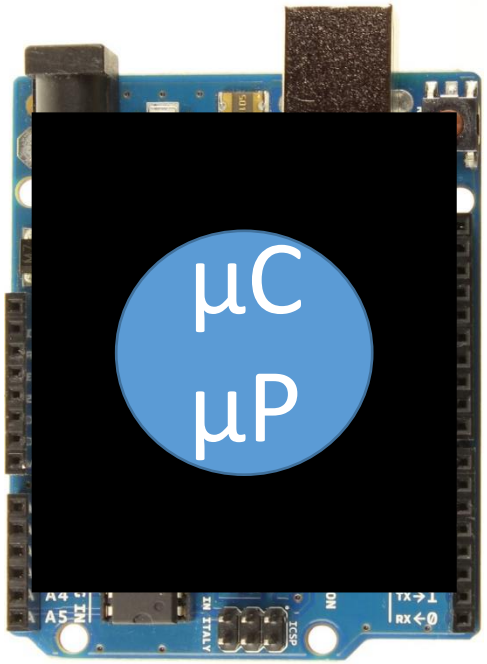
CURSO

ELECTRONICA - ROBOTICA – ARDUINO

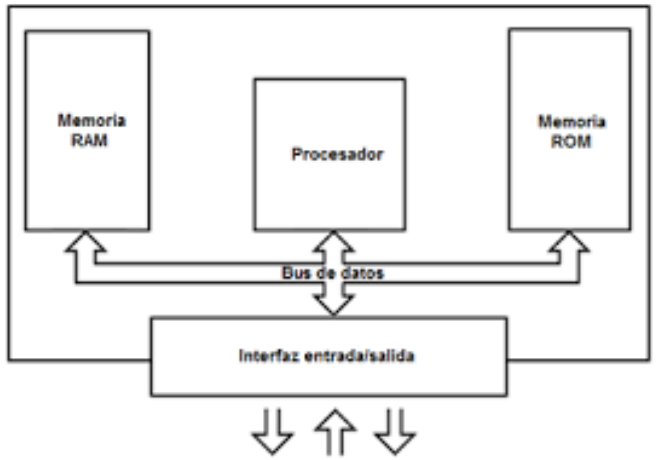
Ing. Brain Naser Soto

MICROCONTROLADOR

ENTRADAS
Periféricos de Entrada
SENSORES:
LUZ
SONIDO
TEMPERATURA
HUMEDAD
VIBRACIONES
ETC.



SALIDAS
Periféricos de Salida
ACTUADORES:
BUZZER
LUCES
INTERRUPTORES
MOTORES
PANTALLAS LCD
BLUETOOTH
ETC.

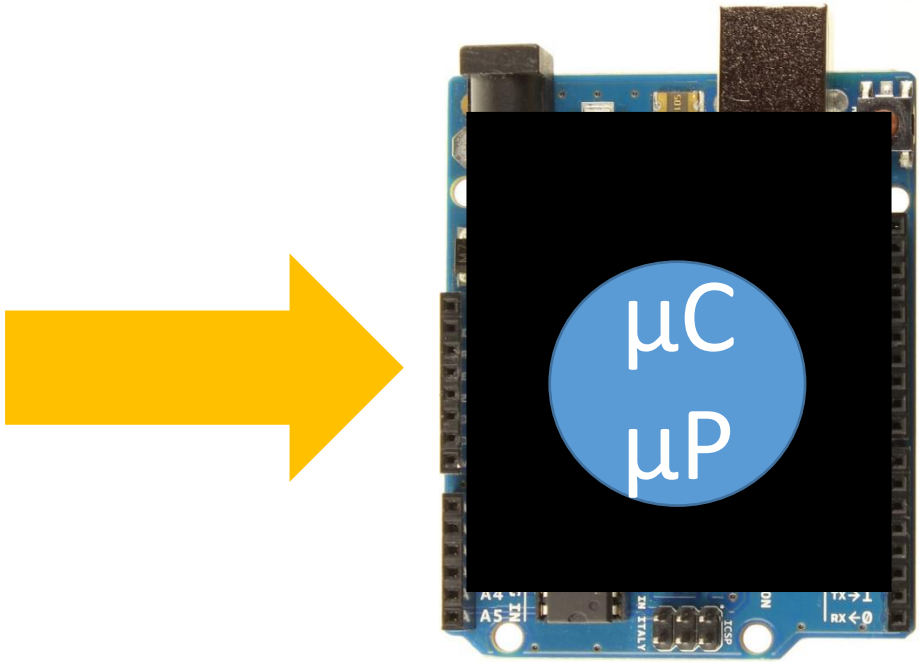


CARACTERÍSTICAS PRINCIPALES ARDUINO:

1.- Qué es Arduino (Concepto)
Hardware
Software
Comunidad (enorme).
2.- Tiene puertos Analógicos y Digitales
3.- Tiene Entorno de Desarrollo (IDE-Multi Plataforma)
4.- Es económico y Flexible
5.- Código fácilmente modificable y actualizado.
6.- Existe mucho Hardware y Shield´s para Arduino
7.- Es Serial, Secuencial.
8.- Trabaja a 16MHz (UNO R3). (16 millones de Ciclos en 1 segundo)

MICROCONTROLADOR

INSTRUCCIONES



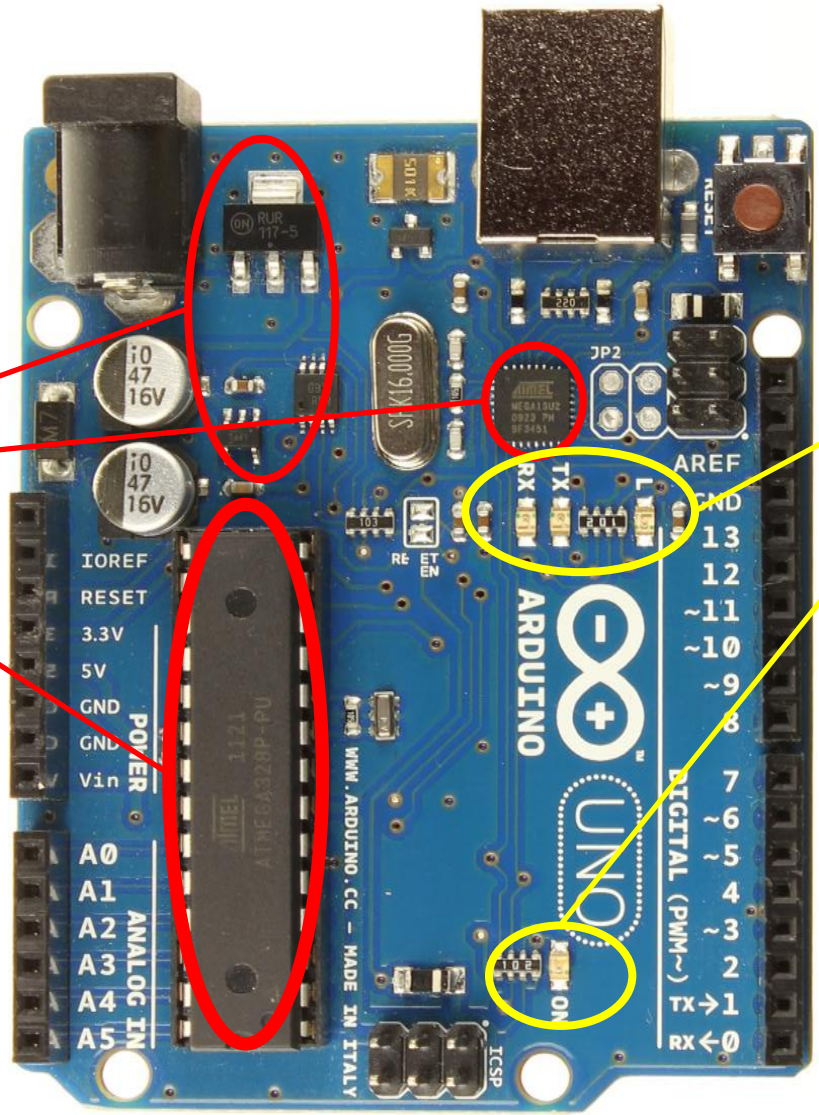
RESULTADOS

CLOCK

ELEMENTOS DE ARDUINO UNO-2

CHIP´s

Circuitos Integrados que componen un Arduino UNO; Regulador, FTDI y uC



LED´s

Monitoreo visual de estado del Arduino Uno.

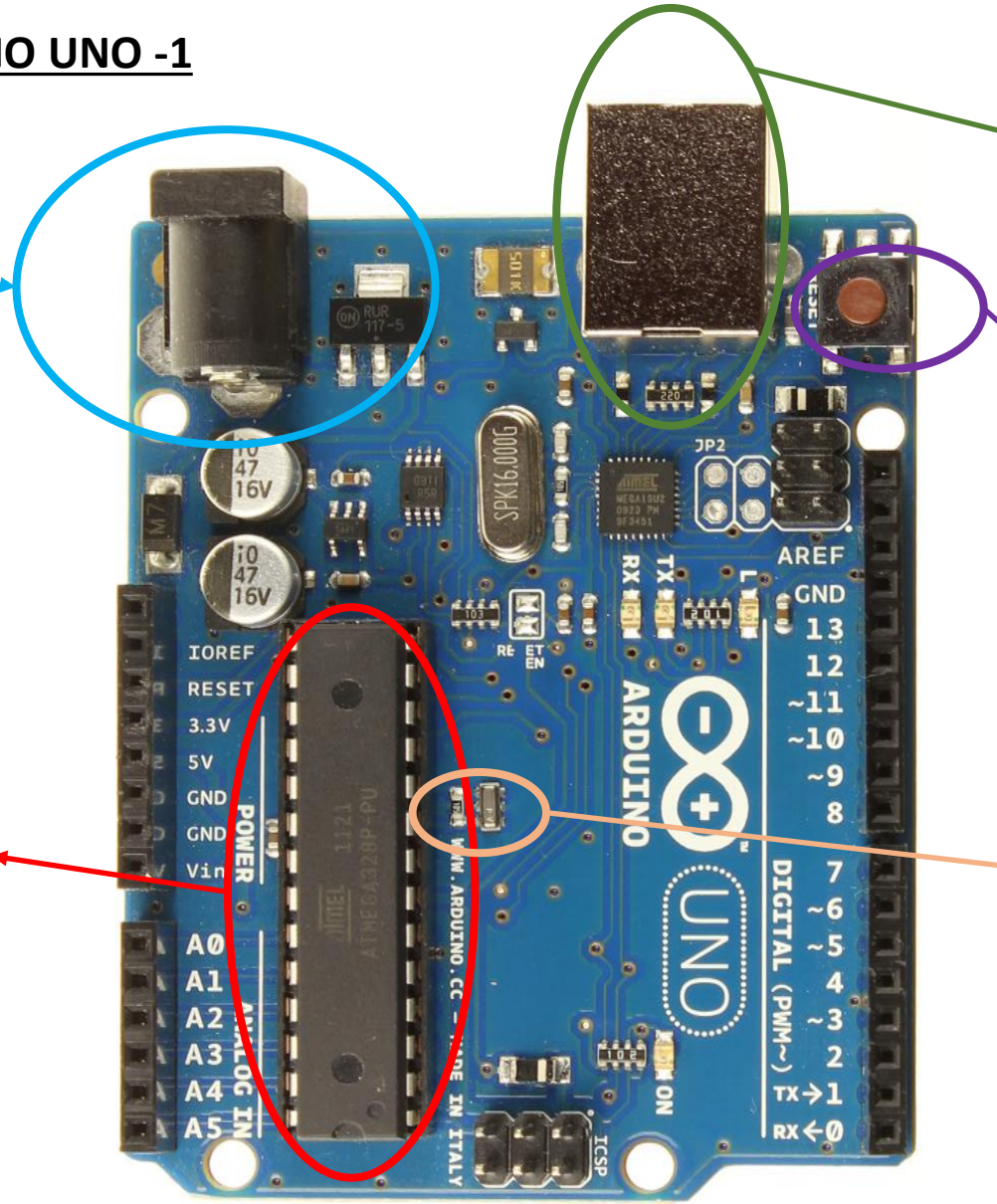
ELEMENTOS PRINCIPALES DE ARDUINO UNO -1

CONECTOR DC

Módulo de energía para la board.

USB

Interface de Comunicación Serial con un PC



RESET

Permite REINICIAR el uC y el código. Pull-Up.

UC

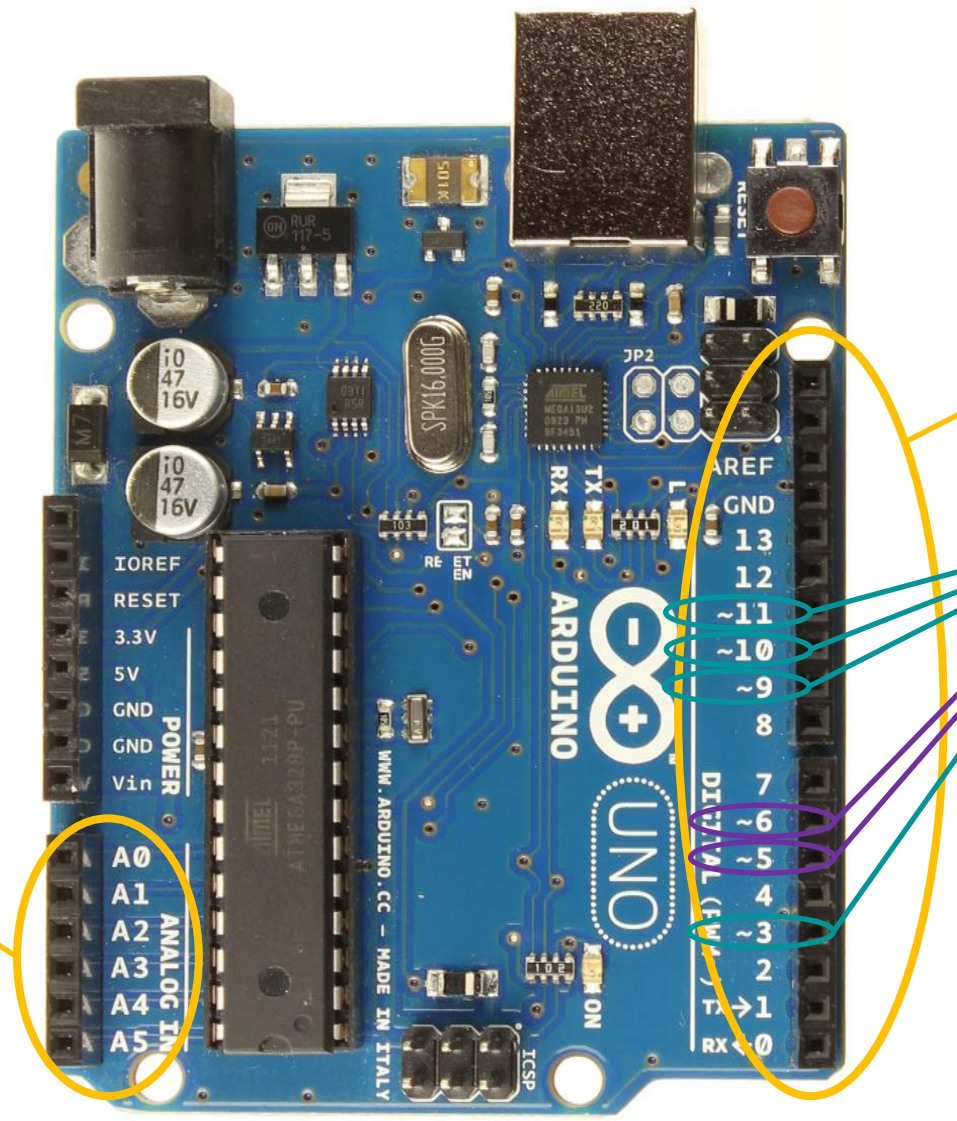
MicroControlador Atmega 328P

OSCILADOR

Reloj a 16MHz. Basado en Crystal.

ELEMENTOS DE ARDUINO UNO-3

Puertos de Entradas y Salidas.



PINES I/O

14 Puertos de Entrada/Salida Digitales. Por defecto son Entradas.

PINES PWM

6 Puertos de Salida PWM marcados con **~**

490 Hz.

980 Hz.

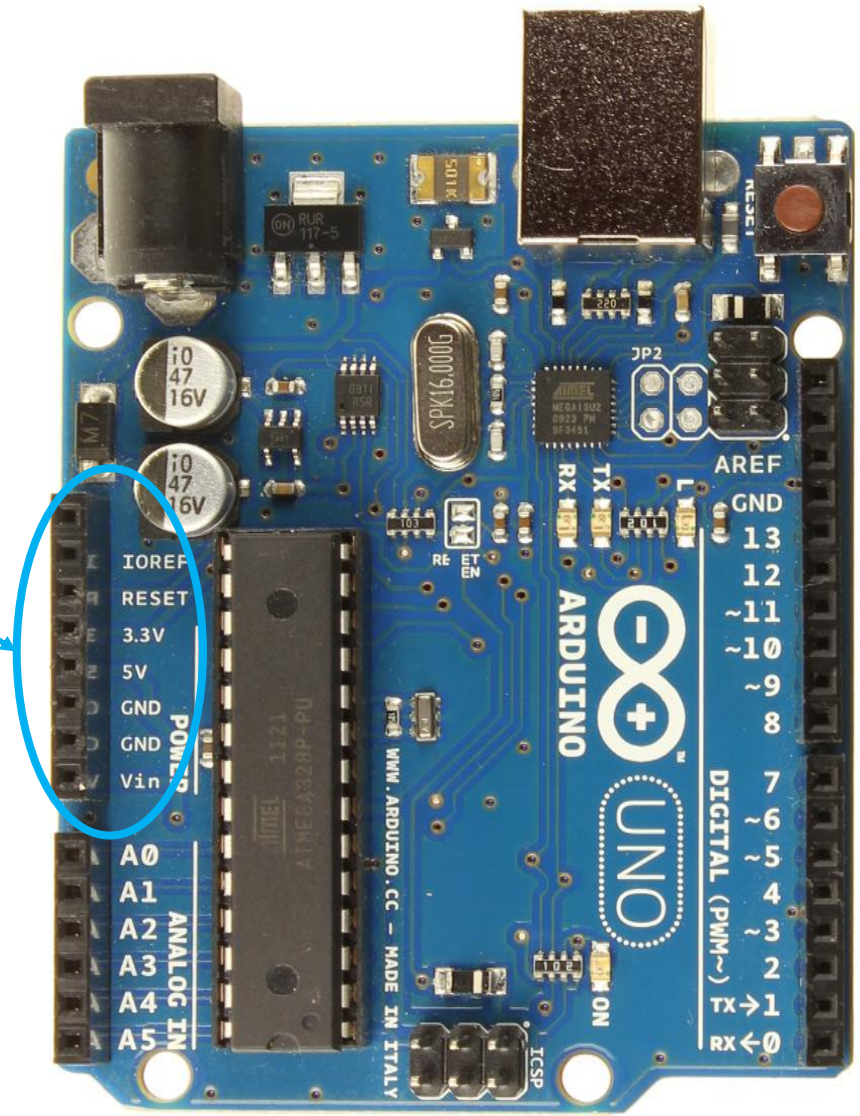
PINES I/O

6 Puertos de Entradas Analógicas, ADC a 10 bit (1024). También se pueden usar como I/O Digitales; por defecto son Entradas Analógicas

ELEMENTOS DE ARDUINO UNO-5

PINES VOLTAJES

Pines de Energía,
Referencias y Reset.

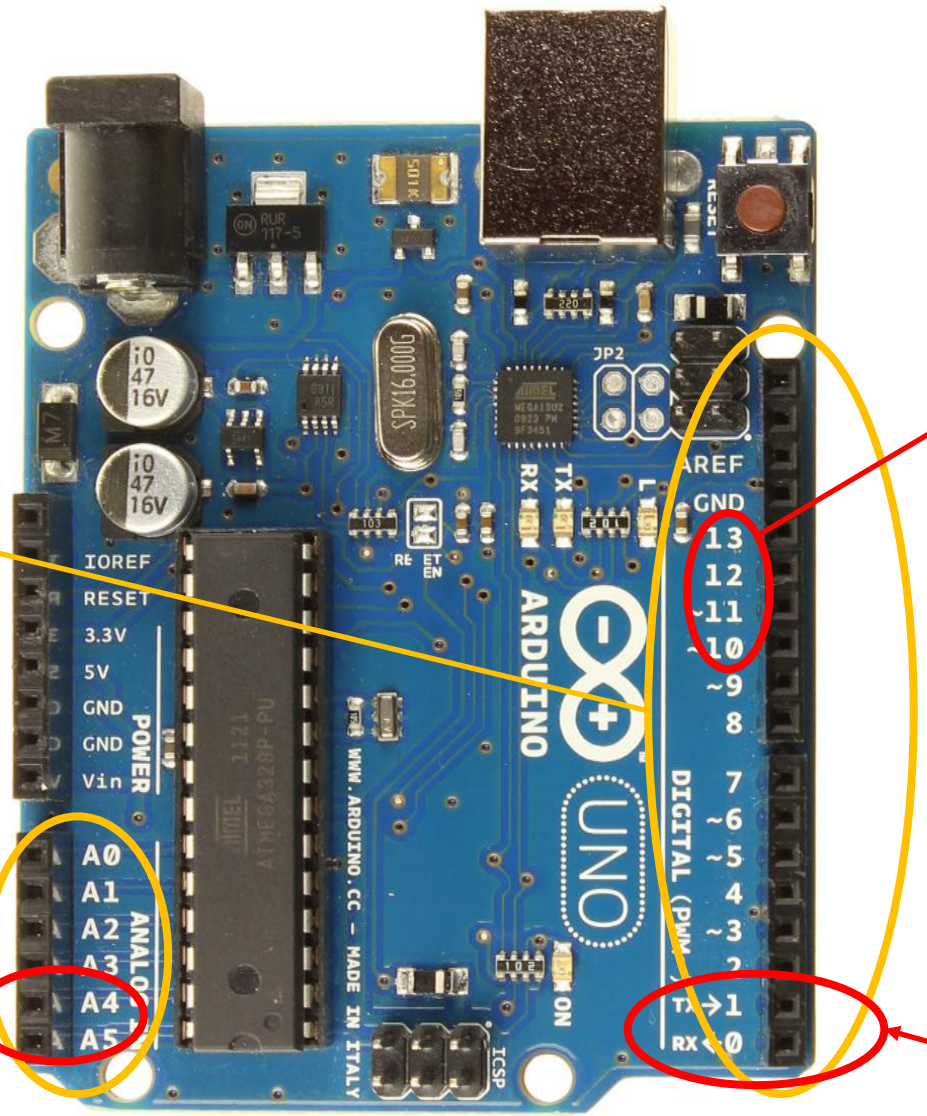


ELEMENTOS DE ARDUINO |UNO-4

PINES I2C

PINES I/O

Puertos o “PINES” de Entrada y Salida para conexión con el mundo físico.



PINES SPI

MOSI,MISO,
CLK, SS

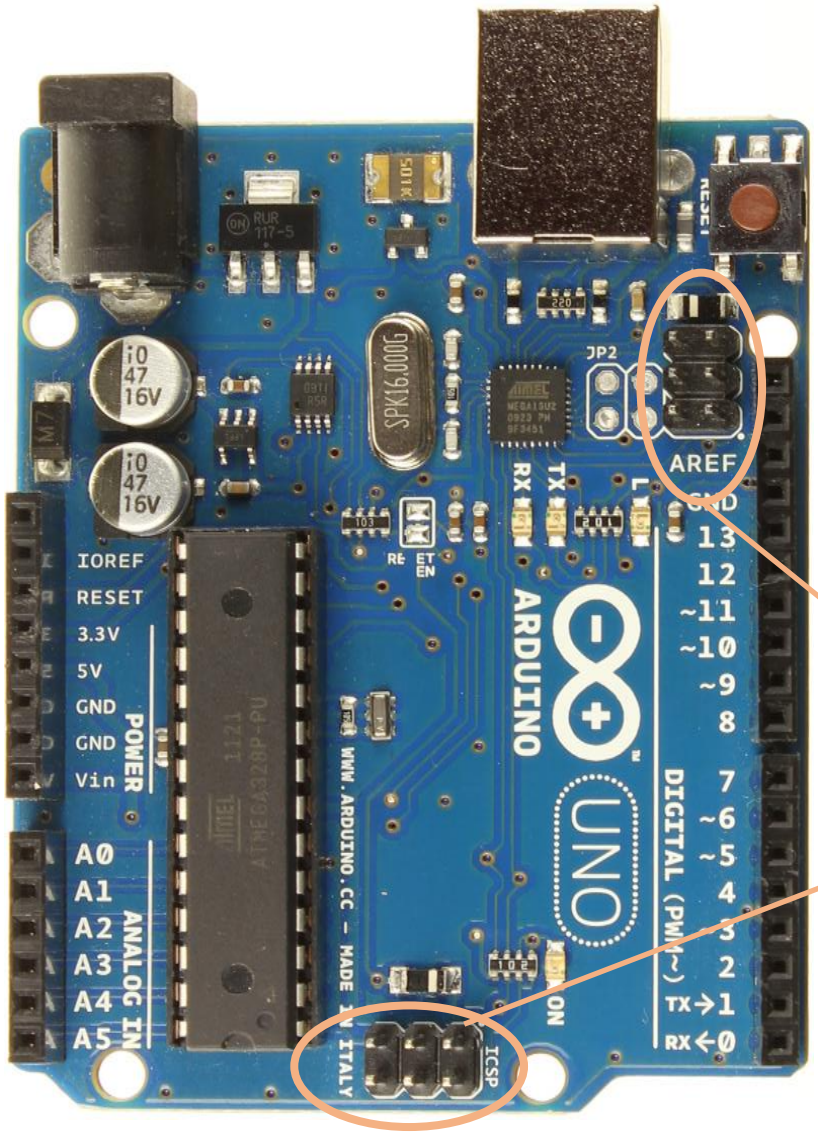
PINES I²C

SDA (A4) y SCL (A5)

PINES UART

Compartido con FTDI.
Future Technology Devices International.

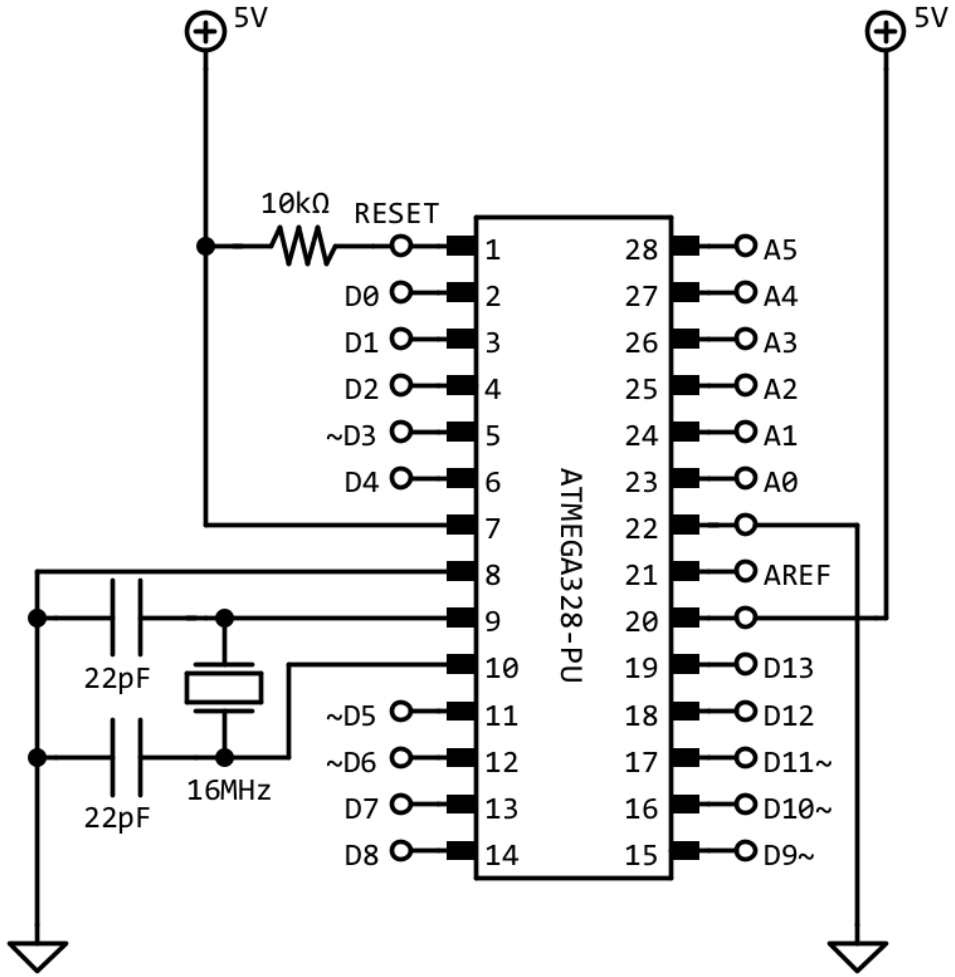
ELEMENTOS DE ARDUINO UNO -1

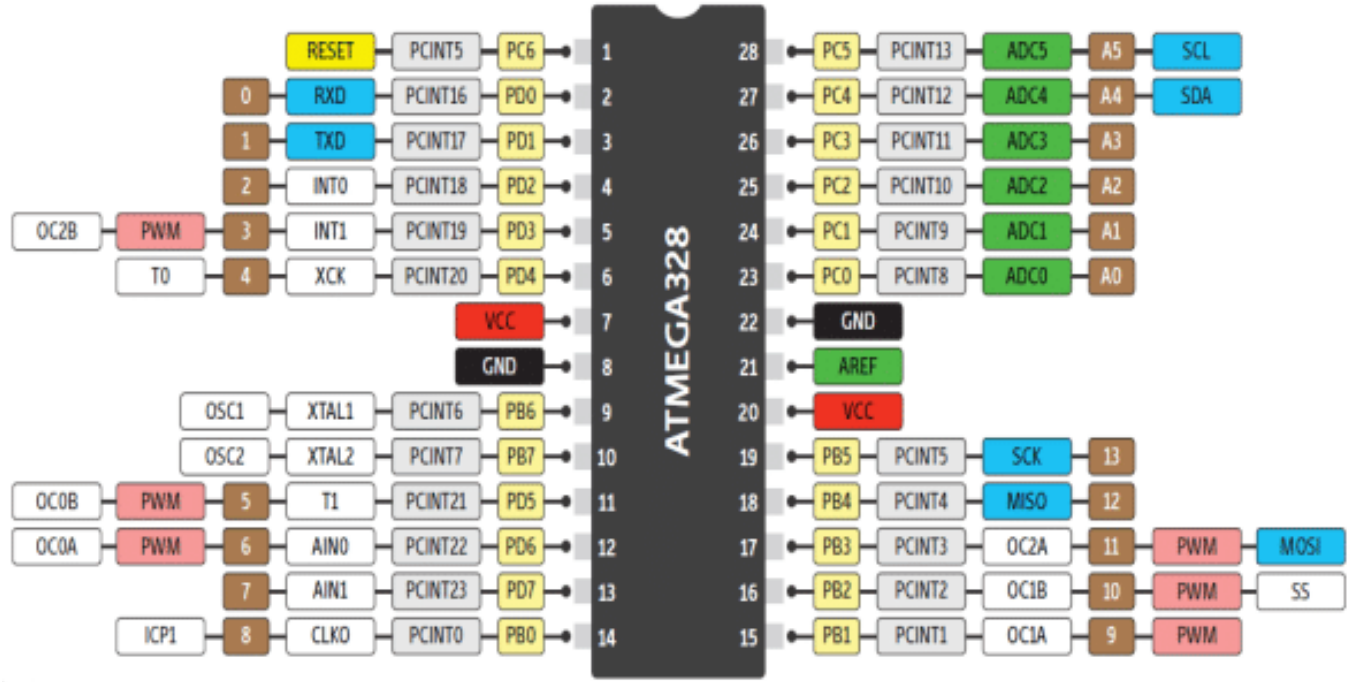


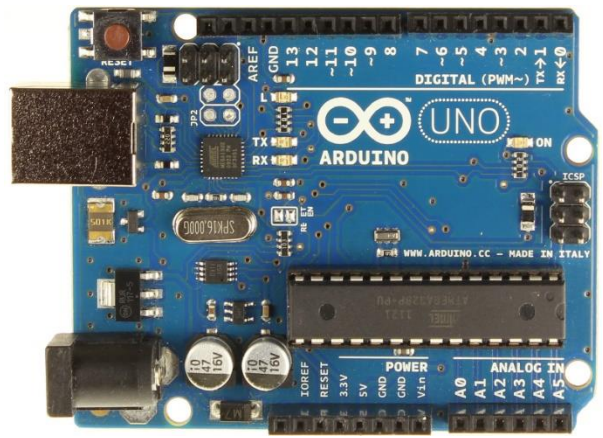
ICSP COM. SERIAL - SPI

Puertos de Comunicación Serial con periféricos externos.
"In Circuit Serial Programming" (Programación Serial En Circuito).

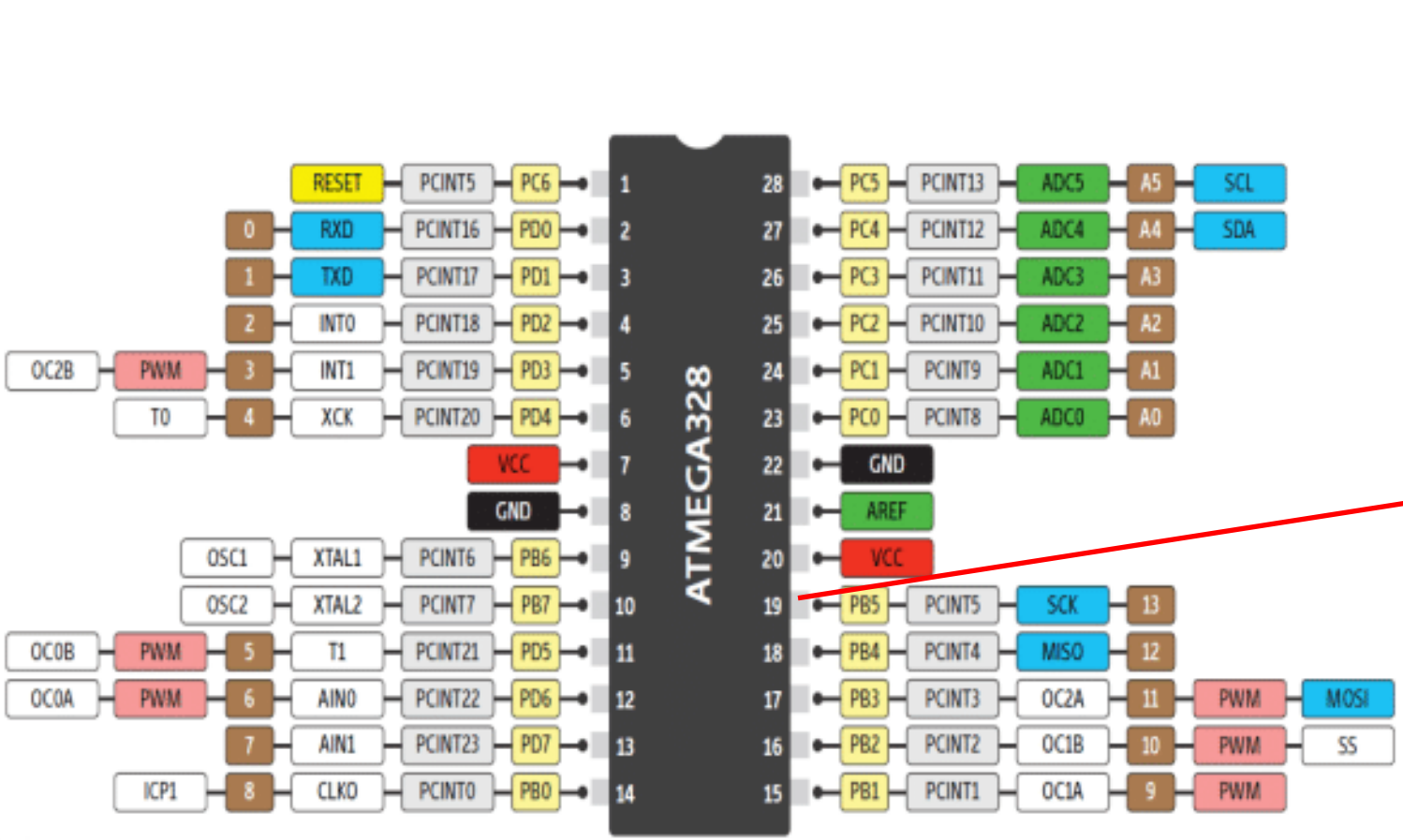
Arduino y ATMEGA 328P

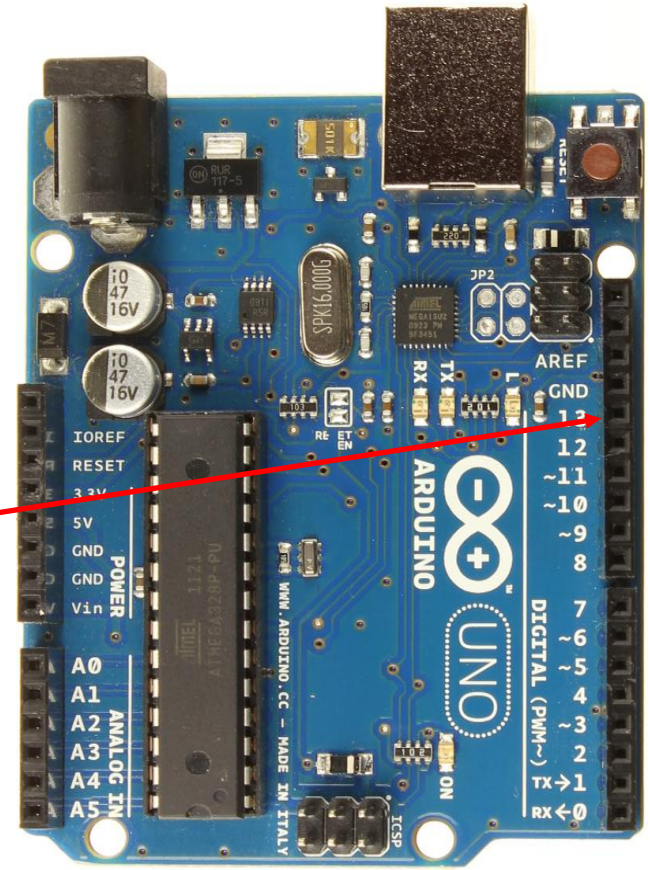




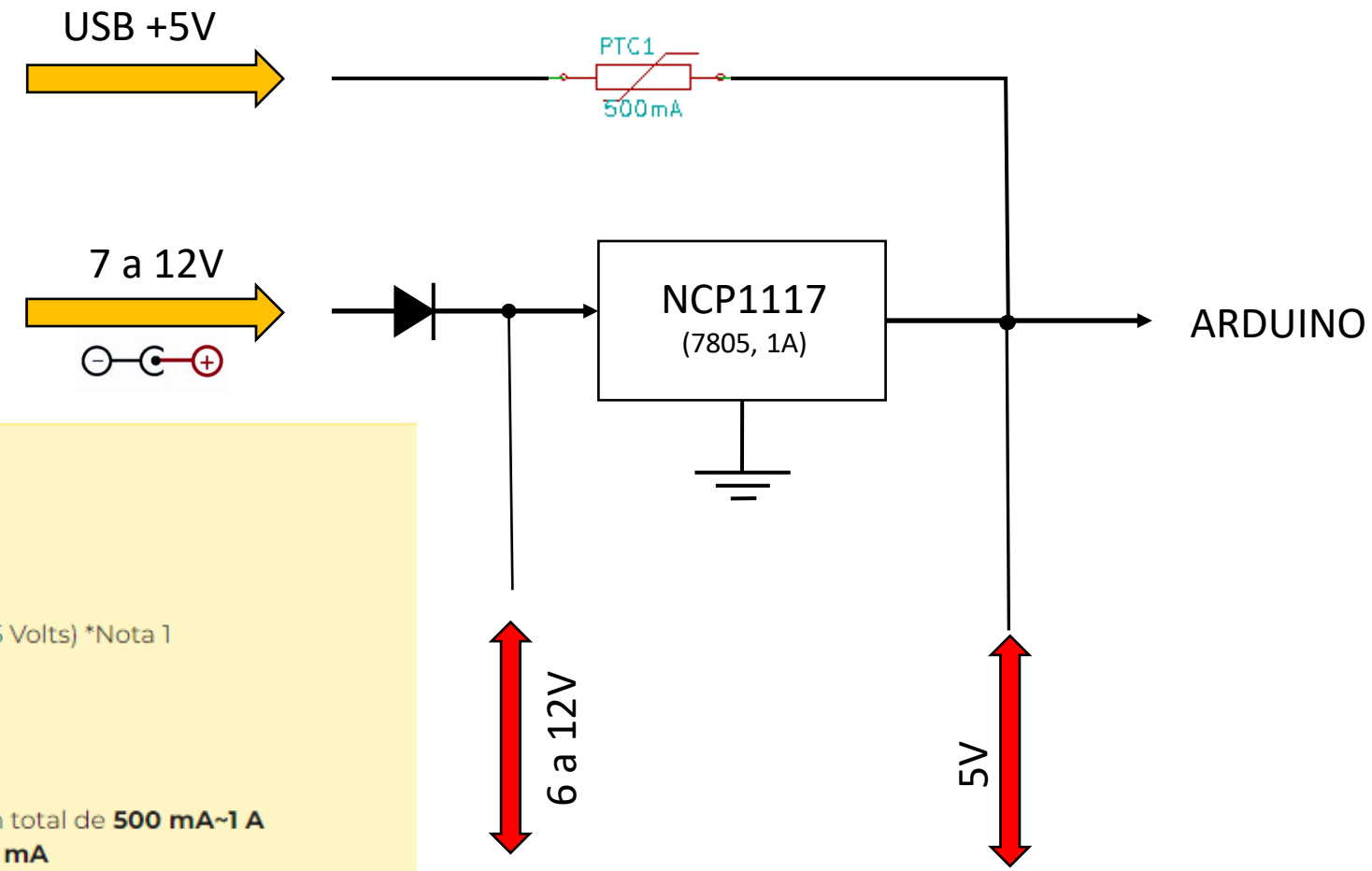


PINOUT ATMEGA 328





4 FORMAS DE ALIMENTAR ARDUINO UNO R3



Limites de voltaje de entrada

- **7~12 V** recomendado
- **6~20 V** limite absoluto
- Pines Entrada/Salida (E/S): **-0.5 V a +5.5 V**

