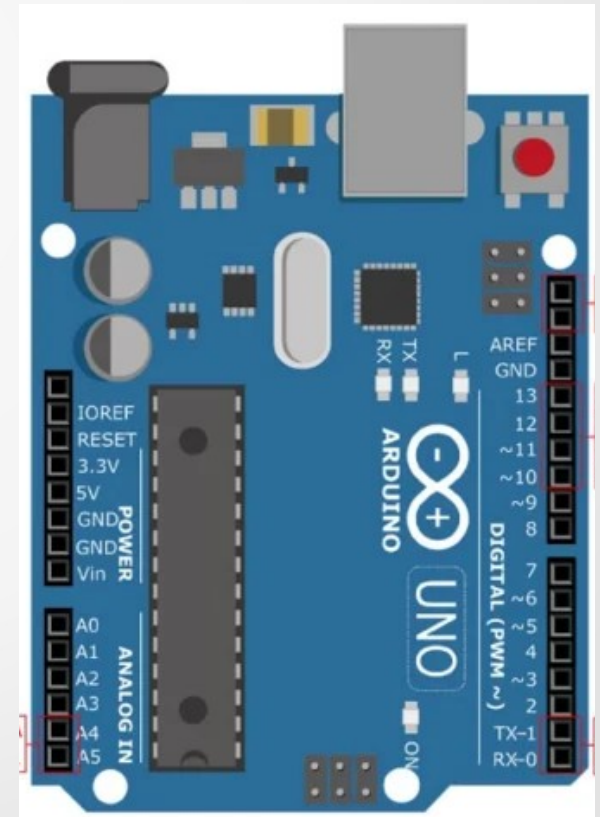


# Interacción con el exterior

- Puertos de Entrada -Salida
- Puertos de comunicación (SCI-SPI I<sup>2</sup>C,etc)
- Entradas y salidas analógicas



# Puertos de comunicación serie

Puertos de comunicación serie,  
Generalidades y aspectos prácticos

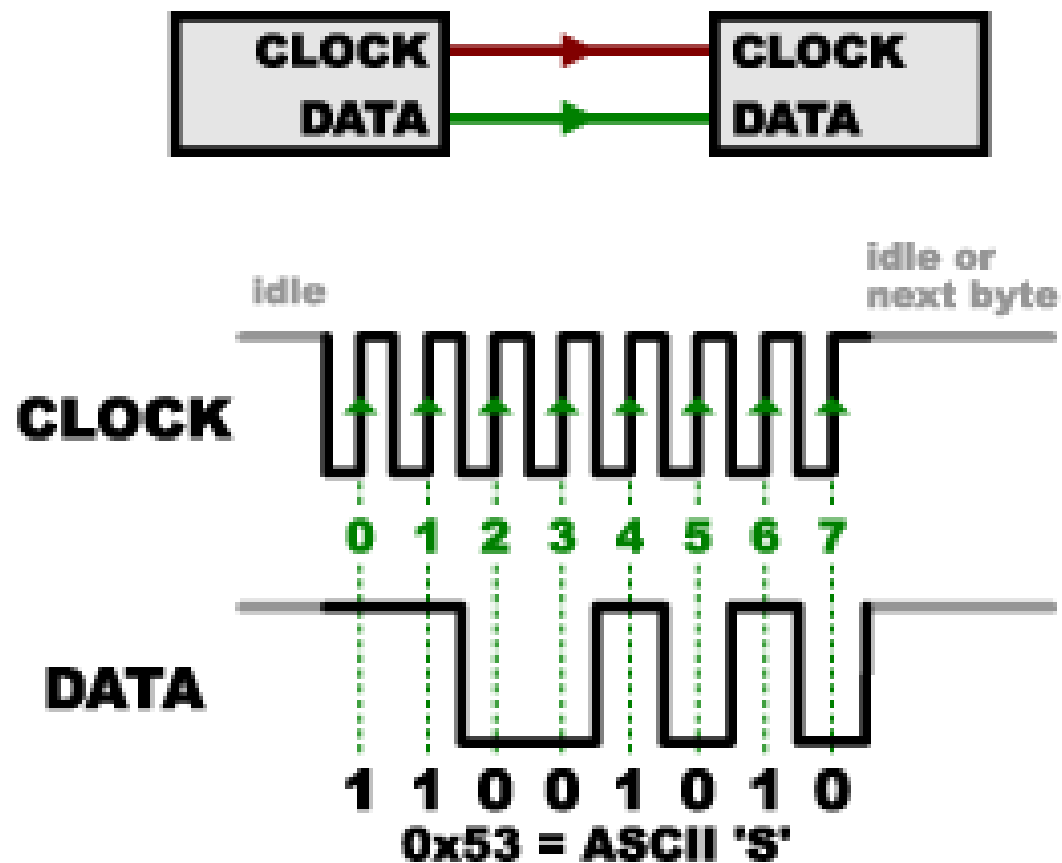
# Temario

- Comunicación serie sincrónica
- Comunicación serie asincrónica
- La señal lógica y su configuración
- El uso de com. Serie en Arduino
- Hardware: la norma RS232
- Hardware: la conexión del Arduino UNO
- Las herramientas de Arduino
- Ejemplo de uso

