Experiment-4

Arduino-1

What is Arduino?

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board.

How to connect Arduino with laptop

- First download the software arduino-1.6.12-windows.exe
- Connect Arduino with laptop by USB cable
- Open Arduino software
- Select Board: Click Tool and select board as Arduino as shown in figure 2.
- Select port: Click Tool and select port as shown in figure 3.

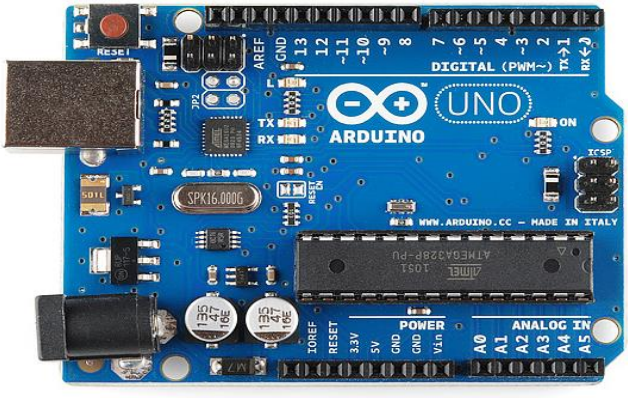


Figure 1: Arduino

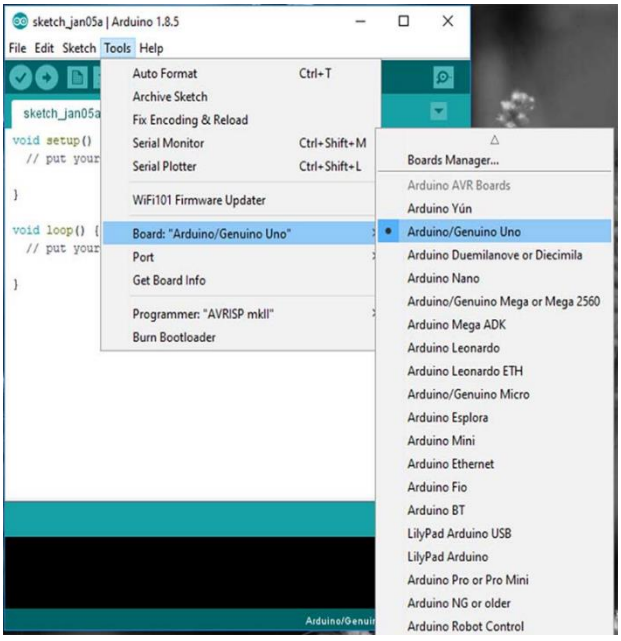


Figure 2

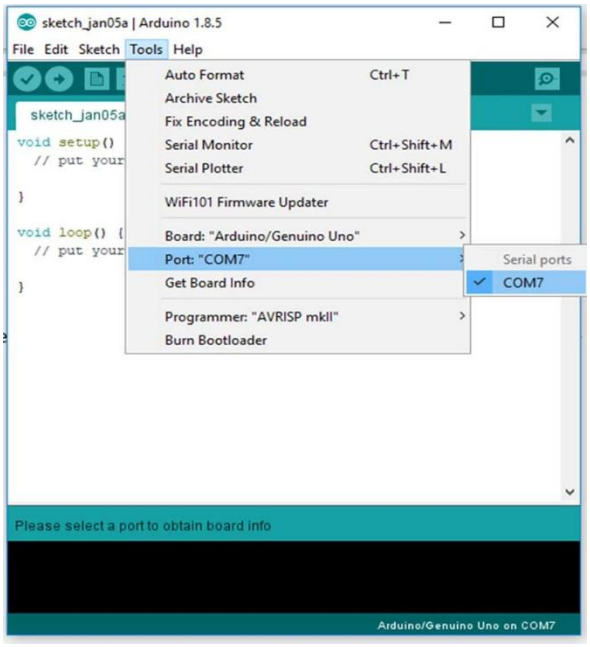


Figure 3

Task 1: Blink internal LED

Hardware required: Arduino Board, USB cable

First connect Arduino with laptop

Code:

```
int LEDPin=LED_BUILTIN; int waitTimeOn=200; // time in mili sec
 int waitTimeOff=200;
void setup()
 {                                          // put your setup code here, to run once:
pinMode(LEDPin,OUTPUT);
}
void loop()
 {                                                  // put your main code here, to run repeatedly:
digitalWrite(LEDPin,HIGH);                 // This supplies 5 volts to the LED anode.
delay(waitTimeOn);                         // Time delay between on and off
digitalWrite(LEDPin,LOW);                  // That takes the LED_BUILTIN pin back to 0 volts
delay(waitTimeOff);                        // Time delay between on and off
}
```

Task 2: Blink internal LED with 10 ms delay

Code:

Change the delay time in above code

Task 3: Blink external LED with 200 ms delay

Hardware required:

1. LED (Light Emitting Diode) 2. Power source (Arduino) 3. Resistor 4. Jumper wires 5. Breadboard

Hardware connection:

1. Connect the positive side of your LED (longer leg) to Arduino digital Pin 13 (or another digital pin, don't forget to change the code to
Match).
2. Connect the negative side of your LED (shorter leg) to a 20K Ohm resistor. Connect the other side of the resistor to ground.
3. Always use resistors between the Arduino and LEDs to keep the LEDs from burning out due to too much current.
4. Upload your code to Arduino and the led starts blinking.