(el máximo real es Vcc + 0.5 V para un arduino de 5 Volts) *Nota 1

Limites de corriente de salida:

- Si es alimentado por USB: un total de **500 mA**
- Si es alimentado por fuente externa o batería: un total de **500 mA~1 A**
- Máximo individual por pin de Entrada/Salida: **40 mA**
- Suma de todas las Entradas/Salidas combinadas

(SIN incluir el pin de "5V"): **200 mA**

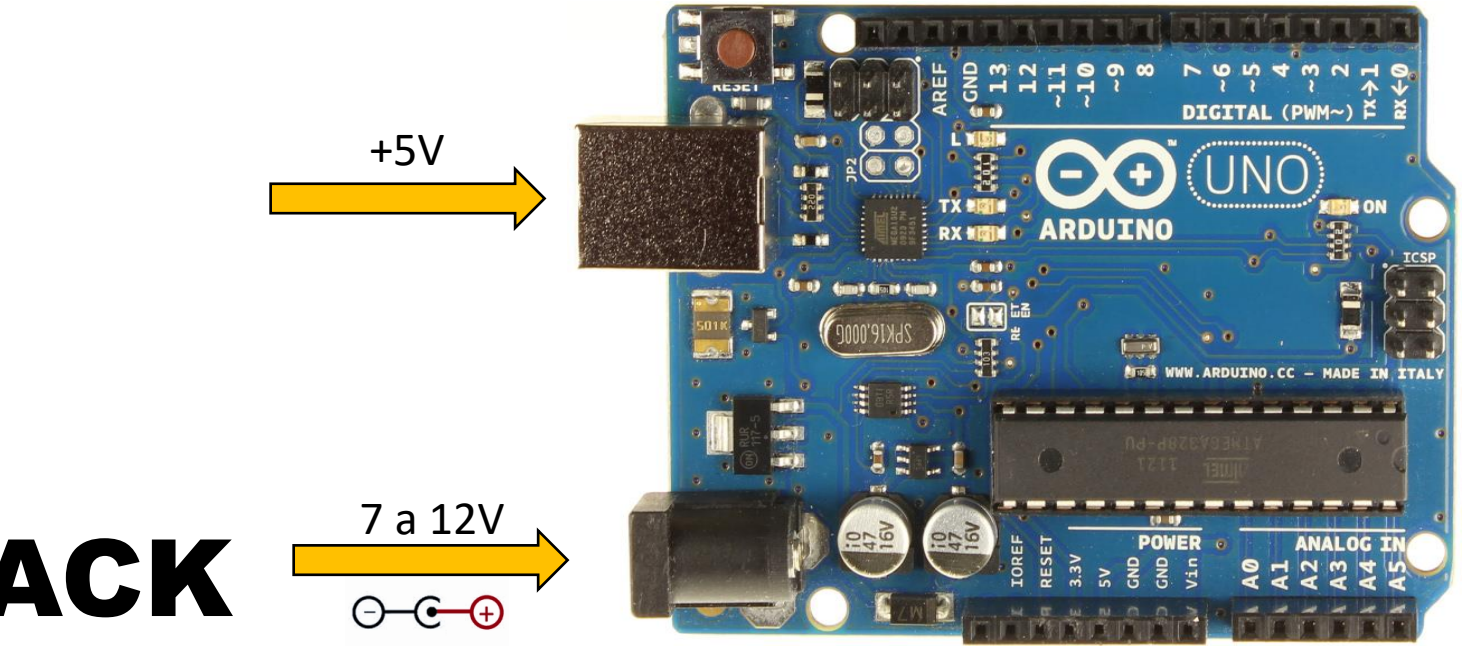
https://www.aladuino.com.mx/blog/limites-de-voltaje-corriente-y-alimentacion-del-arduino/

USB

+5V

DC JACK

7 a 12V



4 FORMAS DE ALIMENTAR ARDUINO UNO R3

1

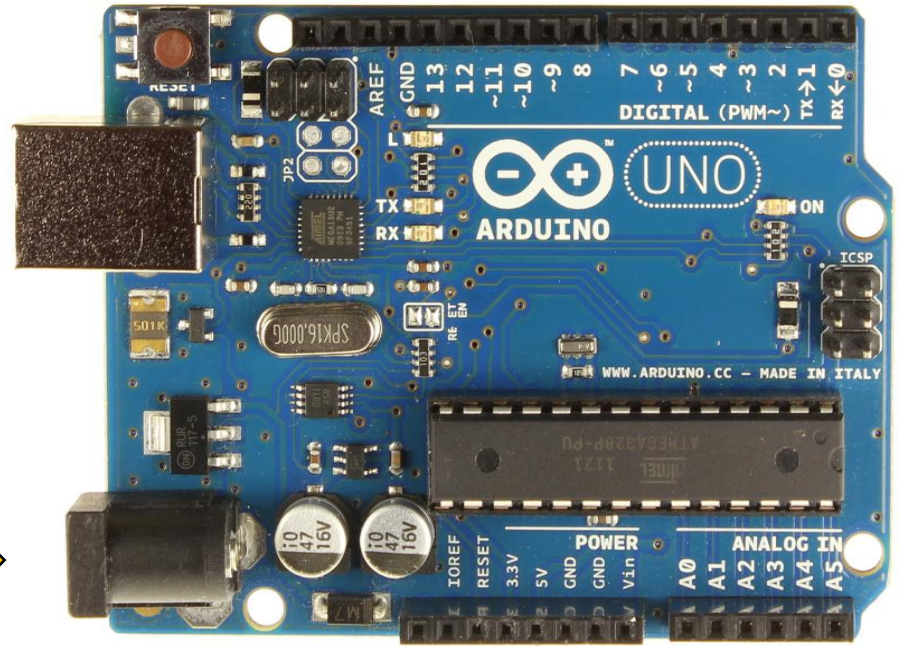
Puerto USB
Cuenta con fusible PPTC (Polymeric Positive Temperature Coefficient device), limitando la corriente a unos 500 mA (USB1.0 y 2.0). El voltaje debe ser de 5V +/-5%

+5V

2

7 a 12V

Jack DC.
De 7 a 12V con positivo en punto central.
Voltajes menores (5 a 7 volts) el Arduino presenta inestabilidad. Voltajes superiores a 12V (hasta 24V), el regulador (NCP1117ST50T3G con encapsulado SOT-223) puede entrar en sobrecalentamiento aún a bajas corrientes. Entrada protegida a inversión de polaridad con diodo.



5V

6 a 12V

3

Pin 5V.
Como ENTRADA: 5V, +/- 5%, sin protecciones.
Como SALIDA: 5V., obtenidos desde el regulador, max 1000mA.

4

Pin Vin.
Como ENTRADA: 6 a 12V, sin protecciones, directo al Regulador, sin pasar por el Diodo Protector, por lo que admite desde 6V.
Como SALIDA: el voltaje será el inyectado al Jack DC – 0.7V, max 1000mA.

PRINCIPALES TIPOS DE CONECTORES USB

Estándar A (2.0)

Estándar A (3.1)

Tipo B (2.0)

Tipo B (3.1)

Mini B (5 pin)

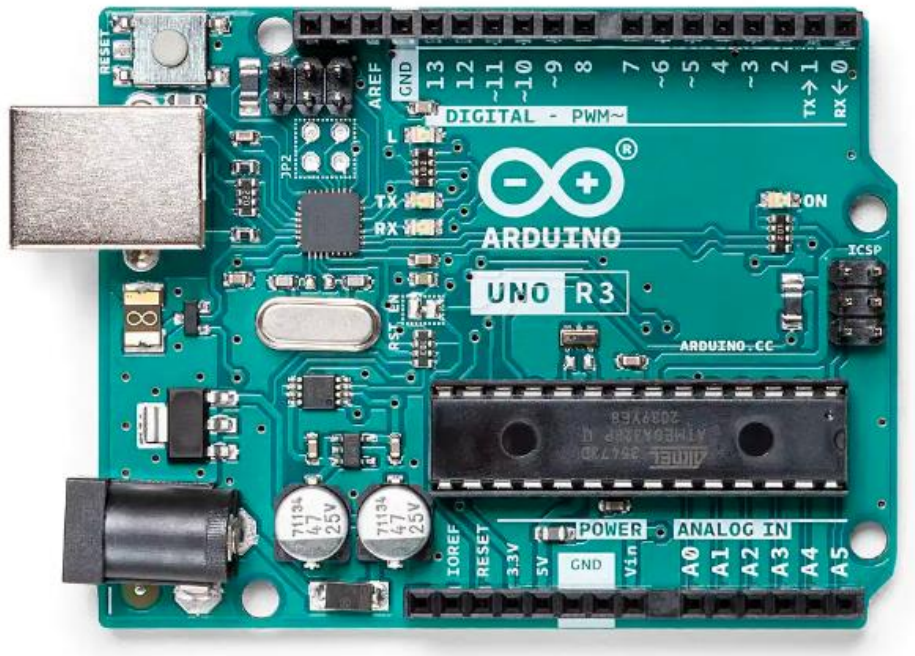
Micro B (5 pin)

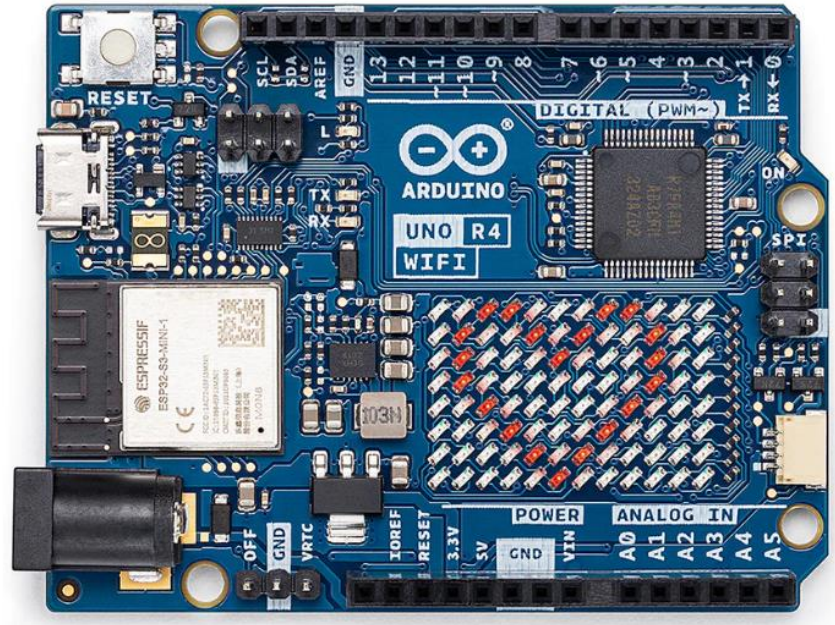
Micro B (10 pin)

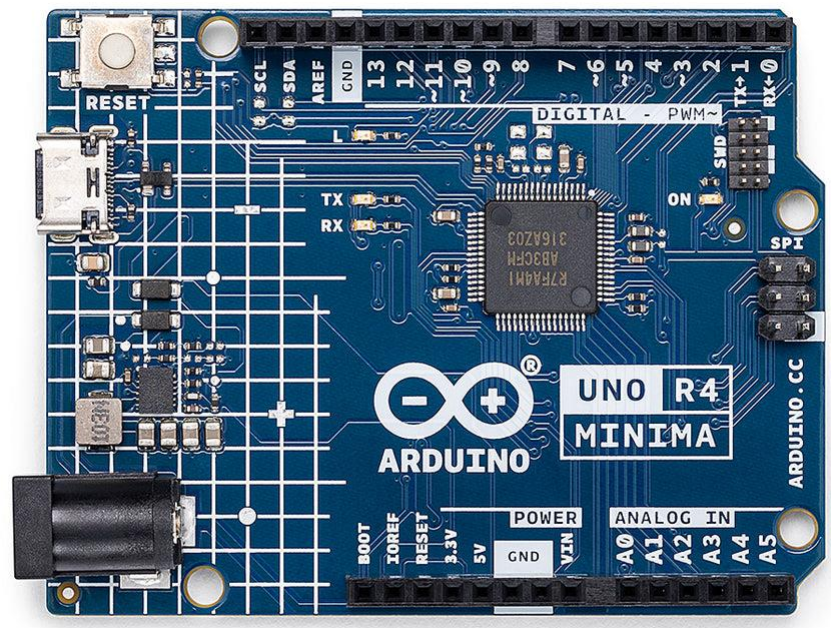
Tipo C (3.1)

Ing. BRAIN NASER SOTO

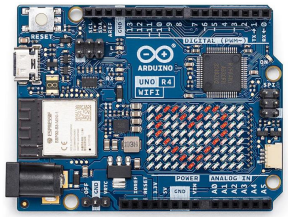
ARDUINO UNO R3 vs UNO R4

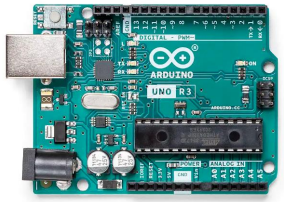






ARDUINO UNO R3 vs UNO R4

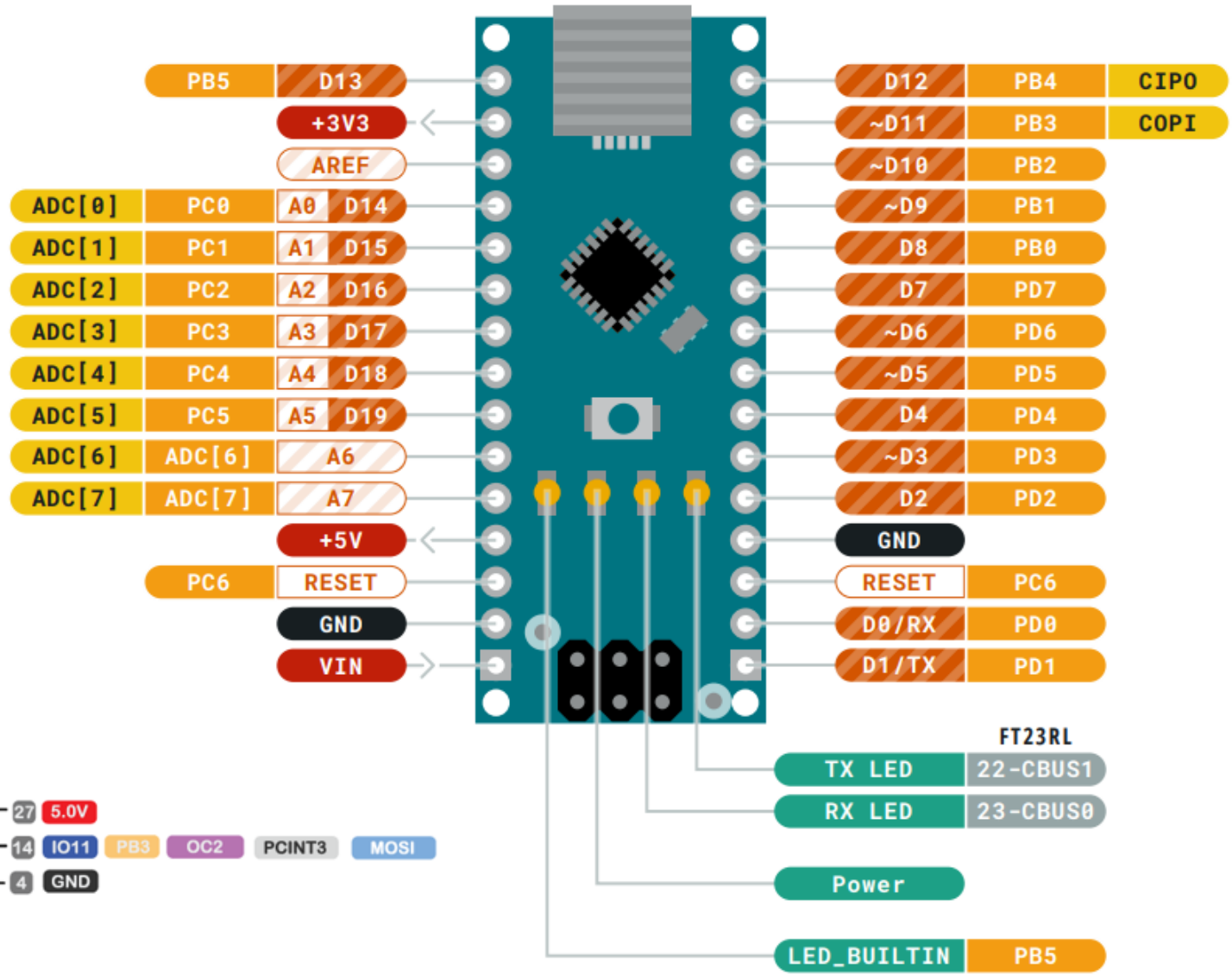




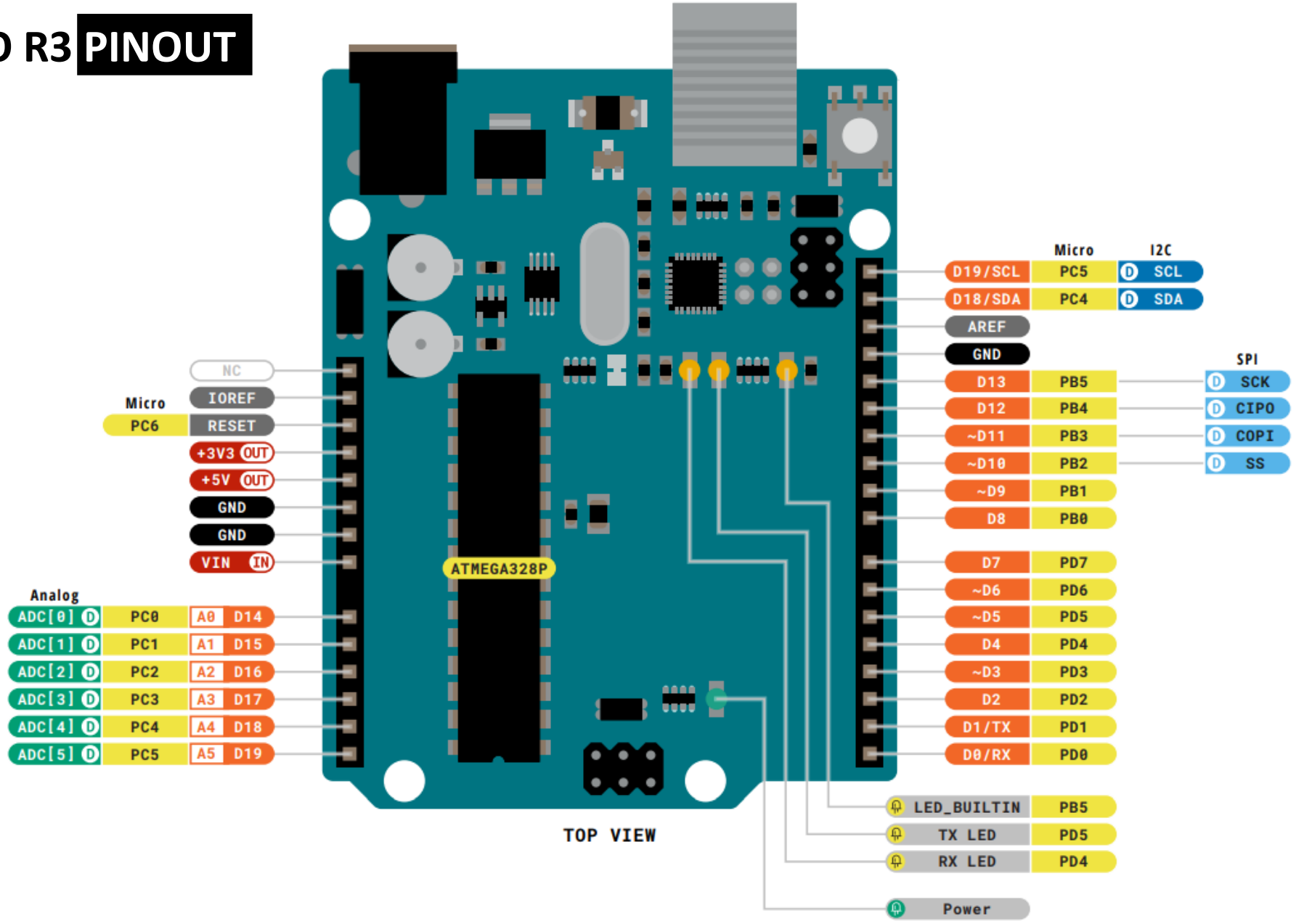
PARAMETRO	ARDUINO UNO R4	ARDUINO UNO R3
Microcontroller	RA4M1 32-bit	ATmega328P 8 bit
Clock Speed	48MHz	16 MHz
External Interrupts	14	2
USB Connector	USB-C (con HID)	USB-B
Real Time Clock (RTC)	Si	No
I/O Pin Source/Skin Current	8mA	20mA
Digital Analog Converter (DAC)	12 bit	No
Analog Digital Converter (ADC)	14 bit	10 bit
CAN Bus Interface	Si	No
WiFi & Bluetooth	Si (sólo R4 WiFi)	No
Flash Memory	256KB	32KB
SRAM	32KB	2KB

										
Fabricante	Arduino	Arduino	Arduino	Arduino	Arduino	Arduino	Arduino	Netduino	Texas Instruments	Fundación Raspberry Pi
Modelo	Pro Mini	Nano	Uno	Mega / Mega 2560	Leonardo	Micro	Due	Netduino 2	Stellaris Launchpad LM4F120	Raspberry Pi Mod.B
Microcontrolador	AVR Atmega 168 ó 328 8bits	AVR ATmega 168 ó 328 8bits	AVR ATmega 328 8bits	AVR ATmega2560 8bits	AVR ATmega 32u4 8bits	AVR ATmega 32u4 8bits	ARM SAM3X8E Cortex-M3 32bits	ARM STMicro STM32F2 Cortex-M3 32bits	ARM LM4F120H5QR Cortex-M4 32bits	ARM Broadcom BCM2835
Frecuencia	16Mhz	16Mhz	16Mhz	16Mhz	16Mhz	16Mhz	84Mhz	120Mhz	80Mhz	700Mhz
Memoria RAM	2KiB	2KiB	2KiB	8KiB	2.5KiB	2.5KiB	96KiB (64+32KiB)	60KiB	32KiB	512MiB
Memoria EEPROM	1KiB	1KiB	1KiB	4KiB	1KiB	1KiB	0	0	-	-
Memoria FLASH	16 ó 32KiB	16 ó 32KiB	32KiB	128 ó 256KiB	32KiB	32KiB	512KiB	192KiB	256KiB	-
Pines digitales entradas/salidas	14/14	14/14	14/14	54/54	20/20	20/20	54/54	20/20	43/43	8/8
Tensión/corriente pines digitales	3.3v ó 5v 40mA	5v 40mA	5v 40mA	5v 40mA	5v 40mA	5v 40mA	3.3v 3~15mA (130mA entre todos)	3.3v~5v 25mA (125mA entre todos)	5v	-
Pines analógicos entradas/salidas	6/0	8/0	6/0	16/0	12/0	12/0	12/2	6/0	-	-
Tensión/resolución pines analógicos	3.3v ó 5v 10bits (1024 valores)	5v 10bits (1024 valores)	5v 10bits (1024 valores)	5v 10bits (1024 valores)	5v 10bits (1024 valores)	5v 10bits (1024 valores)	3.3v 12bits (4096 valores)	5v 12bits (4096 valores)	-	-
Pines con interrupción externa	2	2	2	6	2	2	-	-	-	-
Pines PWM	6	6	6	15	7	7	12	6	-	-
Conexiones Serial / UART	1	1	1	4	1	1	4	4	8	Si
Conexiones I2C / TWI	1	1	1	1	1	1	2	1	4	Si
Conexiones ISP / ICSP	1	1	1	1	1	1	1	1	-	Si
Conexión USB	No (necesita adaptador externo)	Si	Si, USB-B	Si, USB-B	Si, Nativa, MicroUSB	Si, Nativa, MicroUSB	Si, Nativa, MicroUSB	Si, Nativa, MicroUSB	Si, Nativa, MicroUSB	Si, MicroUSB
Conexión USB de depuración	No	No	No	No	No	No	Si, MicroUSB	Si, MicroUSB	Si, MicroUSB	-
Conexión Bluetooth	No	No	No	No	No	No	No	No	No	-
Conexión WiFi	No	No	No	No	No	No	No	No	No	-
Conexión Ethernet	No	No	No	No	No	No	No	No	No	Si
Conexión USB Host	No	No	No	No	No	No	Si	No	Si	Si
Almacenamiento por SD	No	No	No	No	No	No	No	No	No	Si
Corriente en el pin de 5v	-	500mA	500~800mA	500~800mA	500~800 mA	500mA	800mA	-	-	-
Corriente en el pin de 3.3v	-	50mA	50mA	50mA	50mA	50mA	800mA	-	-	-
Voltaje de alimentación por el USB	3.3v ó 5v (sin usb)	5v	5v	5v	5v	5v	5v	5v	5v	5v

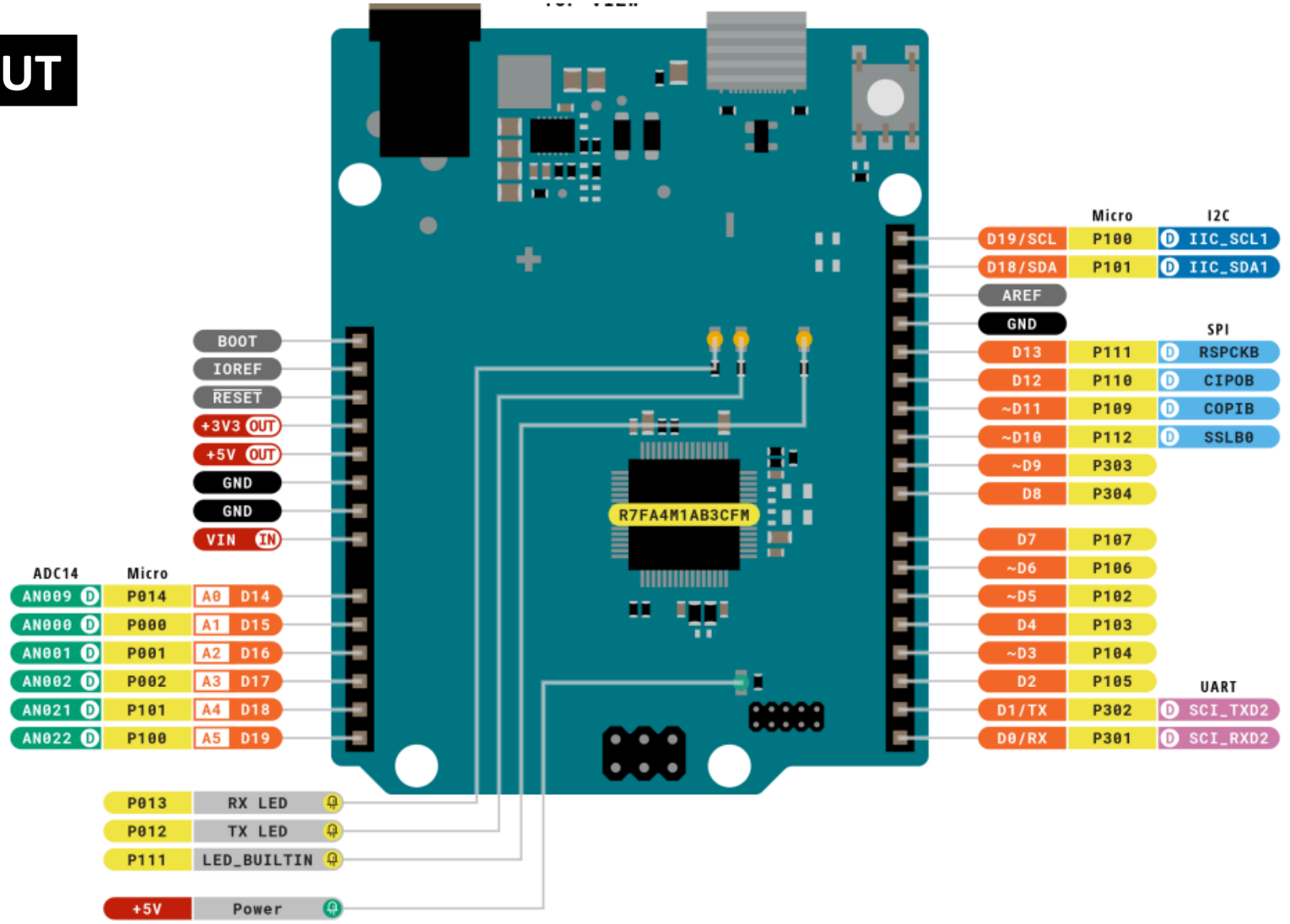
ARDUINO NANO **PINOUT**



ARDUINO UNO R3 PINOUT



ARDUINO UNO R4 PINOUT

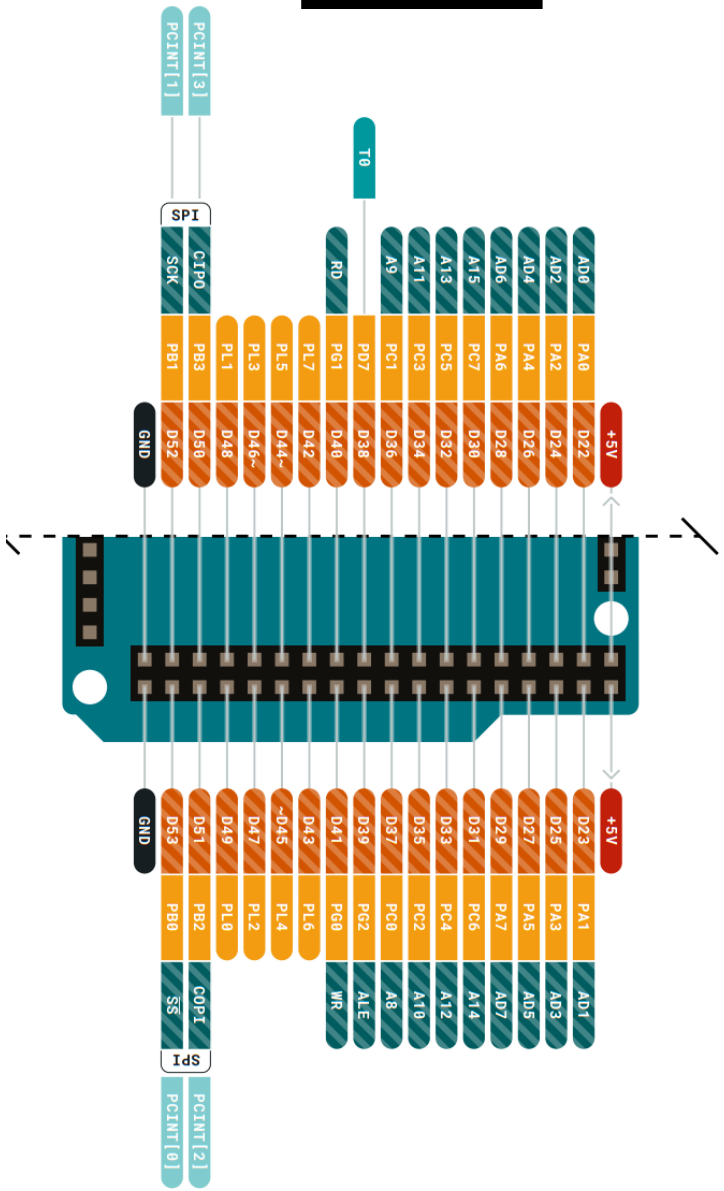


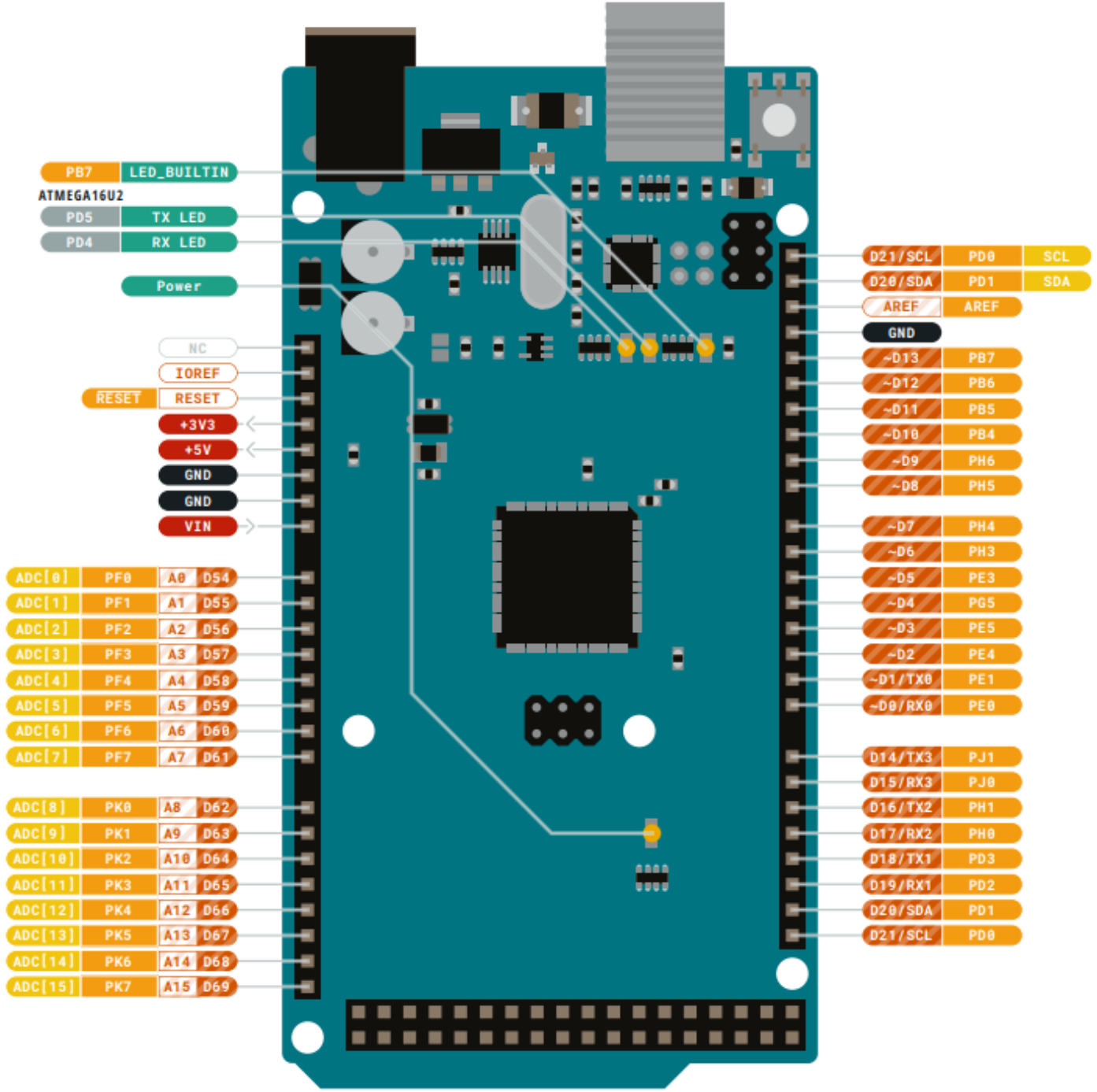
Legend:
Digital
I2C
Other SERIAL
Power
Analog
SPI
Analog
Ground
Main Part
UART/USART
PWM/Timer

ARDUINO

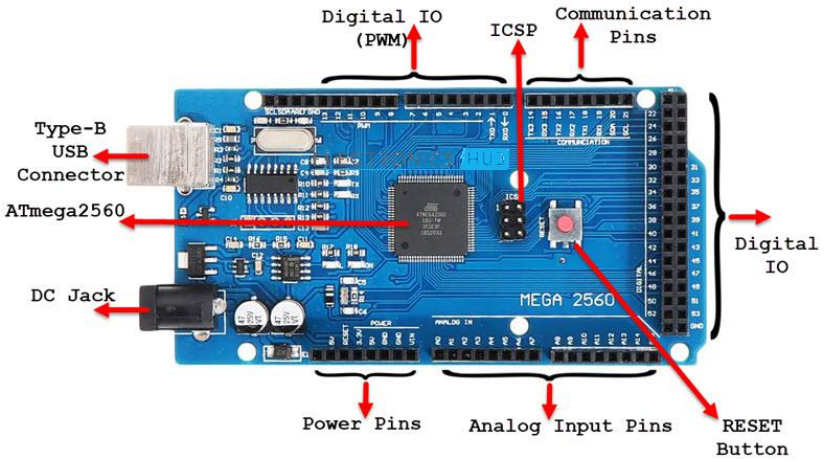
ARDUINO UNO R4 MINIMA
SKU code: ABX00080
Pinout
Last update: 16 Jun, 2023

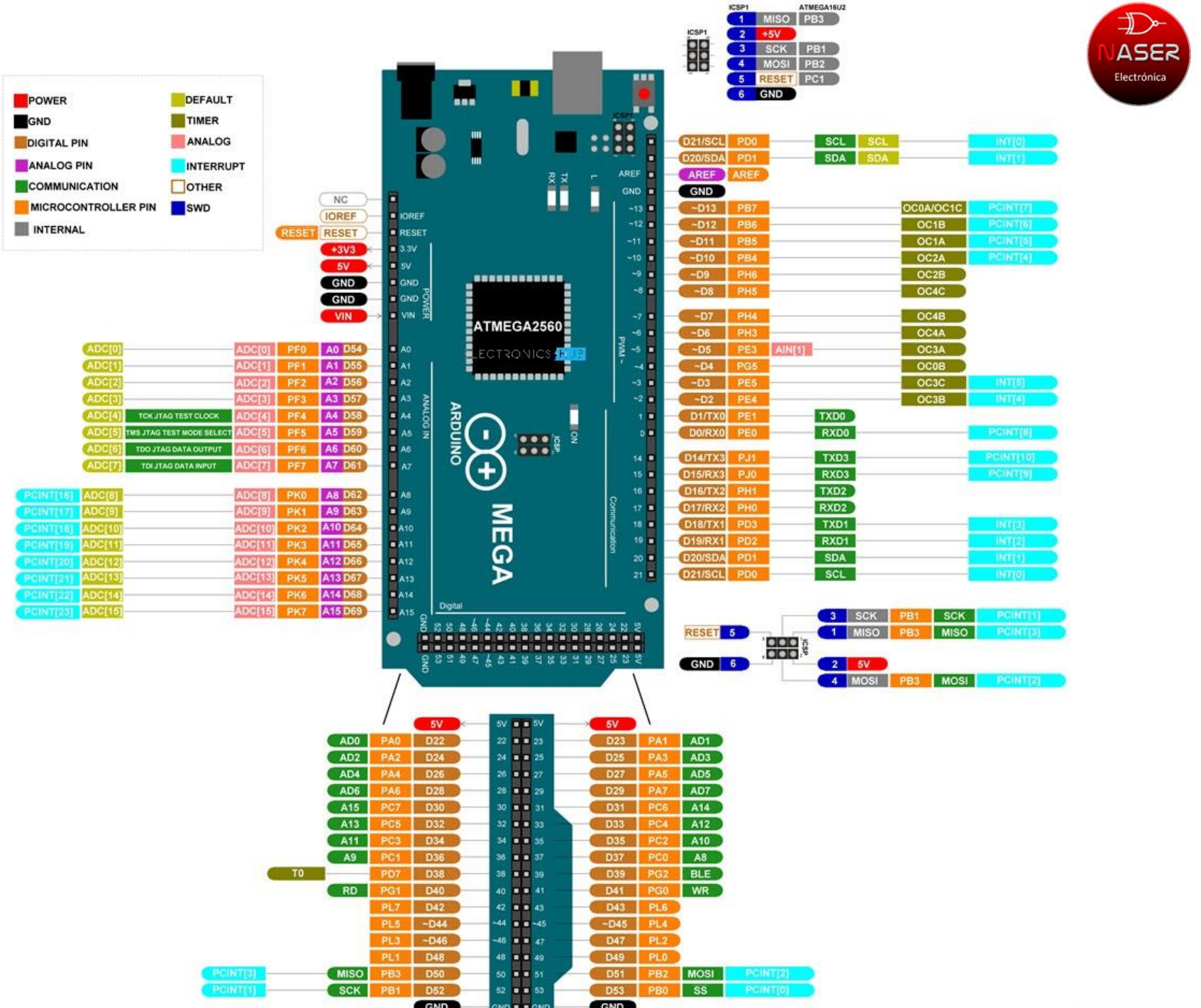
ARDUINO MEGA PINOUT



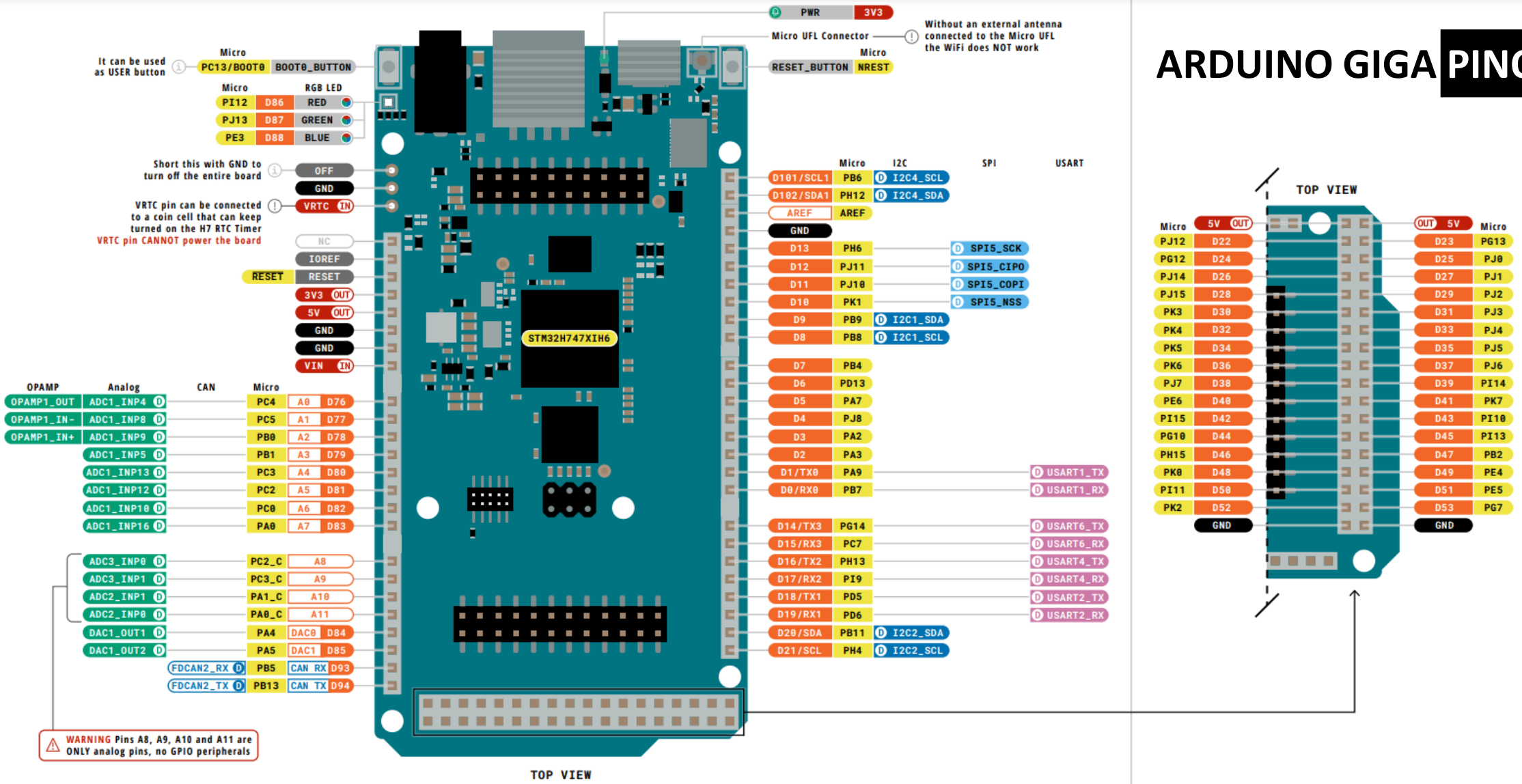


ARDUINO MEGA PINOUT





ARDUINO GIGA PINOUT



Legend:

- Power
- Ground
- Power Input
- Power Output
- GPIO Digital External
- Analog External
- Main Part
- Secondary Part
- Internal Component
- I2C — Default
- SPI — Default
- UART/USART — Default
- Other SERIAL Communication
- Analog — Default
- LED
- RGB LED
- Other

MAXIMUM Total output current sourced or sunked by sum of all I/Os and control pins is 140 mA

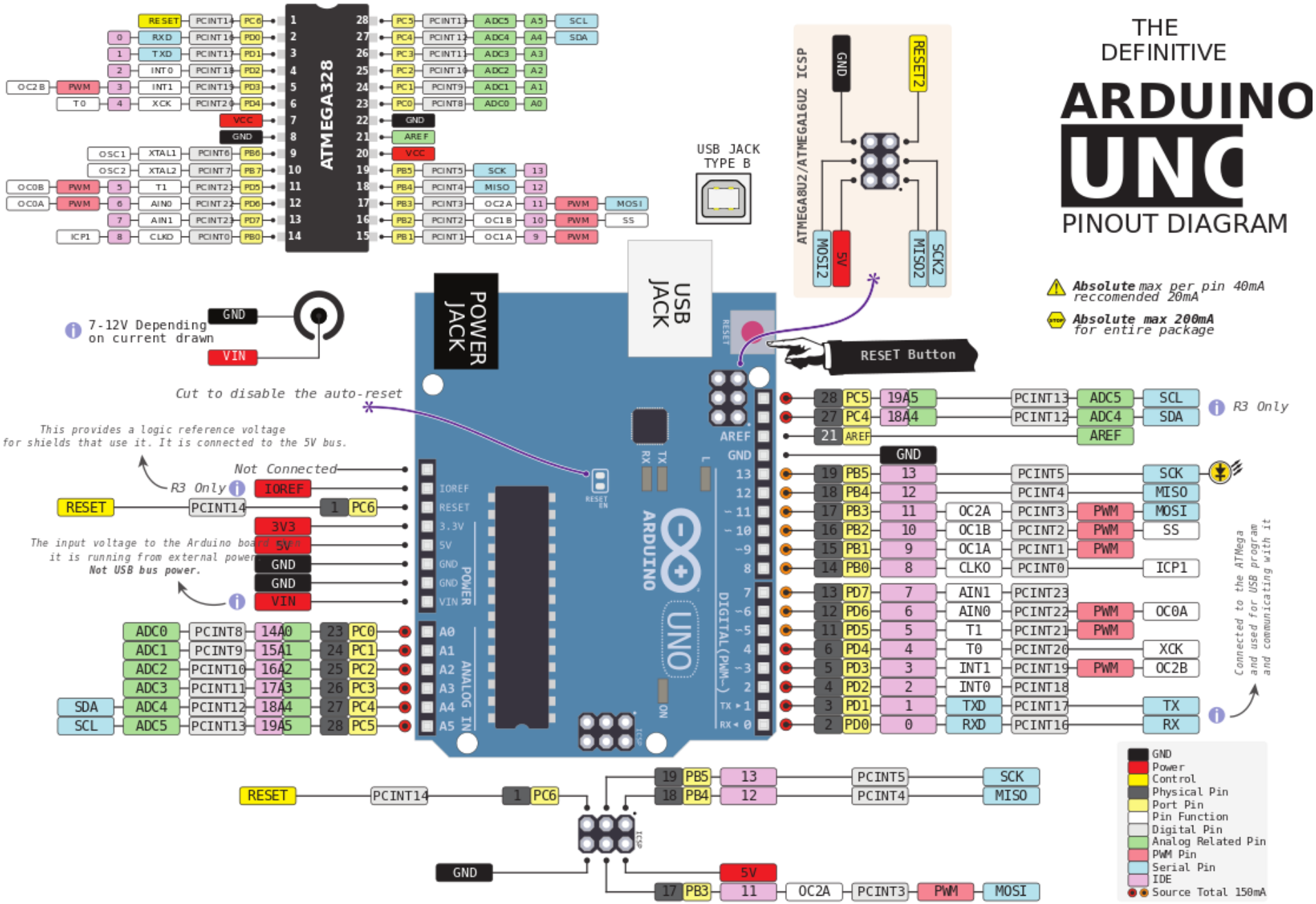
MAXIMUM Output current sourced or sunked by any I/O and control pin is 20 mA

CIPO/COPI have previously been referred to as MISO/MOSI

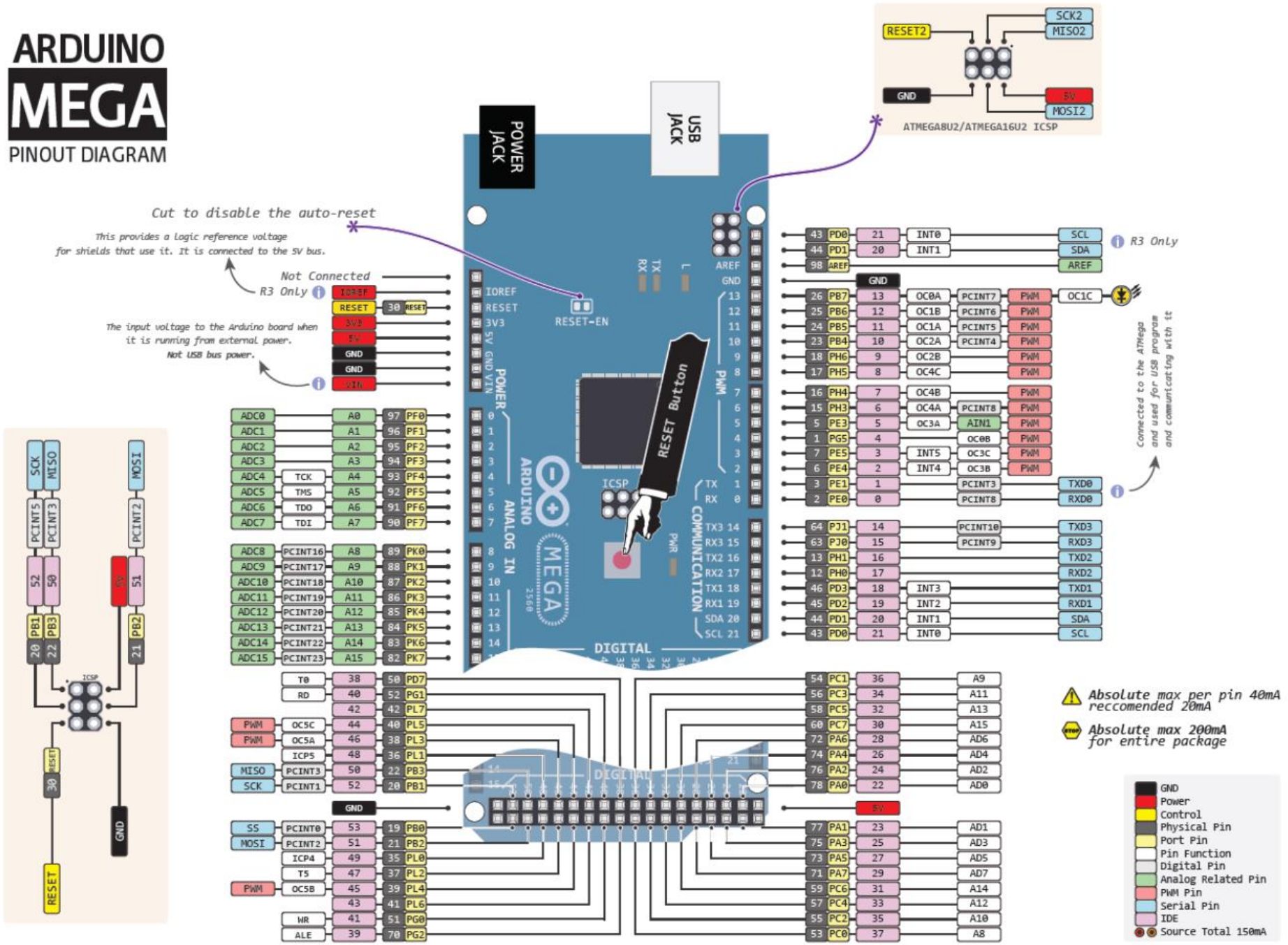
ARDUINO GIGA R1 WIFI BOARD

SKU code: ABX00063
Full Pinout - Page 1 of 9
Last update: 21 Feb, 2023

DOCS.ARDUINO.CC
This work is licensed under the Creative Commons Attribution-ShareAlike 4.0

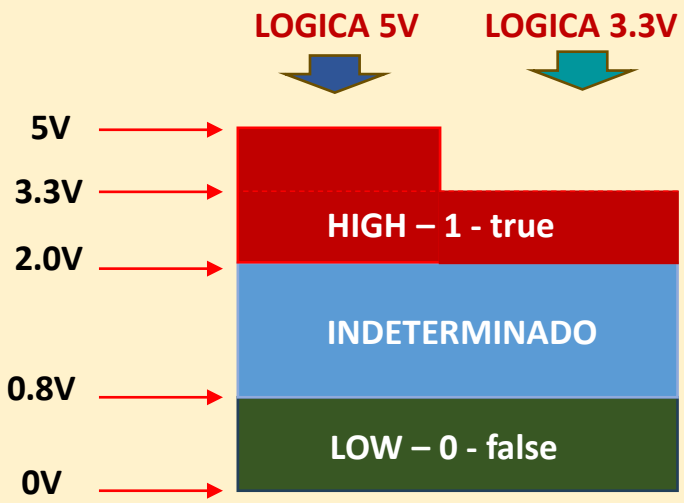

THE
DEFINITIVE
ARDUINO
UNO
PINOUT DIAGRAM
ATMEGA328
ATMEGA8U2/ATMEGA16U2 ICSP
USB JACK TYPE B
POWER JACK
USB JACK
RESET Button
7-12V Depending on current drawn
Cut to disable the auto-reset
This provides a logic reference voltage for shields that use it. It is connected to the 5V bus.
Not Connected
R3 Only
The input voltage to the Arduino board when it is running from external power. Not USB bus power.
Absolute max per pin 40mA reccomended 20mA
Absolute max 200mA for entire package
Connected to the ATMega and used for USB programming and program it
GND
Power
Control
Physical Pin
Port Pin
Pin Function
Digital Pin
Analog Related Pin
PWM Pin
Serial Pin
IDE
Source Total 150mA

ARDUINO MEGA
PINOUT DIAGRAM

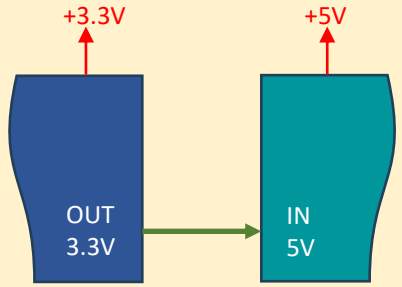


Ing. Brain Naser Soto

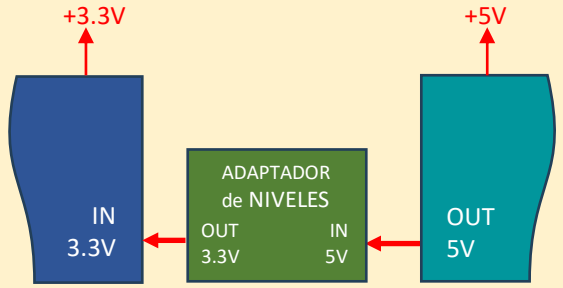
NIVELES LOGICOS



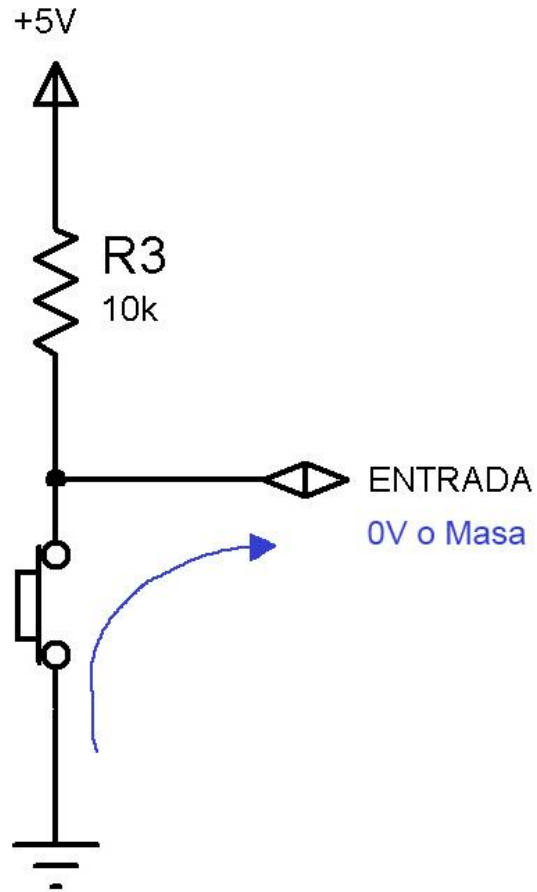
Salida de 3.3V a entrada de 5V

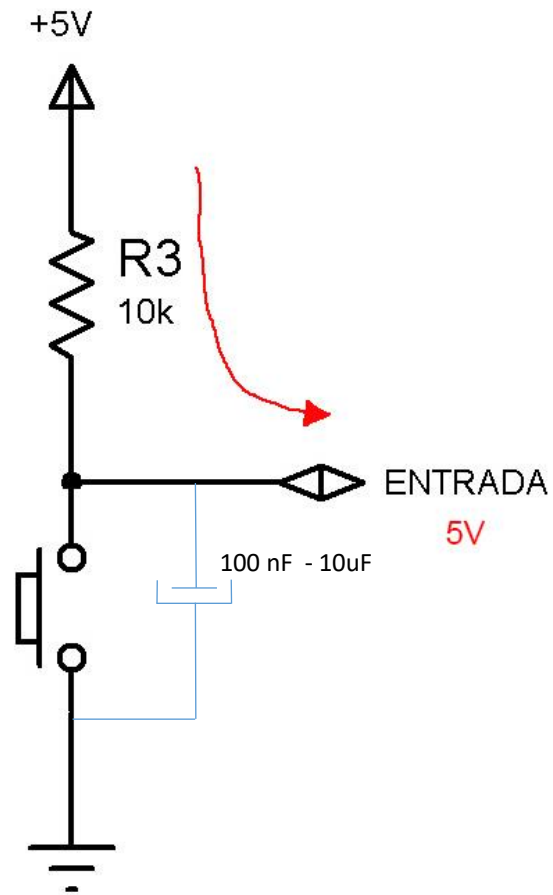


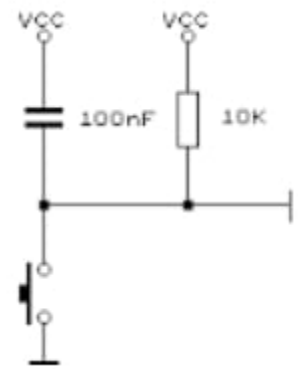
Salida de 5V a entrada de 3.3V



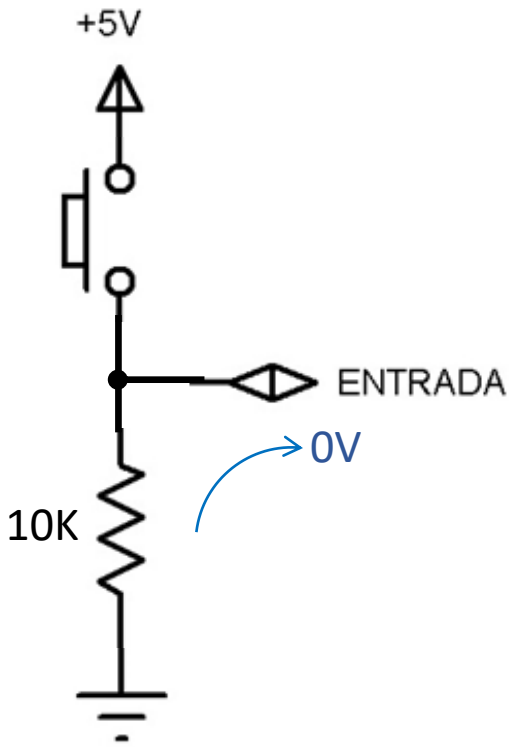
RESISTENCIAS DE PULL UP
INPUT_PULLPU

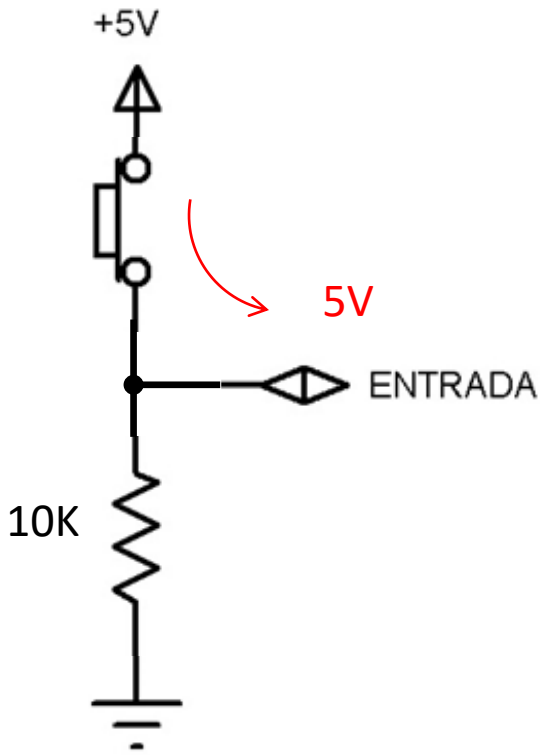


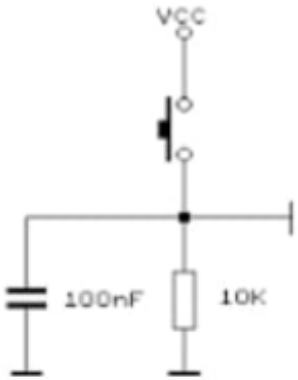




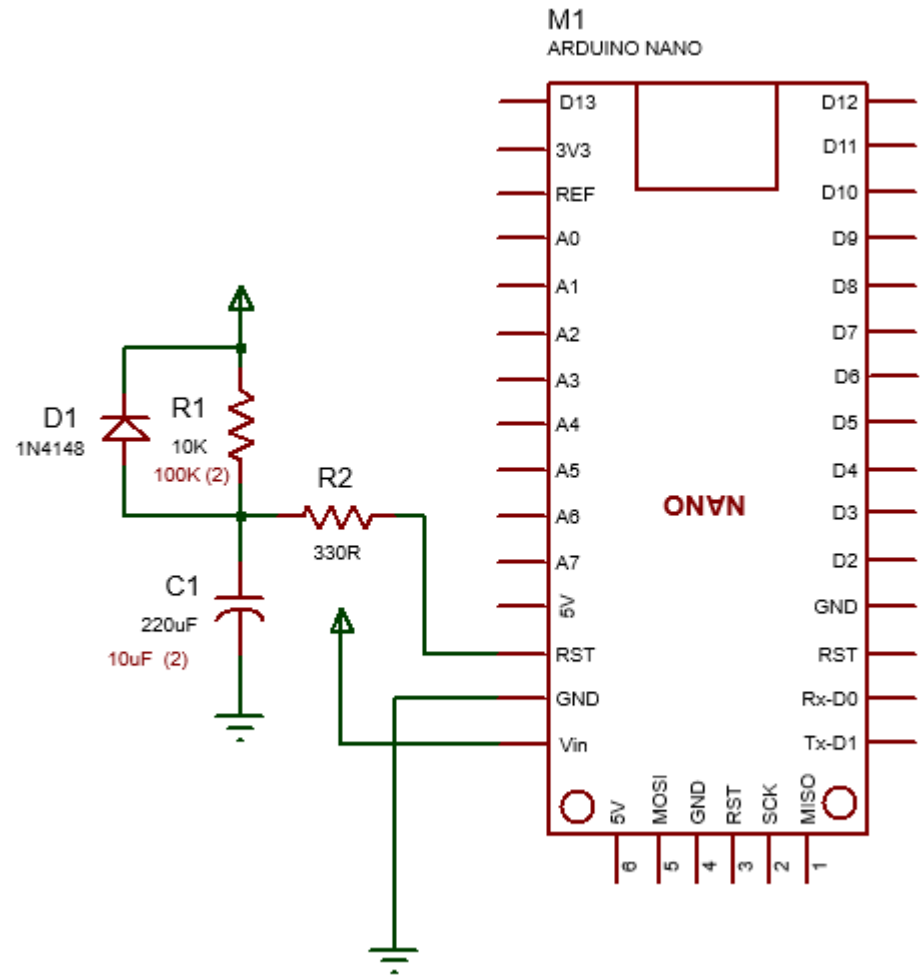
RESISTENCIAS DE PULL DOWN



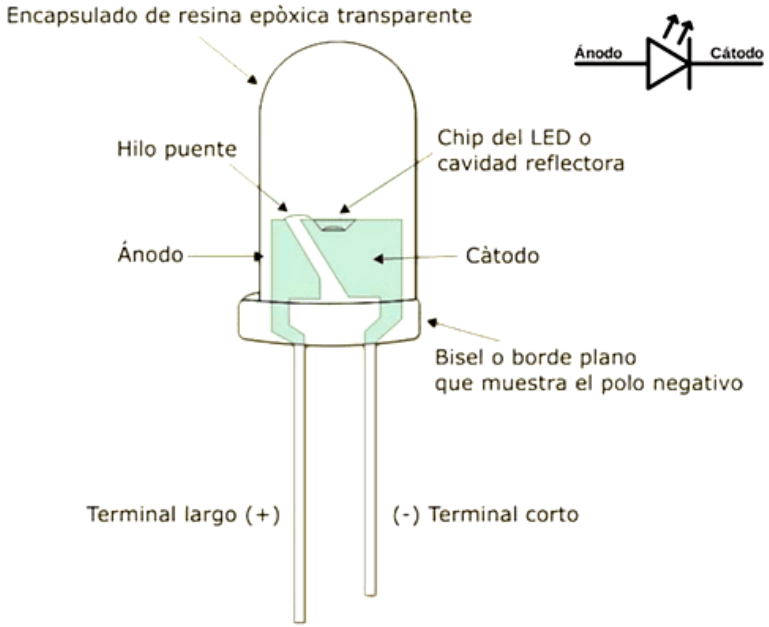


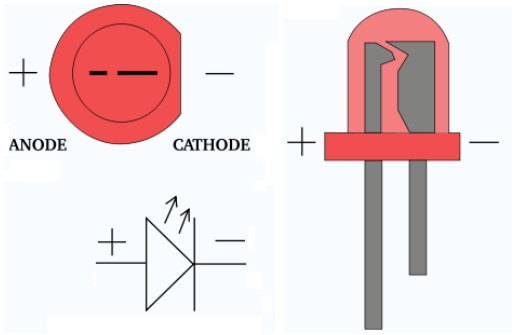


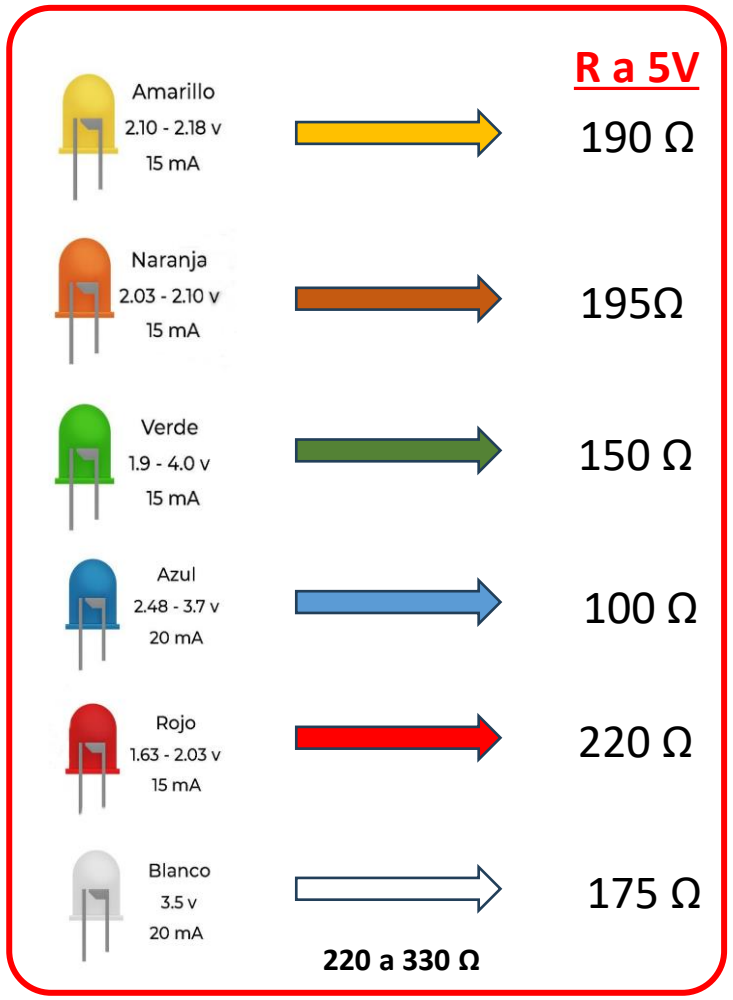
CIRCUITO RESET de INICIO



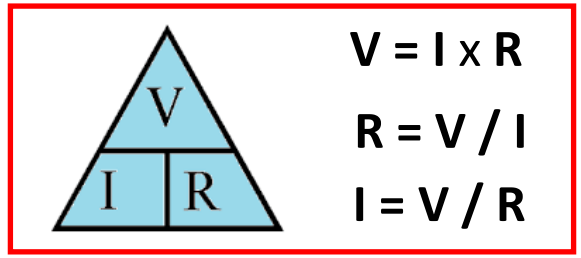
DIODOS LED´S
Light Emitting Diodes







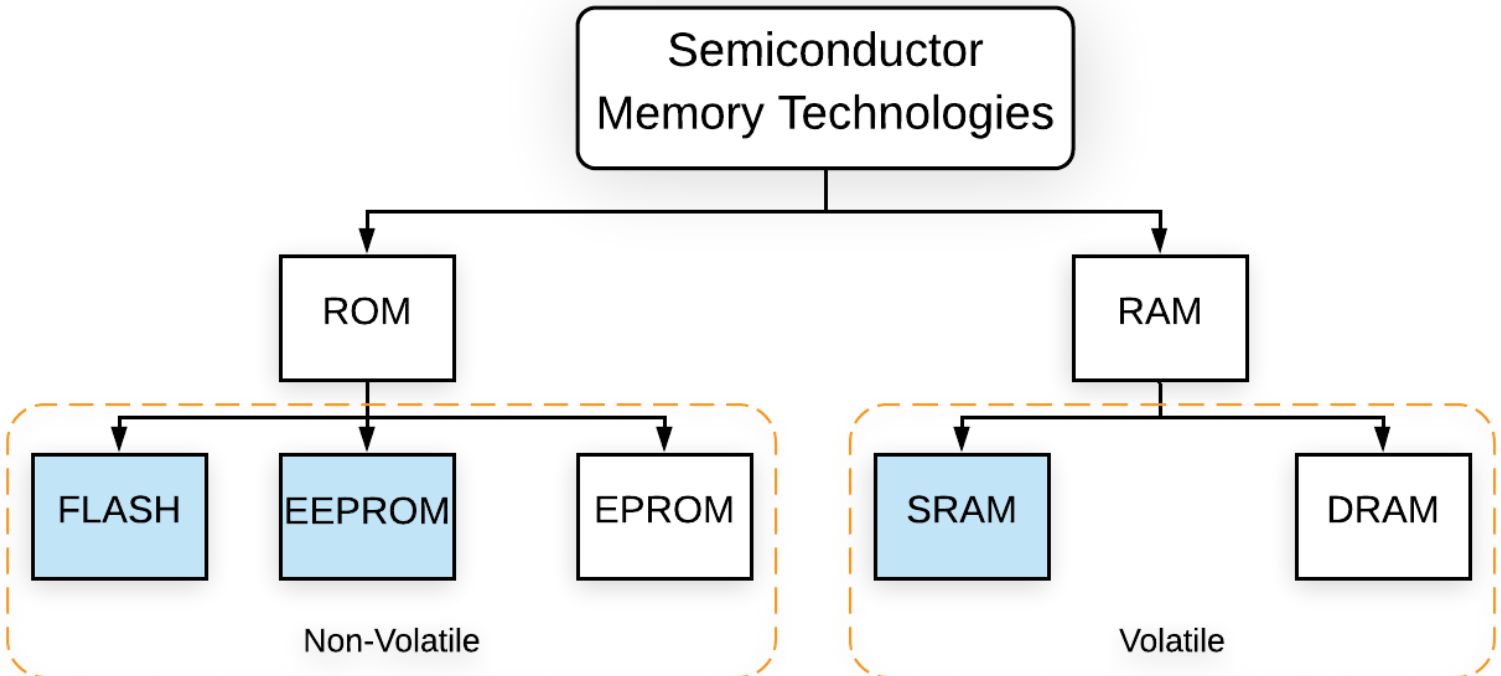
Ley de Ohm



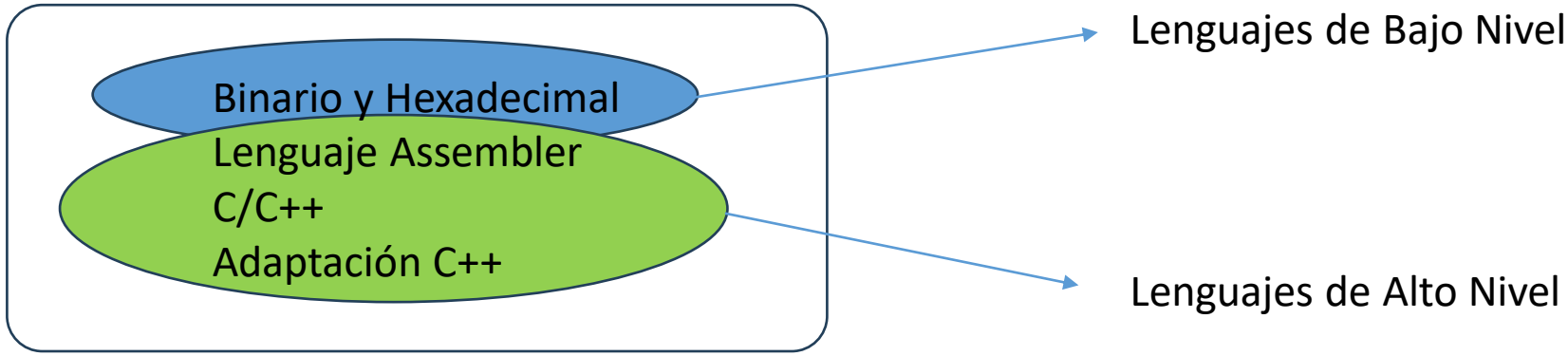
Light Emitting Diodes

	Emitted Colour	Lens	Wave Length (nm)	Pd W (mW)	If mA	If mA (peak)	Vf (V)Min	Vf (V)Typ	Vf (V)Max	IV Min	mcd Typ	View Angle
3mm LEDs	red	diffused	700	45	15	50	1.7	2.1	2.8	3	5	40
	red	waterclear	660	110	30	200	1.5	1.8	2.2	300	500	30
	red	waterclear	625	120	50	100	1.5	2	2.6	1700	2800	15
	red	waterclear	625	130	50	150	2	2.3	2.6	4900	7000	15
	yellow	diffused	585	85	20	160	1.7	2	2.8	1.1	4.5	40
	yellow	waterclear	595	80	15	200	1.5	2	2.6	2000	3000	15
	yellow	waterclear	588	80	15	200	2	2	2.6	4500	6500	15
	orange	diffused	635	100	30	160	1.7	2	2.8	1.11	4.5	40
	green	diffused	565	100	30	160	1.7	2.1	2.8	1.1	6	40
	green	waterclear	570	100	30	160	1.7	2.1	2.4	200	500	70
	green	waterclear	525	130	30	100	3.2	3.5	4	5500	6000	15
	blue/green	waterclear	505	130	30	100	3.2	3.5	4	2800	4000	15
	blue	waterclear	470	130	30	100	3.2	3.5	4	700	1000	15
	blue	waterclear	470	130	30	100	3.2	3.5	4	1700	3000	15
	white	waterclear	–	130	30	100	3.2	3.5	4	700	1000	20
	white	waterclear	–	130	30	100	3.2	3.5	4	2400	3500	20
5mm LEDs	red	diffused	697	45	15	50	1.7	2.1	2.8	2.5	10	36
	red	waterclear	660	110	30	200	1.5	1.8	2.2	200	500	30
	red	waterclear	625	120	50	100	1.5	2	2.6	3000	4000	15
	red	waterclear	625	130	50	150	2	2.3	2.6	7000	10000	15
	red	diffused	643	100	8	150	9	12	15	–	120	35
	red	waterclear	616	135	50	1000	–	2.1	5	7500	20000	–
	red	waterclear	625	500	75	100	–	2.2	2.6	–	30000	16
	yellow	diffused	585	85	20	160	1.7	2	2.8	2	10	36
	yellow	waterclear	615	130	50	150	1.7	2.4	3	1700	1000	15
	yellow	waterclear	585	80	15	200	1.7	1.6	2.2	3000	4000	20
	yellow	waterclear	590	130	50	150	2	2.4	3	7000	9000	15
	yellow	waterclear	585	500	75	100	–	2.2	2.6	–	30000	15
	orange	diffused	635	100	30	160	1.7	2	2.8	2.5	9	36
	orange	waterclear	600	500	75	100	–	2.2	2.6	–	30000	15
	green	diffused	565	100	30	160	1.7	2.1	2.8	2	10	36
	green	waterclear	525	120	30	100	3.2	3.5	4	2500	3000	30
	green	waterclear	525	120	30	100	3.2	3.5	4	4500	6500	15
	green	waterclear	525	120	30	100	3.2	3.5	4	7000	10000	15
	pure green	waterclear	525	500	75	100	–	3.6	4.2	–	30000	15
	blue/green	waterclear	505	120	30	100	3.2	3.5	4	4900	7000	15
	blue	waterclear	470	130	30	100	3.2	3.5	4	1000	1500	20
	blue	waterclear	470	130	30	100	3.2	3.5	4	3500	5000	20
	blue	diffused	470	120	30	100	3.2	3.5	4	600	900	50
	blue	waterclear	470	500	75	100	–	3.6	4.2	–	10000	15
	white	waterclear	–	130	30	100	3.2	3.5	4.3	1000	1500	20
	white	waterclear	–	130	30	100	3.2	3.5	4.3	6000	7000	20
	white	waterclear	–	130	30	100	3.2	3.5	4.3	8000	10000	20
	white	waterclear	–	120	30	150	–	3.3	3.6	17000	18000	15
	white	waterclear	–	500	75	100	–	3.6	4.2	–	30000	15
	pink	waterclear	–	120	30	100	3.2	3.5	4	1400	2000	20

10mm LEDs	red	diffused	635	100	30	160	1.7	2	2.8	5	8	44
	red	waterclear	625	120	50	100	1.7	2	2.6	3000	4000	20
	red	waterclear	625	130	50	150	2	2.3	2.6	7000	10000	20
	green	diffused	565	100	30	160	1.7	2.1	2.8	5	8	44
	green	waterclear	525	120	40	100	3.2	3.5	4	6000	9000	25
	yellow	diffused	585	85	20	160	1.7	2	2.8	5	7	44
	yellow	waterclear	585	80	15	200	1.7	1.8	2.2	2500	3600	25
	yellow	waterclear	590	130	50	150	2.2	2.4	3	6300	9000	20
	blue	waterclear	470	130	30	100	3.2	3.5	4	1400	4500	25
	white	waterclear	–	130	30	100	3.2	3.5	4.3	3500	5000	25
	orange		610	130	50	150	2.4	2.4	3	6300	10000	20
Rectangular 5x2mm	red	diffused	700	45	10	50	1.7	2	2.4	5	15	180
	green	diffused	570	100	10	160	1.7	2.1	2.4	5	15	180
	yellow		590	80	10	160	1.5	2	2.4	5	15	180
Flashing 5 & 10mm	red 5mm	diffused	635	100	30	160	2.5	3	–	5	15	50
	green 5mm	diffused	565	100	30	160	2.5	3	–	5	15	50
	yellow 5mm	diffused	585	80	20	160	2.5	3	–	5	15	50
	blue 5mm	diffused	470	120	30	100	3.2	3.5	4	600	900	50
	Red 5mm	waterclear	622	–	20	100	–	2.13	2.6	–	1000	25.6
	Green	waterclear	520	–	–	100	–	3.5	4.0	–	1800	21.6
	Blue	waterclear	470	–	–	100	–	2.5	4.0	–	550	21.6
	yellow 10mm	diffused	660	–	25	–		12	–	–	50	50
Bi-Colour 3mm 5mm 10mm	red	diffused	626	100	30	160	1.7	2	2.4	2.5	5	75
	green	diffused	570	100	30	160	1.7	2.1	2.4	2.5	8	75
	red	–	625	100	30	160	1.7	2	2.4	2.5	5	75
	green	–	570	100	30	160	1.7	2.1	2.4	2.5	8	75
	red	diffused	625	100	30	160	1.7	2	2.4	2.5	10	54
	green	diffused	565	100	30	160	1.7	2.1	2.4	2.5	10	54
UV / RGB 5mm	UV	diffused	395	80	10-20	100	–	3.7	4.2	30	40	30
	Red	white diffused	625	–	20	100	1.8	2.4	20	192	250	58
	Green		505	–	20	100	3.5	4.5	20	215	280	59
	Blue		460	–	20	100	3.5	4.5	20	115	150	59
Tricolour	red/green/orange											
	3mm	diffused	625	100	30	160	1.7	2	2.4	2.5	5	75
	5mm	diffused	625	100	30	160	1.7	2	2.4	2.5	5	75
	10mm	diffused	625	100	30	160	1.7	2	2.4	2.5	10	75


Semiconductor
Memory Technologies
ROM
RAM
FLASH
EEPROM
EPROM
SRAM
DRAM
Non-Volatile
Volatile

PROGRAMACION



INICIANDO PROGRAMACION EN C/C++

1. INSTRUCCIONES BÁSICAS
2. CONSTANTES y VARIABLES
3. OPERADORES
4. CONTROL CONDICIONAL
5. CONTROL REPETITIVA

www.arduino.cc

1.- Botón para verificar el código.

2.- Botón para compilar y cargar e el controlador.

3.- Crear un nuevo “Sketch”.

4.- Abrir un “Sketch” existente.

5.- Grabar el “Sketch”.

6.- Monitor serial.

