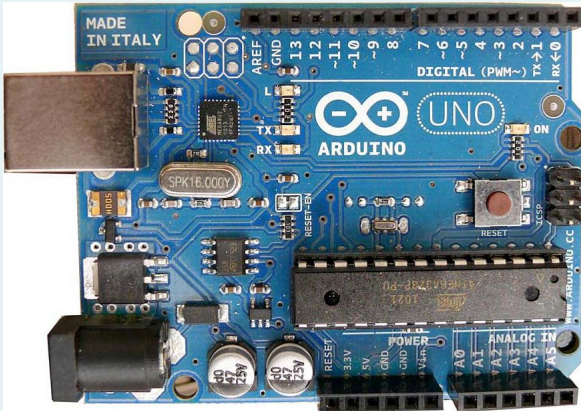


## ¿Qué es?

- Plataforma de cómputo
- Open Source
- Hardware Abstracted Wiring Language
- Programable por USB
- Popular
- Win-Mac-Linux
- Barato
- Todo en una sola placa
- Placa experimental. No para producción

# Arduino UNO

## Esquema

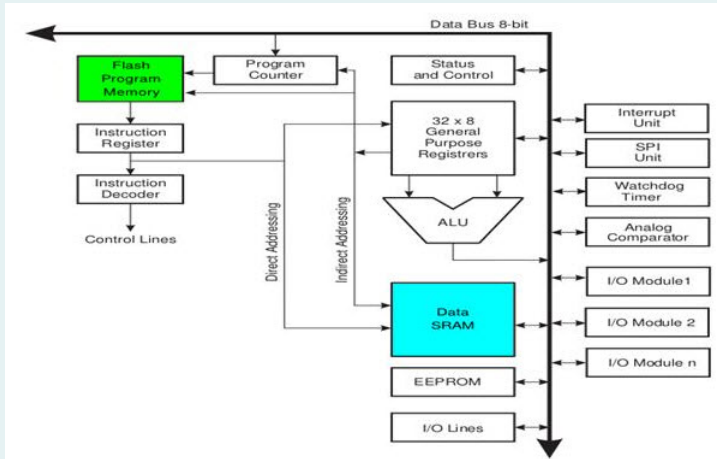


## Especificaciones

- Basado en ATmega328-8bits
- USB
- RISC, 16Mhz, 20 MIPS
- 2KB RAM !!
- 32 K Memory
- 1KB EEPROM
- 6 Ch 10 Bit A/D
- 13 Ch Digitales
- 3 timers
- 5 modos de ahorro de energía
- Arquitectura *Harvard*

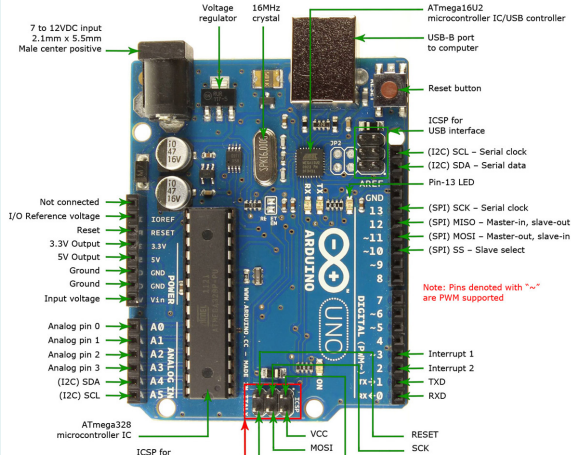
# Arduino UNO

## Arquitectura



# Esquema del Arduino UNO

# Esquema



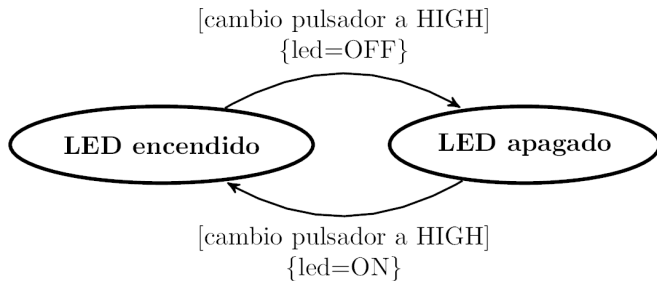
## ¿Qué podemos hacer?

- Digital IO  
(Leds, switches)
- Analog IO  
(resistive sensor data)
- Serial IO  
(Sensores, GPS, arduino, etc)
- Programable desde la Compu

- Crear un circuito que ilumine la pantalla (LED) del celular cuando se presione el botón (pulsador). Presionar el botón nuevamente debe apagar la pantalla.

- Crear un circuito que ilumine la pantalla (LED) del celular cuando se presione el botón (pulsador). Presionar el botón nuevamente debe apagar la pantalla.
- Sí, es una forma linda de decir que vamos a prender y apagar una luz.

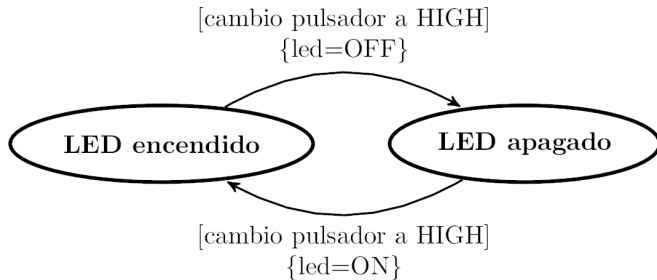
# Máquina de estados



- ¿Cómo se implementaría con interrupciones?

