

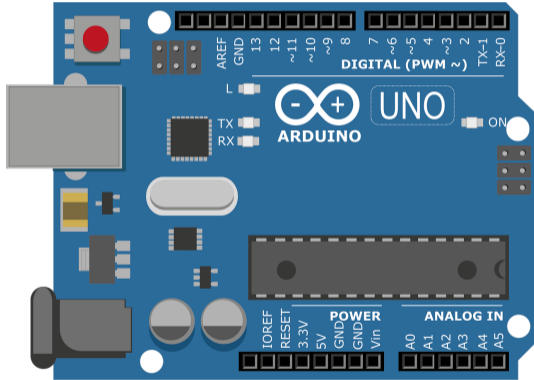
Arduino y protocolo de comunicación SPI

Laboratorio de electrónica - 2C19

13 de Noviembre de 2019



Arduino UNO: Características



Características

- ▶ ATmega328P
- ▶ 16 MHz
- ▶ 5 V
- ▶ 14 D I/O (6 PWM)
- ▶ 6 A I (10 bits)
- ▶ 20 mA por cada I/O
- ▶ Conexión USB
- ▶ Interfaz ICSP

Arduino UNO: Programación

```
void setup()
```

```
{
```

Es una función llamada al comienzo de la ejecución del programa. Inicializa variables, tipos de PIN, librerías, etc

```
}
```

```
void loop()
```

```
{
```

La función corre luego de `setup()` en un *loop* continuo respondiendo a las instrucciones programadas

```
}
```

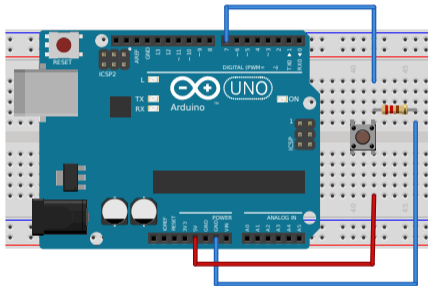
```
float myFunction(int IN1, ...)
```

```
{
```

Puedo definir funciones arbitrarias y llamarlas desde `loop()`

```
}
```

Arduino UNO: digitalWrite(),digitalRead()

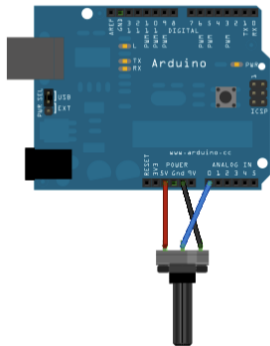


```
int ledPin = 13; // LED connected to digital pin 13
int inPin = 7;   // pushbutton connected to digital pin 7
int val = 0;    // variable to store the read value
```

```
void setup() {
  pinMode(ledPin, OUTPUT); // sets the digital pin 13 as output
  pinMode(inPin, INPUT);  // sets the digital pin 7 as input
}
```

```
void loop() {
  val = digitalRead(inPin); // read the input pin
  digitalWrite(ledPin, val); // sets the LED to the button's value
}
```

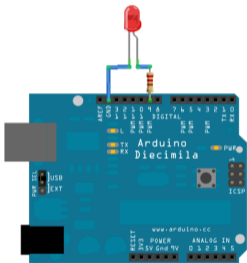
Arduino UNO: analogRead()



```
void setup() {  
  // initialize serial communication at 9600 bits per second:  
  Serial.begin(9600);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  // read the input on analog pin 0:  
  int sensorValue = analogRead(A0);  
  // print out the value you read:  
  Serial.println(sensorValue);  
  delay(1);      // delay in between reads for stability  
}
```

- ▶ número de pines analógicos: A0 a A5 (10 kS/s)
- ▶ comunicación serie exclusiva para la PC

