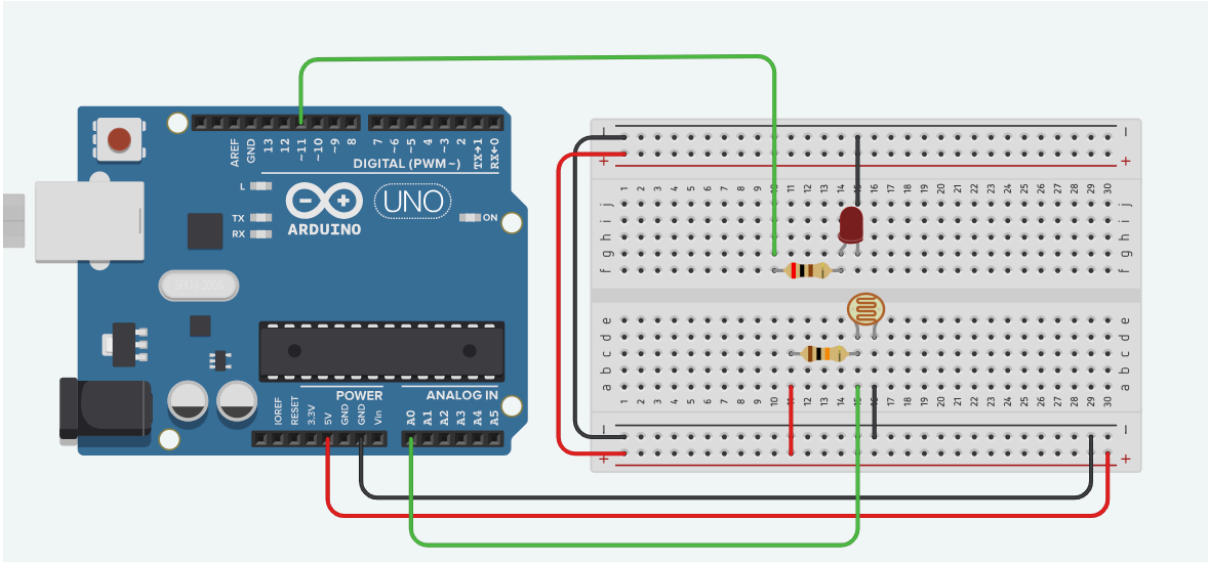
Real-Time Systems and Interfacing Techniques Laboratory
Raghad Afaghani 1192423

Introduction to Arduino

1. Photocells:

Photocells are basically a resistor that changes its resistive value (in ohms Ω) depending on how much light is shining onto the squiggly face.

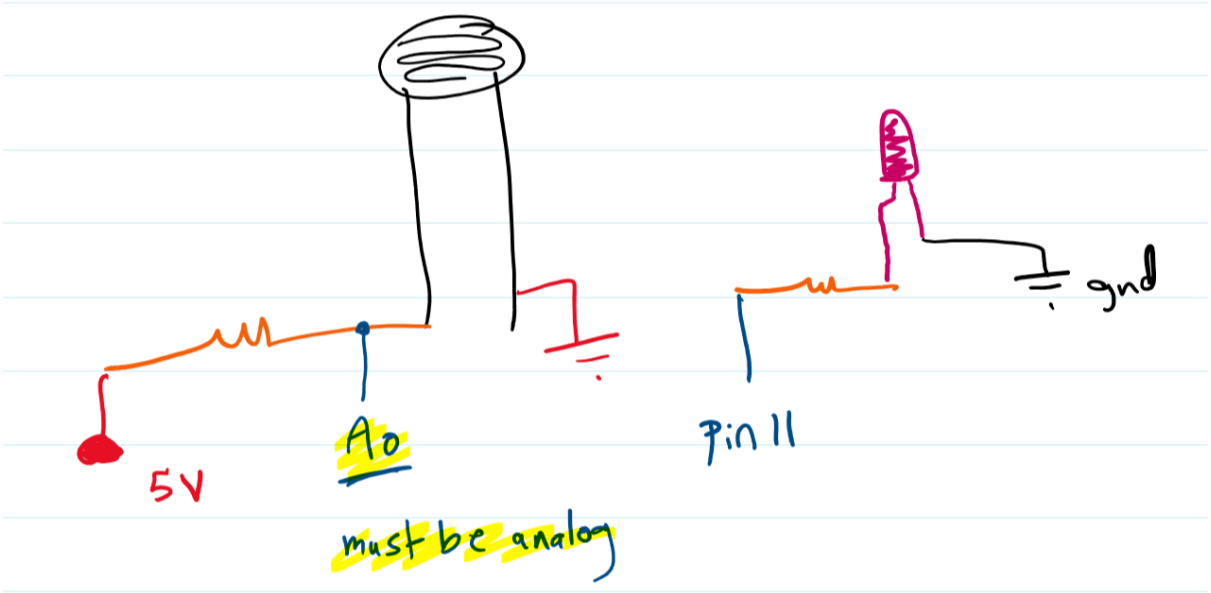
- ↑ light level, then ↓ the Photocell resistance
- ↓ light level, then ↑ the Photocell resistance
- Voltage divider rule: Vo = Vcc (R / (R + Photocell))
- $Vo \propto \frac{1}{\text{Photocell resistance}}$
- $\text{Photocell resistance} \propto \frac{1}{\text{light level}}$



This sketch will take the analog voltage reading and use that to determine how bright the red LED is. The darker it is, the brighter the LED will be! Remember that the LED has to be connected to a PWM pin for this to work, I use pin 11 in this example.