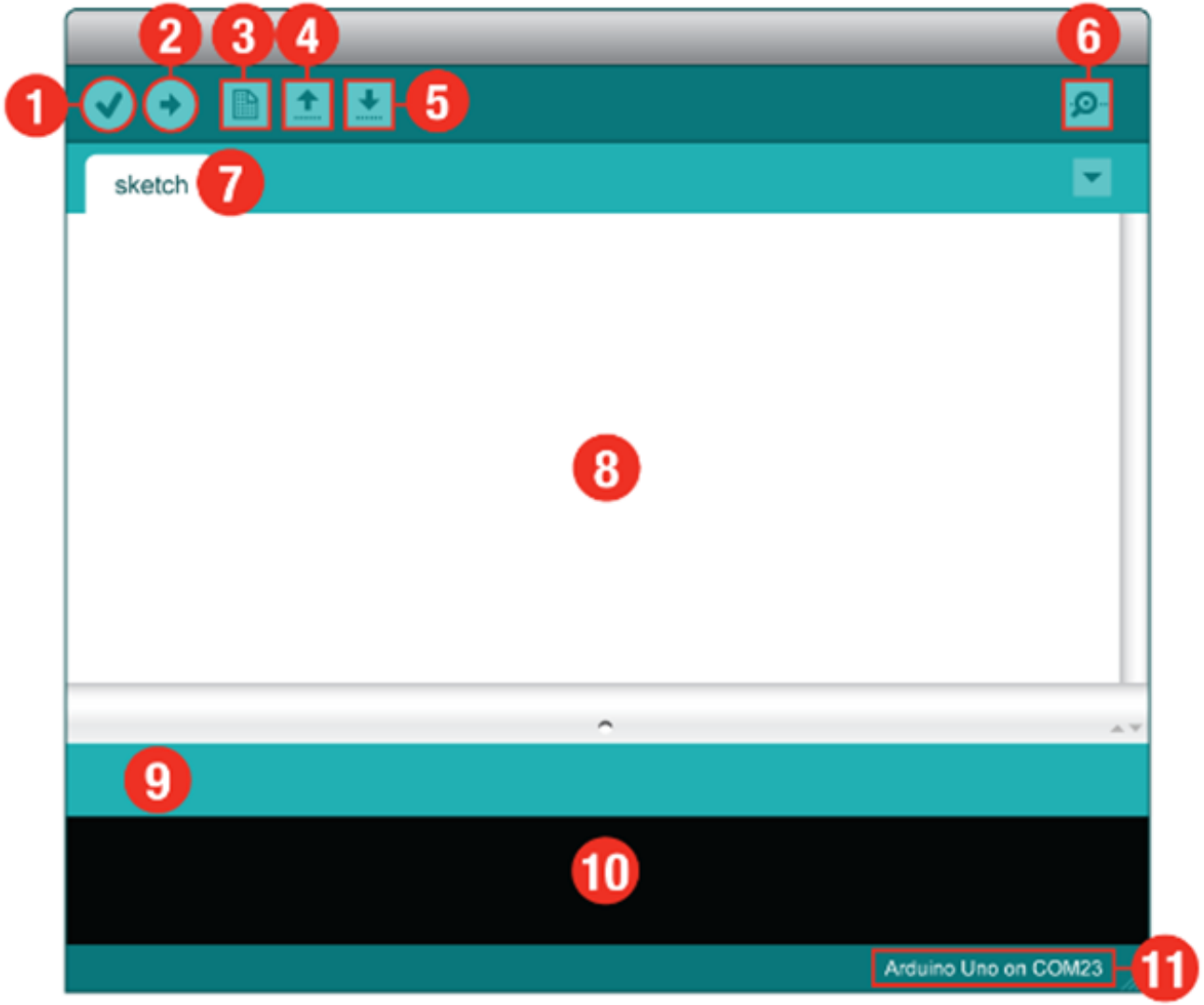
7.- Nombre del Sketch.

8.- Ventana de código.

9.- Ventana de mensajes de verificación y compilación.

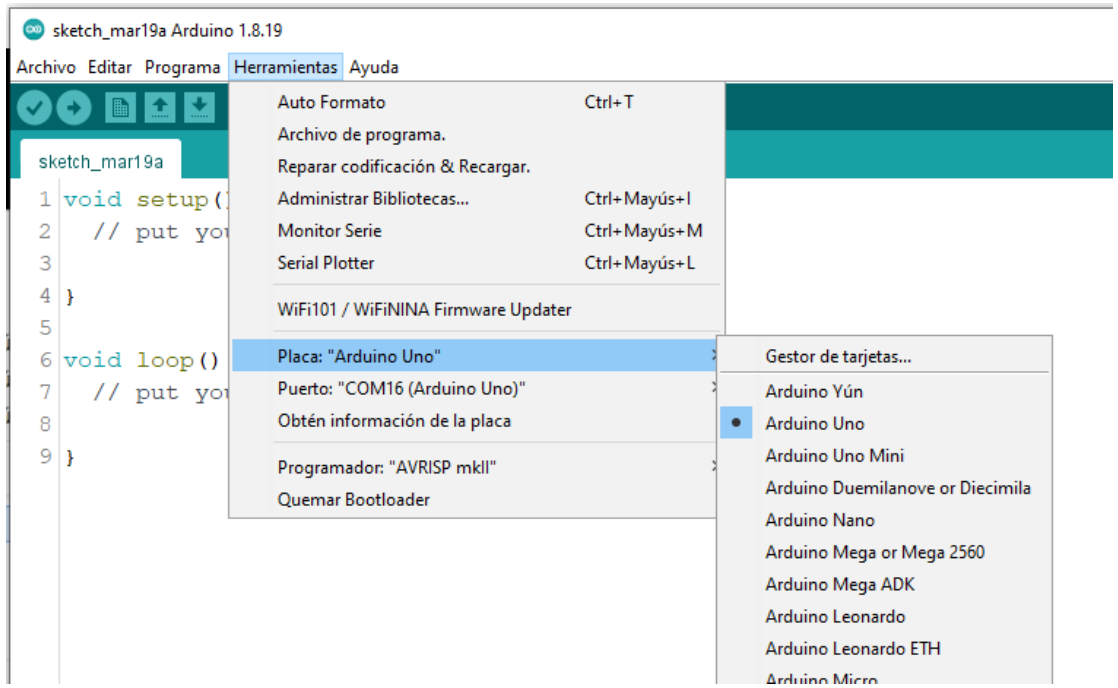
10.- “Consola”, registro del compilador.

11.- Datos de Placa y COM.

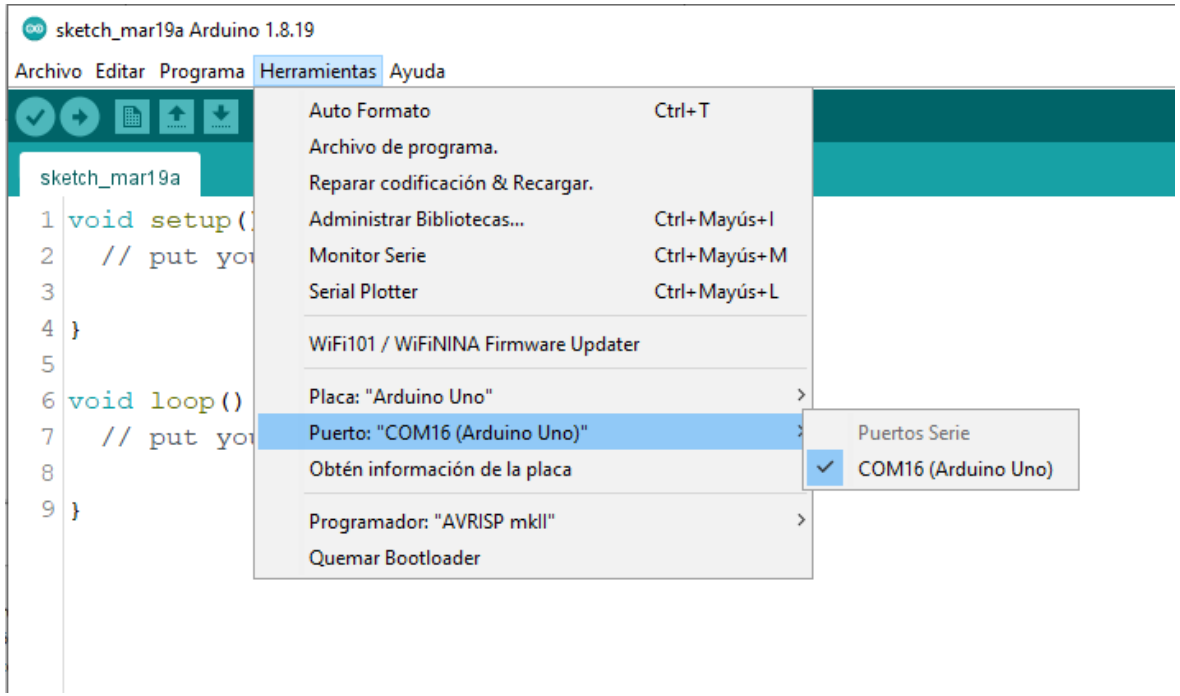


CONFIGURACIÓN DEL IDE

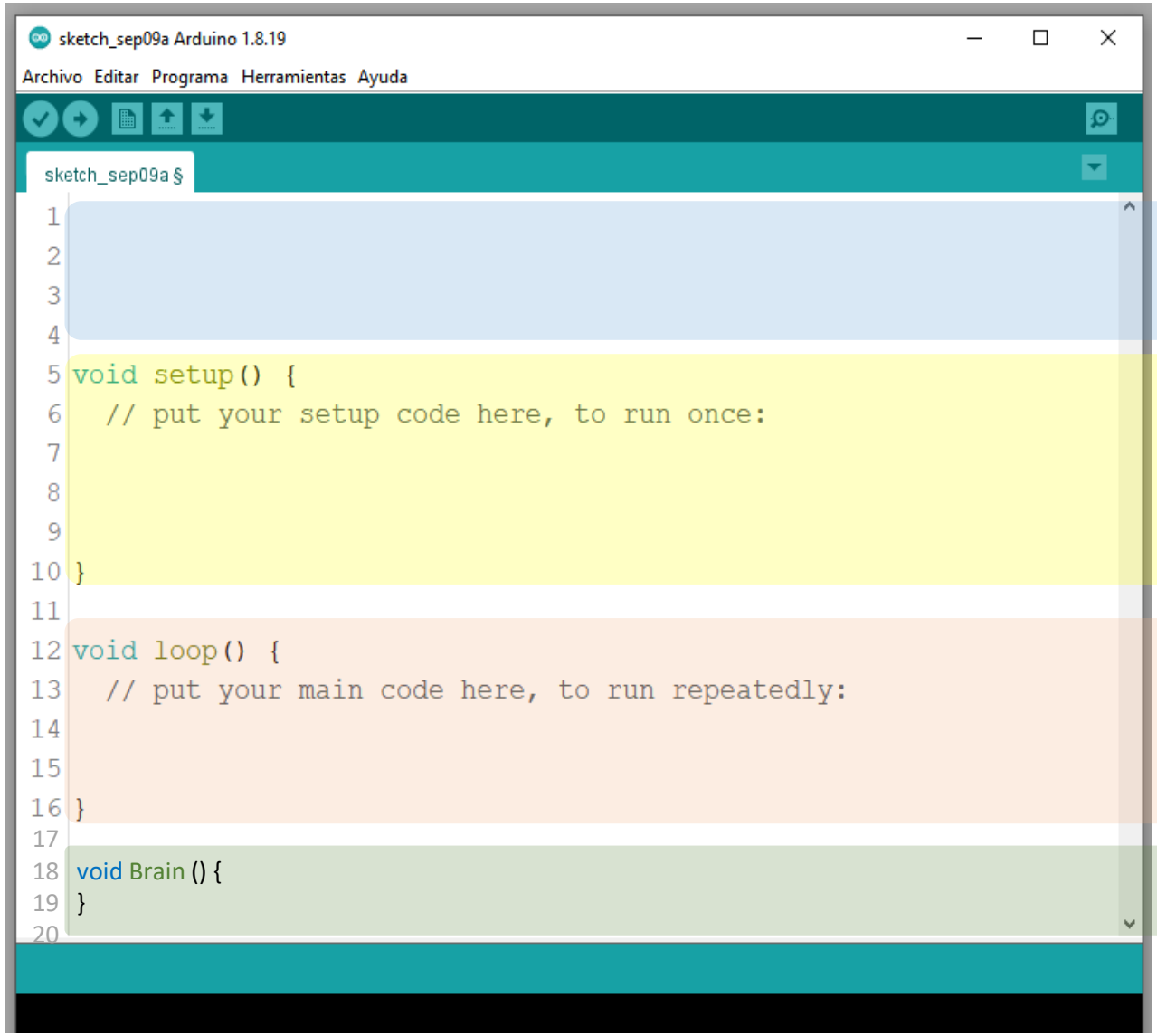
1 Seleccionar Board



2 Seleccionar Puerto USB



La EXTENSIÓN de un código Arduino es **.ino**
Todo archivo de código Arduino, debe estar dentro de una **carpeta** de archivos, sin espacios.



ESTRUCTURA

SCOPE

- **Librerías**
- **Definiciones Macro**
- **Variables GLOBALES**

SETUP

- **Inicialización de Elementos y objetos.**
- **Variables Locales**
- **Código de UNA ejecución.**

LOOP

- **Código de ejecución Cíclica y SECUENCIAL.**
- **Variables Locales**

FUNCTIONS

- **Códigos de Funciones Especiales.**
- **Variables Locales**

Estructura y Flujo

Sintaxis

```
// (Comentario en una linea)
/* (Comentario de multiple linea)*/
#define()
#include <NombreDeLibreria.h>
```

Estructúra básica del programa

```
void setup() {
//Corre una tan sola vez
}
void loop() {
// Se ejecuta repetidamente
}
```

Estructuras de control

```
if (x < 5) { ... } else { ... }
while (x < 5) { ... }
do { ... } while ( x < 5);
for (int i = 0; i < 10; i++)
{ ... }
break;  //sale del bucle inmediatamente
continue; //va a la siguiente iteración
switch (miVariable) {
   case 1:
     ...
     break;
   case 2:
     ...
     break;
   default:
     ...
}
return x; // o "return;" para vacíos
```

Operadores

Operadores generales

```
=  (operador de asignación)
+  (adición)     -  (sustracción)
*  (multiplicación)
/  (división)    %  (módulo)
== (igual a)    != (desigual a)
<  (menor que)  >  (mayor que)
<= (igual o menor que)
>= (mayor o igual que)
&& (y)    || (ó)    ! (negación)
```

Operadores compuestos

```
++ (incremento)
-- (decremento)
+= (suma compuesta)
-= (resta compuesta)
*= (multiplicación compuesta)
/= (división compuesta)
&= (AND binario compuesto)
|= (OR binario compuesto)
```

Operadores a nivel de bit

```
& (AND binario)    | (OR binario)
^ (XOR binario)    ~ (NOT binario)
<< (desplazamiento a la izquierda)
>> (desplazamiento a la derecha)
```

Variables, Datos y Vectores

Conversiones

```
char()        byte()
int()         word()
long()        float()
```

Calificadores

```
static    //persiste entre llamadas
volatile  //usa la RAM
const     //sólo lectura
PROGMEM   //usar la flash
```

Punteros

```
&  (referencia: obtener puntero)
*  (valor: seguir puntero)
```

Constantes

```
HIGH | LOW
INPUT | OUTPUT
true | false
143          //Decimal
0173         //Octal (comenzando en 0)
0b11011111   //Binario
0x7B         //Hex (hexadecimal)
7U           //forzar unsigned
10L          //forzar long
15UL         //forzar long unsigned
10.0         //forzar floating point
2.4e5        //240000
```

Tipos de datos

```
void      vacío
boolean   (0, 1, true, false)
char      (ej. 'a'  -128 a 127)
int       (-32768 a 32767)
long      (-2147483648 a 2147483647)
unsigned char  (0 a 255)
byte           (0 a 255)
unsigned int   (0 a 65535)
word           (0 a 65535)
unsigned long  (0 a 4294967295)
float     (-3.4028e+38 a 3.4028e+38)
double    (igual que los flotantes)
```

Cadenas

```
char S1[8] =
 {'A','r','d','u','i','n','o'};
 //cadena sin terminación
 //puede producir error
char S2[8] =
 {'A','r','d','u','i','n','o','\0'};
 //incluye terminación nula \0
char S3[]="arduino";
char S4[8]="arduino";
```

Vectores y matrices

```
int myInts[6]; //vector de 6 enteros
int myPins[]={2, 4, 8, 3, 6};
int mySensVals[6]={2, 4, -8, 3, 2};
myInts[0]=42;  //asigna al primero
               //en el índice
myInts[6]=12;  //ERROR! El índice va
               //de 0 a 5
```

Funciones Incluidas

I/O Digital

```
pinMode(pin,[INPUT, OUTPUT])
digitalWrite(pin, valor)
int digitalread(pin)
  //Escribe HIGH en entradas para
  //usar los pull-ups
```

I/O Analógicas

```
analogReference([DEFAULT,
    INTERNAL, EXTERNAL])
int analogRead(pin)
analogWrite(pin, valor)  //PWM
```

Advanced I/O

```
tone(pin, freqhz)
tone(pin, freqhz, duracion_ms)
noTone(pin)
shiftOut(pinDatos, pinReloj,
  [MSBFIRST,LSBFIRST], valor)
unsigned long pulseIn(pin,
  [HIGH,LOW])
```

Tiempo

```
unsigned long millis()
  //desbordamiento en 50 dias
unsigned long micros()
  //desbordamiento en 70 minutos
delay(ms)
delayMicroseconds(us)
```

Matemáticas

```
min(x, y)   max(x, y)   abs(x)
sin(rad)    cos(rad)    tan(rad)
sqrt(x)     pow(base, exponente)
constrain(x, valMin, valMax)
map(val, deBAJO, deALTO,
aBAJO,aAlto)
```

Números aleatorios

```
randomSeed(semilla) //long ó int
long random(max)
long random(min, max)
```

Bits y Bytes

```
lowByte(x)    highByte(x)
bitRead(x, bitn)
bitWrite(x, bitn, bit)
bitSet(x, bitn)
bitClear(x, bitn)
bit(bitn)  // bitn: 0=LSB 7=MSB
```

Interrupciones Externas

```
attachInterrupt(interrup, func,
 [LOW, CHANGE, RISING, FALLING])
detachInterrupt(interrupción)
interrupts()
noInterrupts()
```

Bibliotecas

Serie

```
begin([300, 1200, 2400, 4800,
       9600, 14400, 19200, 28800,
       38400, 57600, 115200])
     //Puede ser cualquier número
end()
int available()
byte read()
byte peek()
flush()
print(misDatos)
println(misDatos)
write(misBytes)
flush()
```

EEPROM(#include <EEPROM.h>)

```
byte read(dirInterna)
write(dirInterna, miByte)
```

Servo (#include <Servo.h>)

```
attach(pin, [min_uS, max_uS])
write(ángulo) // 0, 180
writeMicroseconds(uS)
     //1000-2000; 1500 es en medio
read()       //0 - 180
attached()   //regresa boleano
detach()
```

SoftwareSerial(RxPin, TxPin)

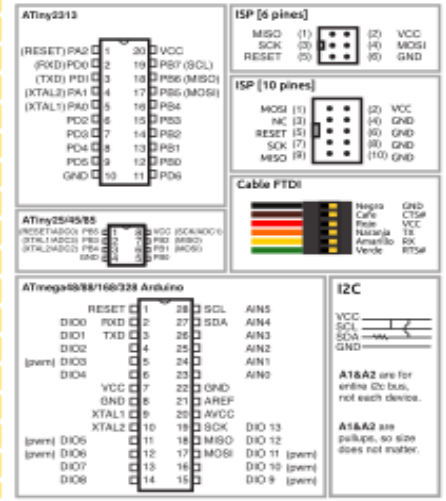
```
  (#include <softwareSerial.h>)
begin(long velocidad) //hasta 9600
char read()     //espera los datos
print(misDatos)
println(misDatos)
```

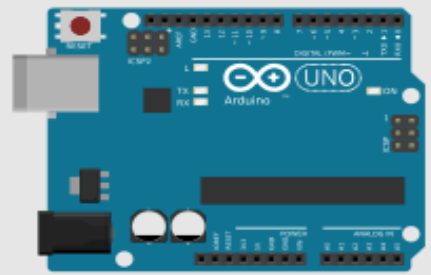
Wire(#include <Wire.h>) //para I²C

```
begin()        //se une a maestro
begin(addr)    //se une a esclavo @dir
requestFrom(dirección, cuenta)
beginTransmission(dir)  // Paso 1
send(miByte)            // Paso 2
send(char * miCadena)
send(byte * datos, tamaño)
endTransmission()       // Paso 3
byte available()   // Num de bytes
byte receive() //Regresa el sig byte
onReceive(manejador)
onRequest(manejador)
```

	ATMega168	ATMega328	ATMega1280
Flash (2k for boobtloader)	16kB	32kB	128kB
SRAM	1kB	2kB	8kB
EEPROM	512B	1kB	4kB

	Duemilanove/ Nano/ Pro/ ProMini	Mega
Pines Digitales	14 + 6 analog (Nano has 14 + 8)	54 + 16 analog
Pines Seriales	0 - RX 1 - TX	0 - RX1 1 - TX1 19 - RX2 18 - TX2 17 - RX3 16 - TX3 15 - RX4 14 - TX4
Interruptores Externos	2 - (Int 0) 1 - (Int 1)	2,3,21,20,19,18 (IRQ0 - IFQ5)
Pines PWM	5, 6 - Timer 0 9,10 - Timer 1 3,11 - Timer 2	0 - 13
ISP	10 - SS 11 - MOSI 12 - MISO 13 - SCK	53 - SS 51 - MOSI 50 - MISO 52 - SCK
I2C	Analog4 - SDA Analog5 - SCK	20 - SDA 21 - SCL





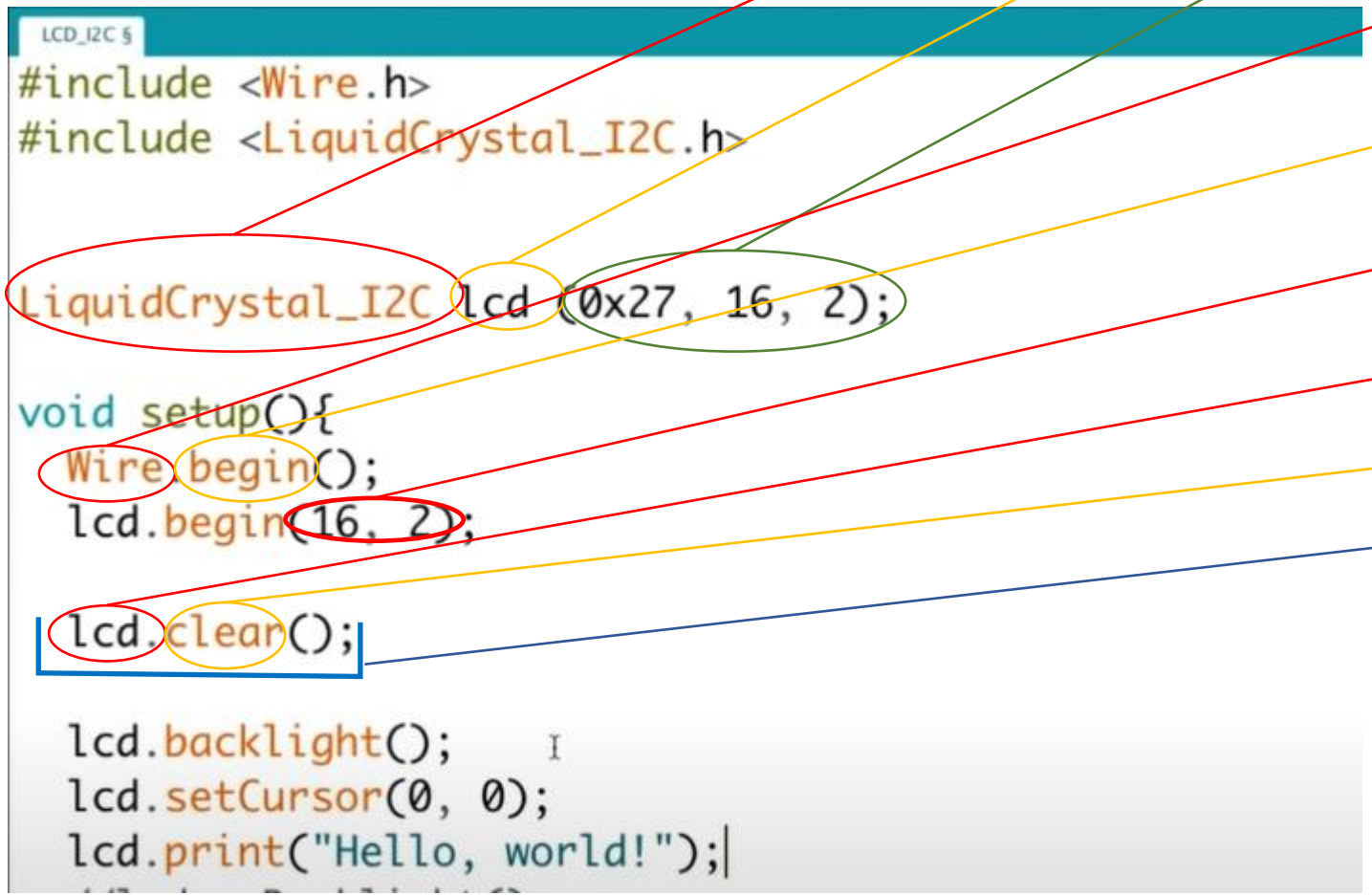
Arduino UNO R3 [Vector extraido de Fritzing]

Este trabajo está bajo licencia Atribución-CompartirIgual 3.0

- Adaptación por Liffiton
- Versión SVG por Frederic Dufourg
- Traducción al español de Antonio Maldonado
- Diseño y adaptación por Karla L. Hdz
- Inspirado en adaptación de Sparkfun Electronics
- Paleta de colores tomada del "Arduino Day"

MAS INFORMACIÓN EN:

NASER Electrónica



"Clase", definida por la Librería

"Objeto", de la Clase

"Constructor", del Objeto

"Clase"

"Método"

"Argumento"

"Objeto"

"Métodos de la Clase (de la librería I2C)"

Completo, es una "Instrucción"

PRIMEROS PASOS (Código)

1 LED INTERNO INTERMITENTE

camelCase

```
pinMode (13,OUTPUT);
digitalWrite (13, HIGH);
delay (1000);
digitalWrite (13, LOW);
delay (1000);
```

LED_BUILTIN

(pin, TIPO)

(pin, ESTADO)

Tiempo en mili Segundos.

Fin de Línea - Instrucción

INPUT
OUTPUT

Palabras Reservadas
Van en Mayúsculas

HIGH – 1 - true
LOW – 0 - false

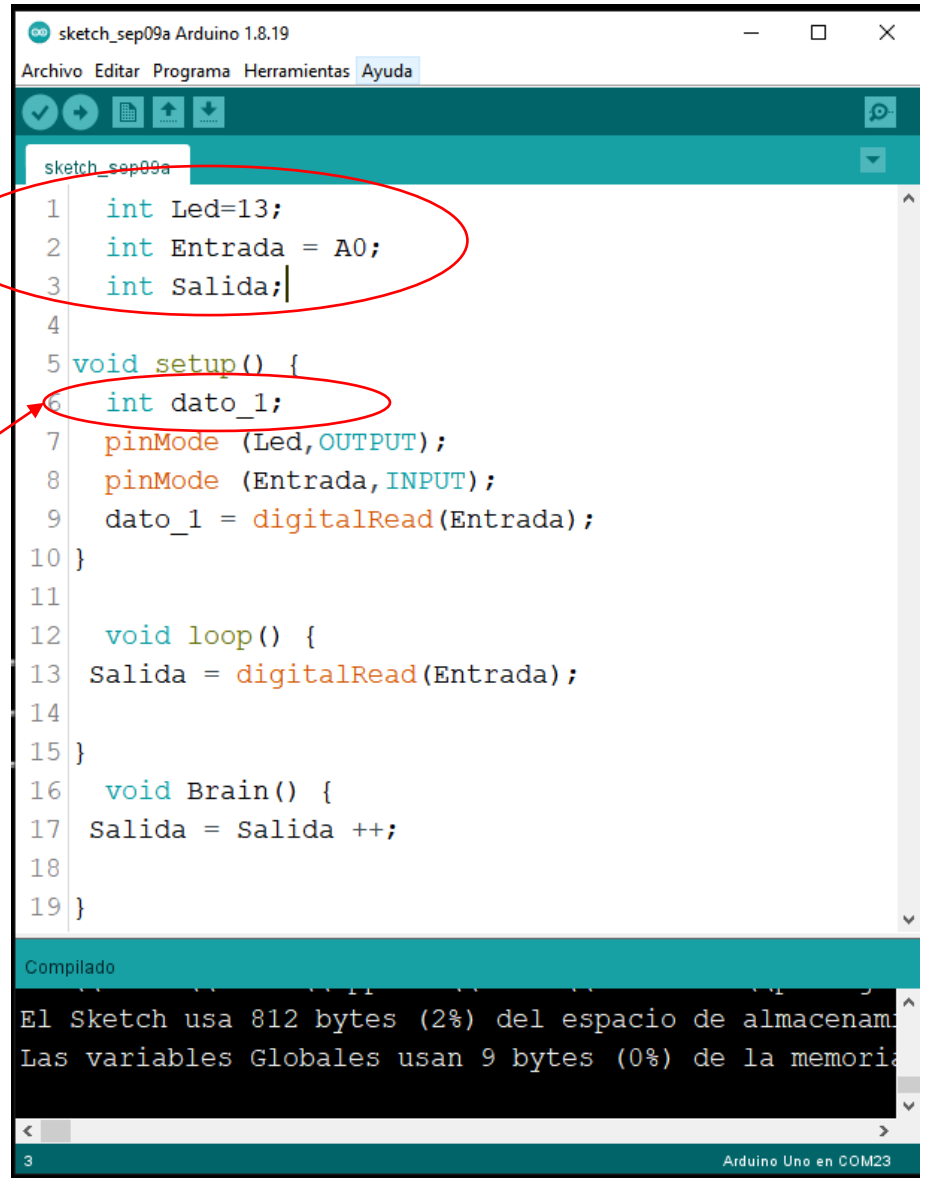
Cualquiera de esas alternativas es un Estado Alto - Verdadero

Cualquiera de esas alternativas es un Estado Bajo - Falso

REGISTROS
NO MODIFICABLES

VARIABLES

- **GLOBALES**
- **LOCALES**



ESTANDARES DE PROGRAMACIÓN
VARIABLES

Permiten tener una estructura clara, ordenada y que otras personas lo entiendan fácilmente

camelCase

Camel case (estilizado como **camelCase**) o **caso camello,** se aplica a frases o palabras compuestas.

La palabra inicia en minúsculas y la compuesta sigue con Mayúscula sin espacio

```
Int controlSwitch = 0
Int ledVerde=1

posicionMotor ()
```

snake_case

Snake Case se refiere al estilo de escritura en el que cada espacio se reemplaza con un carácter de subrayado y la primera letra de cada palabra se escribe en minúsculas.

```
Int control_switch = 0
Int led_verde=1

posicion_motor ()
```

NO USAR
NOMBRES CORTOS

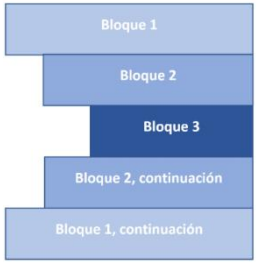
```
Int dist_rd

Int distancia_leida
```

ESTANDARES DE PROGRAMACIÓN
VARIABLES

INDENTACIÓN (Sangría)

Indentación es un anglicismo de uso común en informática.

es utilizar «sangrado», Indentar significa hacer espacios hacia la derecha para mover una línea de código



SEPARAR ACCIONES
EN FUNCIONES

Evitar hacer scroll en el código; un código no debería tener más de 30 líneas; todas las acciones comunes hacerlas como una **función** a llamar (invocar), también llamada **Sub Rutina.**

Ayuda a:
1.- Detectar errores rápidamente
2.- Facilita cambios al código.
3.- Utiliza menos memoria

MONITOR SERIAL

```
int A=10;
int B=2;

void setup() {
  Serial.begin (9600); // Inicializa el Monitor Serial

  Serial.print (“HOLA MUNDO");
  Serial.println (A+B);
 }

void loop() {

}
```

CONSTANTES
REGISTROS de MEMORIA NO MODIFICABLES

INPUT
OUTPUT

Definiciones de Tipo de Puertos

HIGH – 1 - true
LOW – 0 - false

Definiciones de niveles lógicos

```
pinMode (13, OUTPUT);
digitalWrite (13, HIGH);
```

CONSTANTES
REGISTROS de MEMORIA MODIFICABLES

boolean, byte, int , float, long, char… → Tipos de Variables…

- 8 Byte = 64 bit
- 4 Byte = 32 bit de 0 a 4´294´967´296; no tiene "**unsigned**"
- 4 Byte = 32 bit desde -32768 a +32768; **unsigned int** de 0 a 65535.
- 1 Byte = 8 bit
- 1 bite = 1 bit (TODO NÚMERO DISTINTO A CERO ES VERDADERO).

unsigned, static… → Características de Variables…

```
bool Estado = 1; // qué pasa con otros números?

void setup() {
  Serial.begin (9600);
  Serial.println (Estado);
}

void loop() {

}
```

VARIABLES

boolean= variable de 1 bit, sólo 2 estados; true – false.
(TODO NÚMERO DISTINTO A CERO ES VERDADERO).

int = Variable Enteros a 16 bit; desde -32768 a +32768; **unsigned int** de 0 a 65535.
(TODO NÚMERO DISTINTO A CERO ES VERDADERO).

float = Variable con decimales a 32 bit; de 0 a 4´294´967´296; no tiene "**unsigned**"

long= Variable de 64 bits (2 Bytes)

char = Variable de Caracter; ´a´ - B01100010 - 97 - 0x61; Arreglo de Variables; char texto[5] = {´B´, ´R´, ´A´,´I´, ´N´}

string = Variable para cadena de caracteres "ejemplo"; equivalente a char "ejemplo"

VARIABLES DE TIPOS PRIMITIVOS.

Nombre	Tipo	Tamaño	Valor por defecto	Forma de inicializar	Rango
Boolean	Lógico	1 bit	False	Boolean a=true	True-false
Char	Carácter	16 bits	Null	Char a='Z'	Unicode
Byte	Numero entero	8 bits	0	Byte a =0	-128 a 127
Short	Numero entero	16 bits	0	Short a =12	-32.768 a 32.767
Int	Numero entero	32 bit	0	Int a= 1250	-2.147.483.648 a 2.147.483.649
Long	Numero entero	64 bits	0	Long a= 125000	-9*10^18 a 9*10^18
Float	Numero real	32 bits	0	Float a =3.1	-3,4*10^38 a 3,4*10^38
Double	Numero real	64 bits	0	Double a = 125.2333	-1,79*10^308 a 1,79*10^308

OPERADORES ARITMETICOS

+	→	Suma
-	→	Resta
*	→	Multiplicación
/	→	División
%	→	Resto de una división

X=X+1 ⇨ X++

X=X-1 ⇨ X--

OPERADORES COMPUESTOS

X=2
Y=X++ ⇨ **Y=2 y X=3**

X=2
Y=++X ⇨ **X=3 e Y=3**

OPERADORES COMPARATIVOS

==	→	Igual qué (pregunta)
!=	→	Distinto que…
<	→	Menor que…
>	→	Mayor que…
<=	→	Menor o igual que…
>=	→	Mayor o igual que…Tipos

→ X != Y (x es distinto a Y)

OPERADORES LÓGICOS

!	→	NOT (negación del estado)
||	→	OR
&&	→	AND

CONTROL CONDICIONAL IF - ELSE

If (condición) {instrucciones}
If () {} else {}
If () {} else if {}
switch () {case: break;}

CONTROL CONDICIONAL SWITCH - CASE

```
switch (variable){
        case 1: // Código para la opción 1
          break;
        case 2: // Código para la opción 1
          break;
        case 3 // Código para la opción 1
         default: // Código en caso de que nada se cumpla
{
```

CONTROLES REPETITIVOS FOR y WHILE

```
for (;;) {}
```

```
for (inicialización; condición; Iteración)
{
Instrucciones
}
```

```
while () {}
```

```
while (condición)
{
Instrucciones
}
```

CONSTANTES

#define *nombre valor*

#define es un componente útil de C++ que permite al programador dar un nombre a un valor constante (MACRO) antes de compilar el programa. Las constantes definidas en arduino no ocupan espacio en la memoria del programa en el chip. El compilador reemplazará las referencias a estas constantes con el valor definido en el momento de la compilación.