Arduino UNO: analogWrite()



```
int ledPin = 9;    // LED connected to digital pin 9

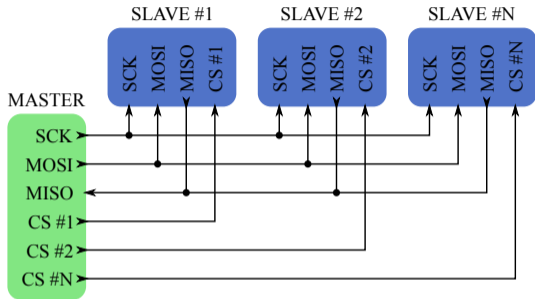
void setup() {
  // nothing happens in setup
}

void loop() {
  // fade in from min to max in increments of 5 points:
  for (int fadeValue = 0 ; fadeValue <= 255; fadeValue += 5) {
    // sets the value (range from 0 to 255):
    analogWrite(ledPin, fadeValue);
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
  }

  // fade out from max to min in increments of 5 points:
  for (int fadeValue = 255 ; fadeValue >= 0; fadeValue -= 5) {
    // sets the value (range from 0 to 255):
    analogWrite(ledPin, fadeValue);
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
  }
}
```

- ▶ pines PWM: 3, 5, 6, 9, 10, 11 @490 Hz (pins 5 and 6: @980 Hz)

Serial Peripheral Interface (SPI)



SCK *Serial Clock*
MOSI *Master Output Slave Input*
MISO *Master Input Slave Output*
SS *Slave Select*

Ventajas

- ▶ Más rápido que otros protocolos seriales *async*
- ▶ Un shift register puede recibir los datos
- ▶ Soporta varios *slaves*

Desventajas

- ▶ Requiere más cables
- ▶ El *master* es el único que controla la comunicación
- ▶ Un SS por cada *slave*

Serial Peripheral Interface (SPI)

El estándar SPI es flexible

- ▶ Permite determinar la velocidad
- ▶ ¿MSB o LSB primero?
- ▶ *Clock*: ¿espera abajo o arriba?
- ▶ *Clock*: ¿ascendente o descendente?

Hay que mirar las especificaciones del componente para escribir el programa de comunicación

```
SPI.beginTransaction(SPISettings(14000000, MSBFIRST, SPI_MODE0));  
(ver Arduino → SPISettings)
```


Serial Peripheral Interface (SPI)

Conexiones

Arduino	MOSI	MISO	SCK	SS
Uno	11	12	13	10
Mega	51	50	52	53

ICSP Mismas conexiones, accesible para los módulos

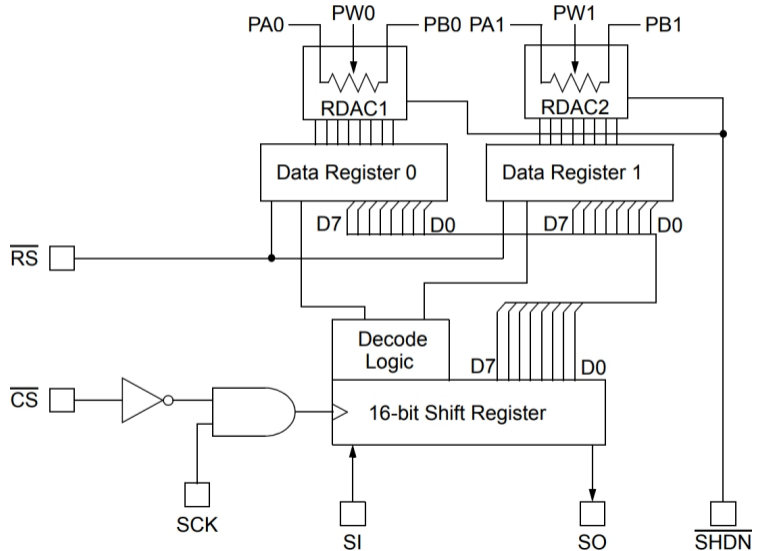
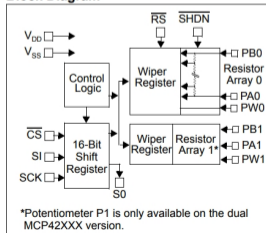