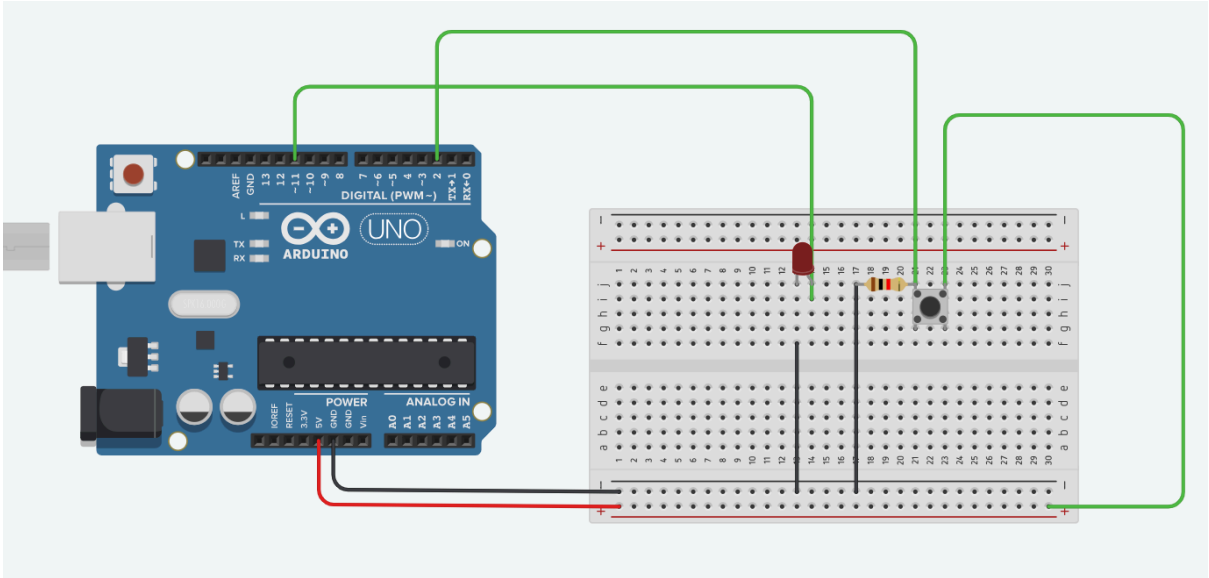
```
int photocellPin = 0;     // the cell and 10K pulldown are connected to A0
int photocellReading;     // the analog reading from the sensor divider
int LEDpin = 11;          // connect Red LED to pin 11 (PWM pin)
int LEDbrightness;

void setup(void) {
```

```
  // We'll send debugging information via the Serial monitor
  Serial.begin(9600);
}

void loop(void) {
  photocellReading = analogRead(photocellPin);
  Serial.print("Analog reading = ");
  Serial.print(photocellReading);  // the raw analog reading

  if (photocellReading > 800) {
          Serial.println(" - Dark");
  } else if (photocellReading < 300) {
          Serial.println(" - Bright");
  } else {
          Serial.println(" - Light");
  }

  // LED gets brighter the darker it is at the sensor
  // now we have to map 0-1023 to 0-255 since that's the range analogWrite uses
  LEDbrightness = map(photocellReading, 0, 1023, 0, 255);
  analogWrite(LEDpin, LEDbrightness);

  delay(500);
}
```



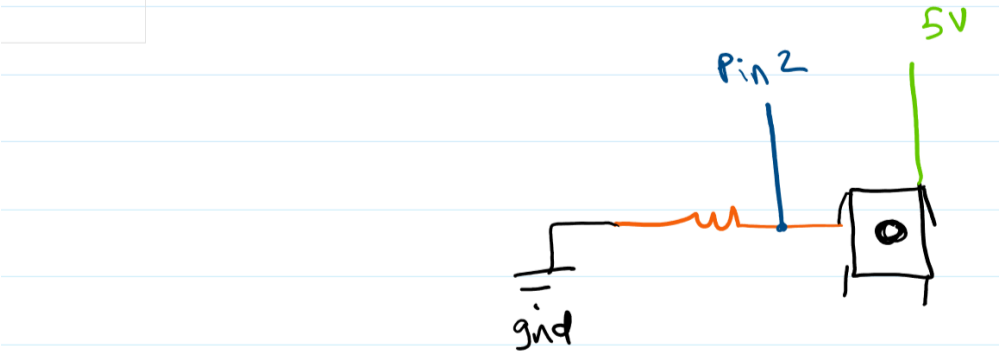
2. Hardware Interrupt

- Int 0: on digital pin 2
- Int 1: on digital pin 3
- attachInterrupt(interrupt, function, mode)
  - interrupt: the number of the interrupt (int): 0/1
  - function: the function to call when the interrupt occurs; this function must take no parameters and return nothing. This function is sometimes referred to as an interrupt service routine.
  - mode: defines when the interrupt should be triggered. Four contstants are predefined as valid values:
    - LOW to trigger the interrupt whenever the pin is low,
    - CHANGE to trigger the interrupt whenever the pin changes value
    - RISING to trigger when the pin goes from low to high,
    - FALLING for when the pin goes from high to low.



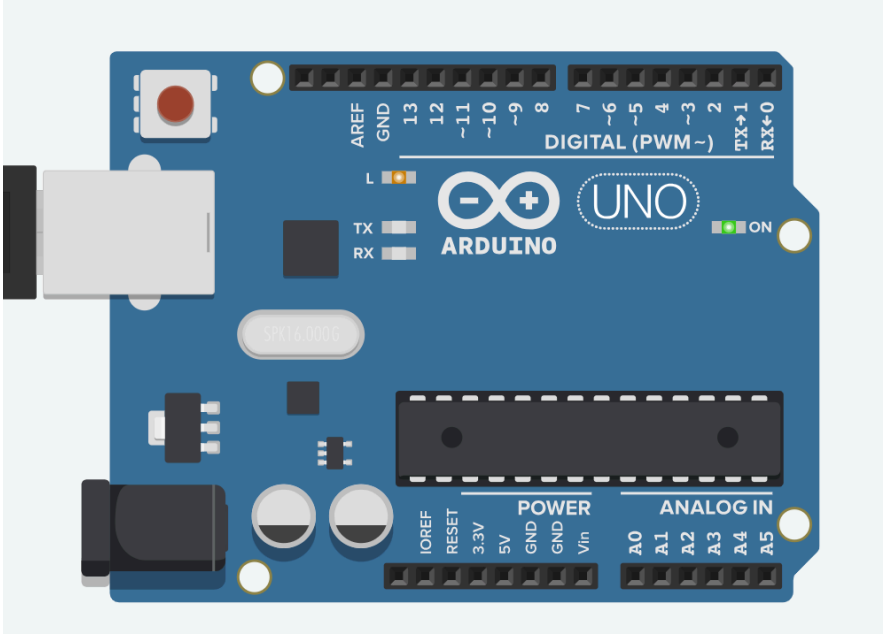

```
// Hardware interrupt example
int pin = 11; // for LED
int pin_button = 2; // for interrupt
volatile int state = LOW;
void setup() {
pinMode(pin, OUTPUT);
pinMode(pin_button, INPUT_PULLUP);
attachInterrupt(0, blink, RISING);
}
void loop() {
digitalWrite(pin, state);
}
void blink() {
state = !state;
```

```
}
```