#define led_verde 13

const *tipo nombre = valor*

La palabra **const** significa constante. Es un calificador de variable que modifica el comportamiento de la variable, haciendo que una variable sea " de solo lectura ". Esto significa que la variable se puede utilizar como cualquier otra variable de su tipo, pero su valor no se puede cambiar. Obtendrá un error del compilador si intenta asignar un valor a una variable **const**.

const float PI = 3,141592;

```
#define ledVerde 13
const float pi = 3.14;
float x;
// ....
x = pi * 2; // it's fine to use consts in math
pi = 7;     // illegal - you can't write to (modify) a constant
```

millis() - micros();

millis()

millis() devuelve el número de milisegundos desde que la placa Arduino empezó a ejecutar, luego de un reinicio o el encendido. Este número se desbordará (volverá a cero), después de aproximadamente 50 días. Retorna la cantidad de milisegundos en un valor long sin signo (unsigned long).

```
unsigned long currentTime=millis();
if(currentTime-previousTime>=interval){
  previousTime=millis();
  //Action
}
```

data type	bytes	min	max
char	1	-128	127
byte	1	0	255
int	2	-32768	32767
unsigned int	2	0	65535
long	4	-2147483648	2147483647
unsigned long	4	0	4294967295

MULTITASKING
Funciones VOID

VARIABLE char

char letra = A

Devuelve la Letra al Monitor Serial

```
char Letra = A;

void setup() {
  Serial.begin (9600);
  Serial.println (Letra);
}

void loop() {

}
```

Devuelve el código Hexa de la Letra A

```
char Letra = A;

void setup() {
  Serial.begin (9600);
  Serial.println (Letra, HEX);
}

void loop() {

}
```

El código ASCII

sigla en inglés de American Standard Code for Information Interchange
(Código Estadounidense Estándar para el Intercambio de Información)

Caracteres de control ASCII

DEC	HEX	Simbolo ASCII	
00	00h	NULL	(carácter nulo)
01	01h	SOH	(inicio encabezado)
02	02h	STX	(inicio texto)
03	03h	ETX	(fin de texto)
04	04h	EOT	(fin transmisión)
05	05h	ENQ	(enquiry)
06	06h	ACK	(acknowledgement)
07	07h	BEL	(timbre)
08	08h	BS	(retroceso)
09	09h	HT	(tab horizontal)
10	0Ah	LF	(salto de linea)
11	0Bh	VT	(tab vertical)
12	0Ch	FF	(form feed)
13	0Dh	CR	(retorno de carro)
14	0Eh	SO	(shift Out)
15	0Fh	SI	(shift In)
16	10h	DLE	(data link escape)
17	11h	DC1	(device control 1)
18	12h	DC2	(device control 2)
19	13h	DC3	(device control 3)
20	14h	DC4	(device control 4)
21	15h	NAK	(negative acknowle.)
22	16h	SYN	(synchronous idle)
23	17h	ETB	(end of trans. block)
24	18h	CAN	(cancel)
25	19h	EM	(end of medium)
26	1Ah	SUB	(substitute)
27	1Bh	ESC	(escape)
28	1Ch	FS	(file separator)
29	1Dh	GS	(group separator)
30	1Eh	RS	(record separator)
31	1Fh	US	(unit separator)
127	20h	DEL	(delete)

Caracteres ASCII imprimibles

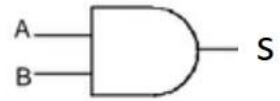
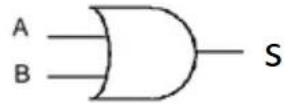
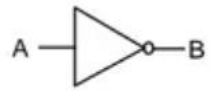
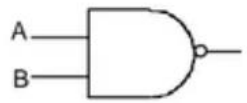
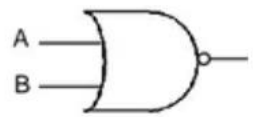
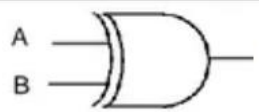
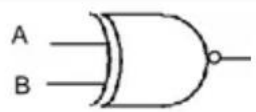
DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
32	20h	espacio	64	40h	@	96	60h	`
33	21h	!	65	41h	A	97	61h	a
34	22h	"	66	42h	B	98	62h	b
35	23h	#	67	43h	C	99	63h	c
36	24h	$	68	44h	D	100	64h	d
37	25h	%	69	45h	E	101	65h	e
38	26h	&	70	46h	F	102	66h	f
39	27h	'	71	47h	G	103	67h	g
40	28h	(	72	48h	H	104	68h	h
41	29h	)	73	49h	I	105	69h	i
42	2Ah	*	74	4Ah	J	106	6Ah	j
43	2Bh	+	75	4Bh	K	107	6Bh	k
44	2Ch	,	76	4Ch	L	108	6Ch	l
45	2Dh	-	77	4Dh	M	109	6Dh	m
46	2Eh	.	78	4Eh	N	110	6Eh	n
47	2Fh	/	79	4Fh	O	111	6Fh	o
48	30h	0	80	50h	P	112	70h	p
49	31h	1	81	51h	Q	113	71h	q
50	32h	2	82	52h	R	114	72h	r
51	33h	3	83	53h	S	115	73h	s
52	34h	4	84	54h	T	116	74h	t
53	35h	5	85	55h	U	117	75h	u
54	36h	6	86	56h	V	118	76h	v
55	37h	7	87	57h	W	119	77h	w
56	38h	8	88	58h	X	120	78h	x
57	39h	9	89	59h	Y	121	79h	y
58	3Ah	:	90	5Ah	Z	122	7Ah	z
59	3Bh	;	91	5Bh	[	123	7Bh	{
60	3Ch	<	92	5Ch	\	124	7Ch	|
61	3Dh	=	93	5Dh	]	125	7Dh	}
62	3Eh	>	94	5Eh	^	126	7Eh	~
63	3Fh	?	95	5Fh	_			

elCodigoASCII.com.ar

ASCII extendido

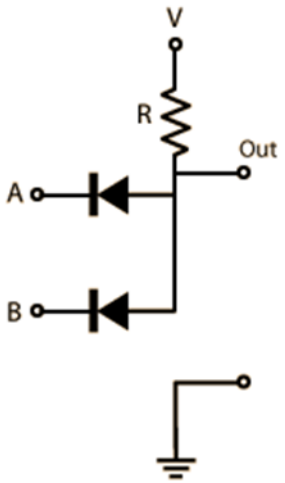
DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
128	80h	Ç	160	A0h	á	192	C0h	└	224	E0h	Ó
129	81h	ü	161	A1h	í	193	C1h	┴	225	E1h	ß
130	82h	é	162	A2h	ó	194	C2h	┬	226	E2h	Ô
131	83h	â	163	A3h	ú	195	C3h	├	227	E3h	Ò
132	84h	ä	164	A4h	ñ	196	C4h	─	228	E4h	õ
133	85h	à	165	A5h	Ñ	197	C5h	┼	229	E5h	Õ
134	86h	å	166	A6h	ª	198	C6h	ã	230	E6h	µ
135	87h	ç	167	A7h	º	199	C7h	Ã	231	E7h	þ
136	88h	ê	168	A8h	¿	200	C8h	╚	232	E8h	Þ
137	89h	ë	169	A9h	®	201	C9h	╔	233	E9h	Ú
138	8Ah	è	170	AAh	¬	202	CAh	╩	234	EAh	Û
139	8Bh	ï	171	ABh	½	203	CBh	╦	235	EBh	Ù
140	8Ch	î	172	ACh	¼	204	CCh	╠	236	ECh	ý
141	8Dh	ì	173	ADh	¡	205	CDh	═	237	EDh	Ý
142	8Eh	Ä	174	AEh	«	206	CEh	╬	238	EEh	¯
143	8Fh	Å	175	AFh	»	207	CFh	¤	239	EFh	´
144	90h	É	176	B0h	░	208	D0h	ð	240	F0h	
145	91h	æ	177	B1h	▒	209	D1h	Ð	241	F1h	±
146	92h	Æ	178	B2h	▓	210	D2h	Ê	242	F2h	‗
147	93h	ô	179	B3h	│	211	D3h	Ë	243	F3h	¾
148	94h	ò	180	B4h	┤	212	D4h	È	244	F4h	¶
149	95h	ò	181	B5h	Á	213	D5h	ı	245	F5h	§
150	96h	û	182	B6h	Â	214	D6h	Í	246	F6h	÷
151	97h	ù	183	B7h	À	215	D7h	Î	247	F7h	¸
152	98h	ÿ	184	B8h	©	216	D8h	Ï	248	F8h	°
153	99h	Ö	185	B9h	╣	217	D9h	┘	249	F9h	¨
154	9Ah	Ü	186	BAh	║	218	DAh	┌	250	FAh	·
155	9Bh	ø	187	BBh	╗	219	DBh	█	251	FBh	¹
156	9Ch	£	188	BCh	╝	220	DCh	▄	252	FCh	³
157	9Dh	Ø	189	BDh	¢	221	DDh	¦	253	FDh	²
158	9Eh	×	190	BEh	¥	222	DEh	Ì	254	FEh	■
159	9Fh	ƒ	191	BFh	┐	223	DFh	▀	255	FFh	

OPERADORES LOGICOS

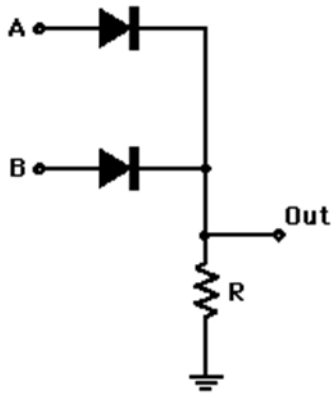
Función	Símbolo	Notación	Tabla de verdad
AND		$C = A \cdot B$	A B S 0 0 0 0 1 0 1 0 0 1 1 1
OR		$C=A+B$	A B S 0 0 0 0 1 1 1 0 1 1 1 1
NOT		$B \quad \overline{A}$	A B 0 1 1 0
NAND		$C \quad \overline{A \; B}$	A B S 0 0 1 0 1 1 1 0 1 1 1 0
NOR		$C \quad \overline{A \quad B}$	A B S 0 0 1 0 1 0 1 0 0 1 1 0
EXOR		$C \quad \overline{A}B \quad A\overline{B}$ $C \quad A \quad B$	A B S 0 0 0 0 1 1 1 0 1 1 1 0
NOR exclusiva		$C \quad \overline{A} \; \overline{B} \quad A \; B$ $C \quad \overline{A \quad B}$	A B S 0 0 1 0 1 0 1 0 0 1 1 1

PUERTAS AND y OR CON DIODOS

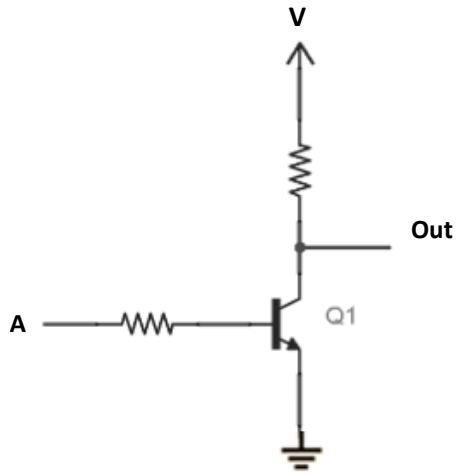
Diode AND Gate



Diode OR Gate

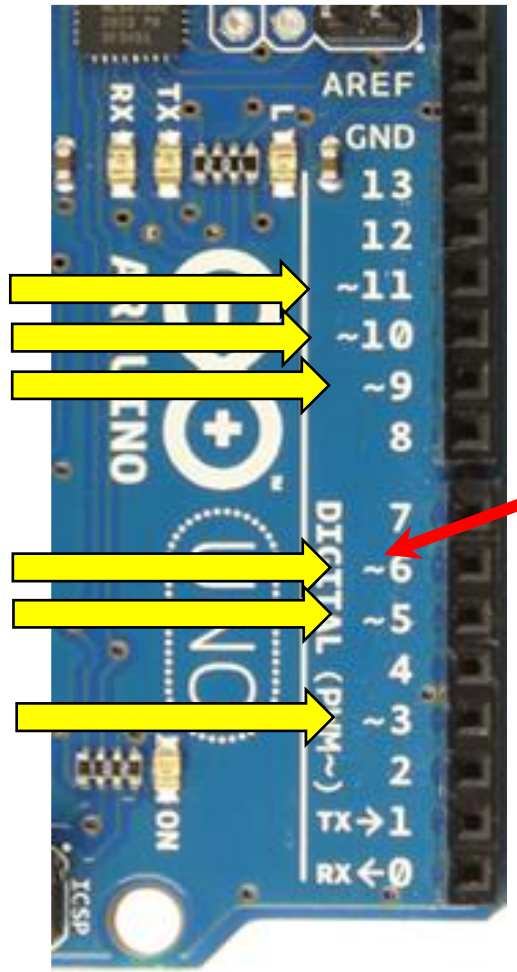


Transistor NOT



PWM = PULSE WIDHT MODULATION
Modulación de Ancho de Pulso

ARDUINO UNO



~ = VIRGULILLA

PWM = PULSE WIDHT MODULATION
Modulación de Ancho de Pulso

ARDUINO UNO

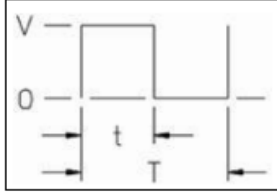
8 bits; 256 valores
0 = 0% Duty Cycle
255 = 100% Duty Cyce

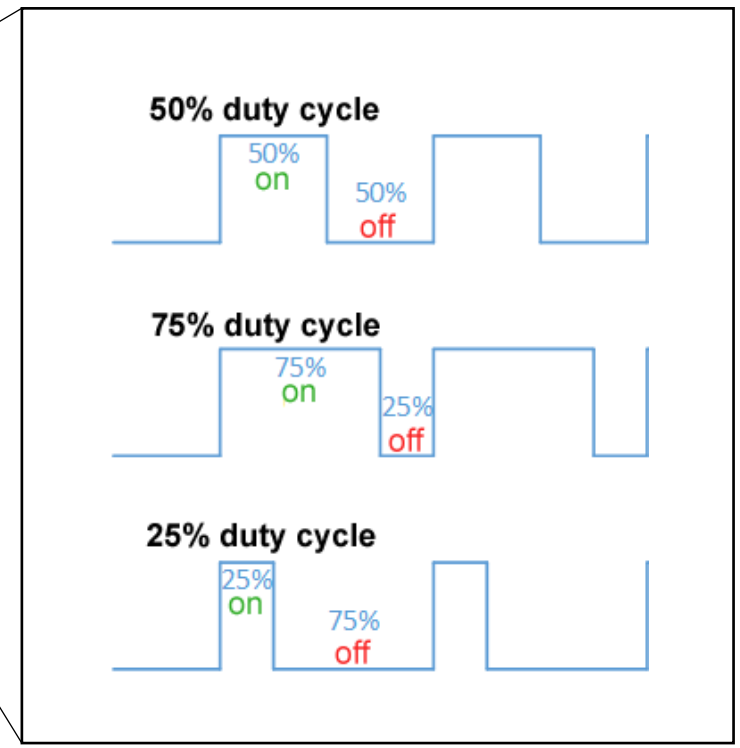
analogWrite (Dx, **valor**);

980 Hz.

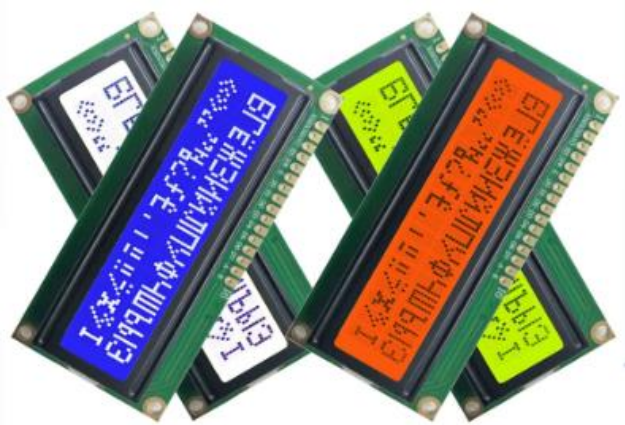
490 Hz.

PWM = PULSE WIDHT MODULATION
Modulación de Ancho de Pulso

Waveform	Crest Factor (C.F.)	AC RMS	AC+DC RMS
	$\sqrt{2}$	$\frac{V}{\sqrt{2}}$	$\frac{V}{\sqrt{2}}$
	$\sqrt{3}$	$\frac{V}{\sqrt{3}}$	$\frac{V}{\sqrt{3}}$
	$\sqrt{\frac{T}{t}}$	$\frac{V}{C.F.} \times \sqrt{1-\left(\frac{1}{C.F.}\right)^2}$	$\frac{V}{C.F.}$

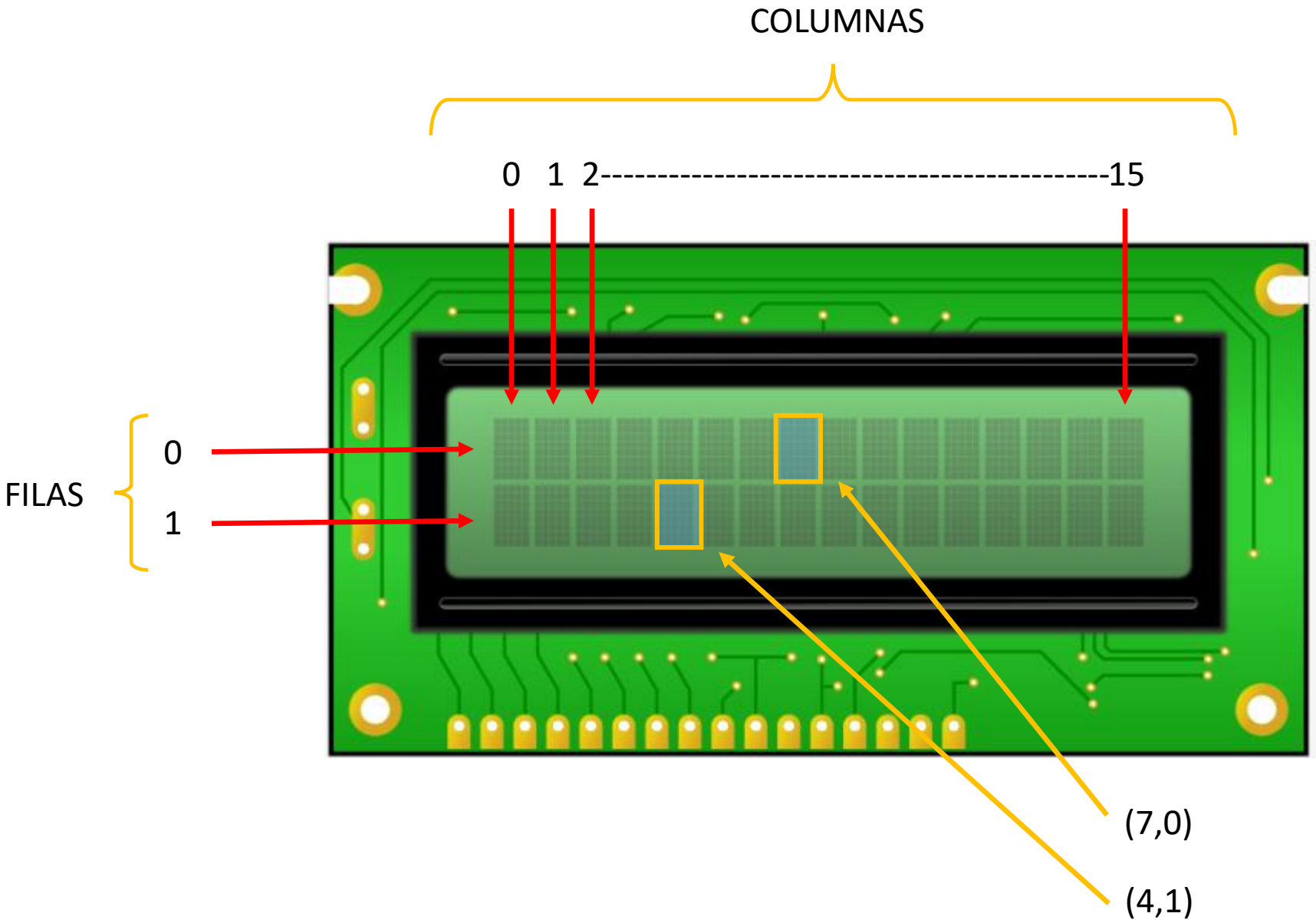


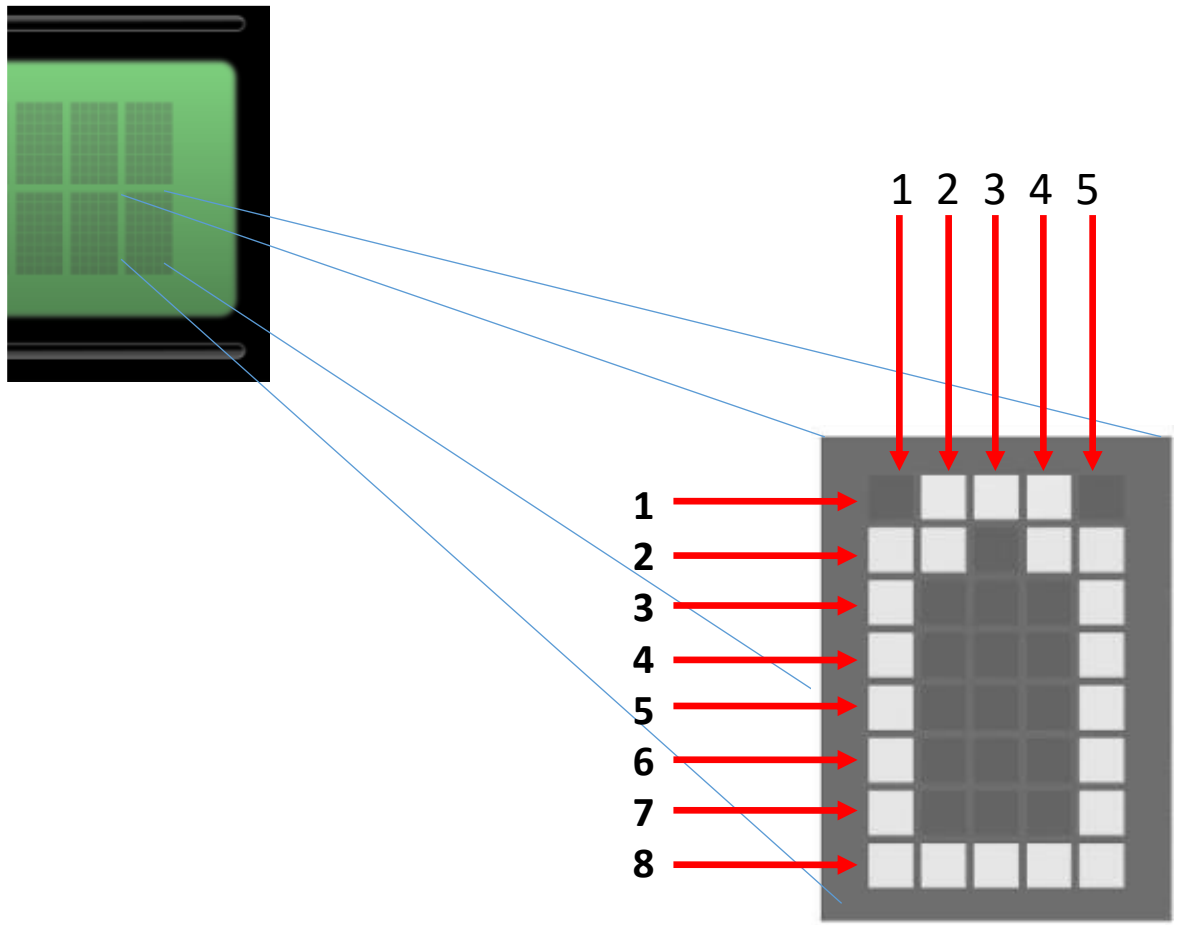
DISPLAY de CRISTAL LIQUIDO
LCD

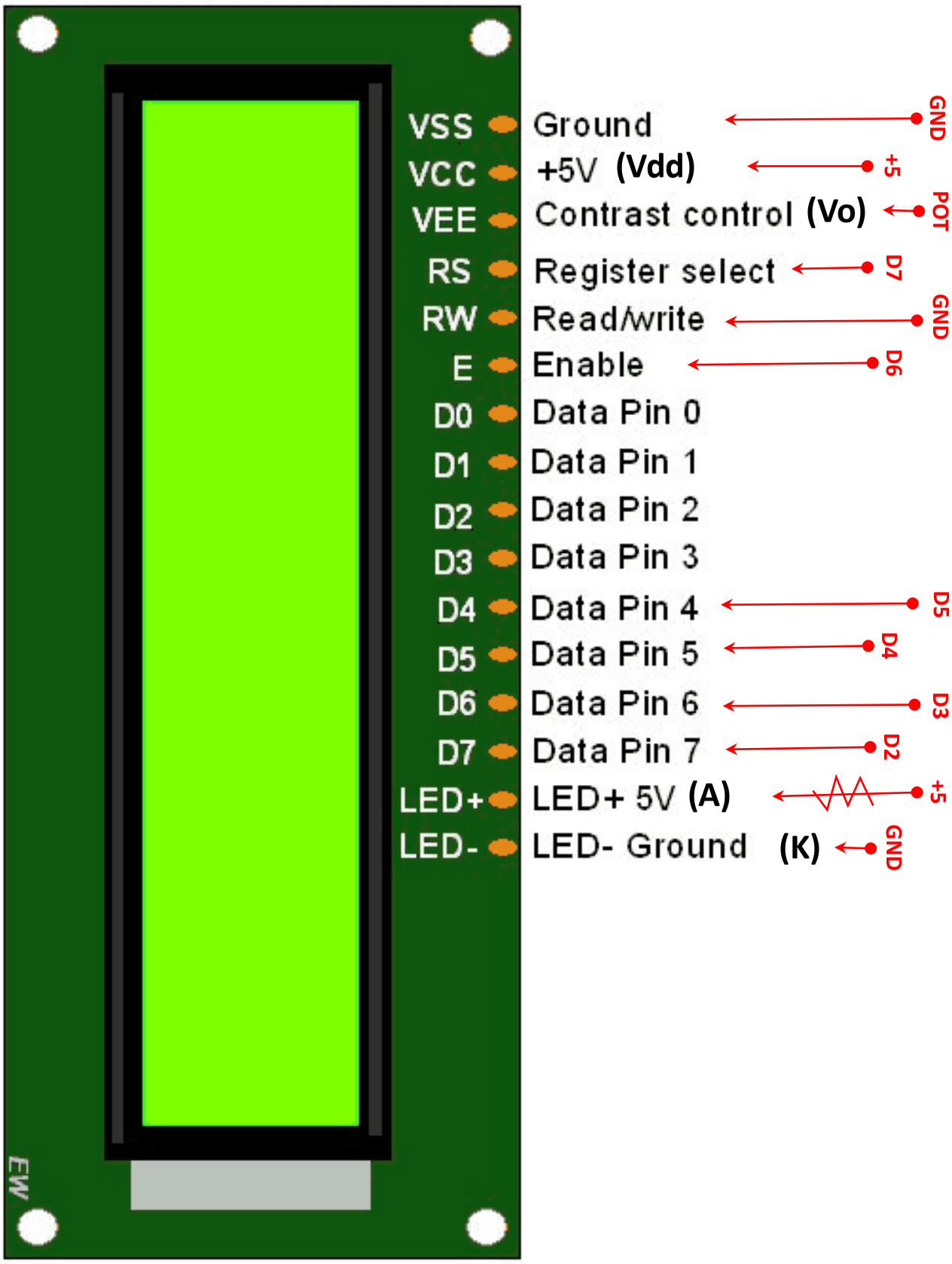


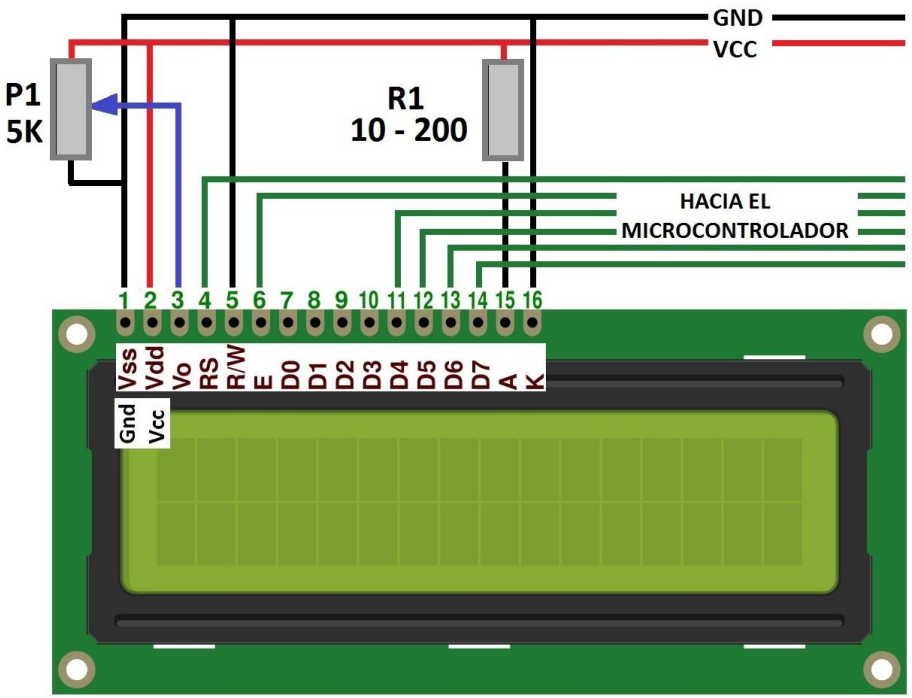
18 x 2

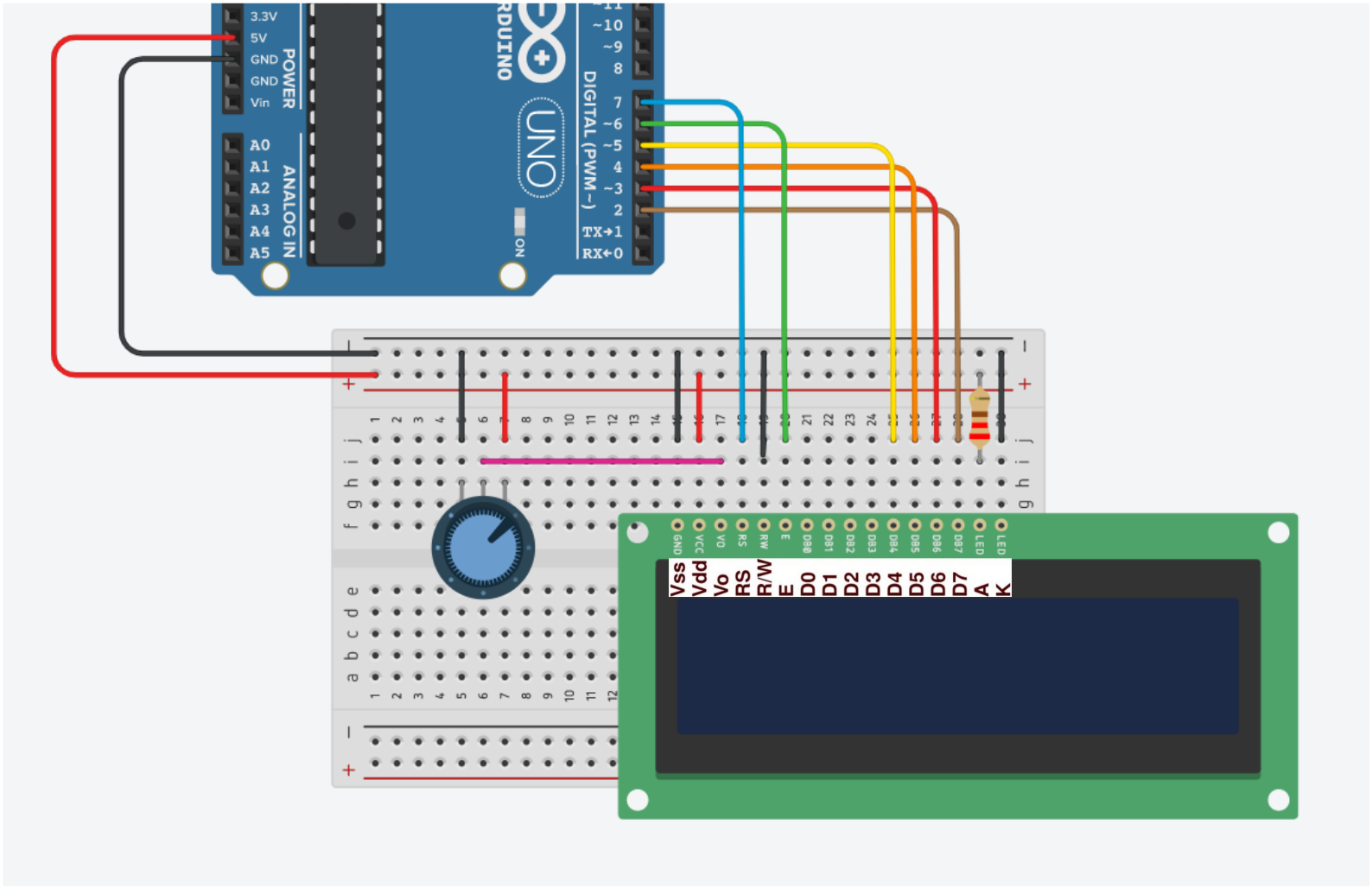
20 x 4


COLUMNAS
0 1 2----------15
FILAS
0
1
(7,0)
(4,1)


1 2 3 4 5
1
2
3
4
5
6
7
8


VSS
VCC
VEE
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7
LED+
LED-
Ground
+5V (Vdd)
Contrast control (Vo)
Register select
Read/write
Enable
Data Pin 0
Data Pin 1
Data Pin 2
Data Pin 3
Data Pin 4
Data Pin 5
Data Pin 6
Data Pin 7
LED+ 5V (A)
LED- Ground (K)
GND
+5
POT
D7
GND
D6
D5
D4
D3
D2
+5
GND
EW


GND
VCC
P1
5K
R1
10 - 200
HACIA EL
MICROCONTROLADOR
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Vss Vdd Vo RS R/W E D0 D1 D2 D3 D4 D5 D6 D7 A K
Gnd Vcc

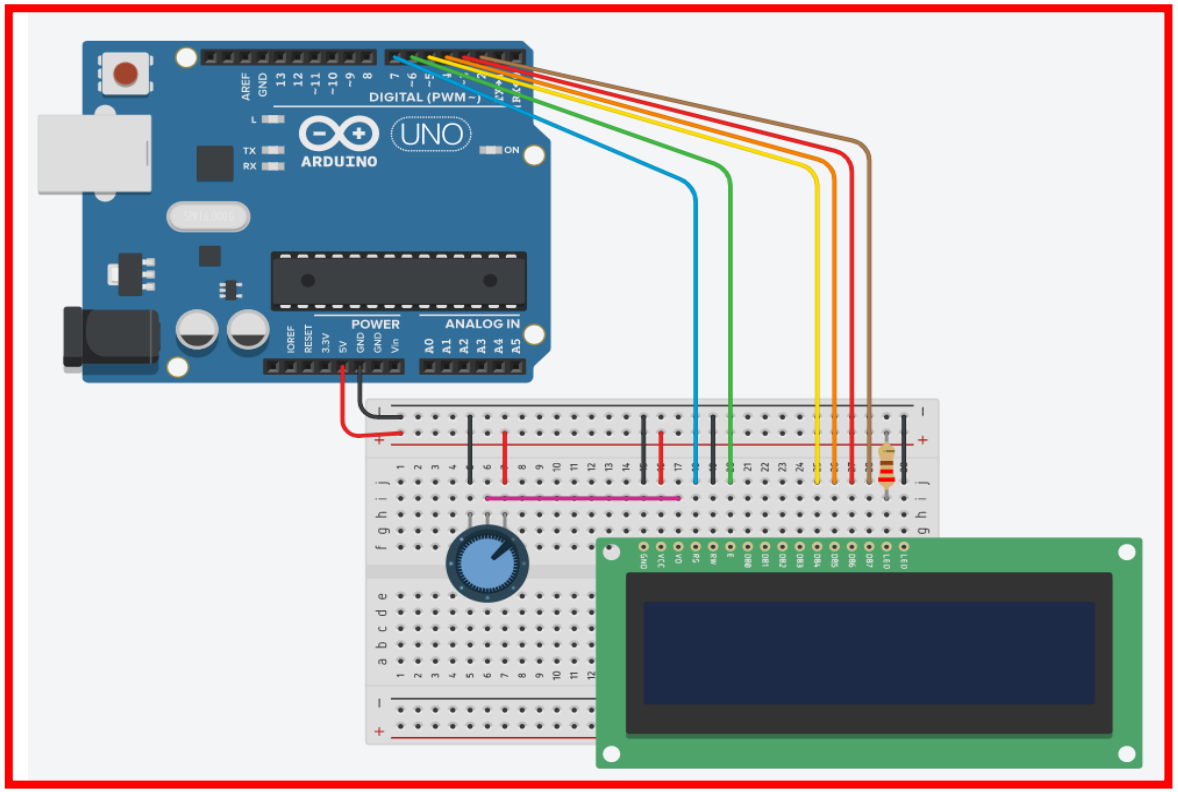

3.3V
5V
GND
GND
Vin
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
UNO
ON
DIGITAL (PWM~)
TX→1
RX←0
Vss Vdd Vo RS R/W E D0 D1 D2 D3 D4 D5 D6 D7 A K

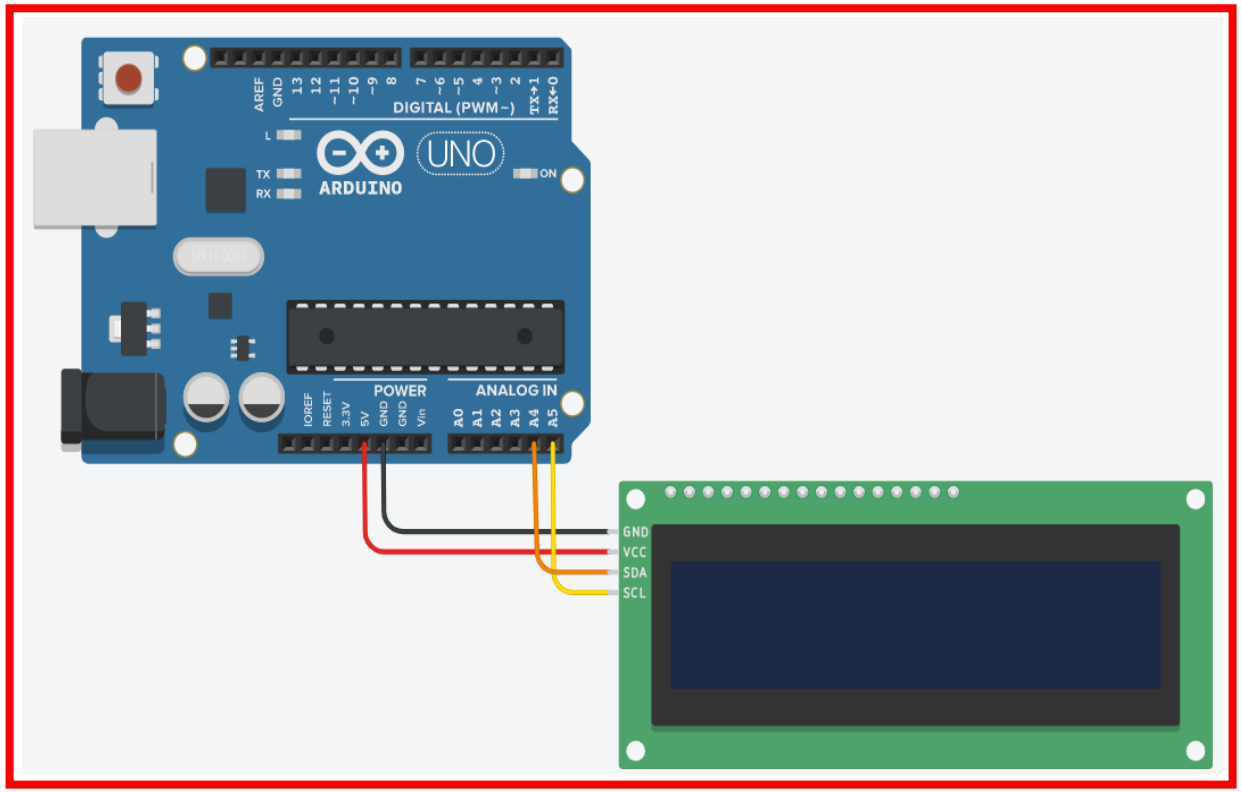
LCD CON COMUNICACIÓN PARALELA

LiquidCrystal lcd(RS, EN, D4, D5, D6, D7)
Definición de pines a utilizar, crea una variable de clase LiquidCrystal, con los pines indicados.
Begin (cols, rows) Inicializa el LCD, especificando el número de columnas (cols) y filas (rows) del LCD.

lcd.clear(); Borra la pantalla LCD y posiciona el cursor en la esquina superior izquierda (posición (0,0)).
lcd.setCursor(col, row); Posiciona el cursor del LCD en la posición indicada por col y row (x,y).
lcd.cursor(); Especifica incluir cursor
lcd.noCursor(); Especifica no incluir cursor
lcd.blink(); Parpadeo del Cursor:
lcd.noBlink(); No parpadeo del Cursor:
lcd.write(); Escribir un carácter en la pantalla LCD, en la ubicación actual del cursor.
lcd.print(); Escribe un texto o mensaje en el LCD, su uso es similar a un Serial.print
lcd.scrollDisplayLeft(); Se desplaza el contenido de la pantalla (texto y el cursor) un espacio hacia la izquierda.
lcd.scrollDisplayRight(); Se desplaza el contenido de la pantalla (texto y el cursor) un espacio a la derecha.
lcd.createChar (num, datos); Crea un carácter personalizado para su uso en la pantalla LCD. Se admiten hasta ocho caracteres de 5x8 píxeles (numeradas del 0 al 7). Donde: num es el número de carácter y datos es una matriz que contienen los pixeles del carácter.

