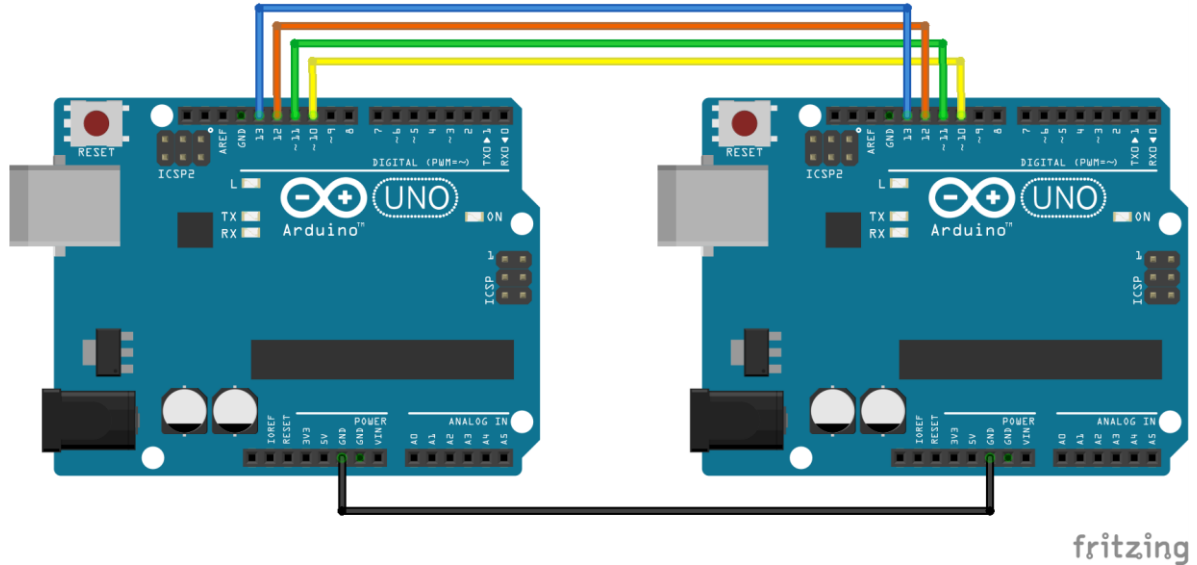


Práctica 1 - Serial Peripheral Interface

Para esta práctica se la librería SPI.h.

1.- Conectar dos Arduino siguiendo el diagrama siguiente:



La asignación de pines quedará de la siguiente manera:

- (SS) : pin 10
- (MOSI) : pin 11
- (MISO) : pin 12
- (SCK) : pin 13
- GND(Tierra): común entre ambos

2.- Cargar el código correspondiente a cada Arduino para su funcionamiento como Master o Esclavo.

3.- Verificar la salida del código en el monitor serial.

Arduino Master

```
#include <SPI.h>

void setup (void) {
  Serial.begin(115200); //Se establece la velocidad de conexión
  digitalWrite(SS, HIGH); // Se deshabilita la línea de selección
  SPI.begin (); //Inicializa el protocolo
  SPI.setClockDivider(SPI_CLOCK_DIV8);/*Se establece el divisor de
  reloj a 8*/
}

void loop (void) {
  char c;
  digitalWrite(SS, LOW); // Se habilita la línea de selección
  // Se envía un mensaje al esclavo
  for (const char * p = "Prueba de envío.\r" ; c = *p; p++) {
    SPI.transfer (c);
    Serial.print(c);
  }
  digitalWrite(SS, HIGH); // Se deshabilita la línea de selección
  delay(2000);
}
```