هدف انا بس بعكس حالة الـ LED عند الضغط على الـ button
باستخدام الـ Hardware Interrupt

3. Software Interrupt




```
// Timer1 overflow interrupt example
#define ledPin 13
void setup() {
  pinMode(ledPin, OUTPUT);
  // initialize timer1
  noInterrupts();  // disable all interrupts
```

```
 TCCR1A = 0;
 TCCR1B = 0;
 TCNT1 = 57723;                  // preload timer 65536-16MHz/256/2Hz
 TCCR1B |= (1 << CS12) | (1 << CS10);  // 256 prescaler
 TIMSK1 |= (1 << TOIE1);         // enable timer overflow interrupt
 interrupts();                   // enable all interrupts
}
ISR(TIMER1_OVF_vect)  // interrupt service routine that wraps a user defined function supplied by attachInterrupt
{
 TCNT1 = 57723;  // preload timer
 digitalWrite(ledPin, digitalRead(ledPin) ^ 1);
}
void loop() {
 // your program here...
}
```

15.11.1 TCCR1A – Timer/Counter1 Control Register A

Bit	7	6	5	4	3	2	1	0	
(0x80)	COM1A1	COM1A0	COM1B1	COM1B0	–	–	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

15.11.2 TCCR1B – Timer/Counter1 Control Register B

Bit	7	6	5	4	3	2	1	0	
(0x81)	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Table 15-6. Clock Select Bit Description

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{I/O}$/1 (no prescaling)
0	1	0	$clk_{I/O}$/8 (from prescaler)
0	1	1	$clk_{I/O}$/64 (from prescaler)
1	0	0	$clk_{I/O}$/256 (from prescaler)
1	0	1	$clk_{I/O}$/1024 (from prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

. Software Interrupt لاستخدام ال blinking of LED

Time = 2 sec $\Rightarrow f = \frac{1}{2} = 0.5$ (using Timer 1 [16 bit])

$$T_1 = 65535 - \frac{f_{osc}}{pre * f}$$

$$65535 - \left(\frac{16000\,000}{256 \times 0.5}\right)$$

كيف دخلناه

لازم احطه بحيث انه الرقم الي بيطلع من الاقواس ما يطلع اكبر من 65535 عشان ما يطلع سالب

```
TCCR1B |= (1 << CS12)|(1 << CS10);        // 1024 prescaler
```

To Do (PWM)

Write a code to generate a square wave on the pin 13 with frequency 1 KHz and duty cycle that is controllable by a potentiometer.

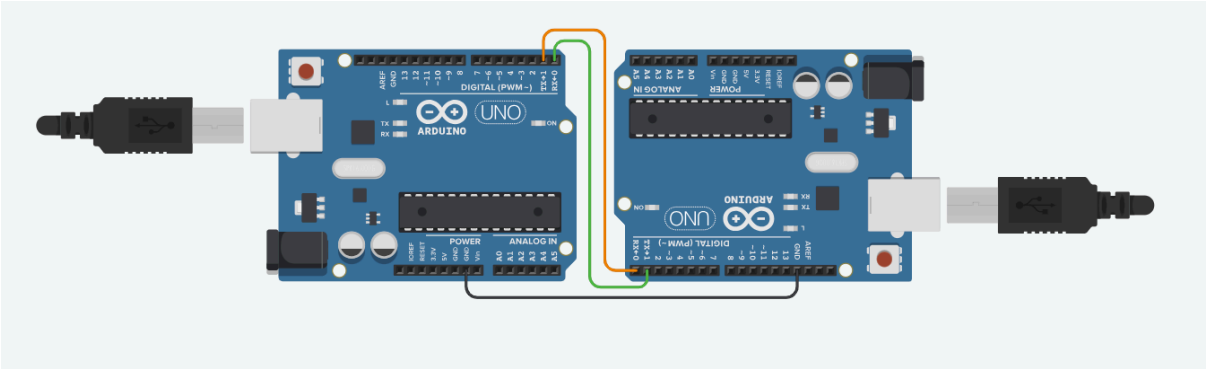
$f = 1000$

$$65535 - \frac{16000\,\sout{000}}{pre * \sout{1000}}$$

$$65535 - 16000$$

$$= 49535$$

Experiment #11
Serial Communication with Arduino



The sender will be responsible of sending "Hello World!" to the receiver Arduino. First, we should specify the serial communication Setup [Baud rate, stop bits, ...] and then send!

Sender

```
void setup() {
  // Setup the Serial at 9600 Baud
  Serial.begin(9600);
}

void loop() {
  Serial.println("Hello World!");  //Write the serial data
  delay(2000);
}
```

The receiver will be responsible of receiving the data and then displaying it. First, we should specify the serial communication Setup [Baud rate, stop bits, ...] and then read the data!