# Comunicación sincrónica



# Comunicación sincrónica

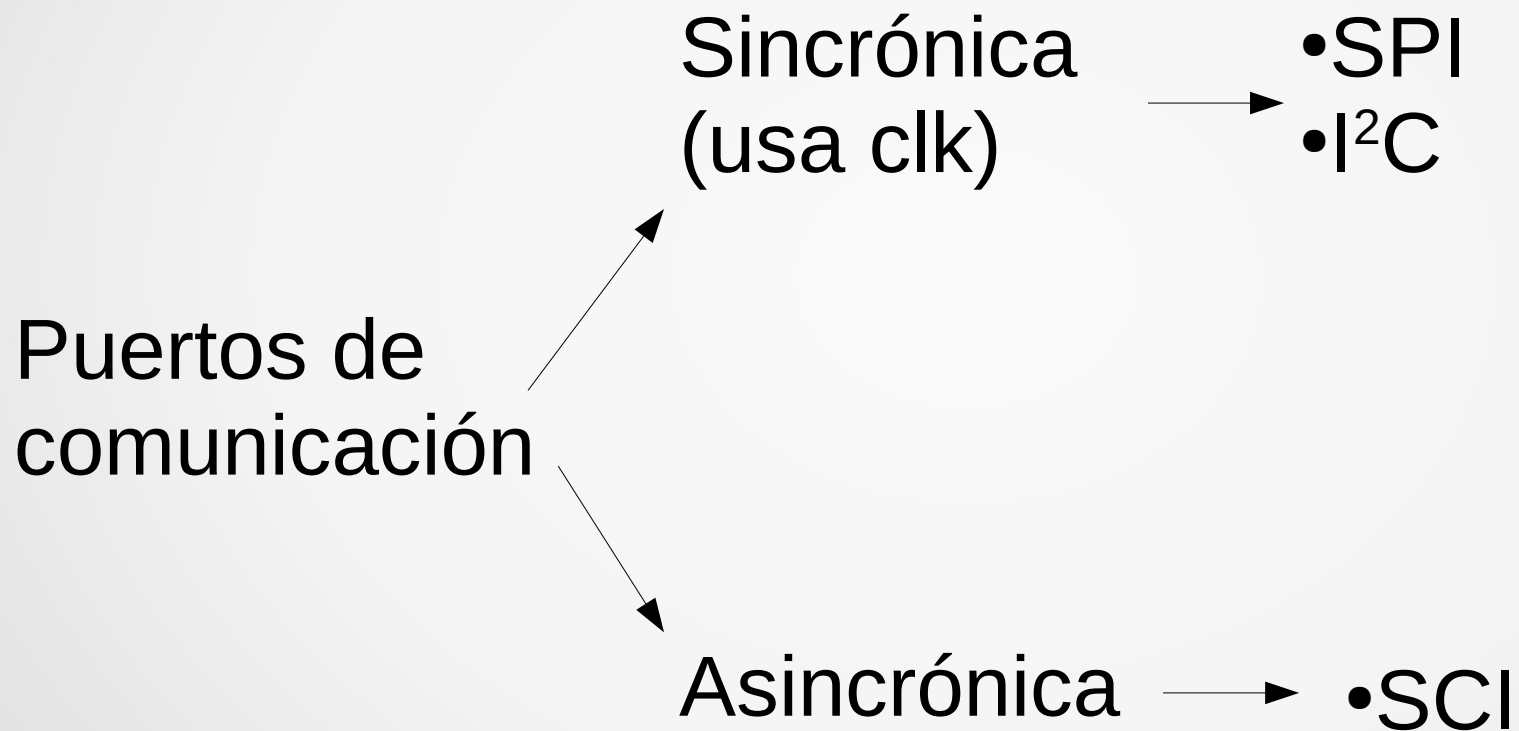




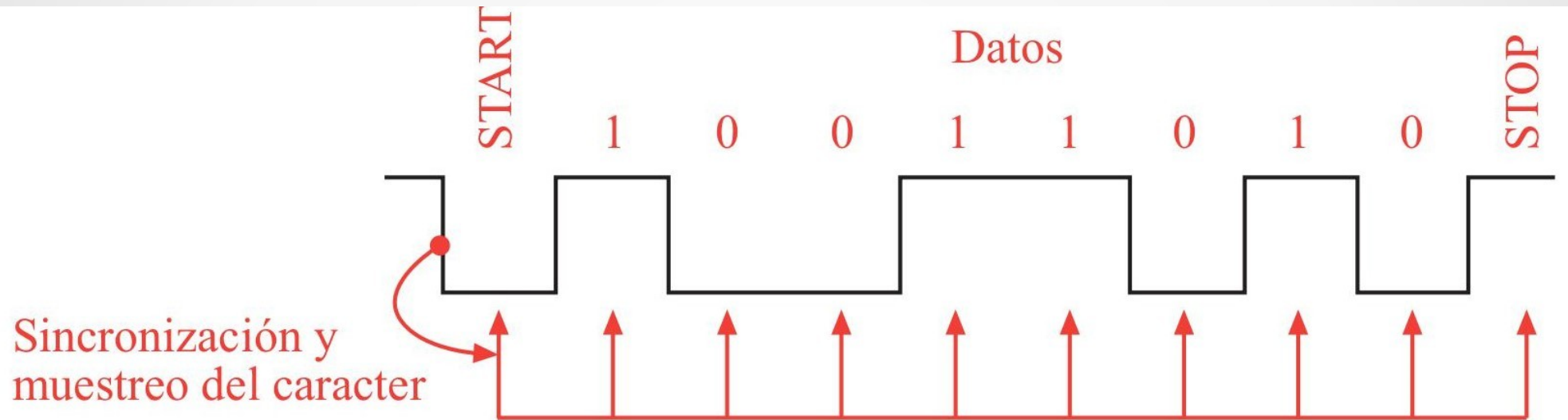
# Comunicación asincrónica



# Clasificación



# Señal lógica





# Configuración

Los parámetros más importantes a configurar de la comunicación serial son:

- La velocidad de transmisión o baud-rate (1200, 9600, 115200 etc)
- El número de bits de datos (5, 6, 7,8 ó 9)
- El número de bits de stop (1 ó 2)
- Y si cuenta con bit de paridad (par, impar o ninguna)

# ¿Cómo se configura en Arduino?

`Serial.begin()`

`Serial.begin(speed)`

`Serial.begin(speed, config)`

`SERIAL_5N1`

`SERIAL_6N1`

`SERIAL_7N1`

`SERIAL_8N1` (the default)

`SERIAL_5N2`

`SERIAL_6N2`

`SERIAL_7N2`

`SERIAL_8N2`

`SERIAL_5E1`: even parity

# ¿Y cómo se usa?

`Serial.print(78)` transmite "78"

`Serial.print(1.23456)` transmite "1.23"

`Serial.print('N')` transmite "N"

`Serial.print("Hello world.")` transmite "Hello world."

`Serial.print(78, BIN)` transmite "1001110"

`Serial.print(78, DEC)` transmite "78"

`Serial.print(78, HEX)` transmite "4E"

`Serial.print(1.23456, 0)` transmite "1"

`Serial.print(1.23456, 4)` transmite "1.2345"

# Programa ejemplo:

```
void setup()
{
  Serial.begin(9600); // 9600 bits por segundo
}

int number = 0;

void loop()
{
  Serial.print("El número es ");
  Serial.println(number); // transmite el número
  delay(500); // retardo
  number++;
}
```



# ¿Y si necesito leer?

```
int incomingByte = 0;

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

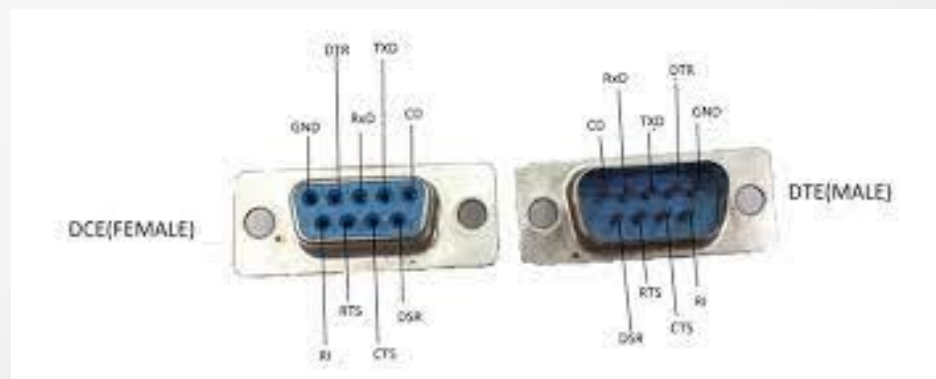
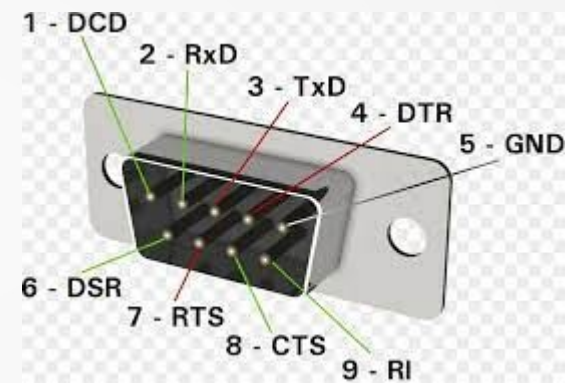
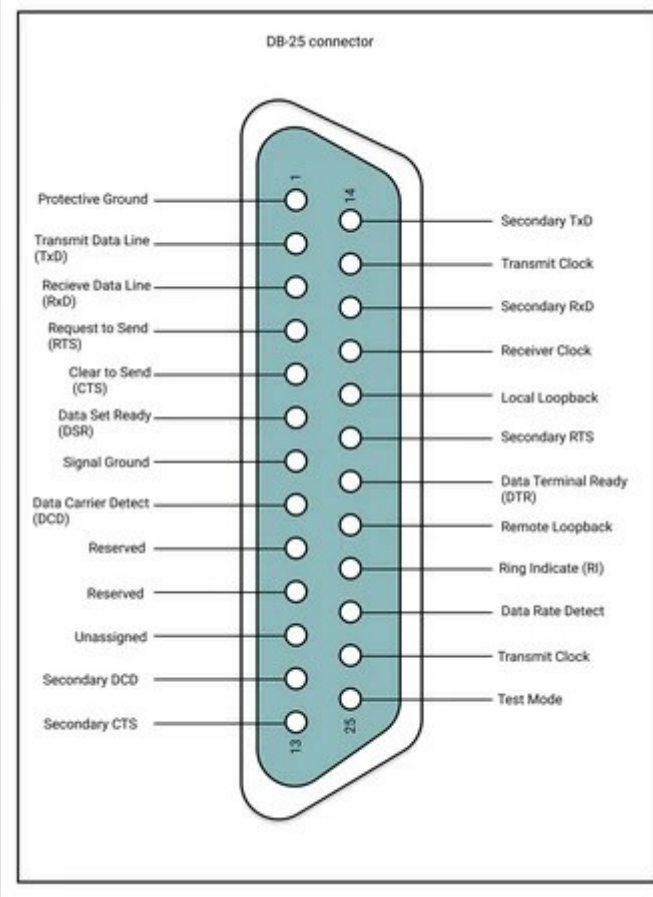
void loop() {
  if (Serial.available() > 0) {
    incomingByte = Serial.read();
  }
}
```