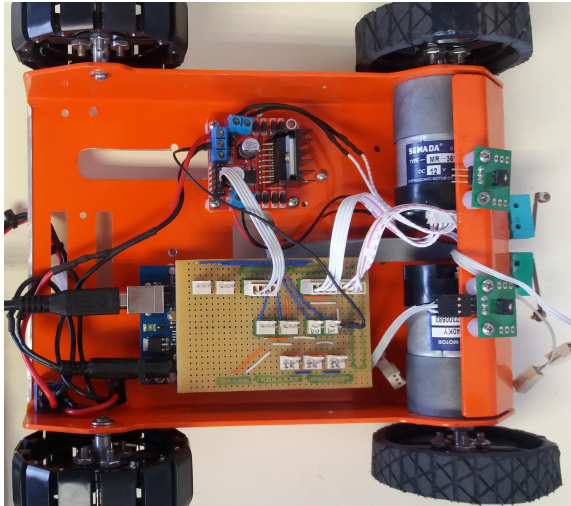


# Máquina de estados



- ¿Cómo se implementaría con interrupciones?
- ¿Y con polling?

# Taller !



# Comportamiento 1

Utilizando: el sensor de línea izquierdo y el sensor de choque.

Lograr el siguiente comportamiento:

- El robot deberá alternar entre: **Girando izquierda** y **Girando derecha** para poder seguir la línea. Se inicia en **Girando derecha** y se mantiene hasta que el sensor detecte un cambio (y produzca ———). En ese momento debe cambiar a **Girando izquierda** hasta otro cambio en el sensor. Y así sucesivamente.

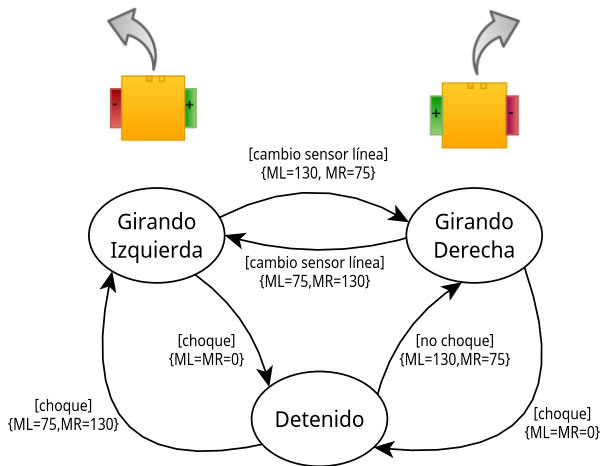
# Comportamiento 1

Utilizando: el sensor de línea izquierdo y el sensor de choque.

Lograr el siguiente comportamiento:

- El robot deberá alternar entre: **Girando izquierda** y **Girando derecha** para poder seguir la línea. Se inicia en **Girando derecha** y se mantiene hasta que el sensor detecte un cambio (y produzca ———). En ese momento debe cambiar a **Girando izquierda** hasta otro cambio en el sensor. Y así sucesivamente.
- El pulsador se comporta como interruptor **ON/OFF**, es decir, mientras el pulsador esté apretado el robot debe detenerse y al soltarse debe seguir en el estado que estaba.

# Máquina de estados – Comportamiento 1



## Comportamiento 2

Utilizando: dos sensores de línea y el sensor de choque. Lograr el siguiente comportamiento:

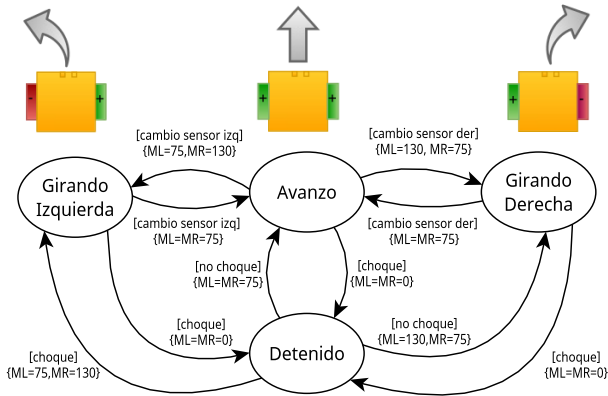
- El robot arrancará configurado en **Avanzando** y cuando detecte que se desvía de la línea deberá corregir su rumbo. Para ello, si alguno de los sensores empieza a sensor **negro**, se debe modificar girar hasta que el mismo sensor vuelve a sensor **blanco** (produciendo nuevamente una interrupción).

## Comportamiento 2

Utilizando: dos sensores de línea y el sensor de choque. Lograr el siguiente comportamiento:

- El robot arrancará configurado en **Avanzando** y cuando detecte que se desvía de la línea deberá corregir su rumbo. Para ello, si alguno de los sensores empieza a sensar **negro**, se debe modificar girar hasta que el mismo sensor vuelve a sensar **blanco** (produciendo nuevamente una interrupción).
- Cuando el pulsador es activado el robot debe configurarse como **Detenido** por 3 segundos y luego continuar con su comportamiento anterior. ¡No se puede utilizar la función `delay()`!, usar en cambio la función `millis()` que indica el paso del tiempo.

## Máquina de estados – Comportamiento 2





# ¿Preguntas?