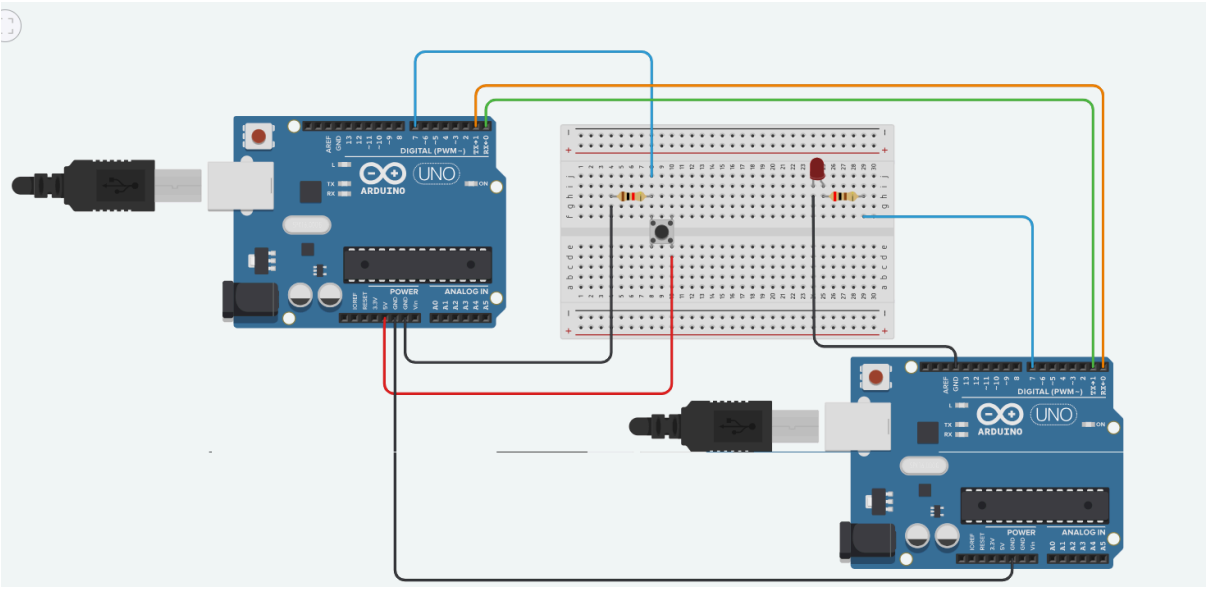
Receiver

```
void setup() {
  // Begin the Serial at 9600 Baud
  Serial.begin(9600);
}
```

```
void loop() {
  if (Serial.available()) {
          String data = Serial.readString();  //Read the serial data and store in var
          Serial.println(data);                 //Print data on Serial Monitor
  }
}
```

هدول الـ sender راح يرسل للمستقبل Hello World
المستقبل راح يقرأها ويعرضها بالـ Serial monitor.

تشبيكة الـ UART : T_x للإرسل مع R_x للمستقبل + العكس
+ GND لـ 2 اردوينو مع بعض



Button الـ جوه
(if pressed or not).
1 Serial بكتب بالـ pressed اذا

Sender

```
int ButtonPin = 7;
void setup() {
  //Setup the Serial at 9600 Baud
  Serial.begin(9600);
```

```
  pinMode(ButtonPin, INPUT);
}

void loop() {
  // Check if the push button is pushed & Send 1 if pushed
  if (digitalRead(ButtonPin) == HIGH) {
          Serial.write(1);
  }
}
```

byteRead (buttonPin = High)

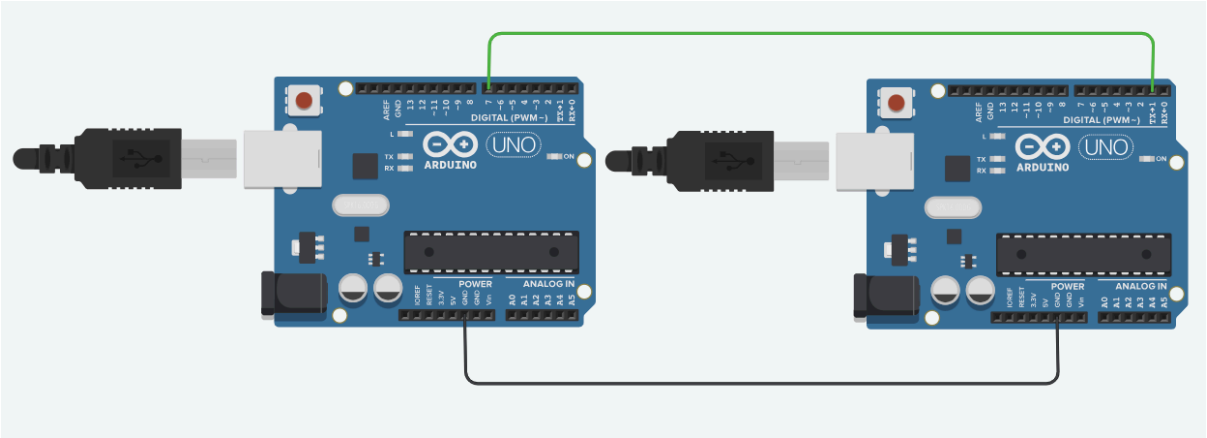
→ Turn the LED on

o.w

→ Turn the LED off.