Arduino Esclavo

```
#include <SPI.h>
char buff [50];
volatile byte indx;
volatile boolean process;

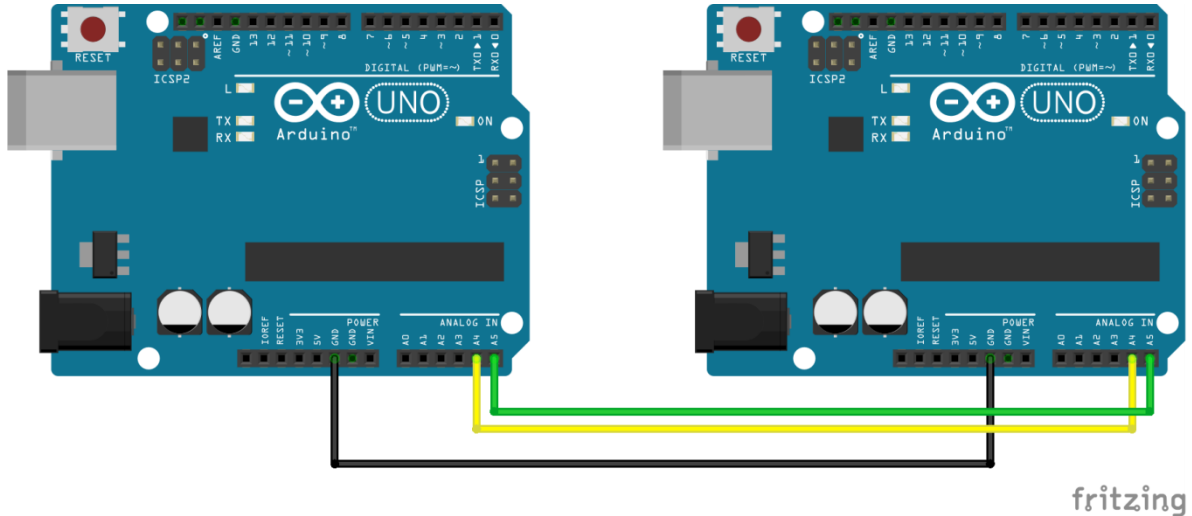
void setup (void) {
  Serial.begin (115200);
  pinMode(MISO, OUTPUT); // Establecemos salida para comunicación con el
  master
  SPCR |= _BV(SPE); // Se habilita el modo esclavo
  indx = 0; // Se inicializa la variable para el mensaje
  process = false;
  SPI.attachInterrupt(); // Se activa la interrupción
}
ISR (SPI_STC_vect) // rutina de interrupción SPI {
  byte c = SPDR; // Se leen los datos del registro SPI
  if (indx < sizeof buff) {
    buff [indx++] = c; // Se guarda el dato en la siguiente posición
    if (c == '\r') // Se verifica el fin del mensaje
      process = true;
  }
}

void loop (void) {
  if (process) {
    process = false; // Se reestablece el proceso
    Serial.println (buff); // Se imprime el mensaje en consola
    indx= 0; // Se reinicia el valor de la variable de posición
  }
}
```

Práctica 2 - Inter-Integrated Circuit

Para esta práctica se utilizará la librería Wire.h.

1.- Conectar dos Arduino siguiendo el diagrama siguiente:



La asignación de pines quedará de la siguiente manera:

- SDA: pin 4
- SCL: pin 5
- GND (Tierra): común entre ambos

2.- Cargar el código correspondiente al Arduino para funcionar como Master o Esclavo.

3.- Verificar la salida del código en el monitor serial.