# Hardware



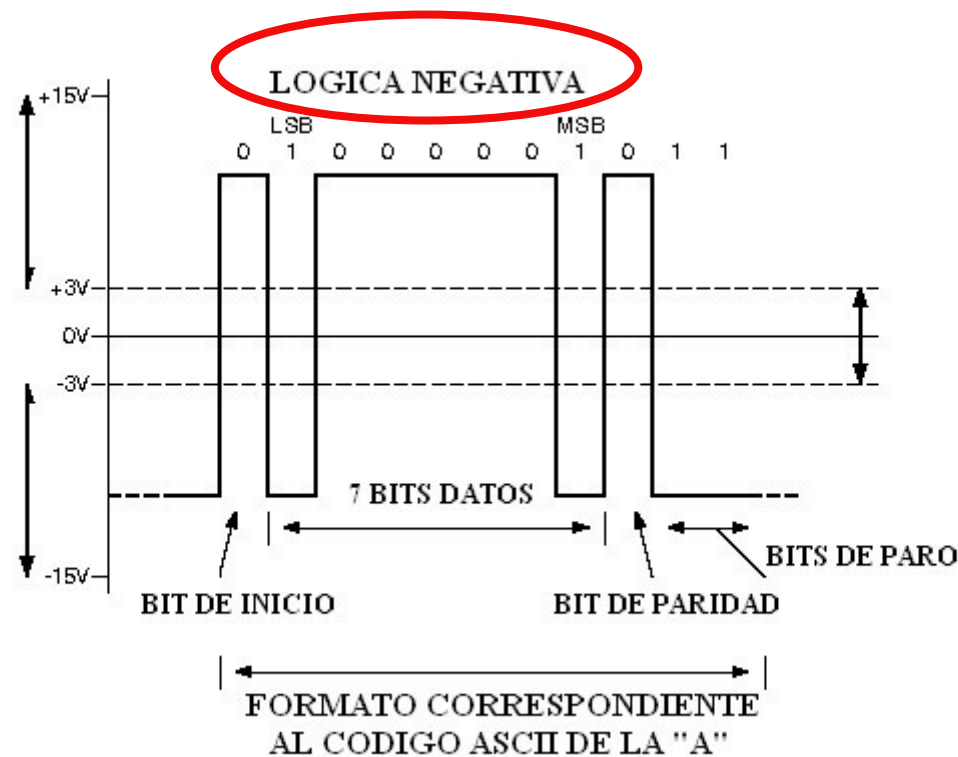
# Norma RS232

La norma determina: El tipo de conectores y el pinout de los mismos



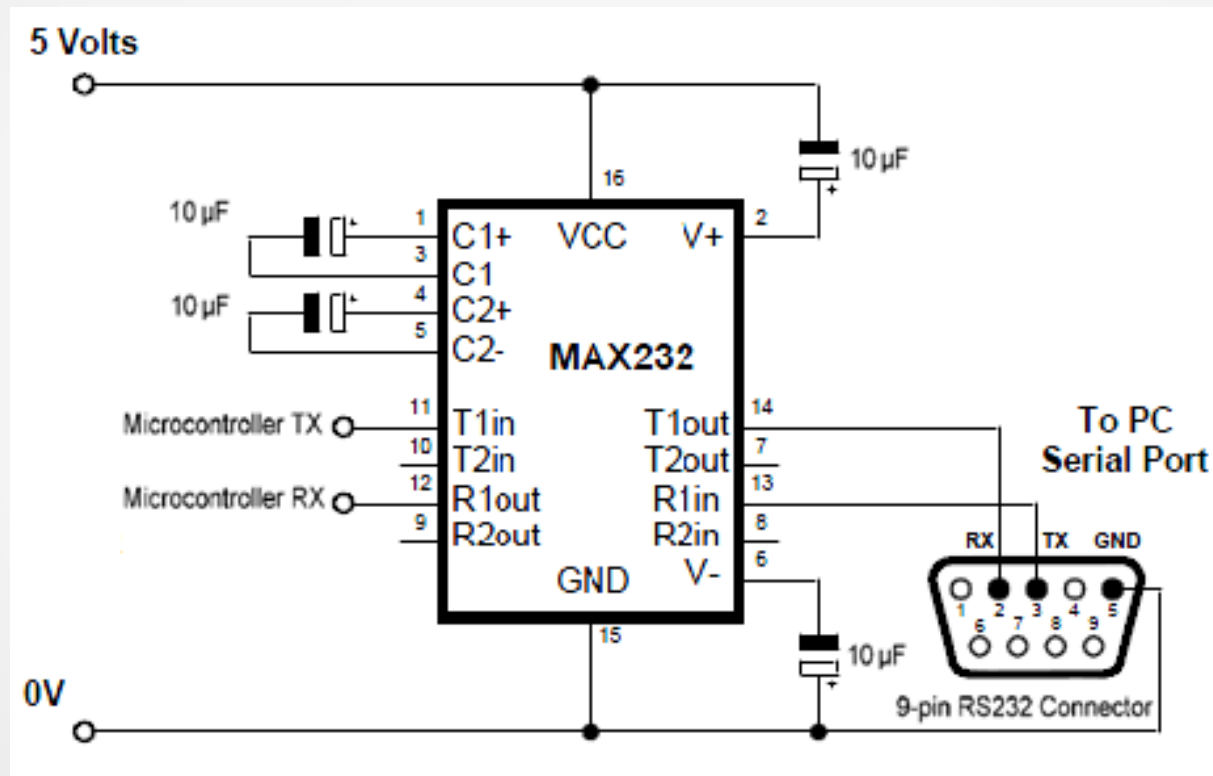
# Norma RS232

La norma determina: El nivel de tensión de las señales y los demás parámetros eléctricos (impedancia de salida, impedancia de entrada, etc)