Receiver

```
byte byteRead;
int LEDpin = 7;

void setup() {
  // Setup the Serial at 9600 Baud
  pinMode(LEDpin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  if (Serial.available()) {
          // Check if 1 is sent, and turn on the LED if yes.
          byteRead = Serial.read();
          if (byteRead == 1) {
          digitalWrite(LEDpin, HIGH);
          }
  }

  digitalWrite(LEDpin, LOW);
}
```



Reciver

```
int readPin = 7;

void setup() {
  // Begin the Serial at 19200 Baud
  Serial.begin(19200);
  pinMode(readPin, INPUT);  //0100 0001
}

void loop() {
  Serial.println(digitalRead(readPin));
}
```

Sender

```
void setup() {
  // Setup the Serial at 300 Baud
  Serial.begin(300);
}

void loop() {
  //Write the character A => will be transmitted as Byte [41H]
  Serial.write('A');
  delay(500);
}
```

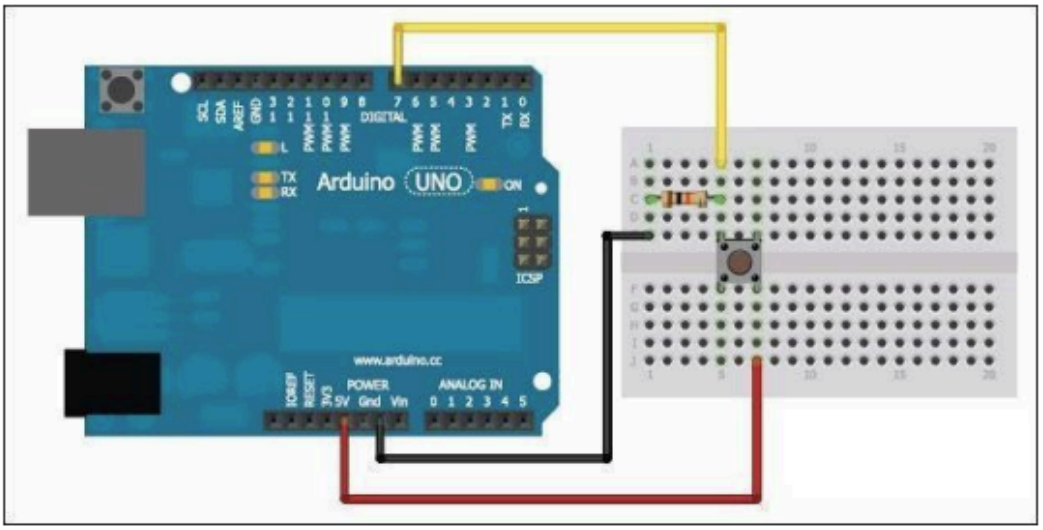


Figure 3: Arduino1 connected with Push button

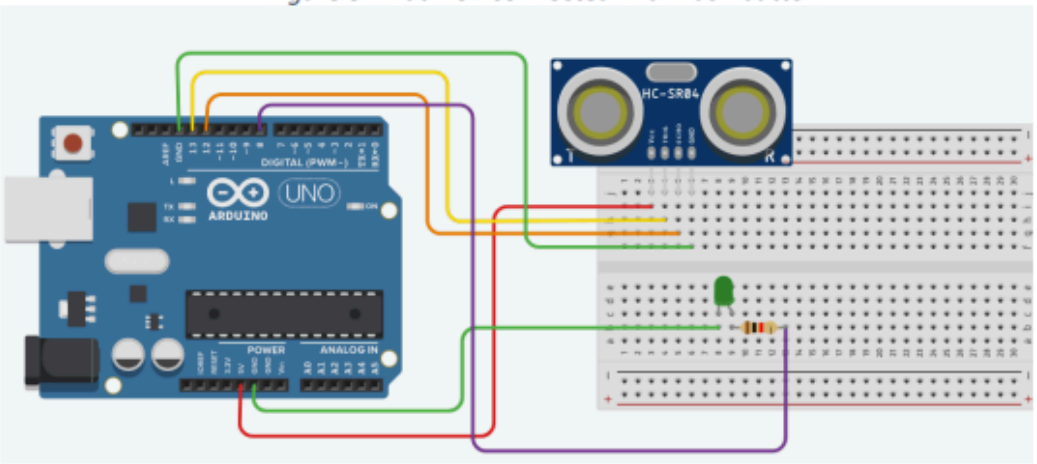


Figure 4: Arduino2 connected with Led

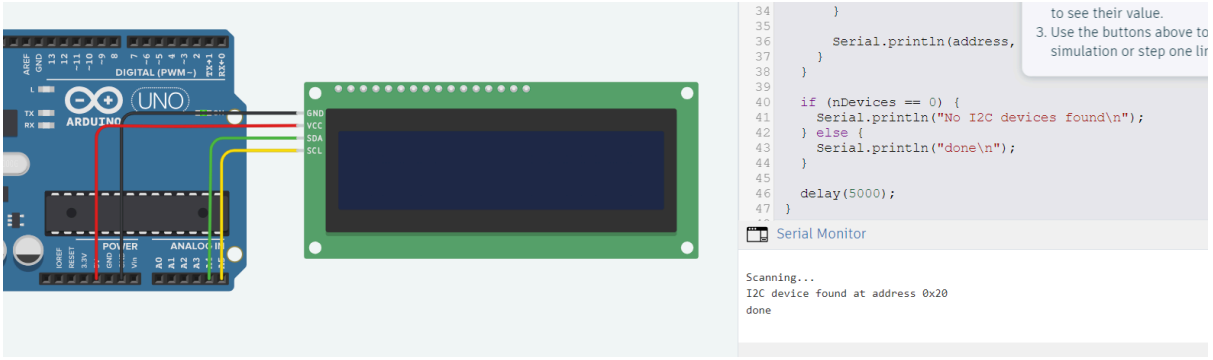
Sender Code:

```
#define BUTTON_PIN 7
void setup() {
// TODO[1]: Setup the Serial at 9600 Baud
Serial.begin(9600);
Serial.setTimeout(10000);
pinMode(BUTTON_PIN, INPUT_PULLUP);
}
void loop() {
// TODO[2]: Check if the push button is pushed & Send 1 if pushed
if(digitalRead(BUTTON_PIN) == HIGH){
Serial.print(1);
delay(1000);
}
}
```

Receiver Code:

```
#define trigPin 13
#define echoPin 12
#define led 8
void setup() {
// TODO[1]: Setup the Serial at 9600 Baud
Serial.begin(9600);
// Ultrasonic and led setup
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(led, OUTPUT);
}
void loop() {
/* Check if data has been sent from the computer: */
if (Serial.available()) {
// TODO[2]: Check if 1 is sent, then run ultrasonic sensor.
// read the incoming byte:
int x = Serial.parseInt();
Serial.println(x);
if (Serial.parseInt() == 1) {
// Ultra Sonic Code
long duration, distance;
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = (duration/2) / 29.1;
Serial.print(distance);
Serial.println(" cm");
digitalWrite(led,HIGH);
}
}
}
```