Arduino Esclavo

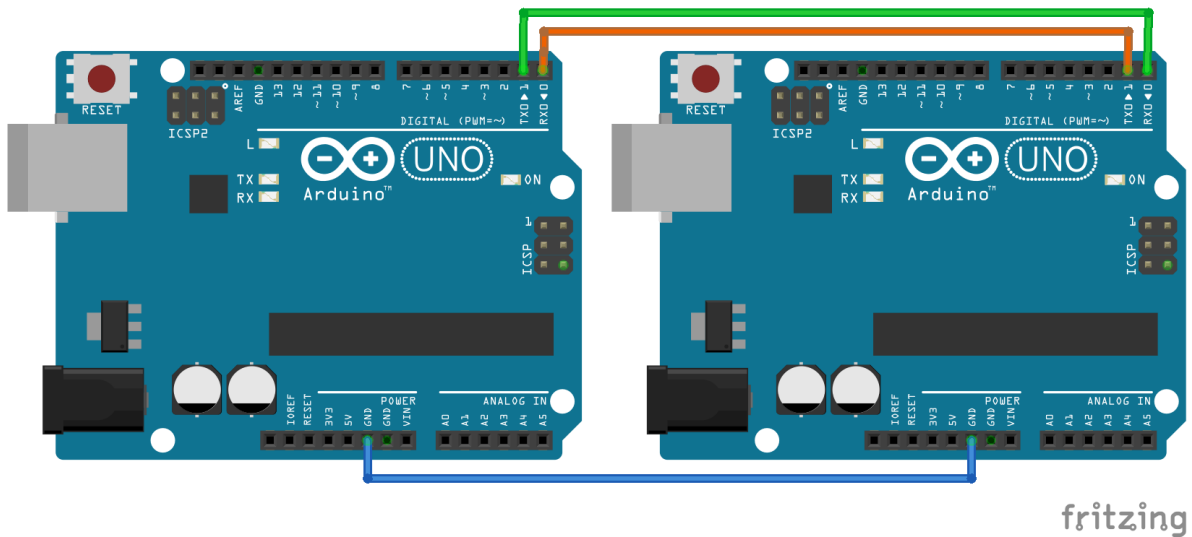
```
#include < Wire.h >
void setup () {
  Wire.begin ( 8 ); // Se establece la dirección 8 para el bus
  Wire.onRequest ( requestEvent ); // Función que se ejecutará a petición
}
void loop () {
  delay ( 100 ); // pausa de 100 milisegundos
}
// Función que se ejecutará cuando el master requiera información
// Esta función se registra como un evento
void requestEvent () {
  Wire.write ( "hello" ); // Respuesta de 6 bytes, esperada por el master
}
```

Arduino Master

```
#include <Wire.h >
void setup () {
  Wire.begin (); // Se inicia el bus (la dirección es opcional para el master)
  Serial.begin ( 9600 ); // Se inicia el puerto serie con velocidad de 9600
}
void loop () {
  Wire.requestFrom ( 8 , 6 ); // Se piden 6 bytes del dispositivo con dirección 8
  while (Wire.available () ) { // Se verifica que haya una conexión disponible
    char c = Wire.read (); // Se recibe un byte como un carácter
    Serial.print (c); // Se imprime el carácter
  }
  delay ( 500 ); // Pausa de 500 milisegundos
}
```

Práctica 3 - UART

1.- Conectar dos Arduino siguiendo el diagrama siguiente:



La asignación de pines quedará de la siguiente manera:

Arduino #1	Arduino #2
RX: pin 0	TX: pin 1
TX: pin 1	RX: pin 0
GND (Tierra)	GND (Tierra)

2.- Cargar el código correspondiente al Arduino transmisor y receptor respectivamente.

3.- Verificar la salida en el monitor serial.

Arduino Transmisor

```
char mensaje[5] = "Hola!"; //Mensaje a enviar

void setup() {
  Serial.begin(9600); // Se inicializa el puerto serie a 9600 bps
}

void loop() {
  Serial.write(mensaje,5); /*Se escribe el mensaje con longitud de
5 caracteres*/
  delay(1000); //Pausa de 1 segundo
}
```

Arduino Receptor

```
char mensaje[10]; //Variable para almacenar el mensaje
  Serial.begin(9600); // Se inicializa el puerto serie a 9600 bps
}

void loop() {
  Serial.readBytes(mensaje,5); //Se lee el mensaje recibido
  Serial.println(mensaje); //Se imprime el mensaje
  delay(1000); // Pausa de 1 segundo
}
```

Referencias:

Serial Communication (2018). Recuperado de: <https://www.arduino.cc/en/Reference/Serial>

SM (2018, Mayo 17). Master Reader/Slave Sender. Recuperado de:
<https://www.arduino.cc/en/Tutorial/MasterReader>