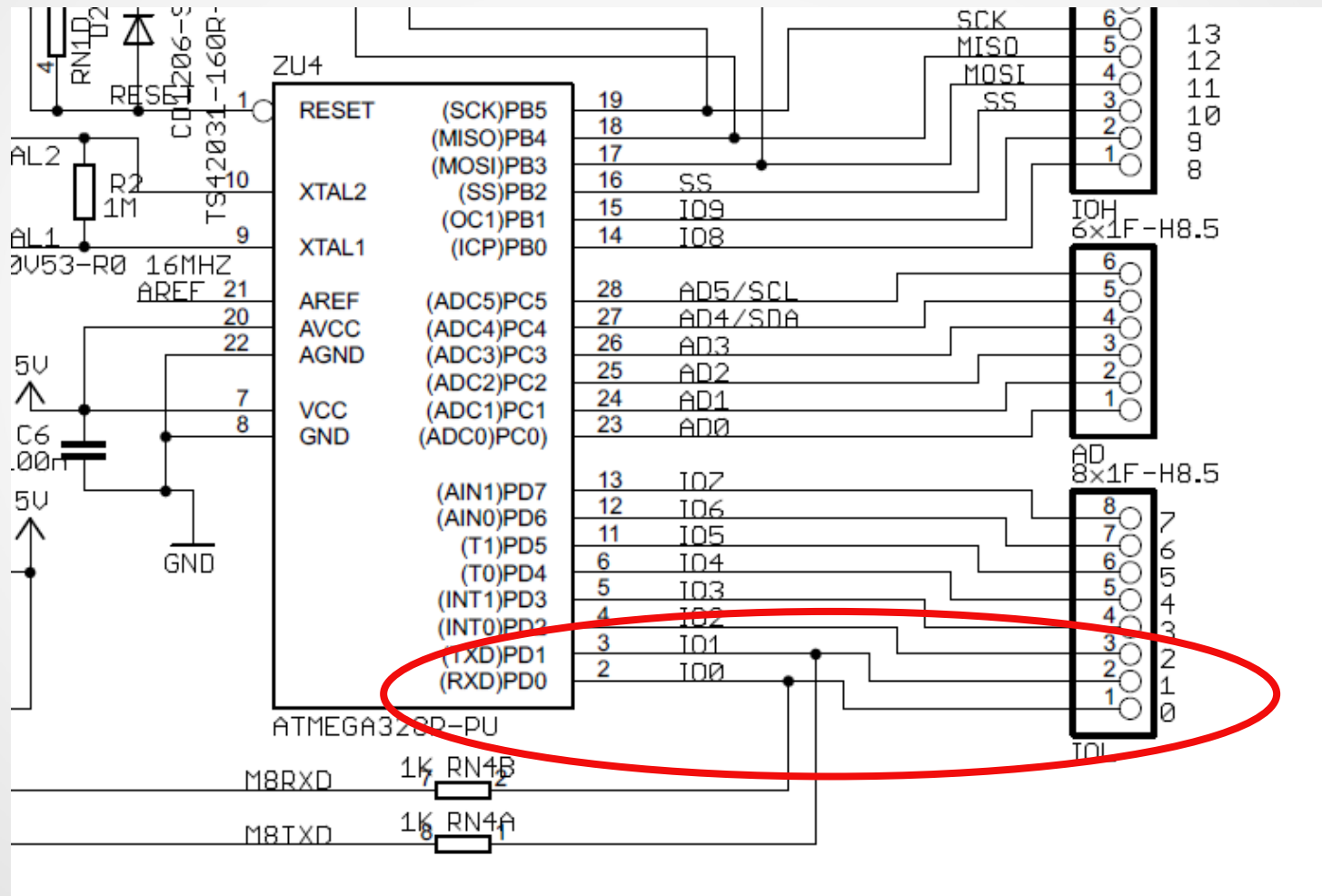




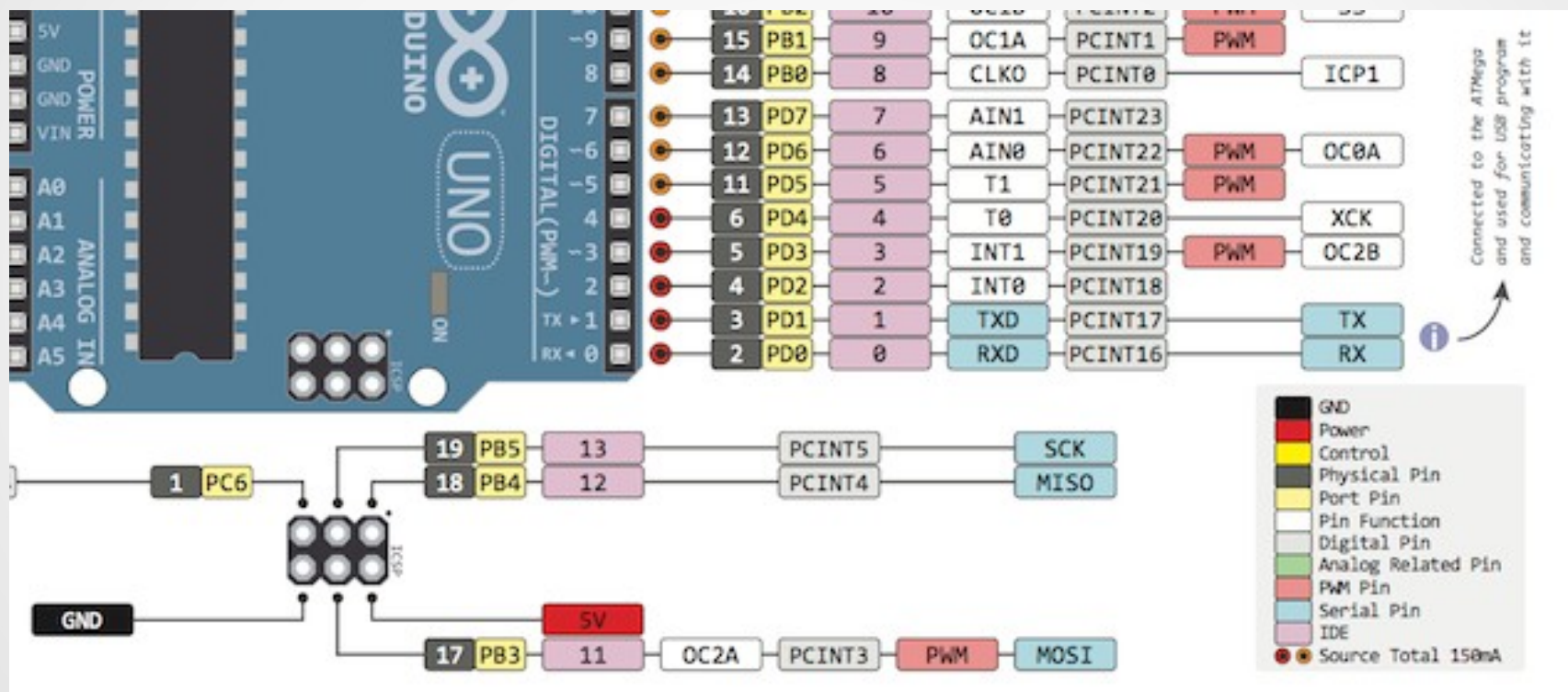
# Norma RS232



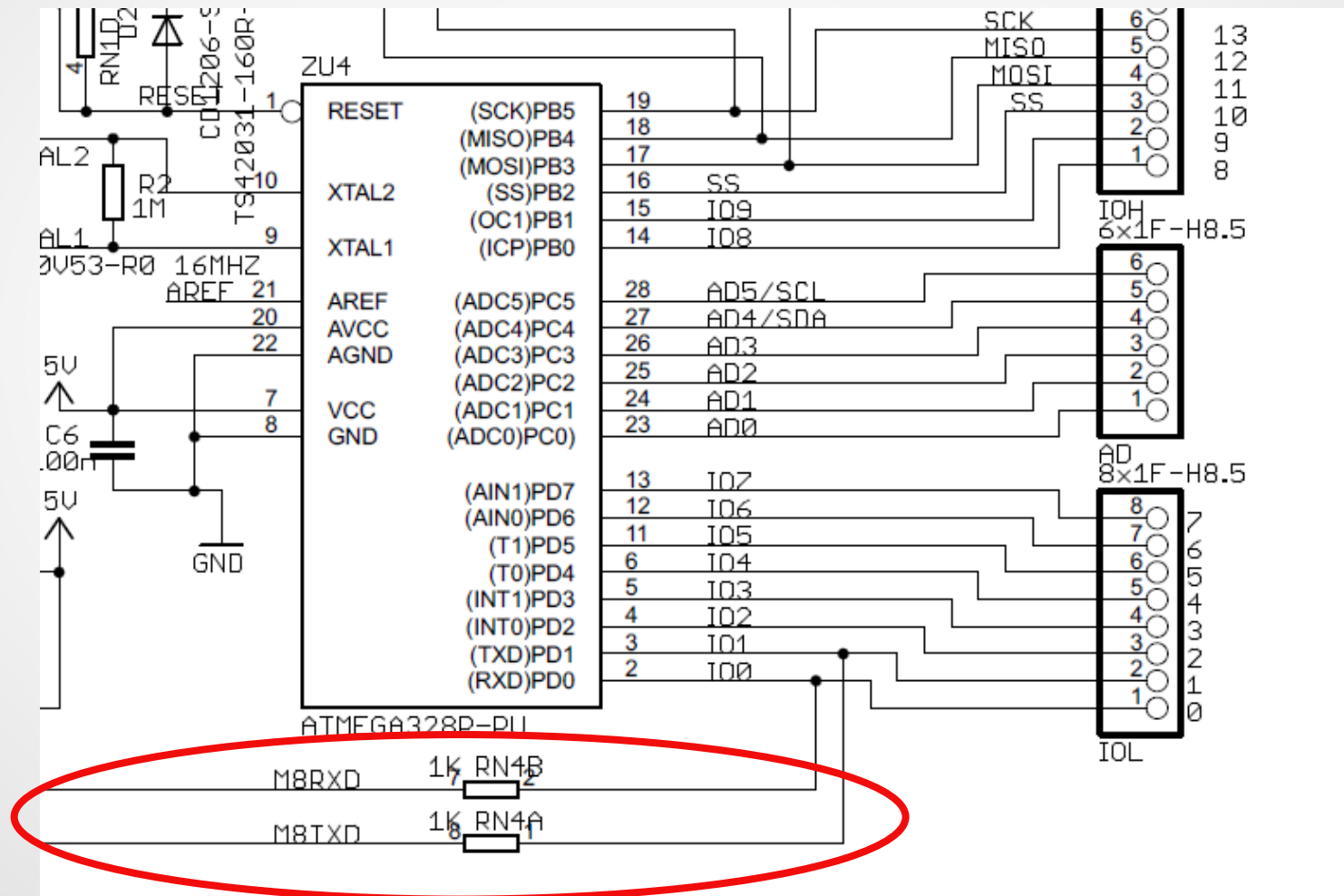