Two wires interface I2C

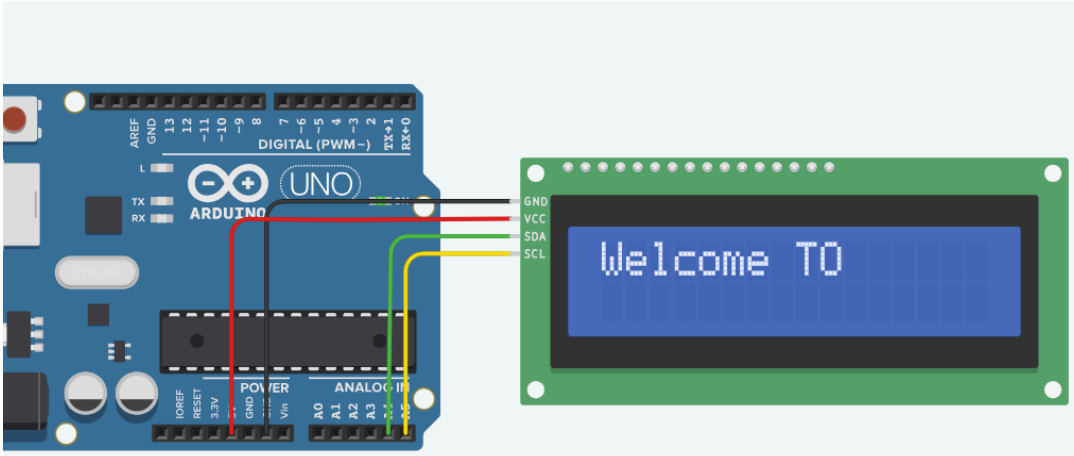


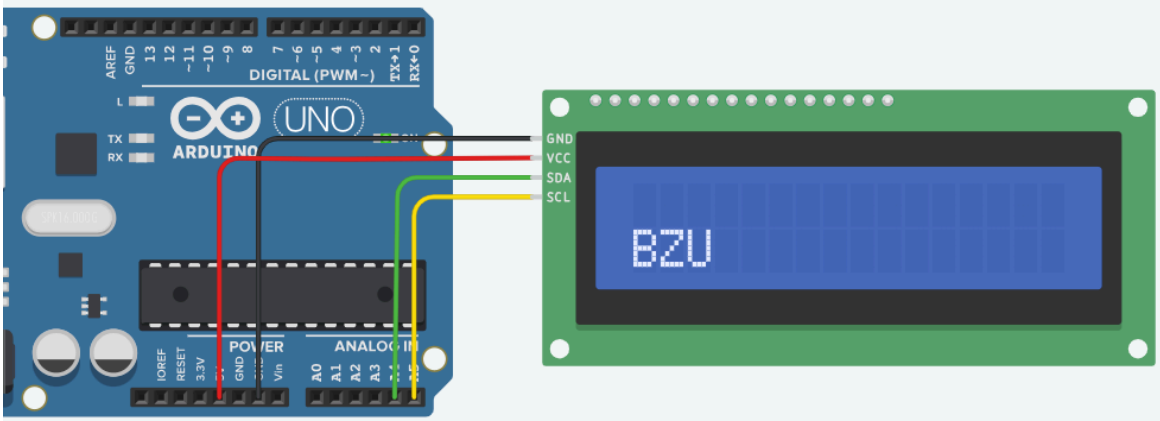
هدوم عشان نعرف ال LCD في أي address مشبوكة

Before displaying text on the LCD, you need to find the LCD I2C address. With the LCD properly wired to the Arduino, upload the following I2C Scanner sketch.

```
#include <Wire.h>

void setup() {
  Wire.begin();
  Serial.begin(9600);
  Serial.println("\nI2C Scanner");
}

void loop() {
  byte error, address;
  int nDevices;

  Serial.println("Scanning...");
  nDevices = 0;

  for (address = 1; address < 127; address++) {
          Wire.beginTransmission(address);
          error = Wire.endTransmission();

          if (error == 0) {
          Serial.print("I2C device found at address 0x");

          if (address < 16) {
          Serial.print("0");
          }

          Serial.println(address, HEX);
          nDevices++;
```

```
            } else if (error == 4) {
            Serial.print("Unknown error at address 0x");

            if (address < 16) {
            Serial.print("0");
            }

            Serial.println(address, HEX);
            }
  }

  if (nDevices == 0) {
            Serial.println("No I2C devices found\n");
  } else {
            Serial.println("done\n");
  }

  delay(5000);
}
```





Part 2.2: Display static text on the LCD Displaying static text on the LCD is very simple. All you have to do is select where you want the characters to be displayed on the screen, and then send the message to the display. Here's a very simple sketch example that displays "Welcome TO BZU".