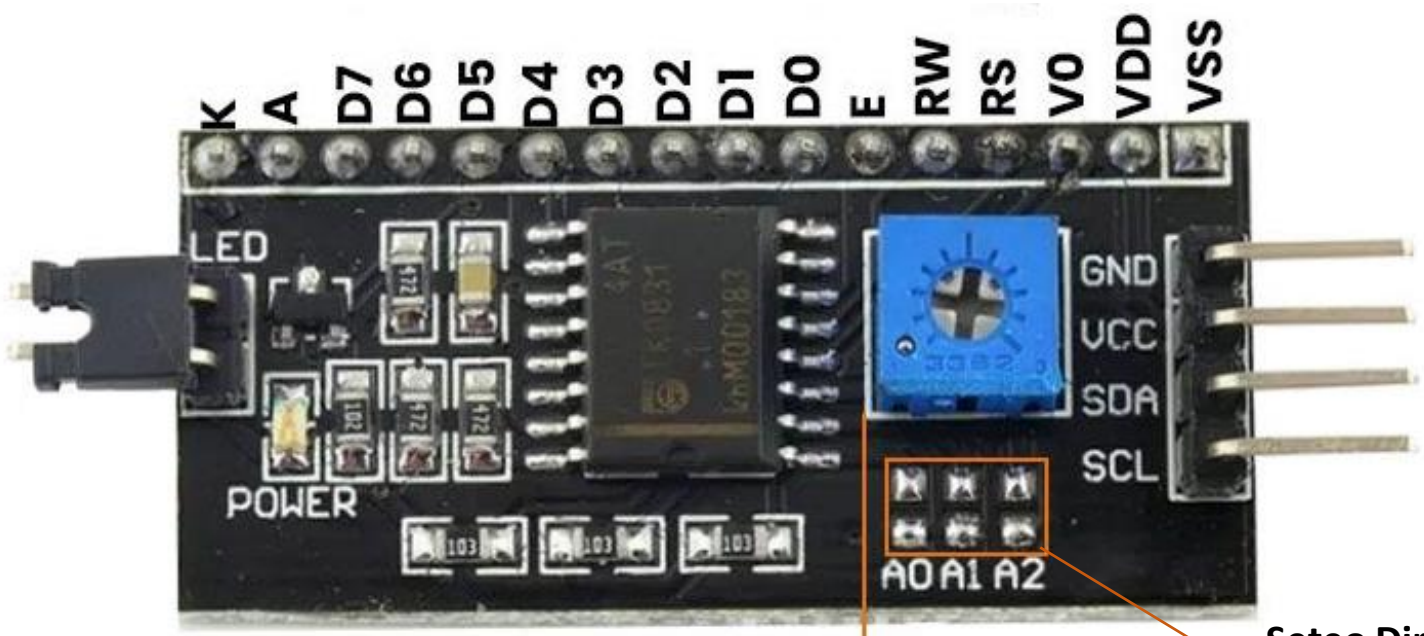
LCD CON COMUNICACIÓN PARALELA vs I2C





MODULO I2C PCF8574

STRAP
BACKLIGHT ON



Al Arduino

Trimpot
Ajuste Contraste

Seteo Dirección del Módulo

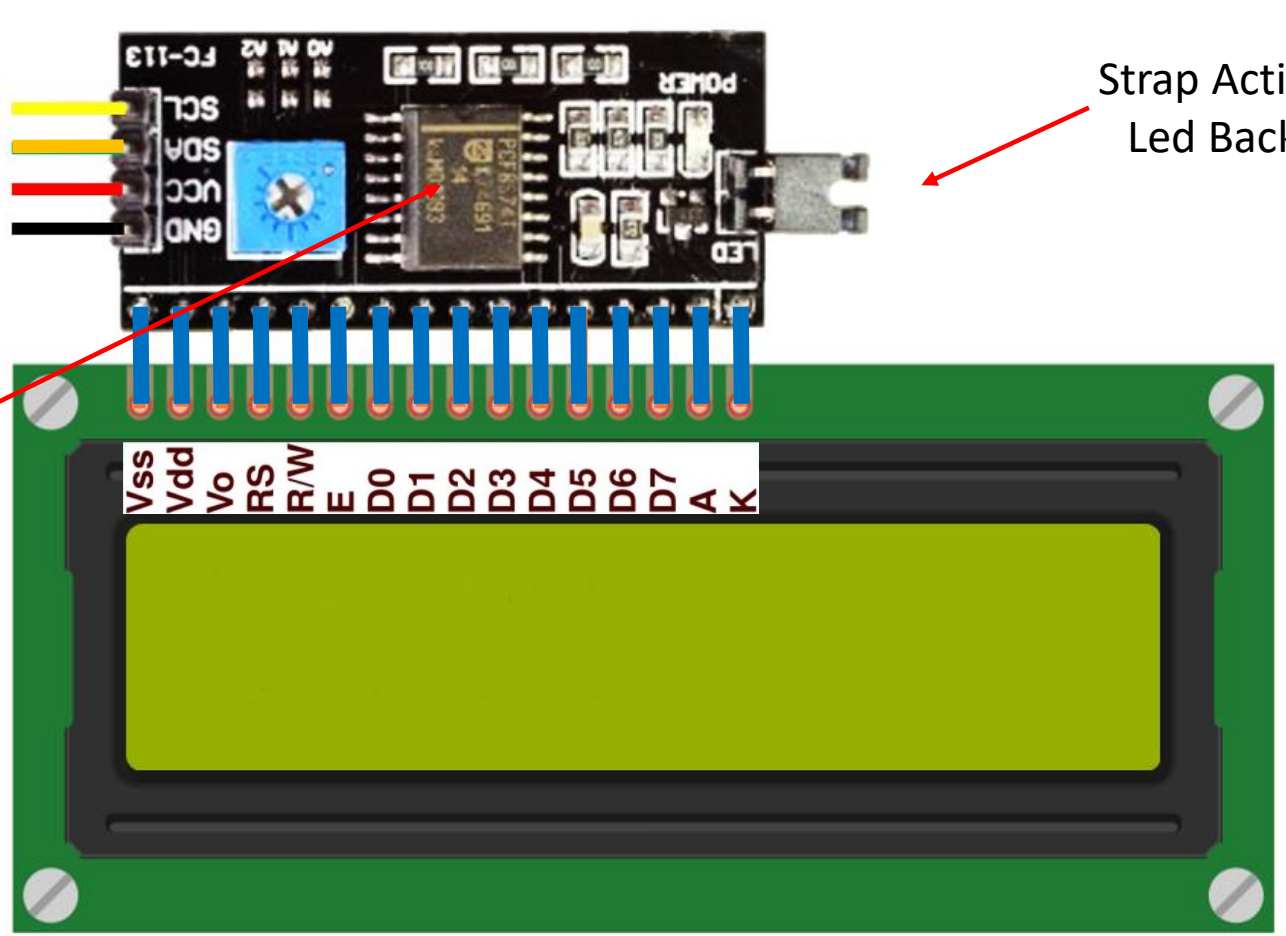
ENTRADAS			DIRECCION I2C
A2	A1	A0	
0	0	0	0X27 <Default>
0	0	1	0X26
0	1	0	0X25
0	1	1	0X24
1	0	0	0X23
1	0	1	0X22
1	1	0	0X21
1	1	1	0X20

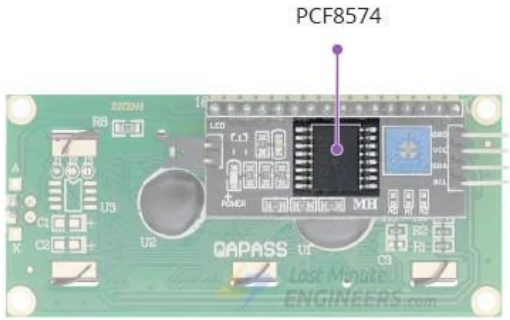
MODULO I2C

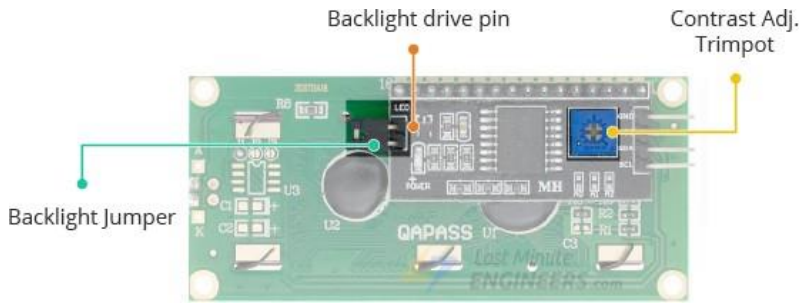
Al ARDUINO

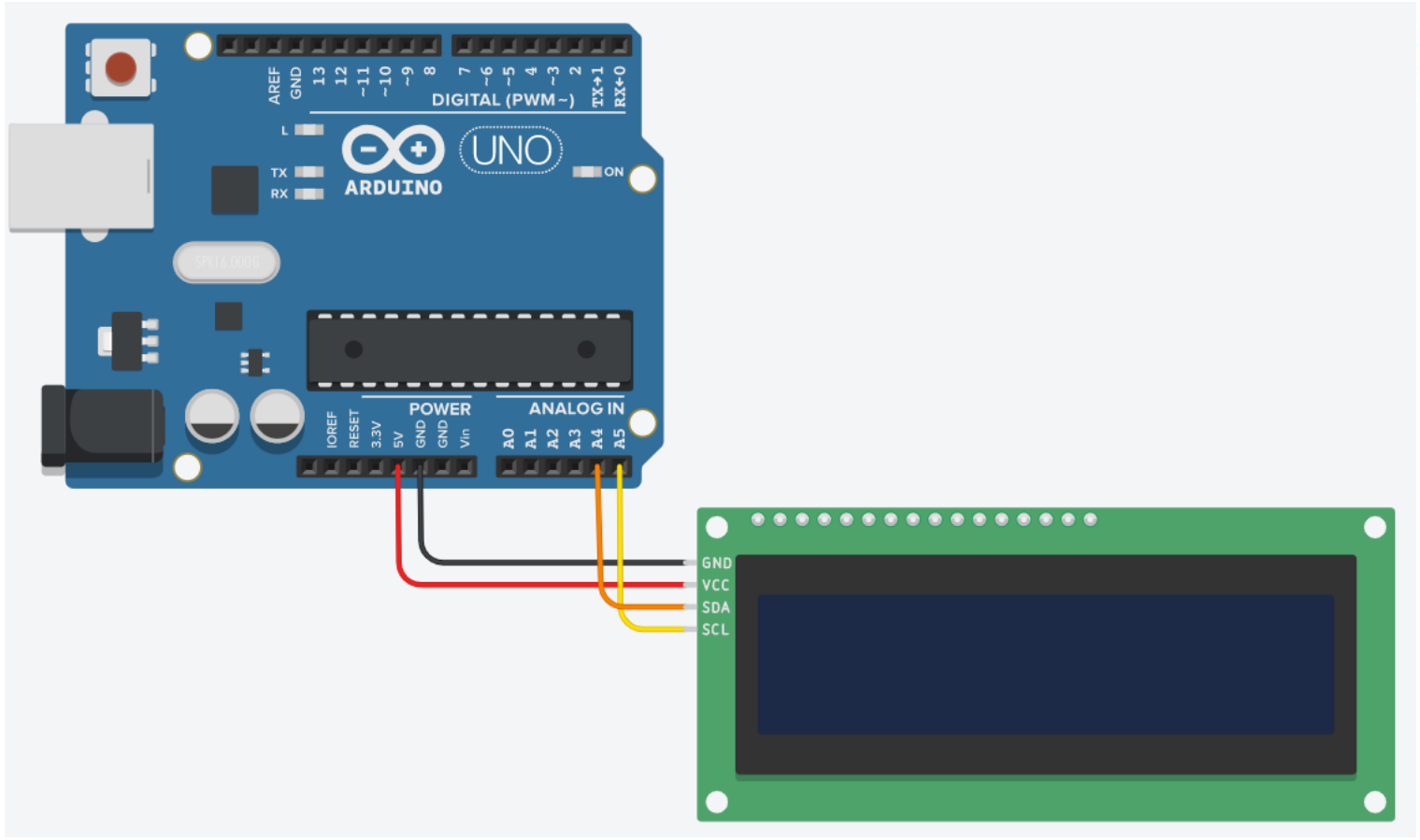
PCF8574

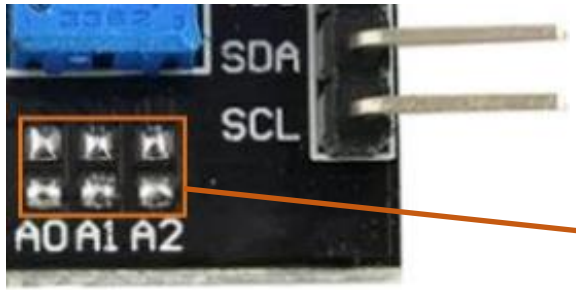
Strap Activación
Led Backlight

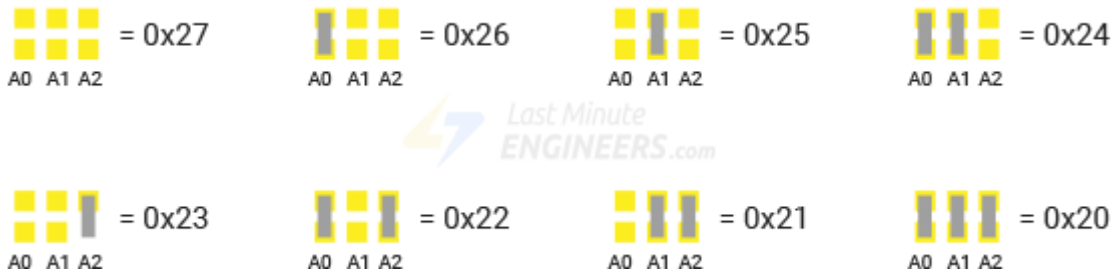



PCF8574


Backlight drive pin
Contrast Adj. Trimpot
Backlight Jumper


DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN
GND
VCC
SDA
SCL


SDA
SCL
A0 A1 A2

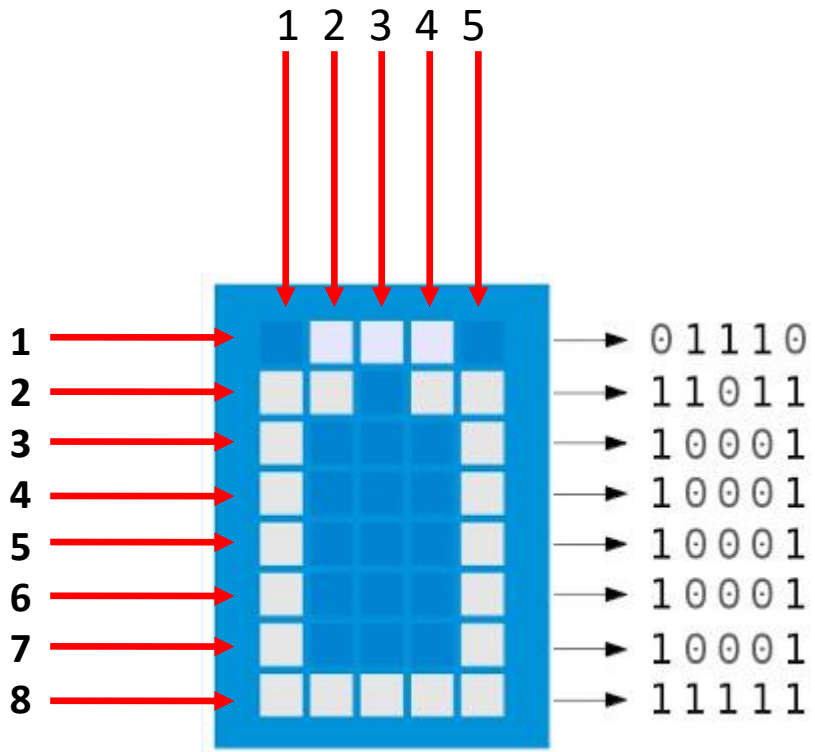

= 0x27
= 0x26
= 0x25
= 0x24
= 0x23
= 0x22
= 0x21
= 0x20
A0 A1 A2

LCD CON COMUNICACIÓN I2C

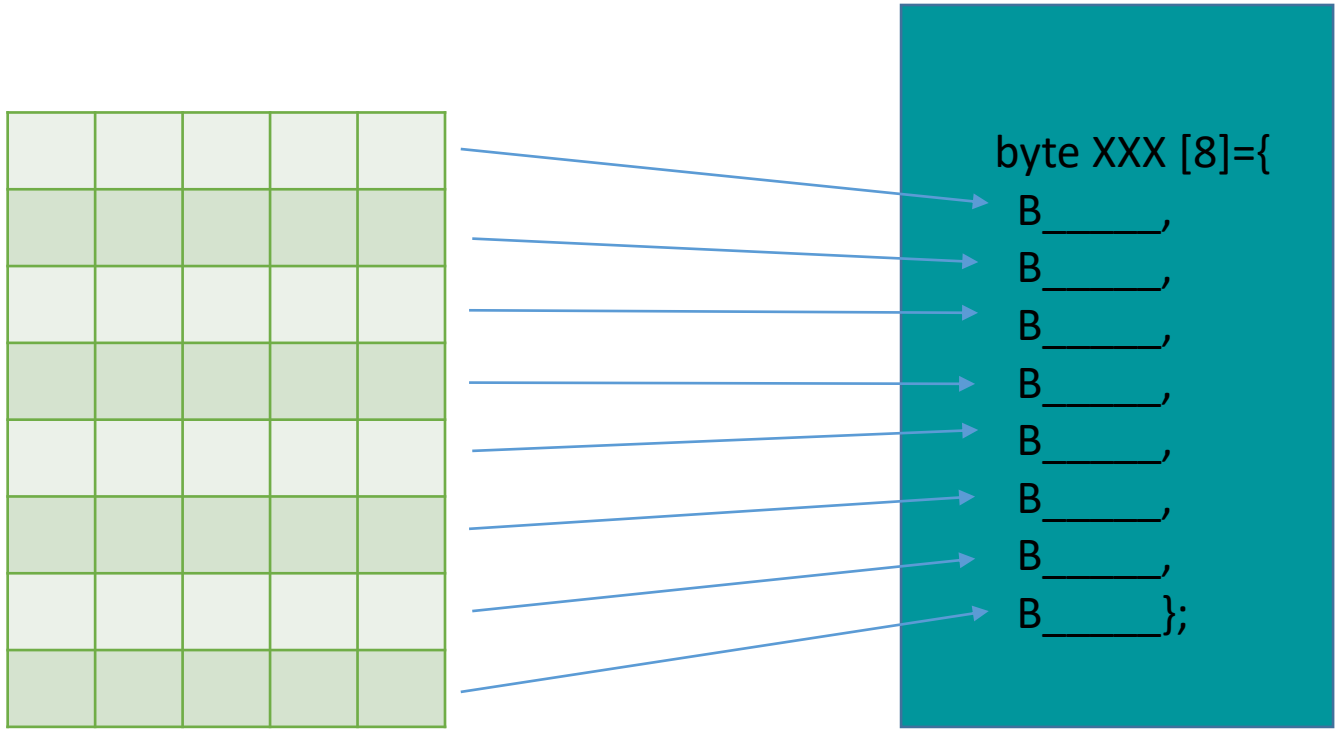
LiquidCrystal_I2C(lcd_Addr, cols, rows)
Declaraciones iniciales del LCD con I2C; Dirección, Columnas, Filas)

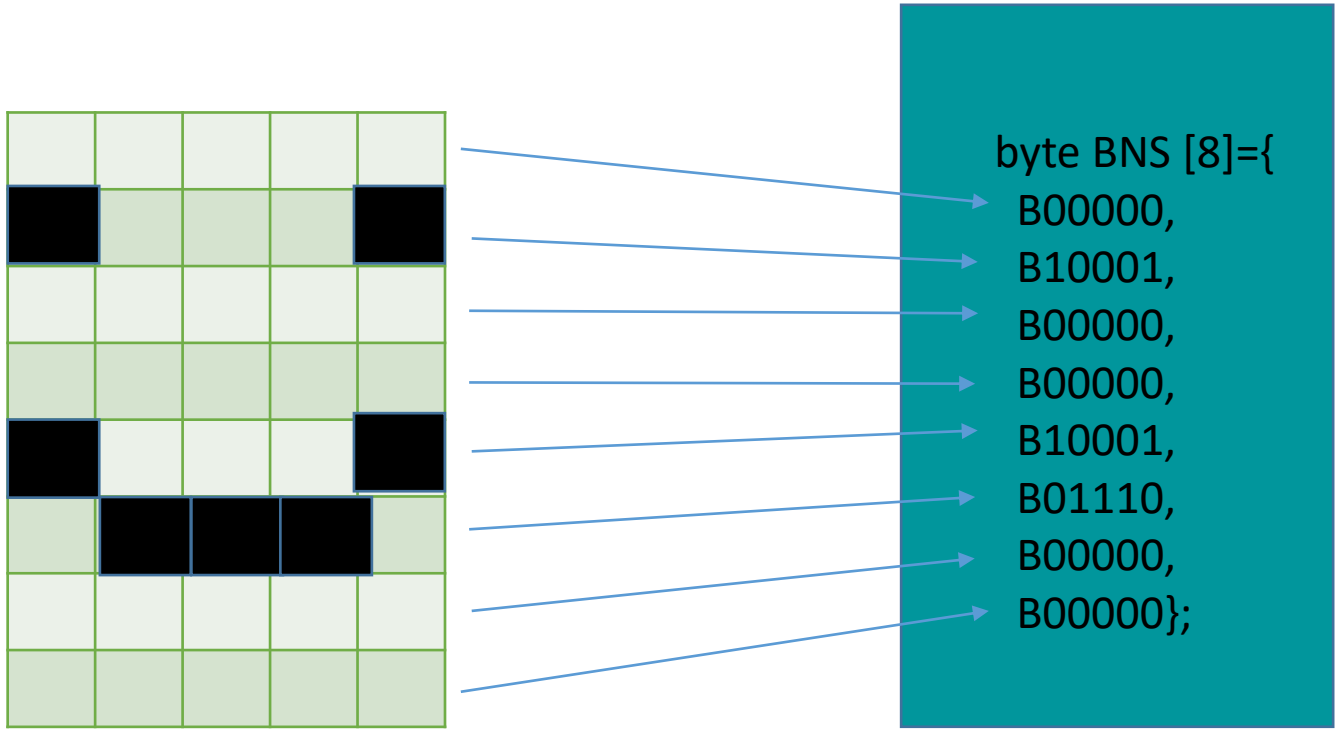
init();	Inicializa el modulo I2C, esta función configura e inicializa el I2C y el LCD.
clear();	Borra la pantalla LCD y posiciona el cursor en posición (0,0).
setCursor(col, row);	Posiciona el cursor en la posición indicada por col y row(x,y).
print();	Escribe un texto o mensaje en el LCD, es similar a un Serial.print
cursor();	Hace visible la posición del Cursor.
noCursor();	Oculta la posición del Cursor.
scrollDisplayLeft();	Desplaza el contenido de la pantalla hacia la izquierda.
scrollDisplayRight();	Desplaza el contenido de la pantalla a la derecha.
autoscroll();	Realiza un scroll en caso de ser requerido.
noAutoscroll();	No reliza un auto scroll, aún cuando parte del texto quede fuera.
leftToRight();	Presentación del texto de Izquierda a Derecha.
rightToLeft();	Presentación del texto de Derecha a Izquierda.
backlight();	Enciende la Luz del Fondo del LCD.
noBacklight();	Apaga la Luz del Fondo del LCD.
blink();	Parpadeo del cursor.
noBlink();	Cancela parpadeo del Cursor.
display();	Activa Pantalla.
noDisplay();	Desactiva Pantalla.

createChar (num, datos); Crea un carácter personalizado para su uso en la pantalla LCD. Se admiten hasta ocho caracteres de 5x8 píxeles (numeradas del 0 al 7). Dónde: **num** es el número de carácter y **datos** es una matriz que contienen los pixeles del carácter.




```
byte BAT [8]={
  B01110,
  B11011,
  B10001,
  B10001,
  B10001,
  B10001,
  B10001,
  B11111};
```

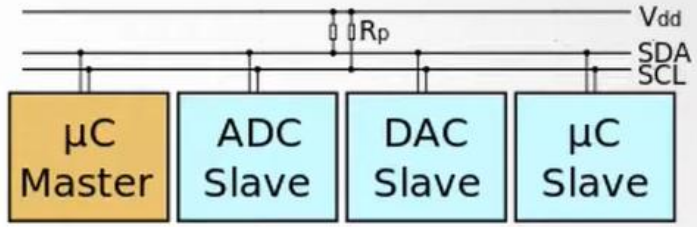

byte XXX [8]={
B_____,
B_____,
B_____,
B_____,
B_____,
B_____,
B_____,
B_____};


byte BNS [8]={
B00000,
B10001,
B00000,
B00000,
B10001,
B01110,
B00000,
B00000};

PROTOCOLO I2C

I²C Inter-Integrated Circuit

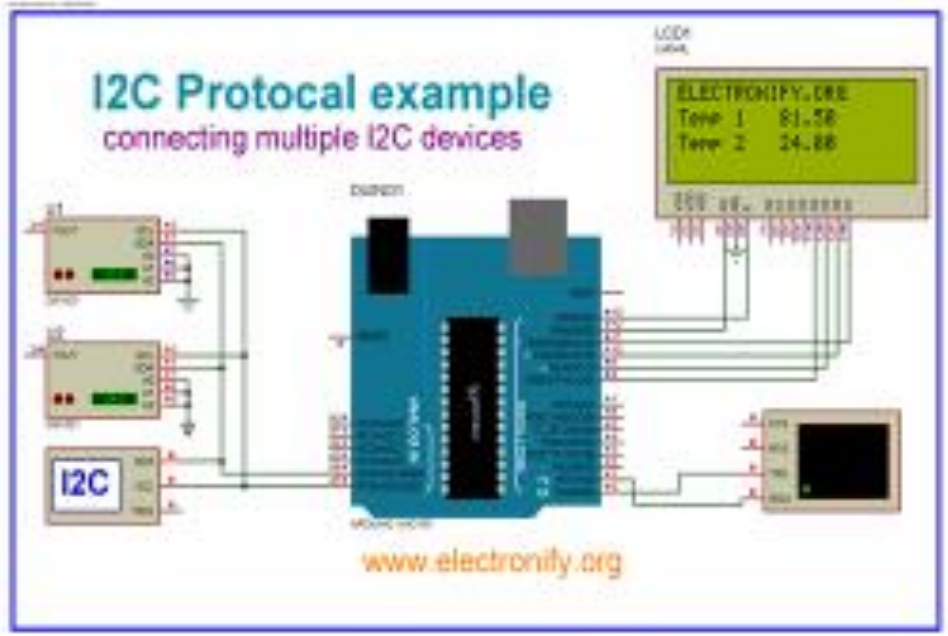
FUNDAMENTOS DE I2C

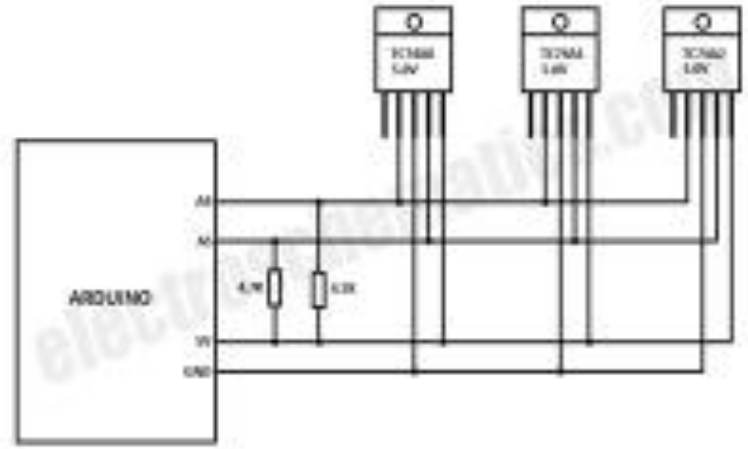


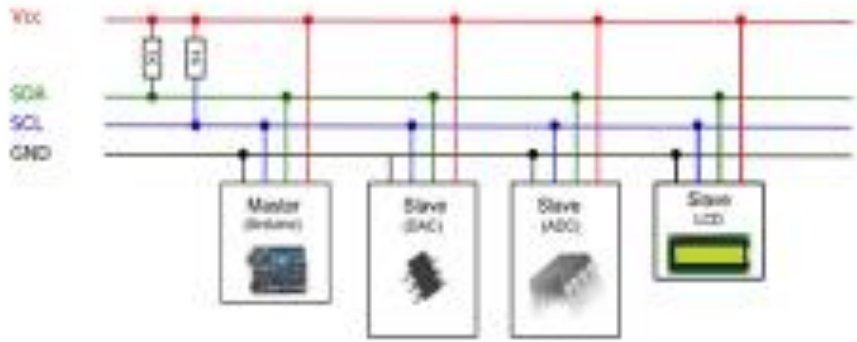
- Desarrollado en 1982 por Philips
- Maestro-Esclavo (Uno o multiples maestros controlando uno o varios esclavos)
- Dos líneas de señal:
  - **SDA (Serial Data)**
  - **SCL (Serial Clock)**

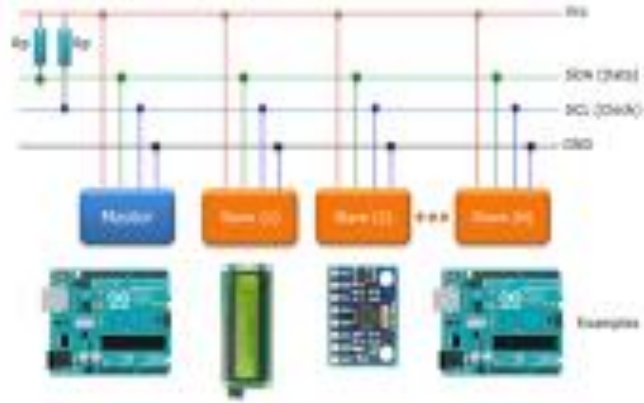
Wires Used	2
Maximum Speed	Standard mode= 100 kbps
	Fast mode= 400 kbps
	High speed mode= 3.4 Mbps
	Ultra fast mode= 5 Mbps
Synchronous or Asynchronous?	Synchronous
Serial or Parallel?	Serial
Max # of Masters	Unlimited
Max # of Slaves	1008

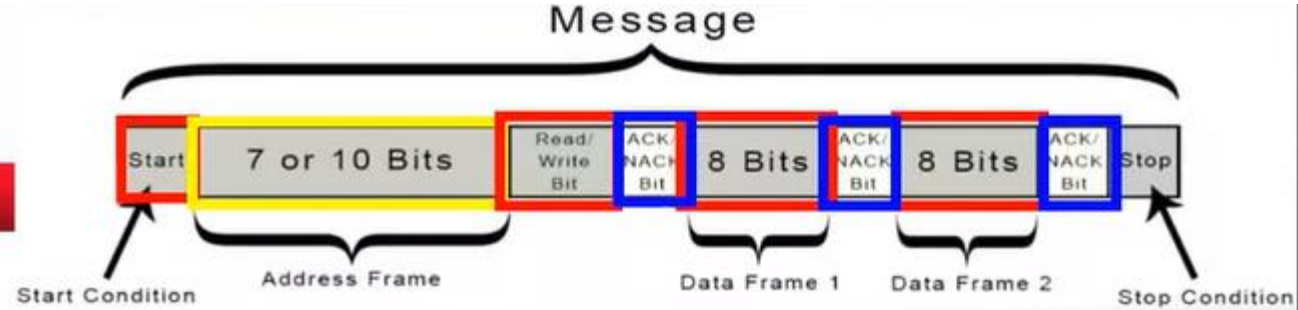
I2C


I2C Protocal example
connecting multiple I2C devices
ELECTRONIFY.ORG
Temp 1 01.50
Temp 2 24.00
I2C
www.electronify.org

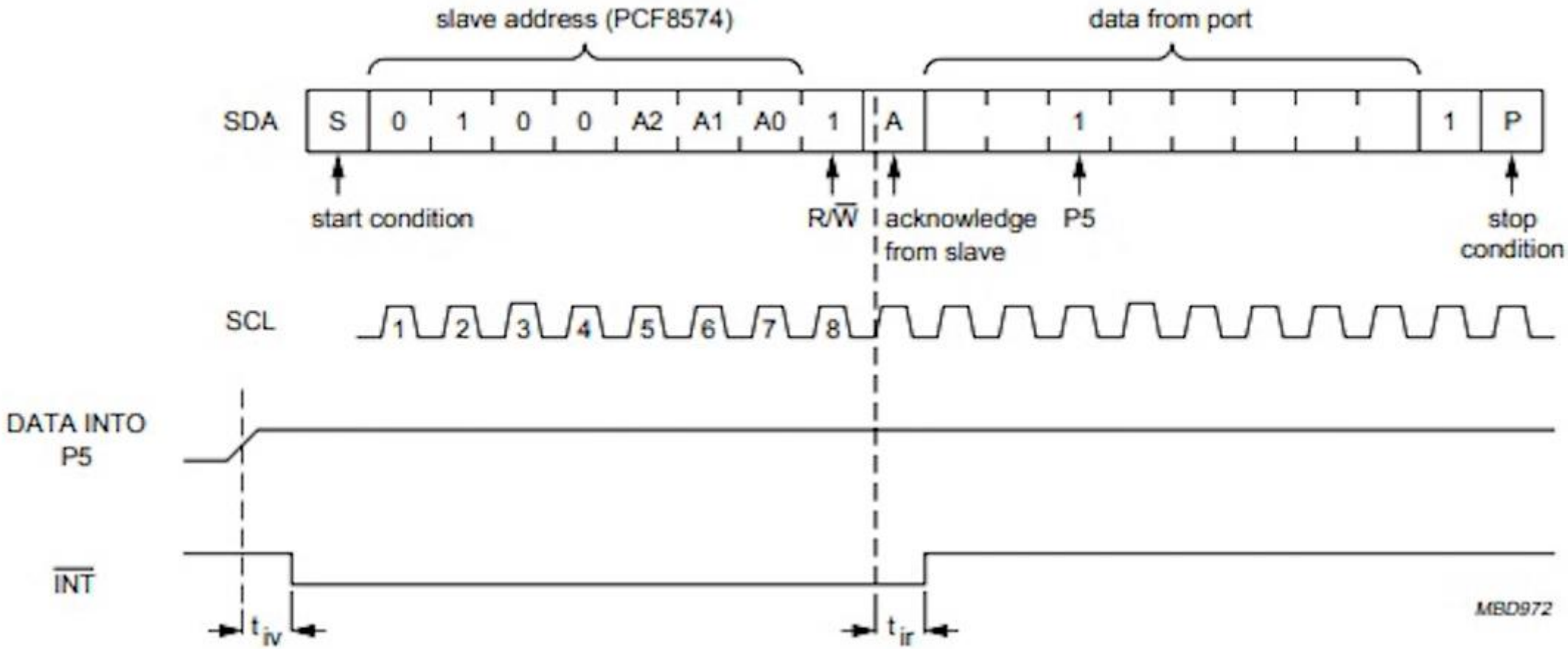

ARDUINO


Vcc
SDA
SCL
GND
Master
Slave
Slave
Slave
LCD

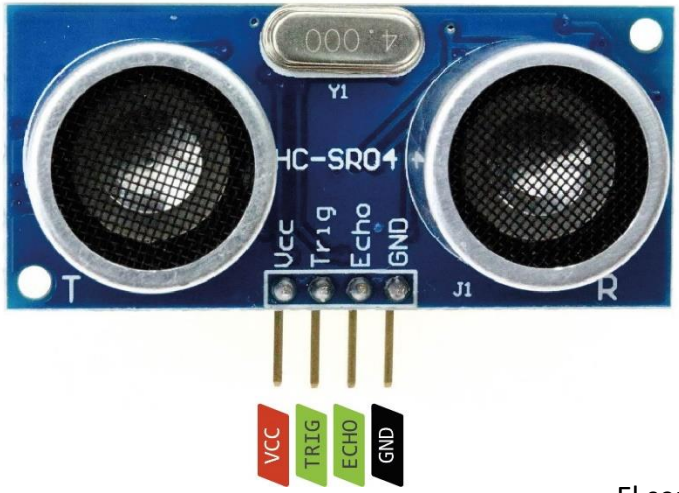




- **Start Condition:** The SDA line switches from a high voltage level to a low voltage level *before* the SCL line switches from high to low.
- **Stop Condition:** The SDA line switches from a low voltage level to a high voltage level *after* the SCL line switches from low to high.
- **Address Frame:** A 7 or 10 bit sequence unique to each slave that identifies the slave when the master wants to talk to it.
- **Read/Write Bit:** A single bit specifying whether the master is sending data to the slave (low voltage level) or requesting data from it (high voltage level).
- **ACK/NACK Bit:** Each frame in a message is followed by an acknowledge/no-acknowledge bit. If an address frame or data frame was successfully received, an ACK bit is returned to the sender from the receiving device.


slave address (PCF8574)
data from port
SDA
S
0
1
0
0
A2
A1
A0
1
A
1
1
P
start condition
R/W
acknowledge from slave
P5
stop condition
SCL
1
2
3
4
5
6
7
8
DATA INTO P5
INT
t iv
t ir
MBD972

MODULO HC-SR04 sensor ultrasónico de distancia

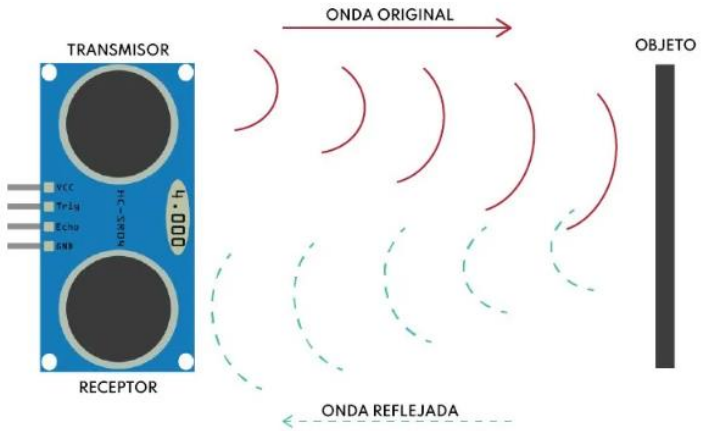


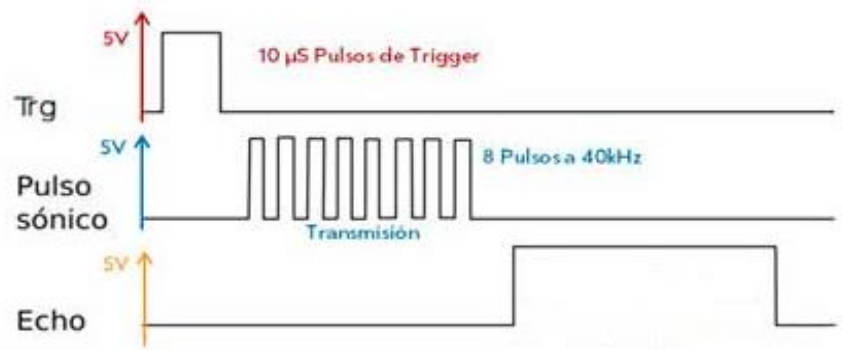
Características y especificaciones

- Modelo: HC-SR04
- Tipo de sensor: proximidad, distancia, presencia
- Principio de funcionamiento: ultrasonido
- Voltaje de alimentación: 5 VDC
- Corriente en reposo < 2mA
- Corriente típica en operación: 15 mA
- Rango de medición: 2 cm a 4 metros
- Angulo de medición: 15º
- Precisión: ± 3 mm
- Frecuencia del pulso ultrasónico: 40 KHz
- Conector: Header macho estándar 0.1 pulgadas
- Interfaz: 4 conexiones
    - Vcc
    - Trigger
    - Echo
    - Gnd
- Tiempo en alto para señal de disparo: 10uS
- Rango de tiempo de señal de eco: 100 uS a 25000 uS
- Tiempo entre medidas: 20 mS

El sensor HC-SR04 posee dos transductores: un emisor y un receptor piezoeléctricos, el emisor piezoeléctrico emite 8 pulsos de ultrasonido(40KHz) luego de recibir la orden en el pin TRIG, las ondas de sonido viajan y rebotan al encontrar un objeto, el sonido de rebote es detectado por el receptor piezoeléctrico, luego el pin ECHO cambia a Alto (5V) por un tiempo igual al que demoró la onda desde que fue emitida hasta que fue detectada, el tiempo del pulso ECO es medido por el microcontrolador y se calcula la distancia al objeto.

MODULO HC-SR04 sensor ultrasónico de distancia





El sensor HC-SR04 posee dos transductores: un emisor y un receptor piezoeléctricos, el emisor piezoeléctrico emite 8 pulsos de ultrasonido(40KHz) luego de recibir la orden en el pin TRIG, las ondas de sonido viajan y rebotan al encontrar un objeto, el sonido de rebote es detectado por el receptor piezoeléctrico, luego el pin ECHO cambia a Alto (5V) por un tiempo igual al que demoró la onda desde que fue emitida hasta que fue detectada, el tiempo del pulso ECO es medido por el microcontrolador y se calcula la distancia al objeto.

HC-SR04 genera un pulso en el pin “echo” cuya duración es proporcional a la distancia medida por el sensor. Para obtener la distancia en centímetros, solamente debemos dividir el tiempo en micro segundos entre 58 para obtener la distancia en centímetros (148 para pulgadas).

La distancia se puede calcular utilizando la siguiente formula:

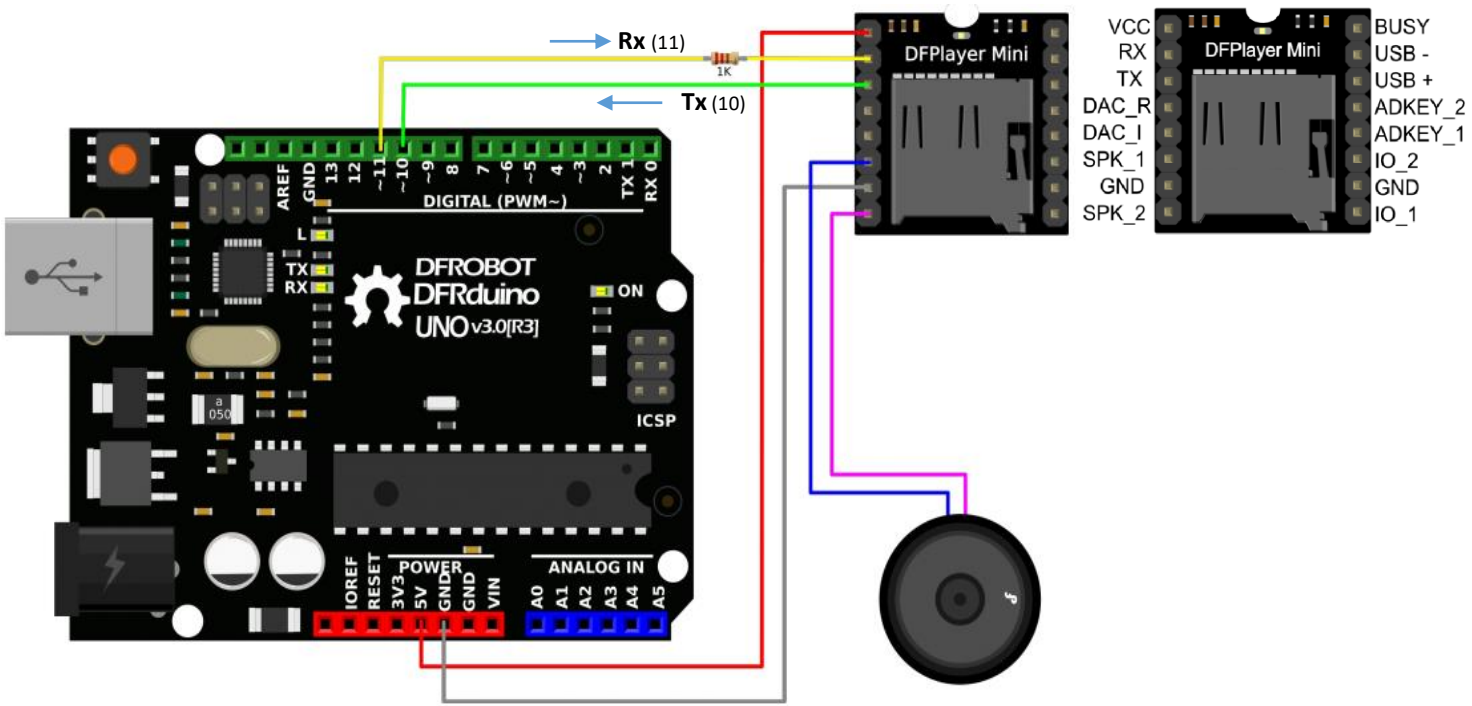
Centímetros: uS / 58 = centimeters
Pulgadas: uS / 148 =inch
rango = high level time * velocity (340M/S) / 2

Si el módulo se energiza de manera independiente al Arduino, se recomienda conectar primero GND y luego Vcc; caso contrario, se pueden producir inestabilidades.

MODULO REPRODUCTOR MP3 DFPlayer

Especificaciones:

- •Tasas de muestreo (kHz): 8/11.025/12/16/22.05/24/32/44.1/48.
- •Salida DAC de 24 bits, soporte para rango dinámico 90dB, soporte SNR 85dB.
- •Compatible con el sistema de archivos FAT16, FAT32, soporte máximo 32G de la tarjeta TF.
- •Función de espera de sonido publicitario, la música se puede suspender para una publicidad.
- •Datos de audio ordenados por carpeta, admite hasta 100 carpetas, cada carpeta puede contener hasta 255 canciones. (ojo, es en orden de escritura).
- •Volumen ajustable de 30 niveles, ecualizador de 6 niveles ajustable.
- •Parlante 4Ω, 3W max.