# Puerto SCI en el Arduino UNO



# Puerto SCI en el Arduino UNO

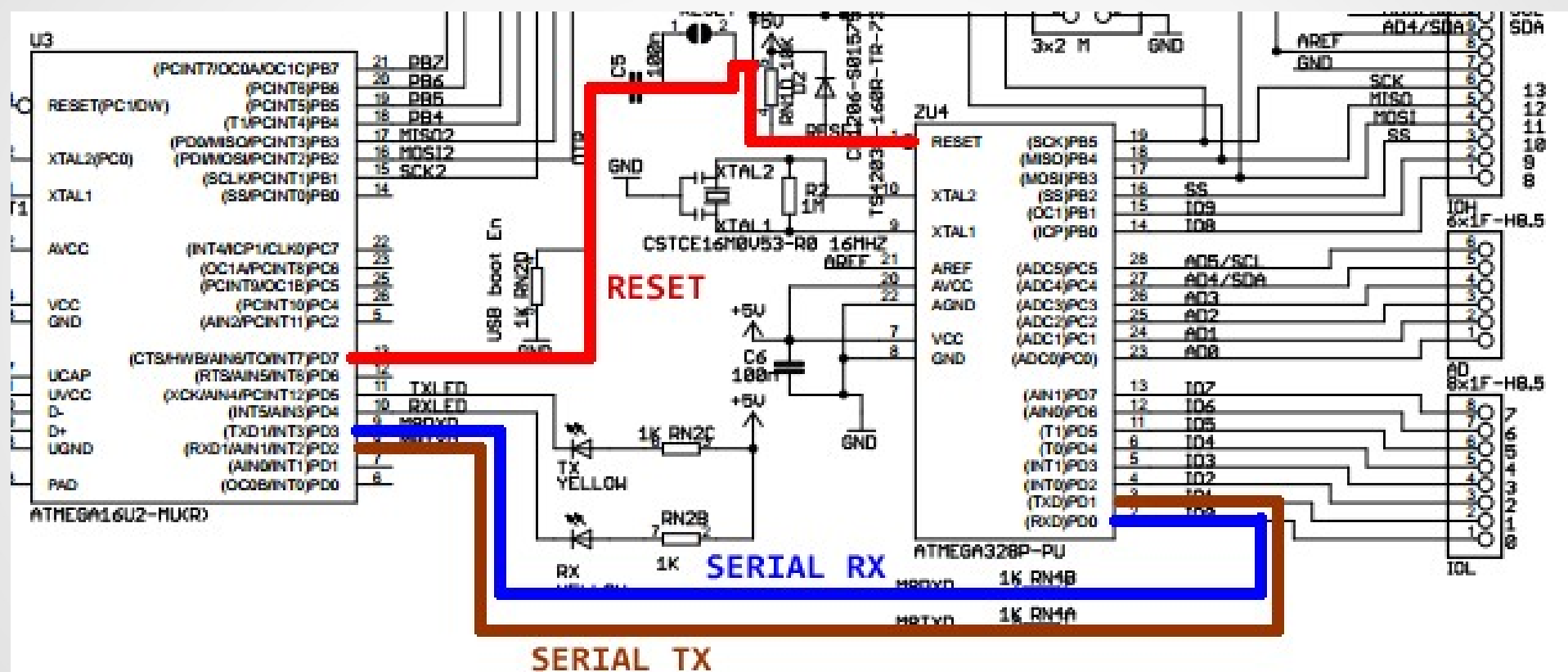