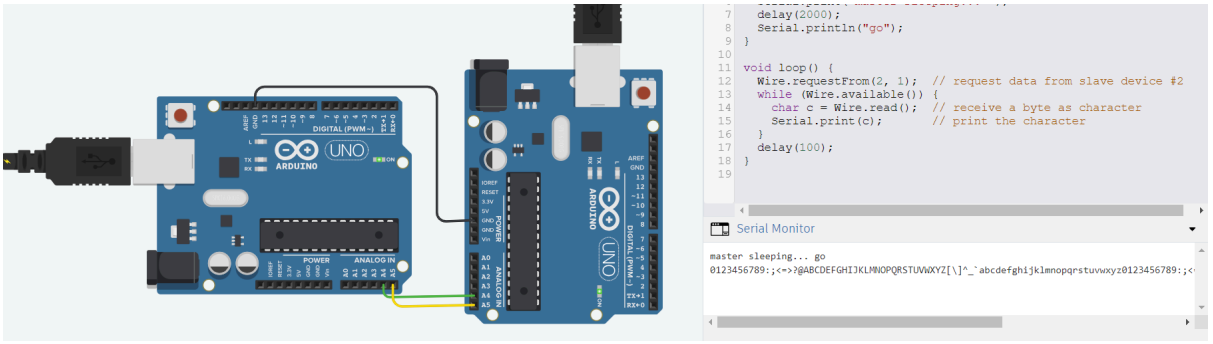
```
#include <LiquidCrystal_I2C.h>

// Set the LCD number of columns and rows
int lcdColumns = 16;
int lcdRows = 2;

// Set LCD address, number of columns and rows
// If you don't know your display address, run an I2C scanner sketch
LiquidCrystal_I2C lcd(0x20, lcdColumns, lcdRows);

void setup() {
  // Initialize LCD
  lcd.init();

  // Turn on LCD backlight
  lcd.backlight();
}

void loop() {
  // Set cursor to the first column, first row
  lcd.setCursor(0, 0);

  // Print message
  lcd.print("Welcome TO");
  delay(1000);

  // Clears the display to print a new message
  lcd.clear();

  // Set cursor to the first column, second row
  lcd.setCursor(0, 1);

  // Print corrected message with regular double quotation mark
  lcd.print("BZU");
  delay(1000);

  // Clears the display for the next iteration
  lcd.clear();
}
```

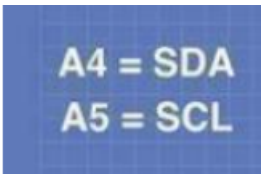
- **lcd.init(): initializesthe display**

- **lcd.backlight():turnsthe LCD backlight on**
- **lcd.setCursor(intcolumn, int row): sets the cursor to the specified column and row**

lcd.print(message):displaysthe message or variable on the display

- **lcd.clear(): clearsthe display**

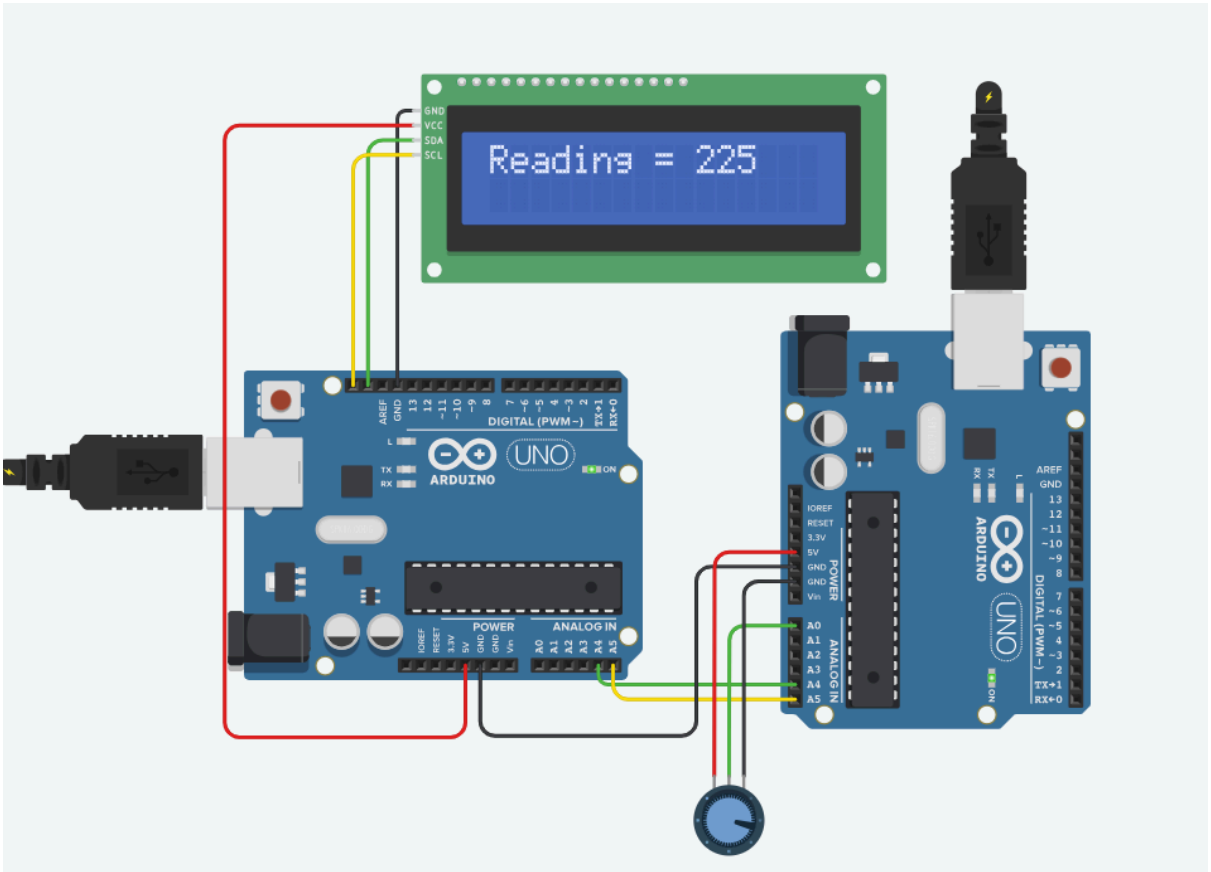




Our example will send data from slave to master. The data is characters starting from '0' and ending with 'z'.

```
// master side
#include <Wire.h>
void setup() {
  Wire.begin();              // join i2c bus (address optional for master)
  Serial.begin(9600);  // start serial for output
  Serial.print("master sleeping... ");
  delay(2000);
  Serial.println("go");
}

void loop() {
  Wire.requestFrom(2, 1);  // request data from slave device #2
  while (Wire.available()) {
          char c = Wire.read();  // receive a byte as character
          Serial.print(c);   // print the character
  }
  delay(100);
}
```

```
// slave side code
#include <Wire.h>
void setup() {
  Wire.begin(2);              // join i2c bus with address #2
  Wire.onRequest(requestEvent);  // register event
  Serial.begin(9600);                  // start serial for output
}

void loop() {
  delay(100);
}

// function that executes whenever data is received from master
// this function is registered as an event, see setup()
void requestEvent() {
  static char c = '0';
  Wire.write(c++);
  if (c > 'z')
            c = '0';
}
```