MCP41xxx: Potenciómetro digital

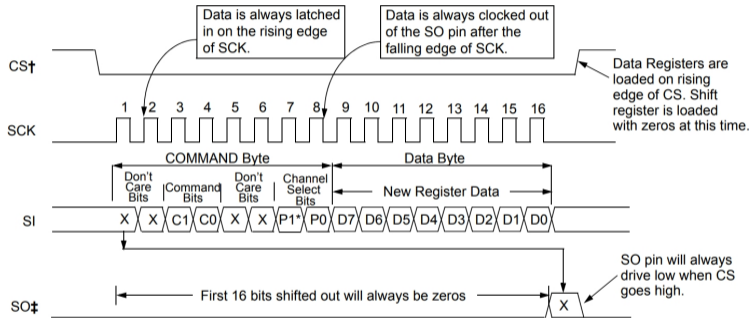
Features

- 256 taps for each potentiometer
- Potentiometer values for 10 k Ω , 50 k Ω and 100 k Ω
- Single and dual versions
- SPI™ serial interface (mode 0,0 and 1,1)
- ± 1 LSB max INL & DNL
- Low power CMOS technology
- 1 μ A maximum supply current in static operation
- Multiple devices can be daisy-chained together (MCP42XXX only)
- Shutdown feature open circuits of all resistors for maximum power savings
- Hardware shutdown pin available on MCP42XXX only
- Single supply operation (2.7V - 5.5V)
- Industrial temperature range: -40°C to +85°C
- Extended temperature range: -40°C to +125°C

Block Diagram



MCP41xxx: Potenciómetro digital

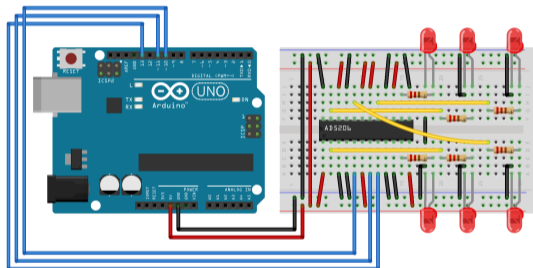


† There must always be multiples of 16 clocks while CS is low or commands will abort.

‡ The serial data out pin (SO) is only available on the MCP42XXX device.

* P1 is a 'don't care' bit for the MCP41XXX.

AD5206: Ejemplo

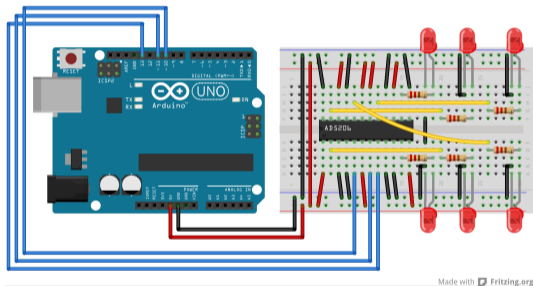


Made with  Fritzing.org

digitalPotWrite()

```
void digitalPotWrite(int address, int value) {  
  seleccionarChip();  
  esperar();  
  transferirAddress();  
  transferirValue();  
  esperar();  
  soltarChip();  
}
```

AD5206: Ejemplo



setup()

```
#include <SPI.h>

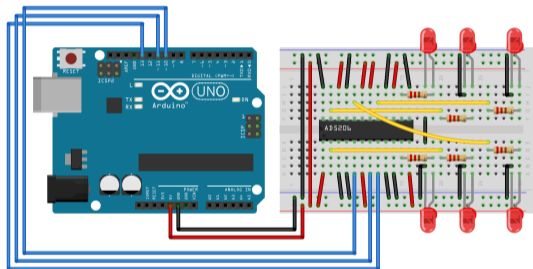
const int slaveSelectPin = 10;

void setup() {

    pinMode(slaveSelectPin, OUTPUT); // pin 10 es salida

    SPI.begin(); // inicializa SPI
}
```

AD5206: Ejemplo



Made with  Fritzing.org

loop()

```
void loop() {  
  digitalPotWrite(address, value); // fijo una posición del wiper  
  esperar;  
}
```