# Puerto SCI en el Arduino UNO

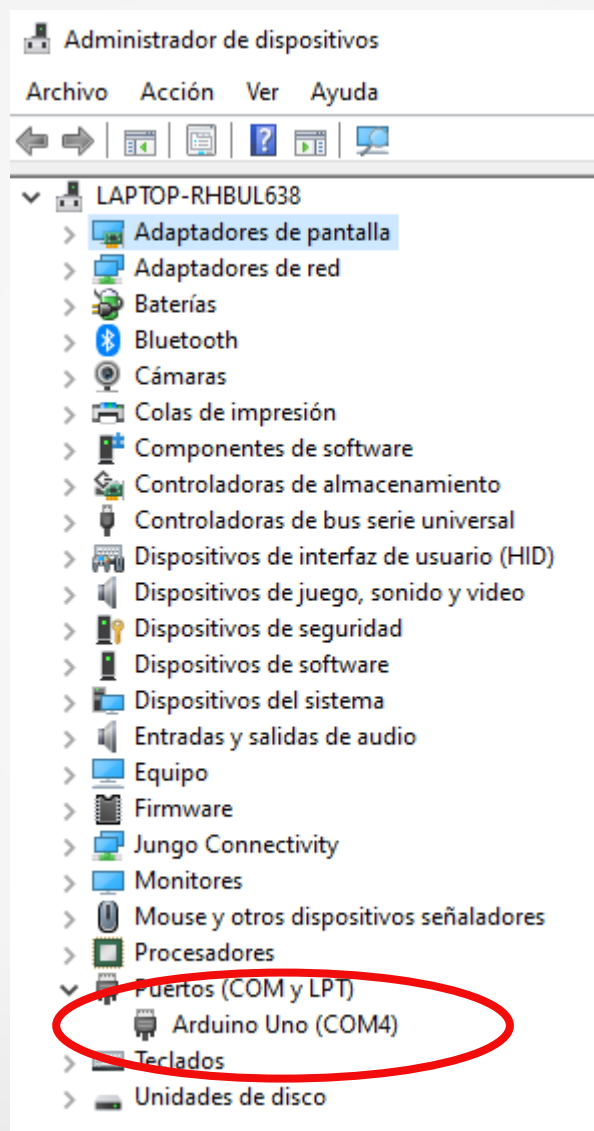




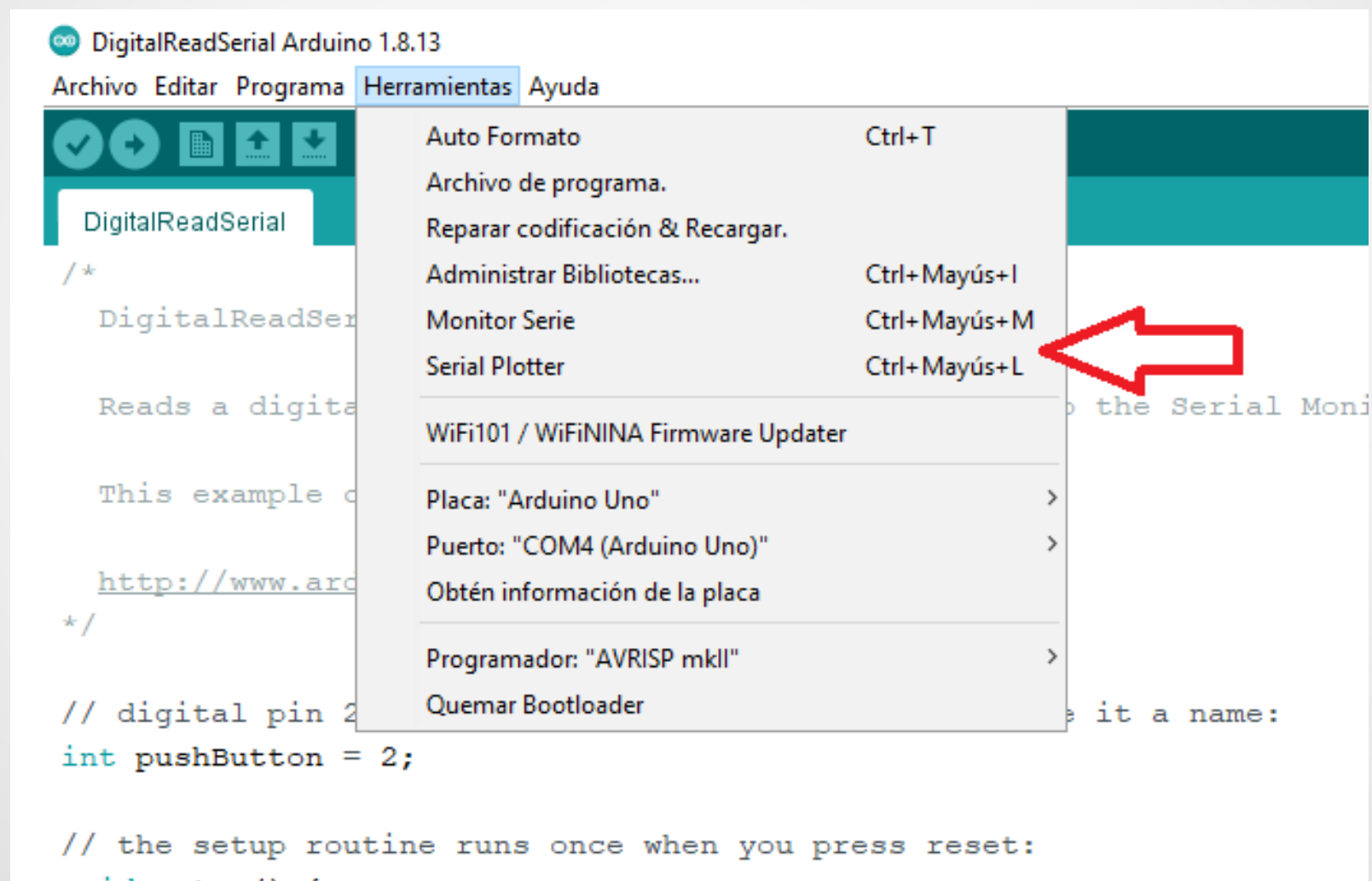
# Puerto SCI en el Arduino UNO



# Puerto SCI en el Arduino UNO



# Puerto SCI en el Arduino UNO



# Puertos de comunicaciones serie

Fin.....