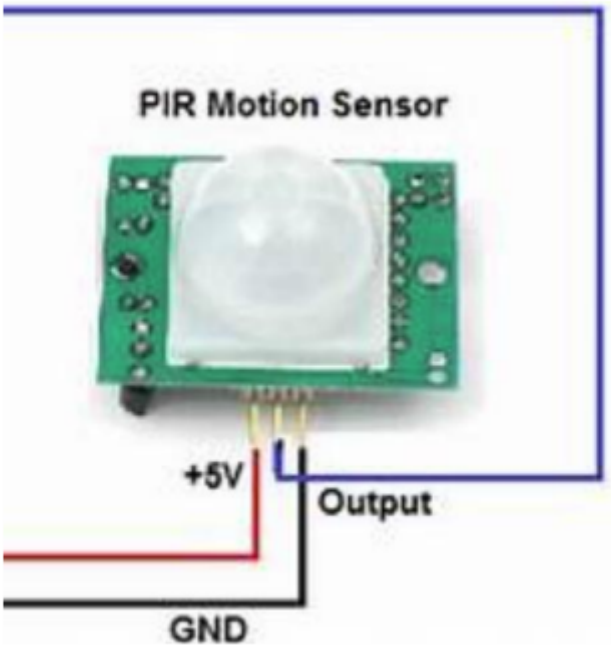


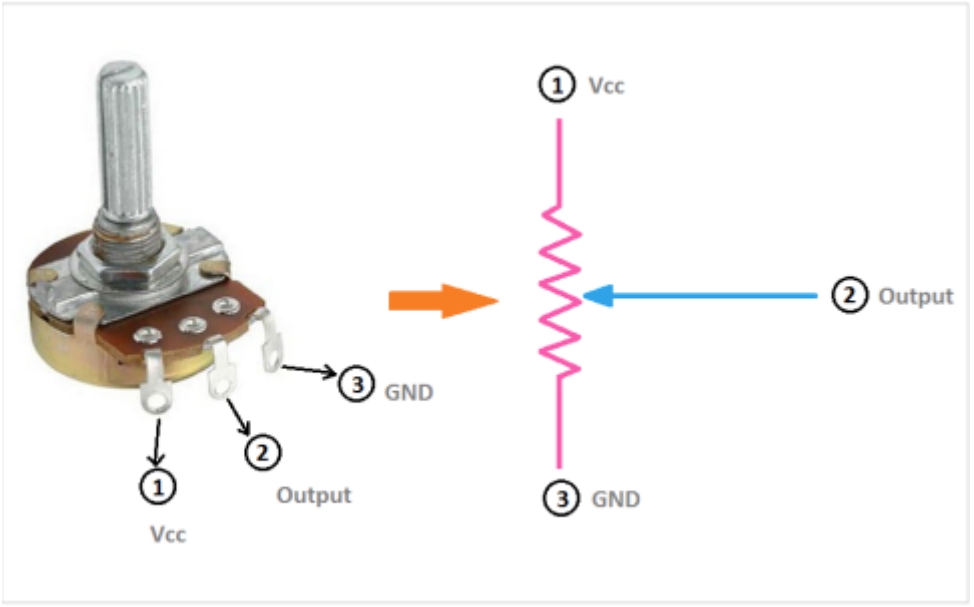
In this part we want to read analog value from slave Arduino and write the value on LCD connected to Arduino master device:

```
// master side
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// set the LCD number of columns and rows
int lcdColumns = 16;
int lcdRows = 2;

// set LCD address, number of columns and rows
// if you don't know your display address, run an I2C scanner sketch
LiquidCrystal_I2C lcd(0x20, lcdColumns, lcdRows);

void setup() {
  Wire.begin();               // join i2c bus (address optional for master)
  Serial.begin(9600);  // start serial for output
  Serial.print("master sleeping... ");
  delay(2000);
  Serial.println("go");
  // initialize LCD
  lcd.init();
  // turn on LCD backlight
  lcd.backlight();
}

void loop() {
  Wire.requestFrom(50, 2);  // request data from slave device #50
  int res;            //result from I2c reading
  byte MSB = Wire.read();   /* receive a byte MSB(NOTE: the
  function is blocking, so it would not continue the code until a
  byte reach)*/
  byte LSB = Wire.read();   /* receive a byte LSB(NOTE: the
  function is blocking, so it would not continue the code until a
  byte reach)*/
  res = ((MSB << 8) | LSB);
  Serial.print("MSB = ");
  Serial.print(MSB);
  Serial.print(" , LSB = ");
  Serial.print(LSB);
  Serial.print(" , analog value = ");
  Serial.println(res);  // print the analog value
  // set cursor to first column, first row
```

```
  lcd.setCursor(0, 0);
  // print message
  lcd.print("Reading = ");
  lcd.print(res);
  delay(1000);
  lcd.clear();
}
```

```
// slave side code
#include <Wire.h>
void setup() {
  Wire.begin(50);           // join i2c bus with address #50
  Wire.onRequest(requestEvent);  // register event
  Serial.begin(9600);               // start serial for output
}

void loop() {
  delay(100);
}

// function that executes whenever data is received from master
// this function is registered as an event, see setup()
void requestEvent() {
  int value = analogRead(A0);
  Wire.write(value >> 8);           // send the MSB
  Wire.write(value & 0x00ff);  // send the LSB
}
```


PIR Motion Sensor
+5V
Output
GND


1 Vcc
2 Output
3 GND
3 GND
2
1
Output
Vcc