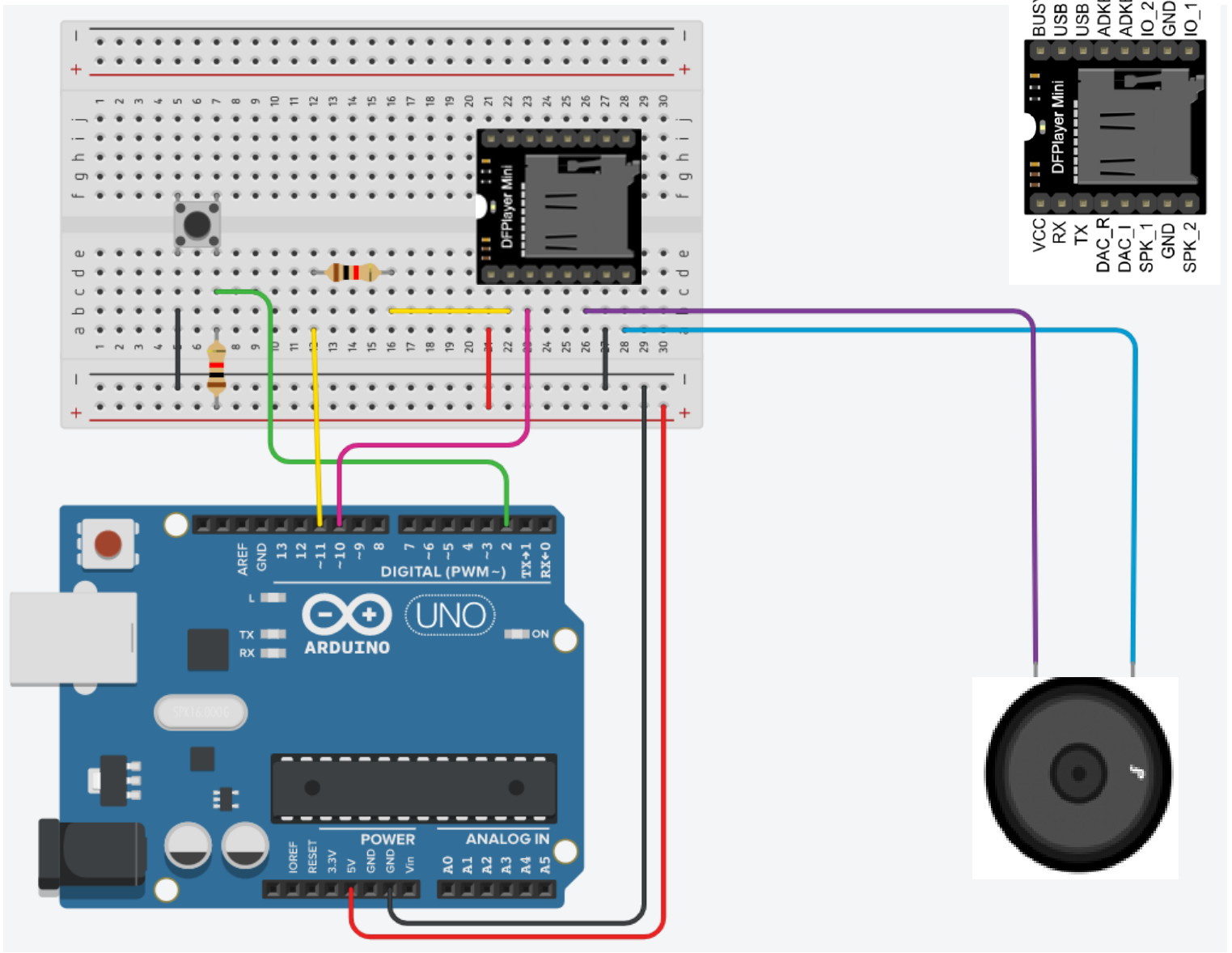


Pin	Description	Note
VCC	Input Voltage	DC3.2~5.0V;Type: DC4.2V
RX	UART serial input	
TX	UART serial output	
DAC_R	Audio output right channel	Drive earphone and amplifier
DAC_L	Audio output left channel	Drive earphone and amplifier
SPK2	Speaker-	Drive speaker less than 3W
GND	Ground	Power GND
SPK1	Speaker+	Drive speaker less than 3W
IO1	Trigger port 1	Short press to play previous（long press to decrease volume）
GND	Ground	Power GND
IO2	Trigger port 2	Short press to play next（long press to increase volume）
ADKEY1	AD Port 1	Trigger play first segment
ADKEY2	AD Port 2	Trigger play fifth segment
USB+	USB+ DP	USB Port
USB-	USB- DM	USB Port
BUSY	Playing Status	Low means playing \High means no

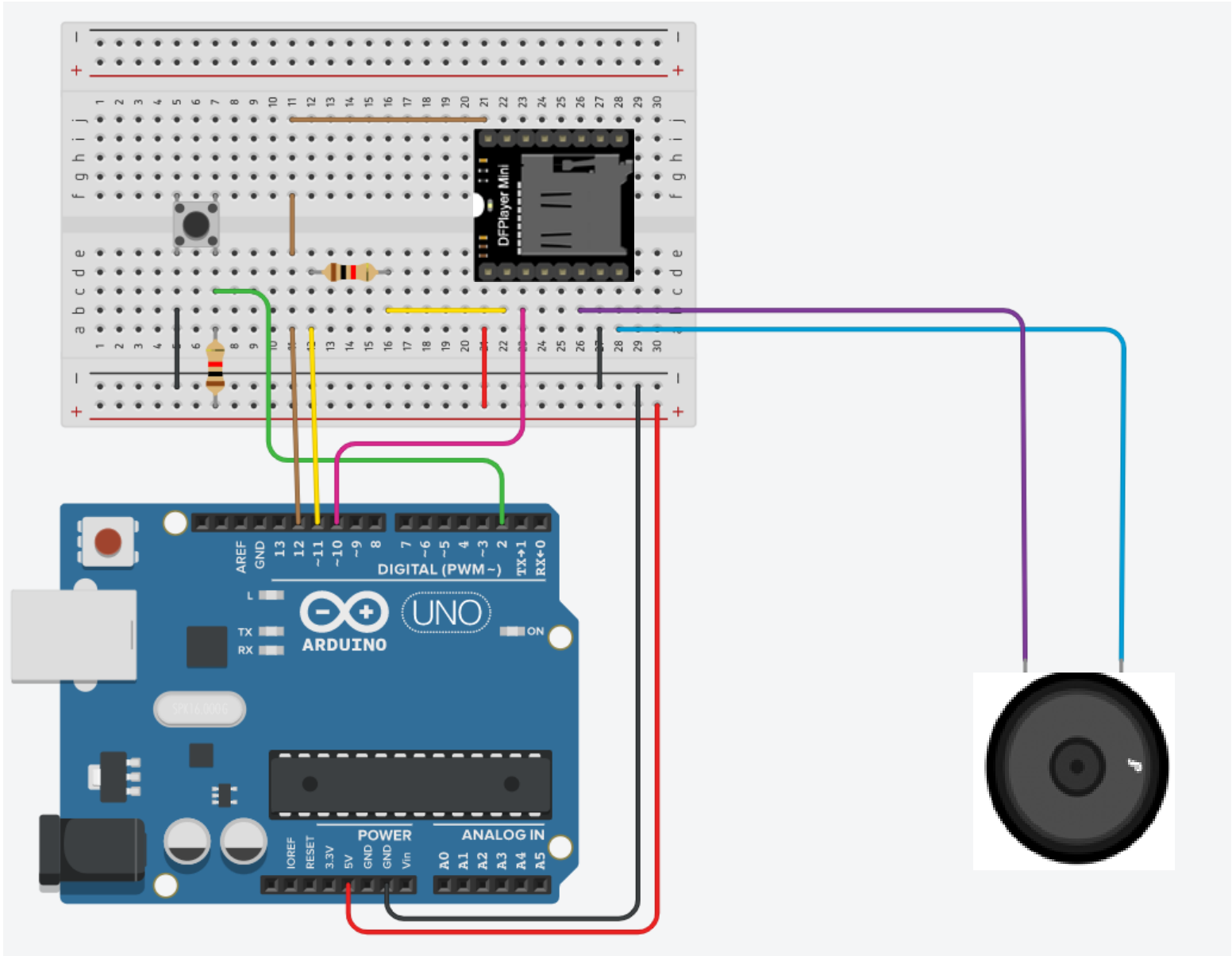
https://www.dfrobot.com/product-1121.html

https://wiki.dfrobot.com/DFPlayer_Mini_SKU_DFR0299

MODULO DFPlayer Básico



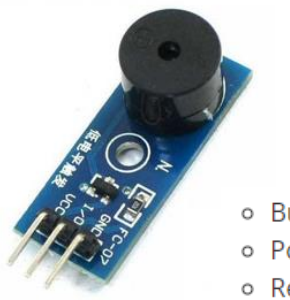
MODULO DFPlayer con Busy



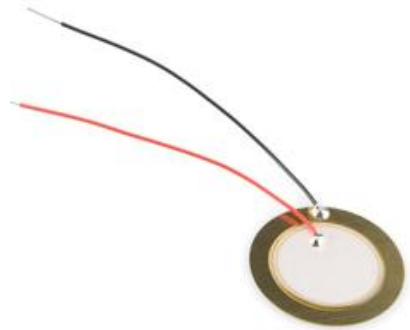
BUZZER (ZUMBADOR) Activos y Pasivos.

- Tipo: Activo
- Diametro: Φ12mm
- Alto: 9.5mm
- **Voltaje disponible:**

Voltaje	3VDC	5VDC	12VDC
Rango de voltaje	2.7-3.5VDC	4-7VDC	10-13VDC
Corriente (mA)	<=32mA	<=32mA	<=25mA
Salida sonido	>=85dB	>= 85dB	>= 85dB
Frec. Resonancia	2300Hz +-300	2300Hz +-300	2300Hz +-300
Rango temperatura	-20°C a 45°C	-20°C a 45°C	-20°C a 45°C



- Buzzer pasivo 5VDC
- Posee 3 pines, VCC +, GND -, I/O señal
- Requiere de una señal de frecuencia entre 2KHz y 5KHz
- Este modulo no enciende colocando positivo y negativo
- Voltaje de trabajo: 5VDC
- Tamaño compacto: 33 x 13 x 13 mm

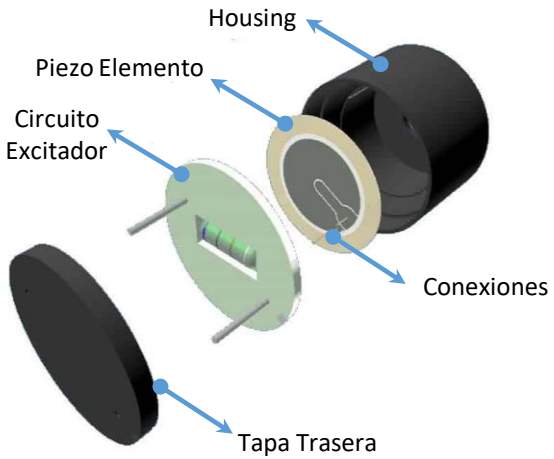


- Tipo: Disco piezoelectrico
- Voltaje Vpp: 3.28V
- Voltaje de operación: 1 a 30VDC
- Frecuencia: 2000Hz +/- 500
- Resistencia de resonancia: <300 ohm
- Capacitancia 120Hz: 12,000pF
- Salida de sonido hasta 10cm: >60dB aproximado

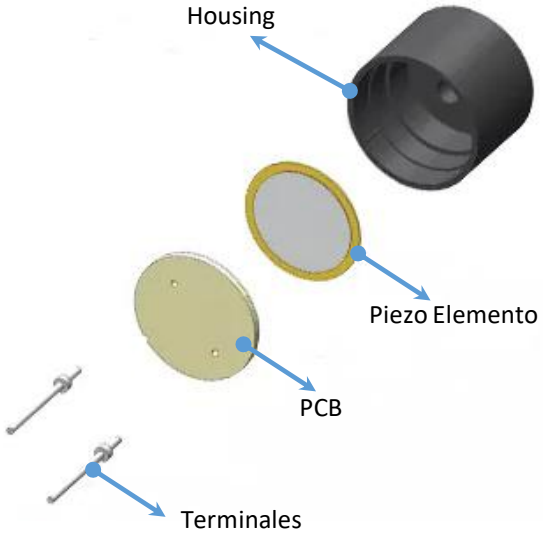
- Tipo: Buzzer piezoelectrico
- Nivel de sonido: 95dB
- Voltaje de operación: 3 a 24VDC
- Corriente de trabajo: 20mA
- Frecuencia: 3900Hz +/- 500
- Diámetro: 22.5mm
- Altura: 11mm
- Distancia entre agujeros de montaje: 30mm

BUZZER (ZUMBADOR) Activos y Pasivos.

ACTIVO
(KY-012)

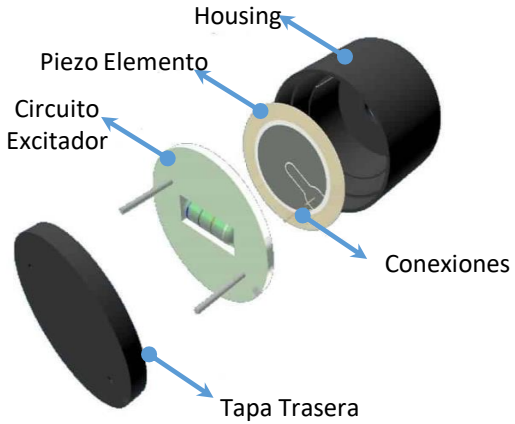


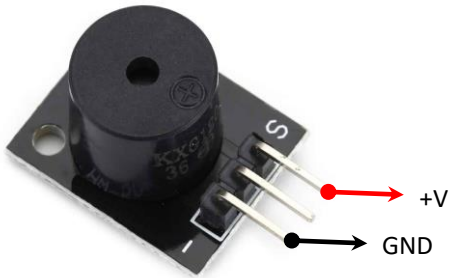
PASIVO
(KY-006)



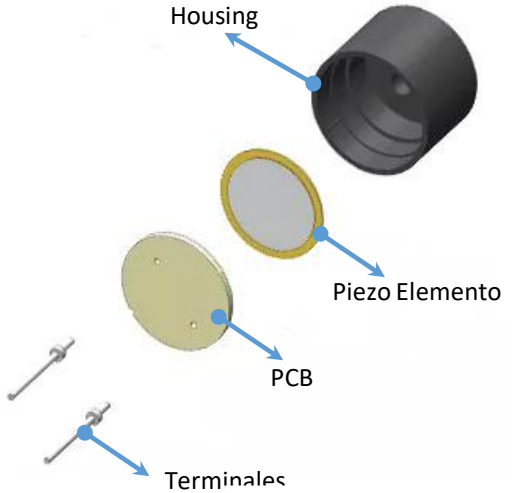
BUZZER (ZUMBADOR) Activos y Pasivos.

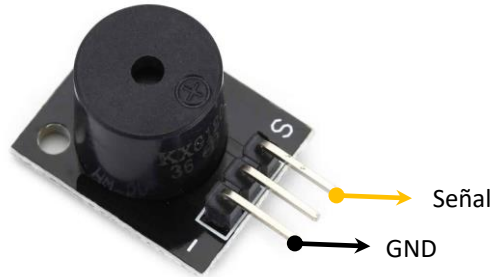
KY-012
Activo



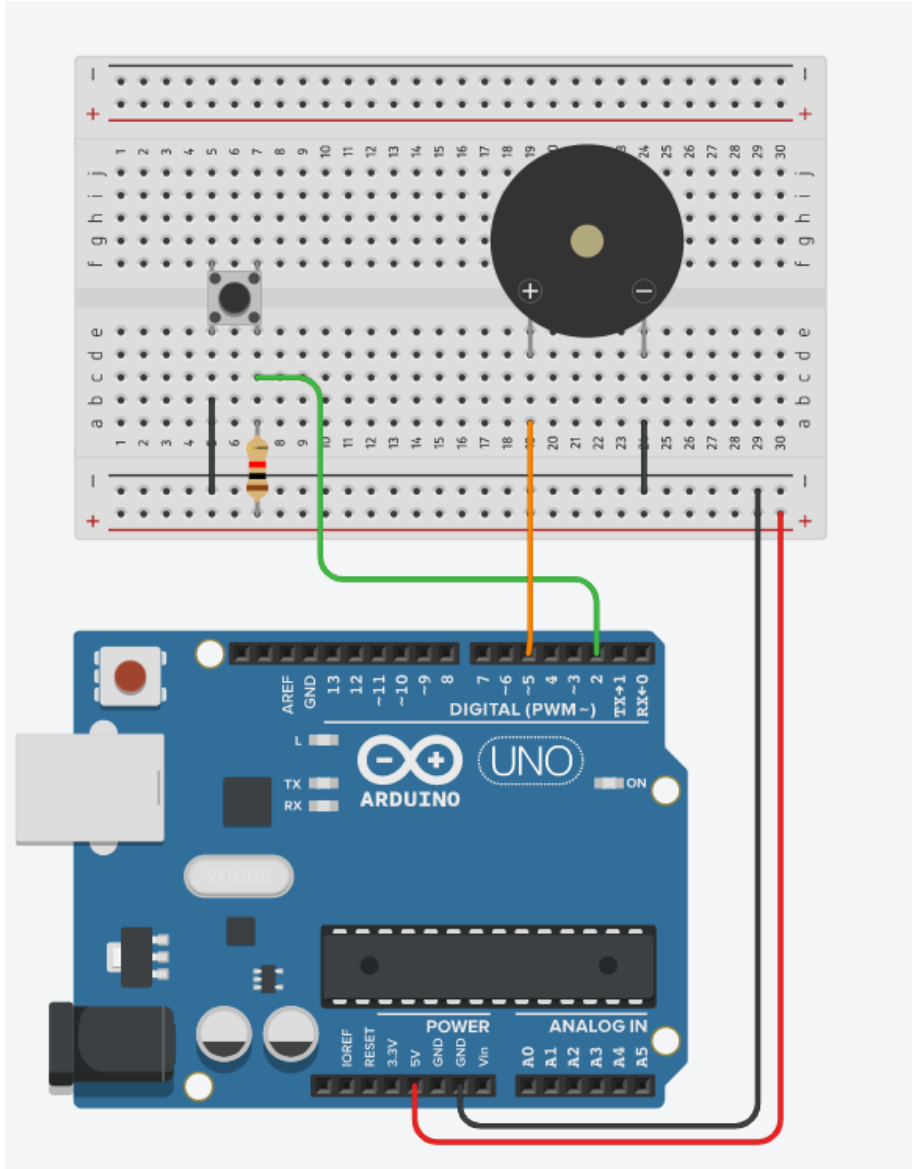


KY-006
Pasivo



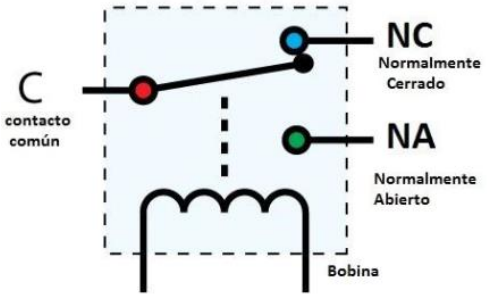


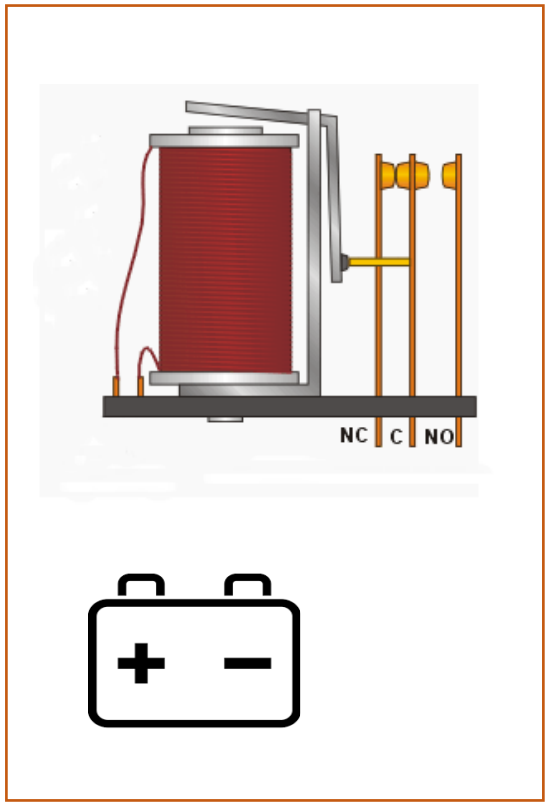
Melodías: https://github.com/robsoncouto/arduino-songs

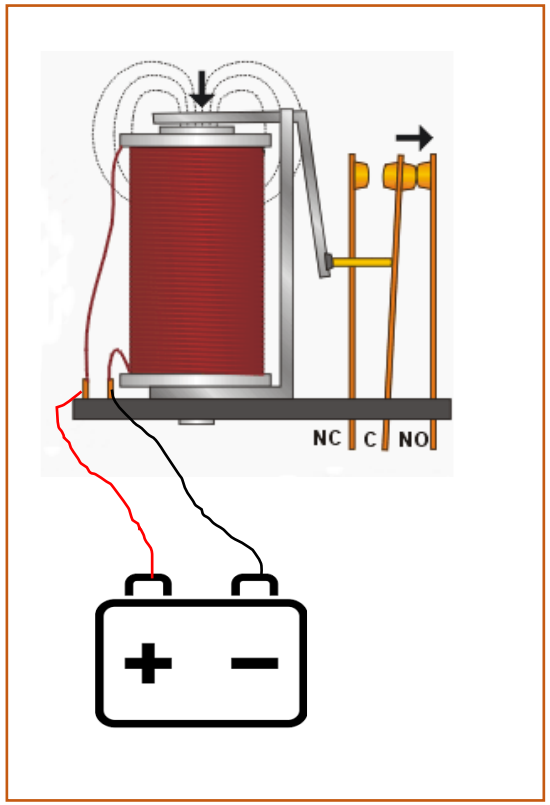

DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN

RELÉS (RELAY´S)

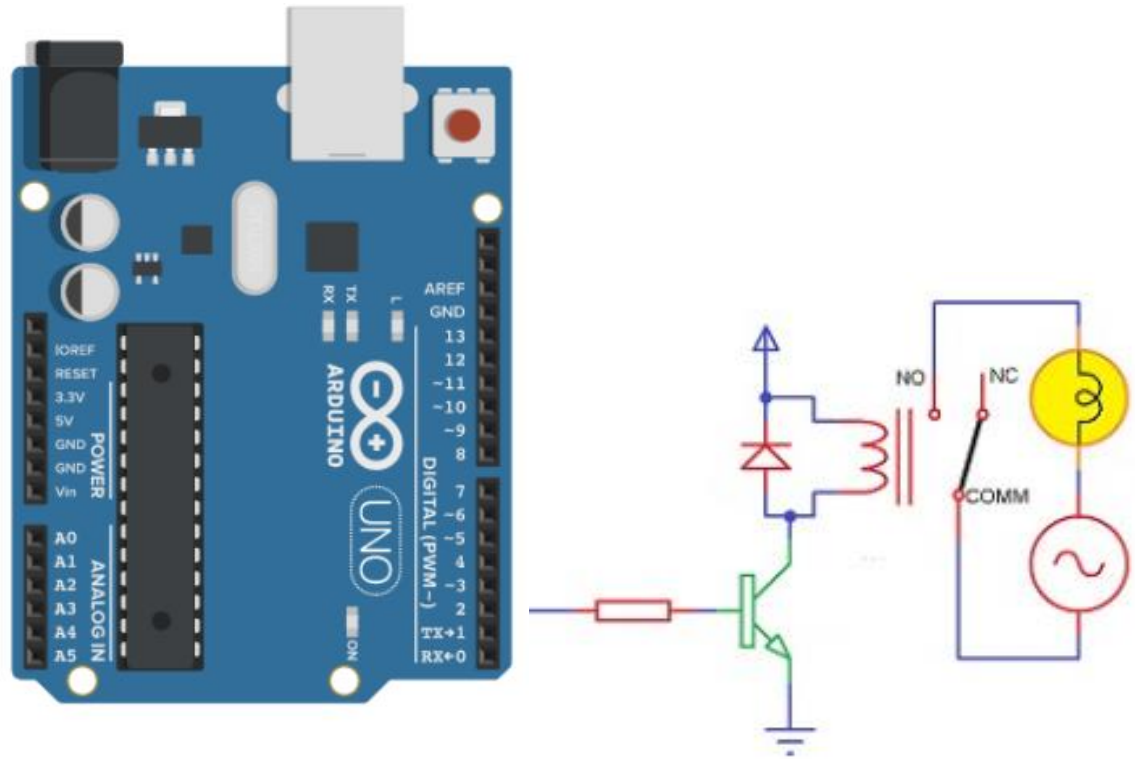
RELÉS (RELAY´S)







UTILIZACIÓN DE RELÉS CON ARDUINO



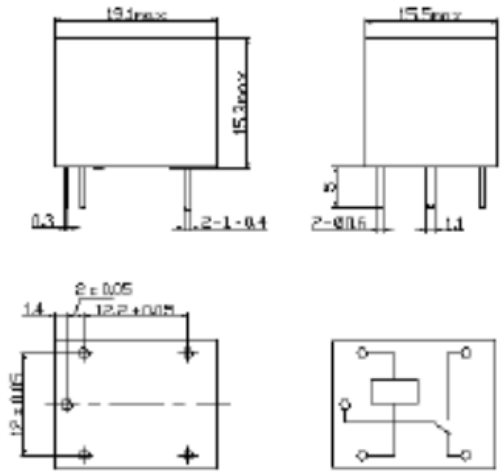
1. MAIN FEATURES

- Switching capacity available by 10A in spite of small size design for highdensity P.C. board mounting technique.
- UL,CUL,TUV recognized.
- Selection of plastic material for high temperature and better chemical solution performance.
- Sealed types available.
- Simple relay magnetic circuit to meet low cost of mass production.

4. RATING

CCC	FILE NUMBER:CH0052885-2000	7A/240VDC
CCC	FILE NUMBER:CH0036746-99	10A/250VDC
UL /CUL	FILE NUMBER: E167996	10A/125VAC 28VDC
TUV	FILE NUMBER: R9933789	10A/240VAC 28VDC

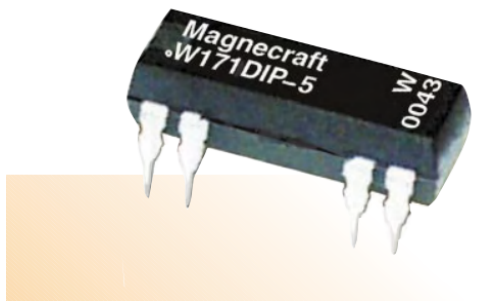
5. DIMENSION(unit:mm) DRILLING(unit:mm) WIRING DIAGRAM

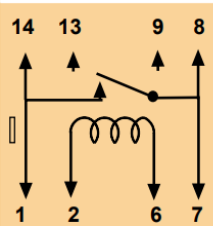


CLASS 171DIP

DUAL IN - LINE PACKAGE REED RELAY

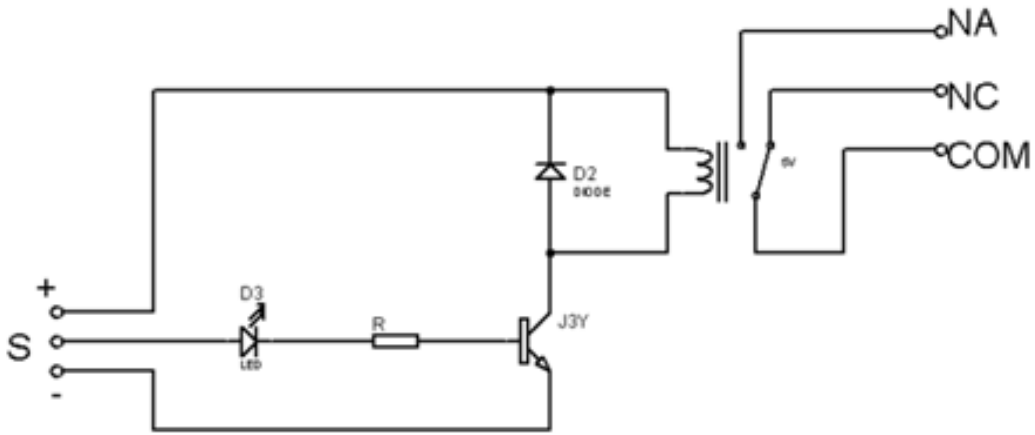
SPST-N.O. OR N.C., DPST-N.O. 0.5 AMP

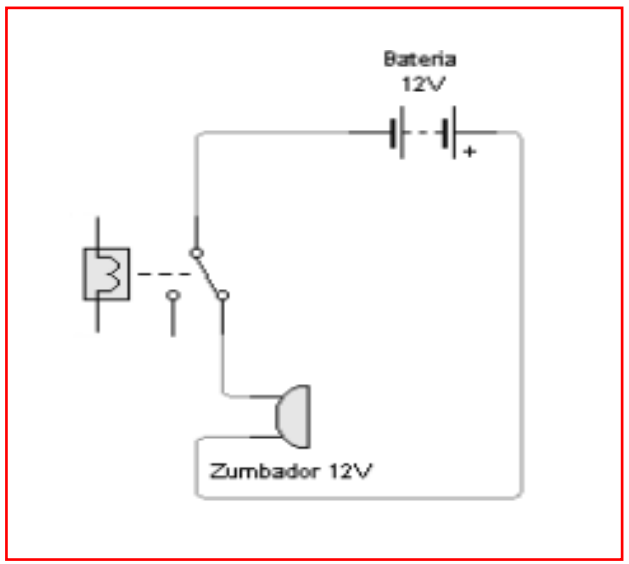


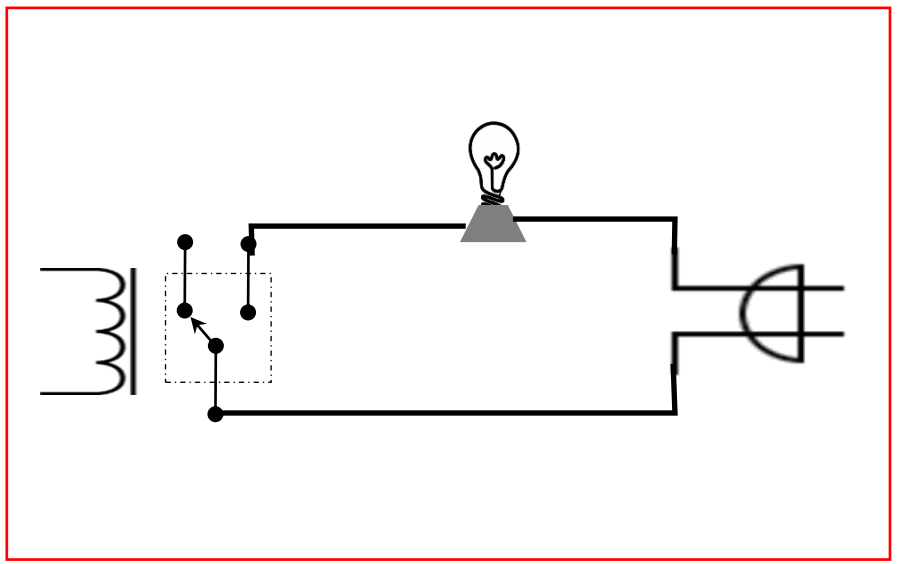
WIRING DIAGRAMS (TOP VIEWED)	STANDARD PART NUMBERS	COIL MEASURED @ 25˚C		
		NOMINAL INPUT VOLTAGE	NOMINAL RESISTANCE (OHMS)	NOMINAL POWER (mW)
SPST - N. O.				
	W171DIP-2	5	500 Ω	50
	W171DIP-4	12	1200 Ω	120
	W171DIP-5	24	2200 Ω	270

CONTACTS	
Contact Material:	Rhodium
Contact Resistance:	200 milliohms max.
Contact Rating:	0.5 amp 100 VDC (10VA)
	1.5 amps max continuous carry current
TIMING	
Operate time:	1 mS or less @ nominal voltage
Release time:	1 mS or less @ nominal Voltage
DIELECTRIC STRENGTH	
Across Open Contacts:	150 V rms
Between Mutually Insulation Points:	500 V rms
Insulation Resistance:	1000 megohms min. @ 100 VDC
Capacitance:	1.0 pf typical contact to contact
TEMPERATURE	
Operating:	-40˚C to +85˚C @ rated operation
Storage:	-40˚C to +105˚C

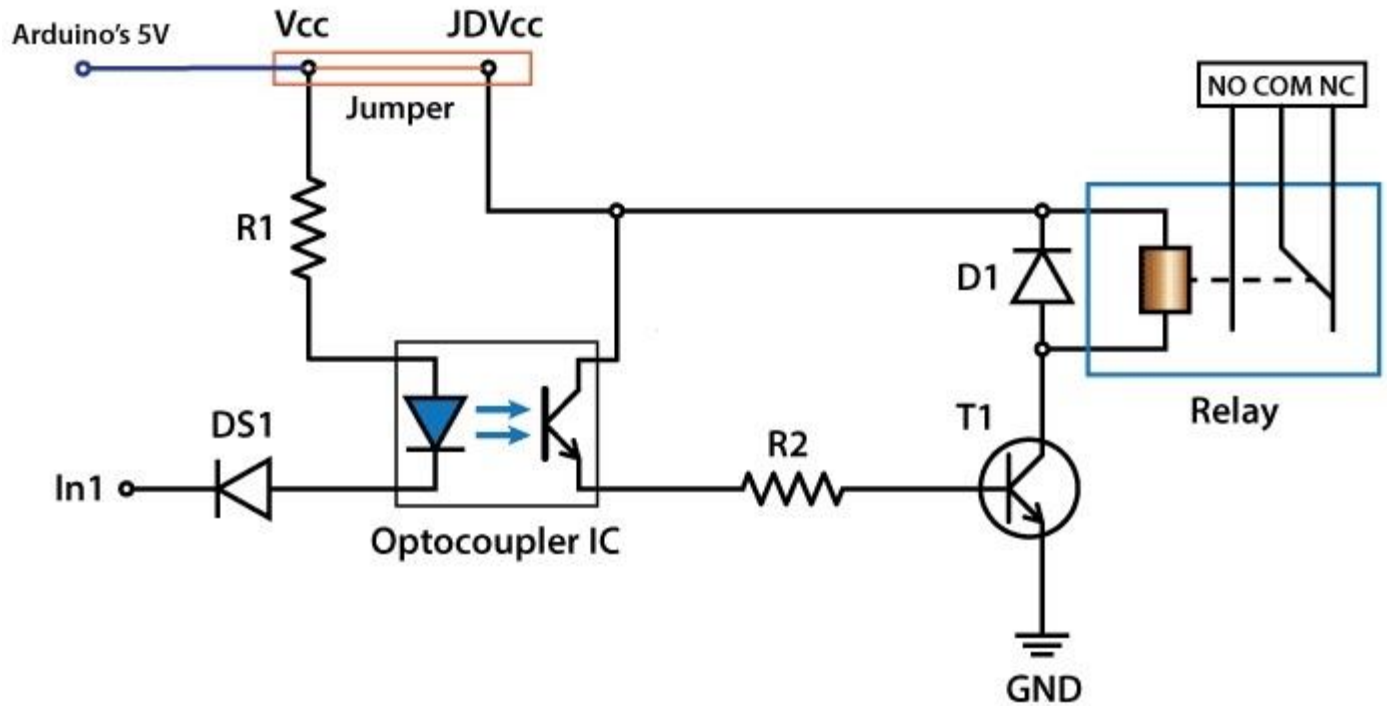
MODULO RELÉ NC/NO



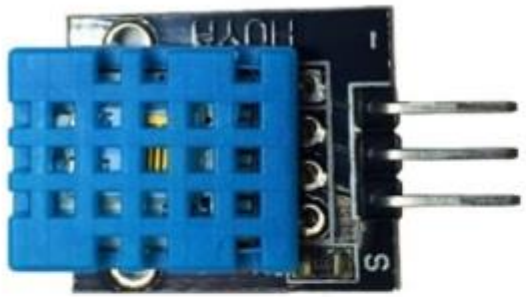




MODULO RELÉ OPTOACOPLADO



SENSOR TEMPERATURA y HUMEDAD DHT11/22



GROUND
SIGNAL
VCC+

DHT11

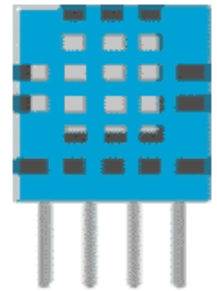
GROUND
NOT USED
SIGNAL
VCC+

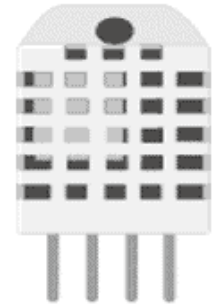
GROUND
SIGNAL
VCC+

DHT22

GROUND
NOT USED
SIGNAL
VCC+

SENSOR TEMPERATURA y HUMEDAD DHT11/22





	DHT11	DHT22
Operating Voltage	3 to 5V	3 to 5V
Max Operating Current	2.5mA max	2.5mA max
Temperature Range	0-50°C / ± 2°C	-40 to 80°C / ± 0.5°C
Humidity Range	20-80% / 5%	0-100% / 2-5%
Sampling Rate	1 Hz (reading every second)	0.5 Hz (reading every 2 seconds)
Advantage	low cost	More Accurate

Librería a utilizar: dht.h

SENSOR TEMPERATURA y HUMEDAD DHT11/22

HUMEDAD RELATIVA

DHT11 y 22 miden *Humedad Relativa* . La humedad relativa es la relación entre la cantidad de vapor de agua en el aire y el punto de saturación del vapor de agua en el aire.

En el punto de saturación el vapor de agua comienza a condensarse y acumularse en las superficies.

El punto de saturación depende de la temperatura del aire. El aire frío puede contener menos vapor de agua antes de saturarse, y el aire caliente puede contener más vapor de agua antes de saturarse.

Una HR del 100 %, se produce condensación;
en una HR del 0 %, el aire está completamente seco.

Densidad de Vapor de Agua

% Humedad Relativa $$RH = \left(\frac{\rho_w}{\rho_s} \right) \cdot 100\%$$

Densidad de Vapor a Saturación

SENSOR TEMPERATURA y HUMEDAD DHT11/22

HUMEDAD Y TEMPERATURA EN DHT11

El DHT detecta el vapor de agua midiendo la resistencia eléctrica entre dos electrodos. El componente sensor de humedad es un sustrato que retiene la humedad con electrodos aplicados a la superficie, el sustrato libera iones que aumentan la conductividad entre los electrodos. El cambio de resistencia entre los dos electrodos es proporcional a la humedad relativa. Una humedad relativa más alta disminuye la resistencia entre los electrodos, mientras que una humedad relativa más baja aumenta la resistencia entre los electrodos.

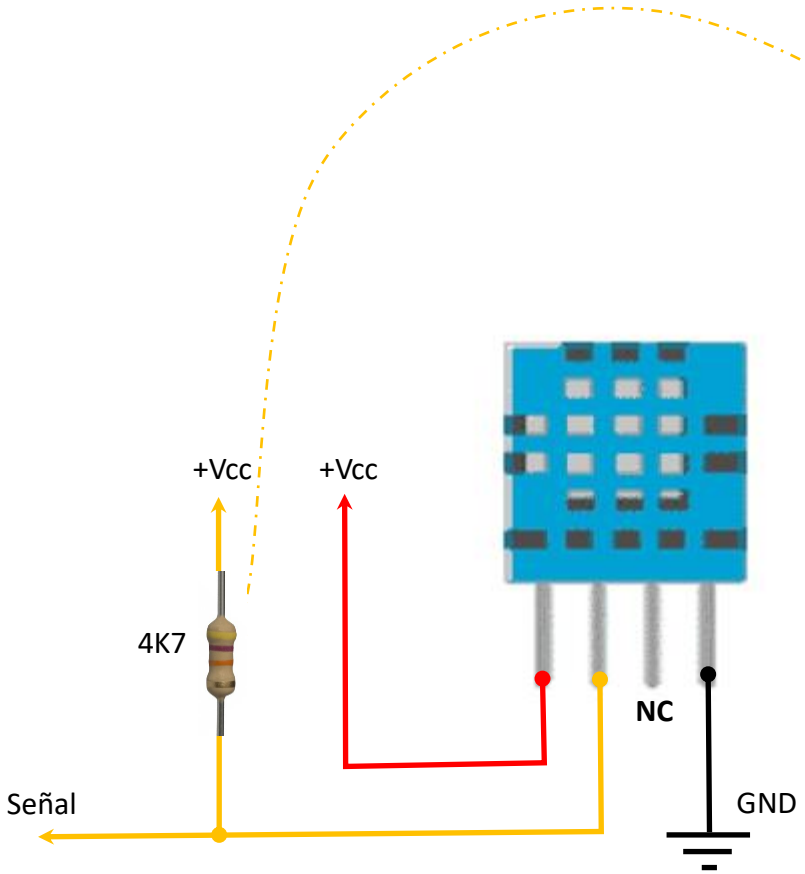
El DHT mide la temperatura con un sensor de temperatura NTC montado en superficie (termistor) integrado en la unidad.

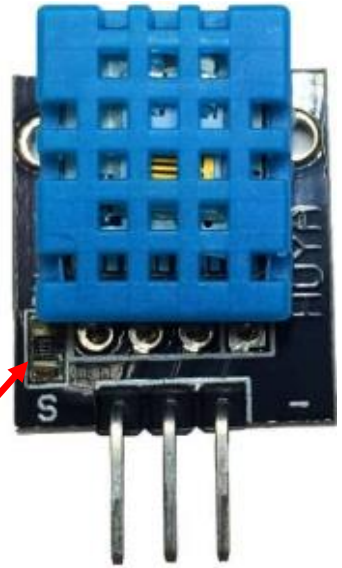
CONFIGURACION:

El DHT usa solo un cable de señal para transmitir datos al Arduino. La alimentación proviene de cables separados de 5 V y de tierra. Se necesita una resistencia pull-up de 10 K ohmios entre la línea de señal y la línea de 5 V para asegurarse de que el nivel de la señal se mantenga alto de forma predeterminada (ver datasheet del fabricante).