Code

```
int LEDPin=13; int waitTimeOn=200;// time in mili sec
int waitTimeOff=200;// time in mili sec
void setup()
{
```

```
pinMode(LEDPin,OUTPUT);
}
void loop()
{
digitalWrite(LEDPin,HIGH);
delay(waitTimeOn);
digitalWrite(LEDPin,LOW);
delay(waitTimeOff);
}
```

Task 4: Blink external LED with 200 ms delay in on to off and 20 ms delay in off to on

Code:

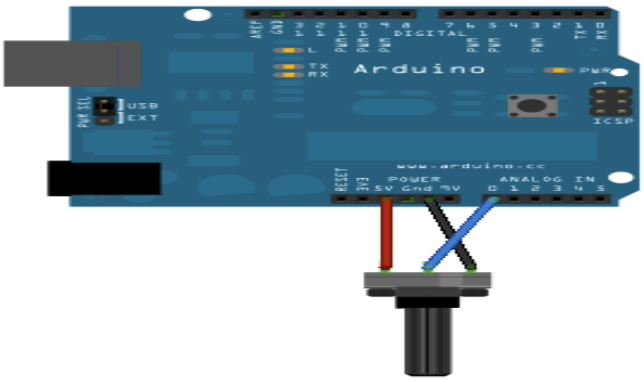
Change the delay time in above code

Task 5: Change the intensity of LED

Code:

```
int led = 9;                          // the PWM pin the LED is attached to
int brightness = 0;                   // how bright the LED is
int fadeAmount = 5;                   // how many points to fade the LED by
void setup()
 {
  pinMode(led, OUTPUT);     // declare pin 9 to be an output:
}
void loop()
 {
  analogWrite(led, brightness);           // set the brightness of pin 9
  brightness = brightness + fadeAmount;       // change the brightness for next time through the loop:
  // reverse the direction of the fading at the ends of the fade:
  if (brightness <= 0 || brightness >= 255)
 {
    fadeAmount = -fadeAmount;
  }
  delay(30);      // wait for 30 milliseconds to see the dimming effect
}
```

Task 6: Connect the 10K pot and vary the resistance.
Hardware required- Arduino Board**,** 10k ohm potentiometer
Circuit:



Connect the three wires from the potentiometer to your board. Center port of potentiometer connect to analog input. Left side port of potentiometer goes to ground and right side one goes to 5 V supply. By turning the shaft of the potentiometer, you change the amount of resistance on either side of the wiper which is connected to the center pin of the potentiometer. This changes the voltage at the center pin. When the resistance between the center and the side connected to 5 volts is close to zero (and the resistance on the other side is close to 10 kilohms), the voltage at the center pin nears 5 volts. When the resistances are reversed, the voltage at the center pin nears 0 volts, or ground. This voltage is the analog voltage that you're reading as an input.

The microcontroller of the board has a circuit inside called an analog-to-digital converter *or* ADC that reads this changing voltage and converts it to a number between 0 and 1023. When the shaft is turned all the way in one direction, there are 0 volts going to the pin, and the input value is 0. When the shaft is turned all the way in the opposite direction, there are 5 volts going to the pin and the input value is 1023. In between, analog Read() returns a number between 0 and 1023 that is proportional to the amount of voltage being applied to the pin.

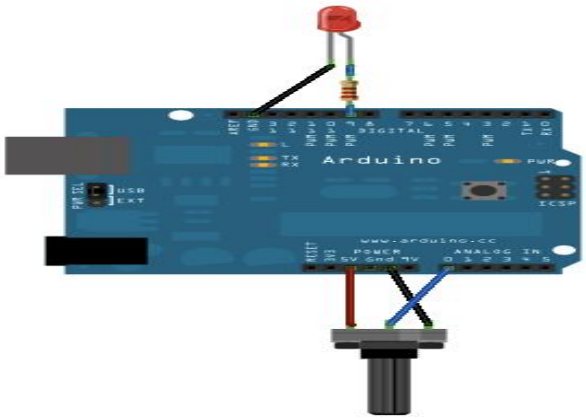
Code:

```
void setup()
{
  // put your setup code here, to run once:
// initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

void loop()
 {
  // put your main code here, to run repeatedly:
// read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
  float voltage = sensorValue * (5.0 / 1023.0);
  // print out the value you read:
```

```
  Serial.println(sensorValue);
  delay(1);
}
```

Task 7: With the help of pot regulate the brighten of external connected LED.
Hardware required: Arduino Board **,** 10k ohm potentiometer,LED,220 ohm resistor.
Circuit:



In the sketch above, after declaring two pin assignments (analog 0 for our potentiometer and digital 9 for your LED. Arduino has an analog Read range from 0 to 1023, and an analog Write range only from 0 to 255, therefore the data from the potentiometer needs to be converted to fit into the smaller range before using it to dim the LED.

Code:

```
const int analogInPin = A0;  // Analog input pin that the potentiometer is attached to
const int analogOutPin = 9; // Analog output pin that the LED is attached to
int sensorValue = 0;        // value read from the pot
int outputValue = 0;        // value output to the PWM (analog out)
void setup()
 {
// initialize serial communications at 9600 bps:
Serial.begin(9600);
}
void loop()
 {
 // read the analog in value:
sensorValue = analogRead(analogInPin);
// map it to the range of the analog out:
outputValue = map(sensorValue, 0, 1023, 0, 255);
 // change the analog out value:
analogWrite(analogOutPin, outputValue);
  // print the results to the Serial Monitor:
float voltage = sensorValue * (5.0 / 1023.0);
Serial.print("sensor = ");
```

```
Serial.print(sensorValue);
Serial.print("\t output = ");
Serial.println(outputValue);
  // wait 2 milliseconds before the next loop for the analog-to-digital
delay(2);
}
```