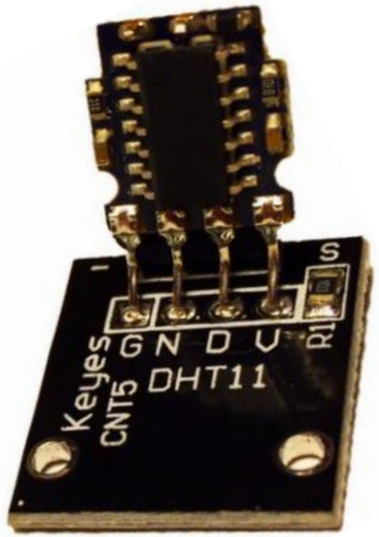
Hay dos versiones diferentes del DHT11 con las que te puedes encontrar. Un tipo tiene cuatro pines y el otro tipo tiene tres pines y está montado en una placa de circuito impreso pequeña. La versión montada en PCB incluye la resistencia pull-up de 10K Ohm SMD.

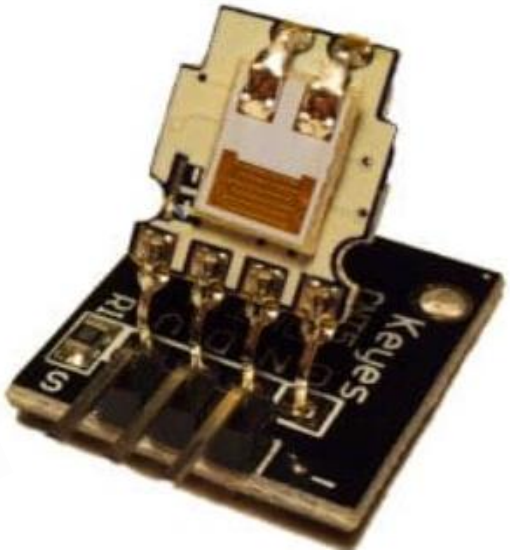
SENSOR TEMPERATURA y HUMEDAD DHT11/22





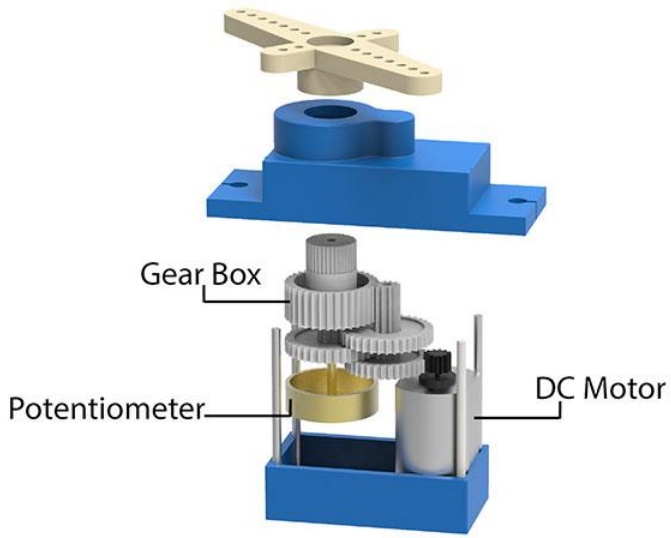
Resistencia
Pull-Up: 4K7

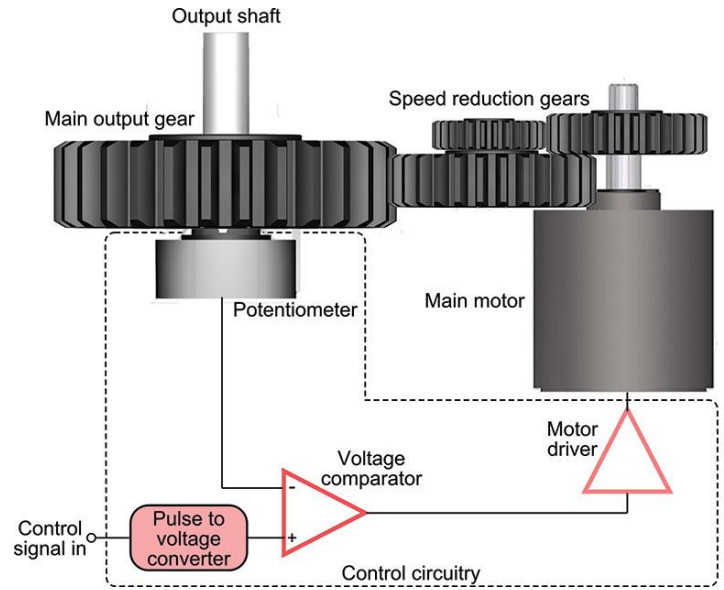




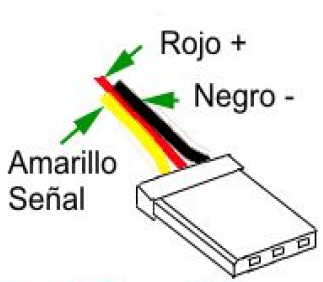
https://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino/

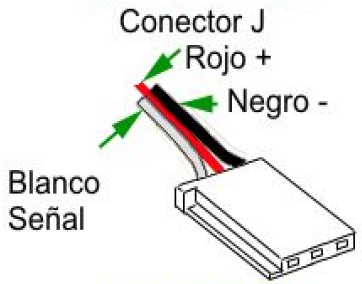
SERVOMOTORES

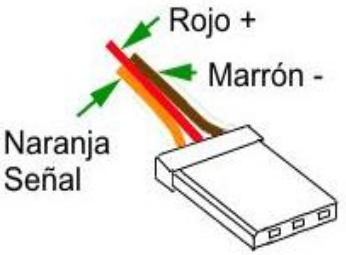


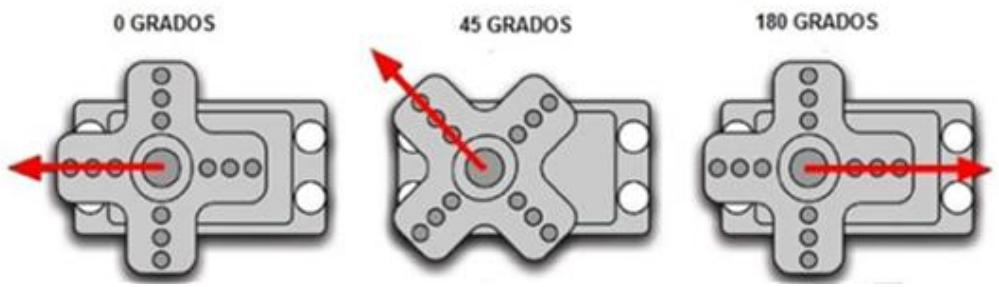


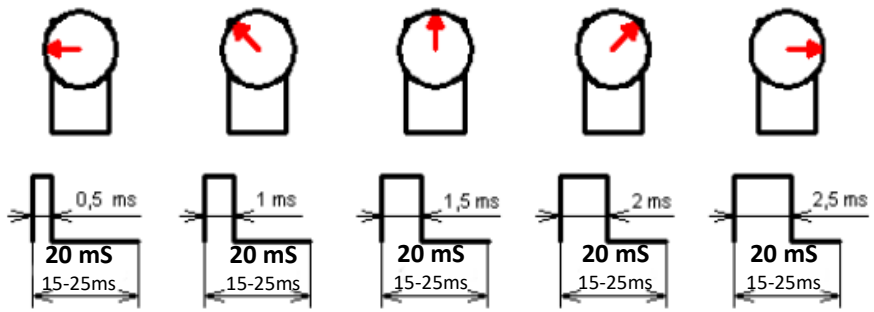
SERVOMOTORES

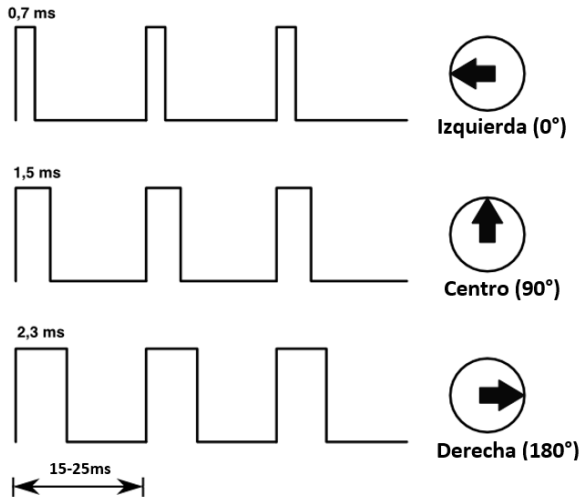




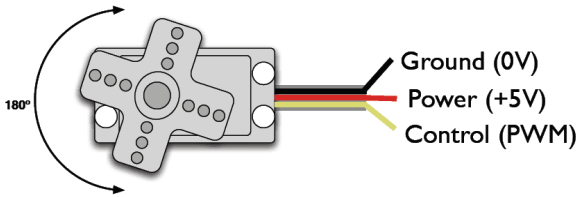








SERVOMOTORES



SG90 9 g Micro Servo

Basic Information
Modulation: Analog
Operating Voltage = 4.0 to 7.2 volts
Torque: 4.8V: 25.0 oz-in (1.80 kg-cm)
Speed: 4.8V: 0.10 sec/60°
Weight: 0.32 oz (9.0 g)
Dimensions: Length: 0.91 in (23.1 mm) Width: 0.48 in (12.2 mm) Height:1.14 in (29.0 mm)
Motor Type: 3-pole
Gear Type: Plastic
Rotation/Support: Bushing

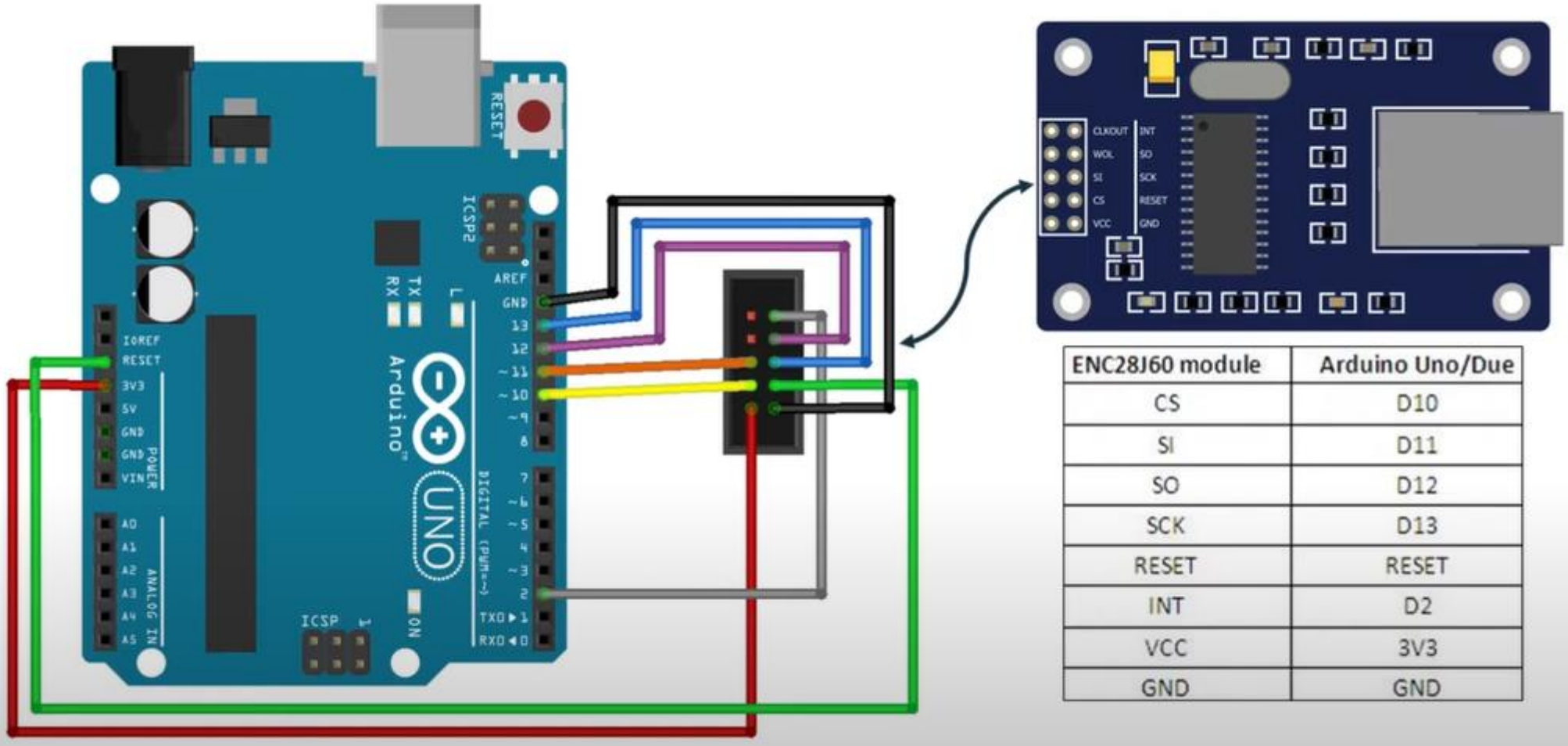
Additional Specifications
Rotational Range: 180°
Pulse Cycle: ca. 20 ms
Pulse Width: 500-2400 µs

SERVOMOTORES

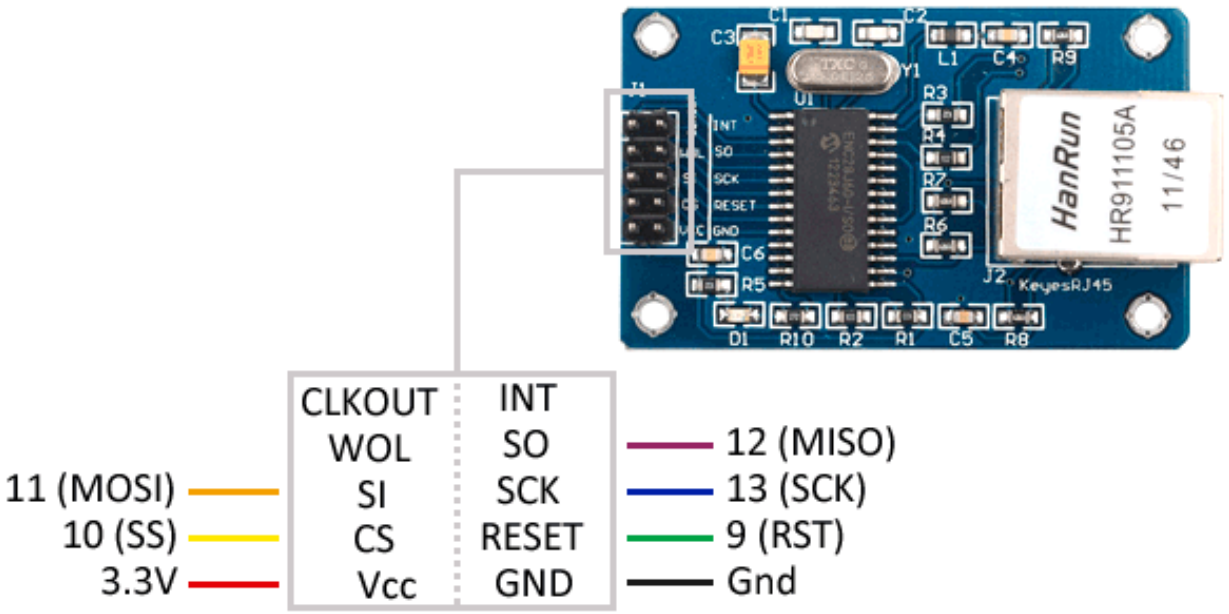
Características:

– Voltaje de alimentación: 4.2V – 6.0V DC
– Señal de salida (control): 1.5 ms ± 0.5 ms
– Corriente de salida: 15mA (5V)
– 3 canales
– Tamaño: 46*32*17mm
– Peso: 8g
– Modos de ajuste: Manual, automático, con una mediana

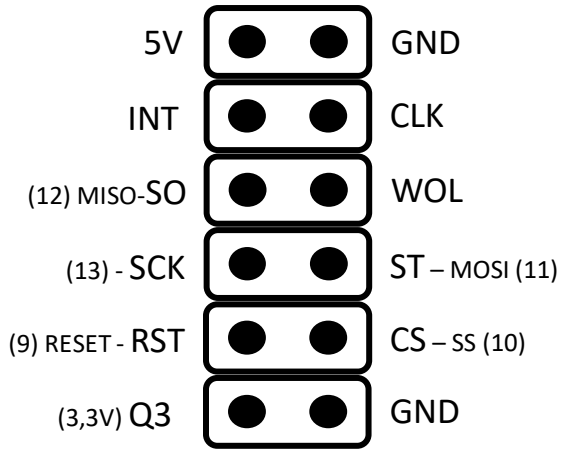
CONEXIONES ENTRE EL MÓDULO ETHERNET ENC28J60 Y ARDUINO UNO



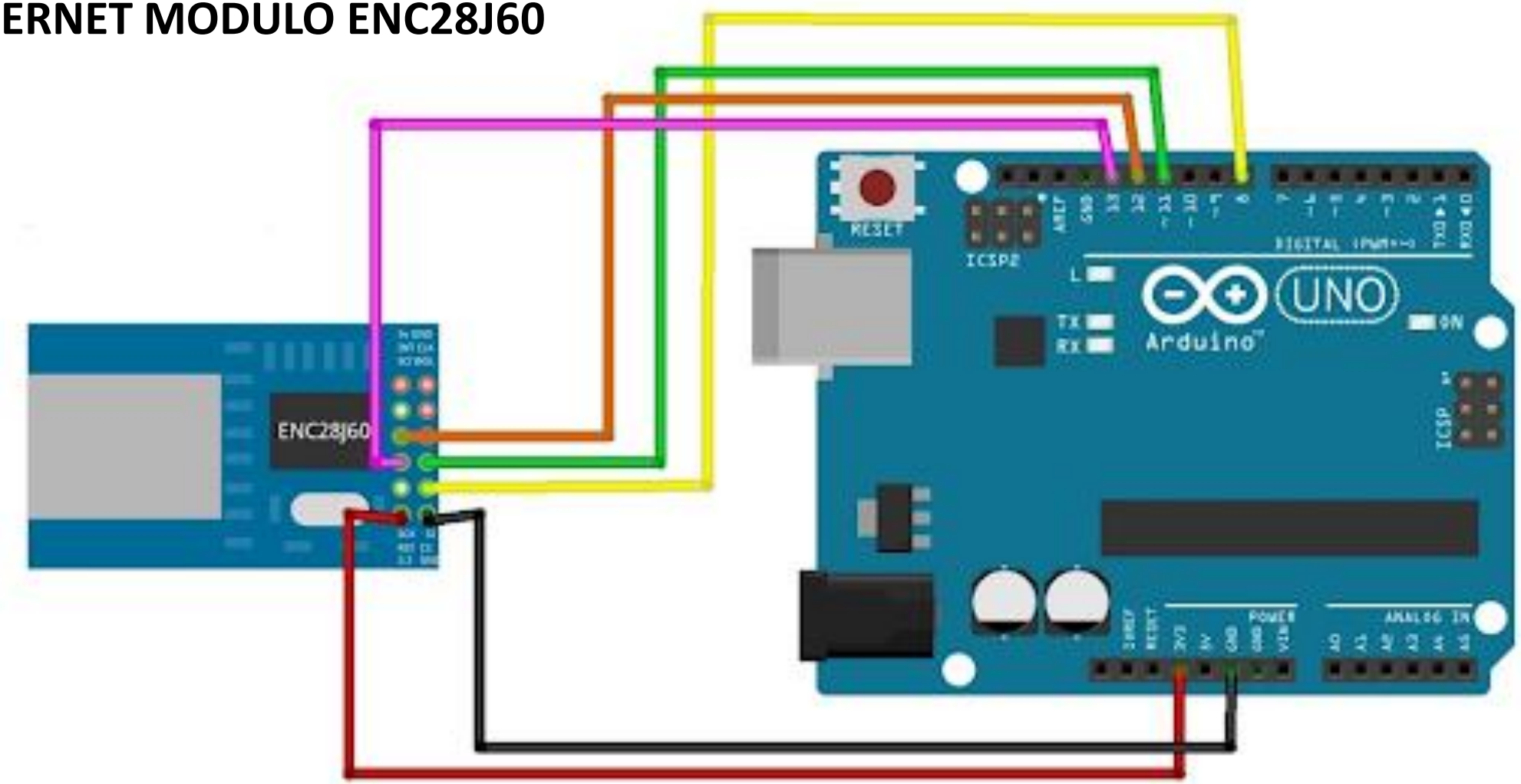
ENC28J60 module	Arduino Uno/Due
CS	D10
SI	D11
SO	D12
SCK	D13
RESET	RESET
INT	D2
VCC	3V3
GND	GND



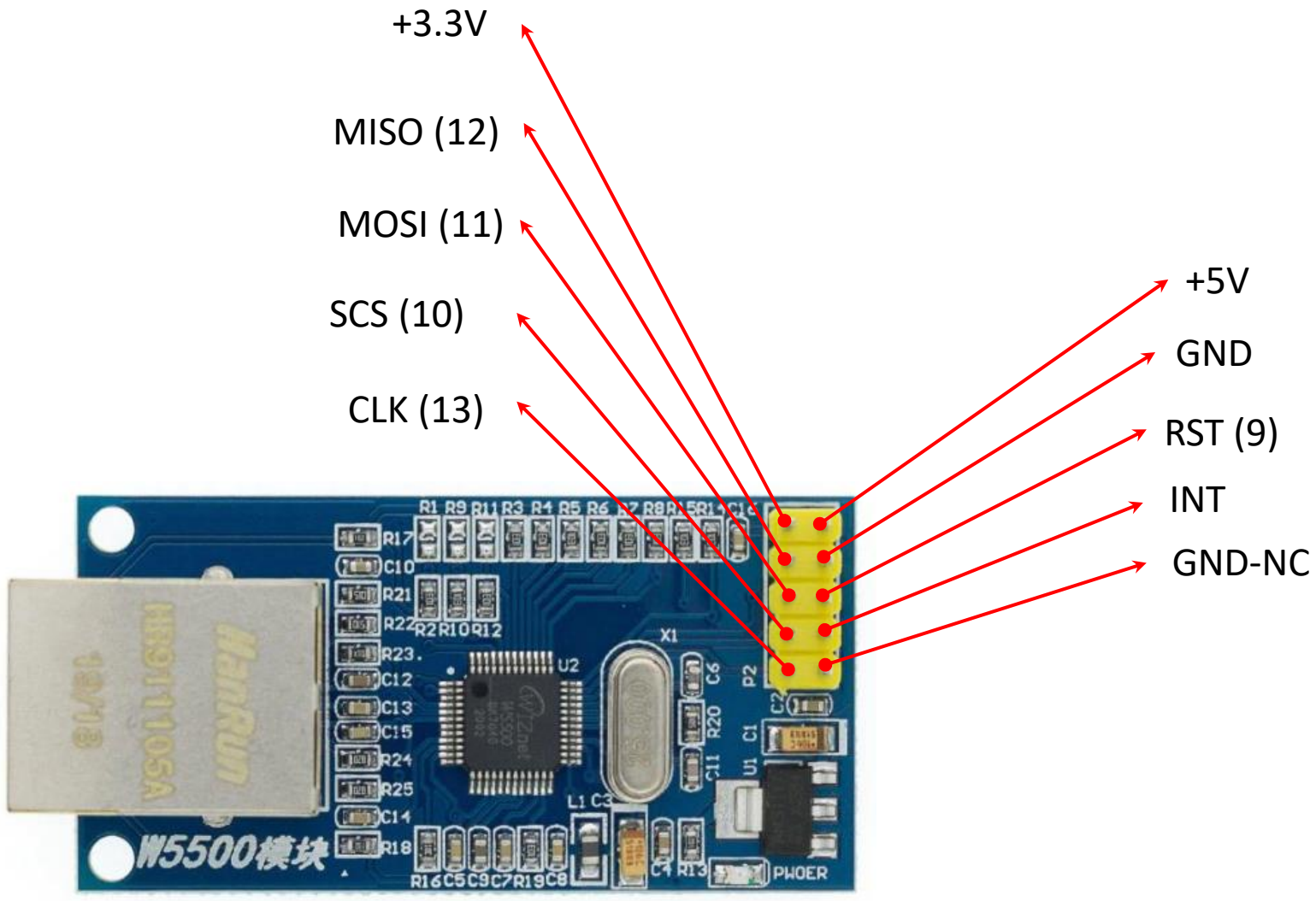
Pin Name	Description
V_{CC}	Module supply voltage
CLKOUT	Programmable clock output
ENC-WOL	Wake on LAN
RESET	Active low reset input
ENC-INT	Active low interrupt output pin
CS	SPI chip select
MOSI	SPI data input
MISO	SPI data output
SCK	SPI clock
GND	Module ground reference



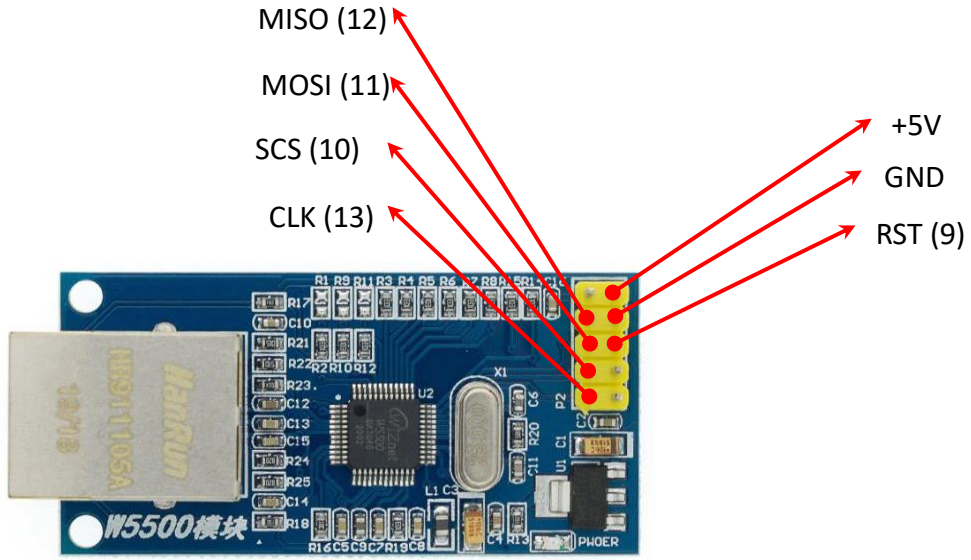
ETHERNET MODULO ENC28J60



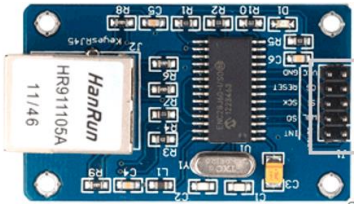
ETHERNET MODULO W5500

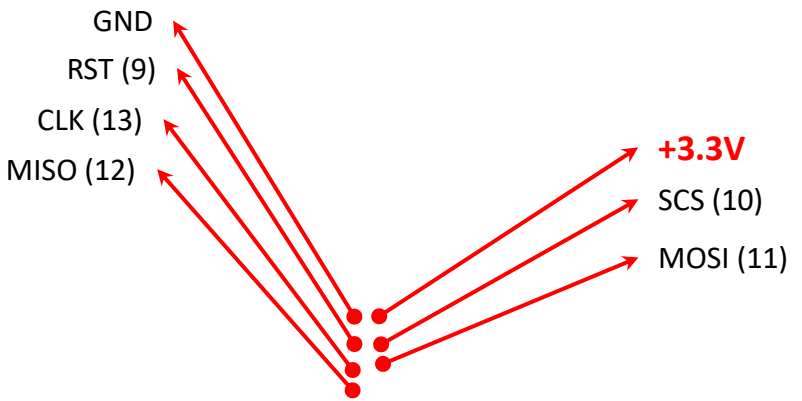


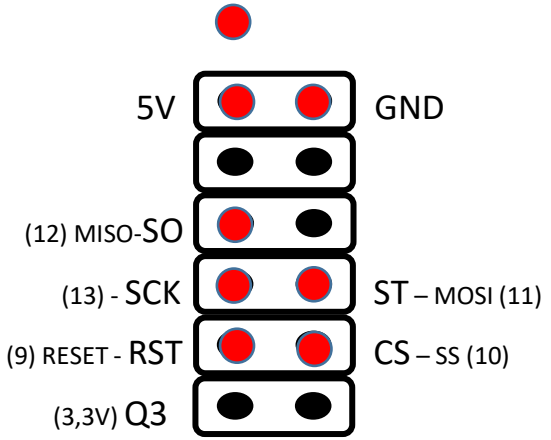
MODULOS ETHERNET



W5500

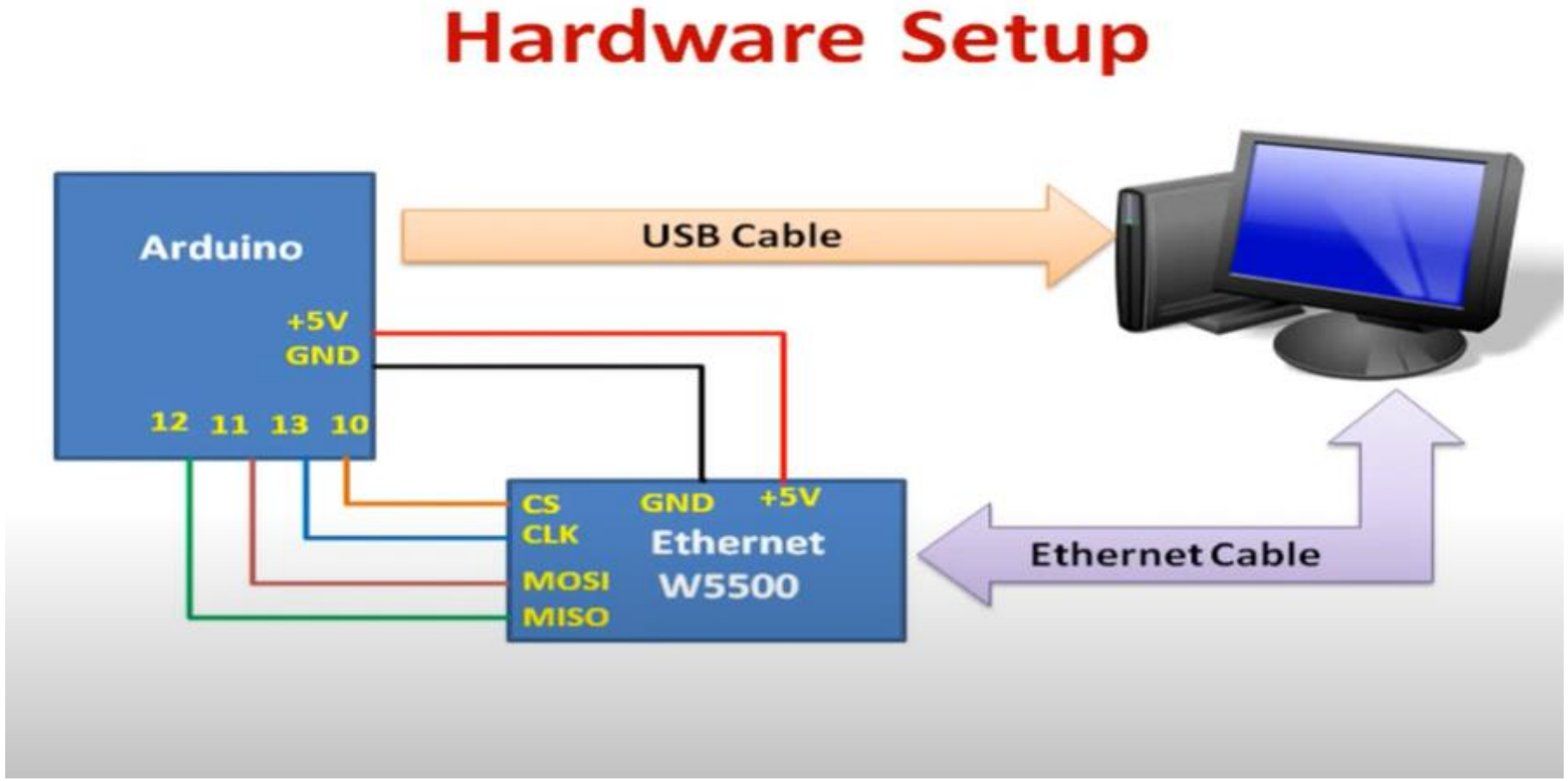




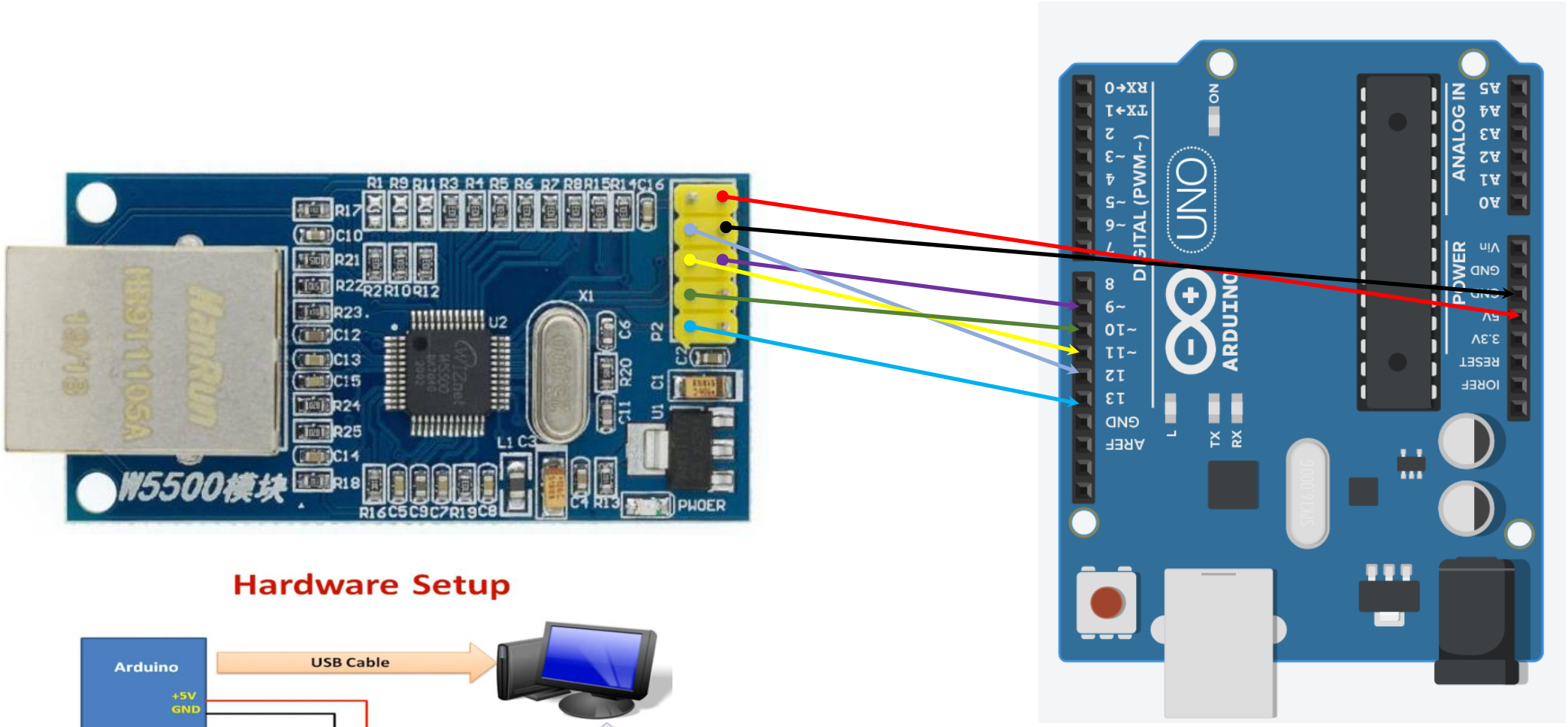


ENC28J60

ETHERNET MODULO W5500



ETHERNET MODULO W5500



TECLADO MATRICIAL
4x4 Membrana

Teclado
Matriz 4x4

Pantalla LCD
16x02

Pantalla LCD
Lcd Display 20x4

Interfaz Serial I2c
p/ Display Lcd

Display 7 Segmentos

Matriz Led 8x8
Driver Max7219

MODULO FC-37
SENSOR LLUVIA - NIEVE

Sensor Nivel De Agua

Sensor Humedad
Higrometro

MODULO MQ-4
Sensor Gas Metano

Módulo MQ-7 Sensor Gas
Monóxido de Carbono

MODULO HC-SR04
SENSOR ULTRASONIDO

Sensor de sonido KY-038

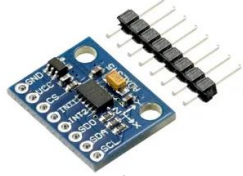
Sensor Detector Movimiento
PIRr HC-SR501

Sensor De Movimiento
RCWL-0516

SENSOR
VIBRACIÓN SW-420

Sensor de Vibración
Piezoelectrico

SENSOR DE INCLINACION
TILT SWITCH



ACELERÓMETRO
GIROSCOPIO DIGITAL
Adxl345 3 Ejes Gy-291

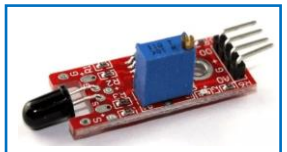
Módulo DHT11, Sensor
Temperatura y Humedad
Relativa

Sensor de Luz - LDR

Sensor Infrarojo
Ky-022

Sensor Proximidad
Obstaculos Infrarojo

Sensor TCRT5000
Seguimiento Línea
Blanco Y Negro



Sensor Llama-Flama
4 Pines LM393

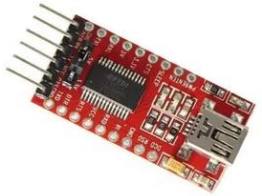
Módulo Buzzer
Zumbador Digital

BUZZER 95 dB
3-24V (12v)

MODULO RTC-Ds3231
I2C Reloj Tiempo Real

MODULO ETHERNET LAN
Enc28j60 Rj45

MODULO SERIAL 485
Rs485 TTL



Conversor Usb A Serial
TTL FT232 Ftdi 5.5v/3.3v

Conversor Adaptador
Usb Serial Ttl

Módulo Lector
Memoria Micro Sd

MOTOR DC
con Caja Reductora
Motor Gear dc

SERVO MOTOR
SG90

Motor Paso a Paso

MODULO L298n-Puente H
Controlador de Motores

Controlador De Motores
Mini Puente H (L298n)

Módulo Medición de
Velocidad / Efecto Hall

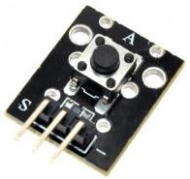
Módulo Medidor
Velocidad

Módulo LED RGB

MODULO JOYSTICK
Robótica Ps2

Modulo Relay 5v, 1 Canal
Trigger Hi/Lo

Módulo Encoder
Rotativo



Módulo Botón
Pulsador

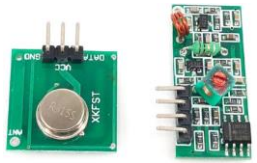
Modulo Relay 5v, 1 Canal
Trigger Hi/Lo

MODULO ACS712-5A
Sensor De Corriente

MODULO S50 TAG RFID
NFC 13,56Mhz

Módulo Wifi

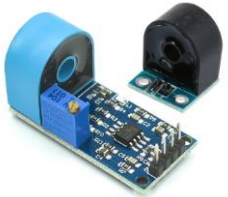
MODULO BLUETOOTH
HC-06



Módulo Tx – Rx
315 Mhz.

Modulo Gprs Gsm
Sim800L

Módulo Medidor
Voltaje



Sensor de Corriente
CA ZMCT103C

KY-027

SENSOR MAGIC LIGHT CUP
Vibración y Nivel

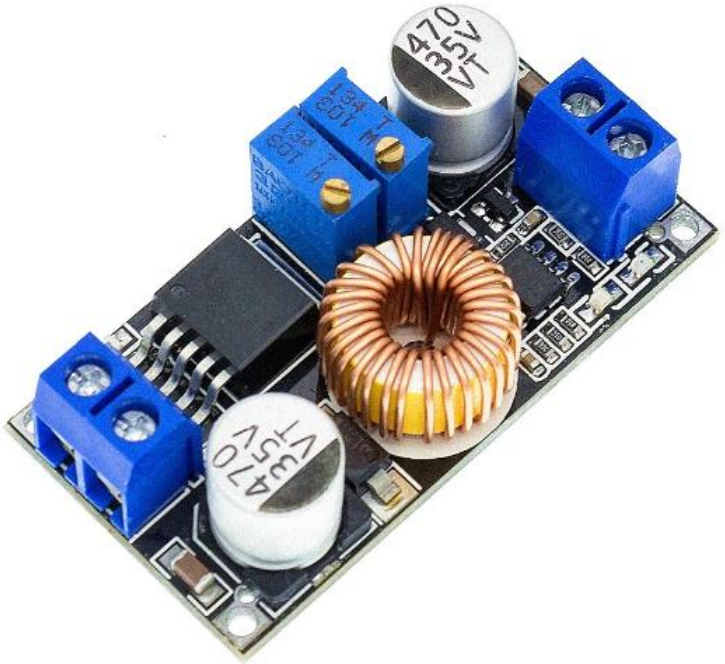
KY-035

SENSOR EFECTO HALL

KY-031

SENSOR DE GOLPE-TTL

MODULO BUCK DC/DC 4015



Tipo: **BUCK**
XL 4015 Corriente constante, **5A, BUCK DC-DC**

Voltaje de entrada: **8.0V-36V**

Voltaje de salida: **1.25-32 V**
Corriente Max. Salida: **5A**
Eficiencia: **95%**

Frecuencia de funcionamiento: 180 KHz
Limitación Corriente: Si
Protección Cortocircuito: Si
Protección Térmica: Si