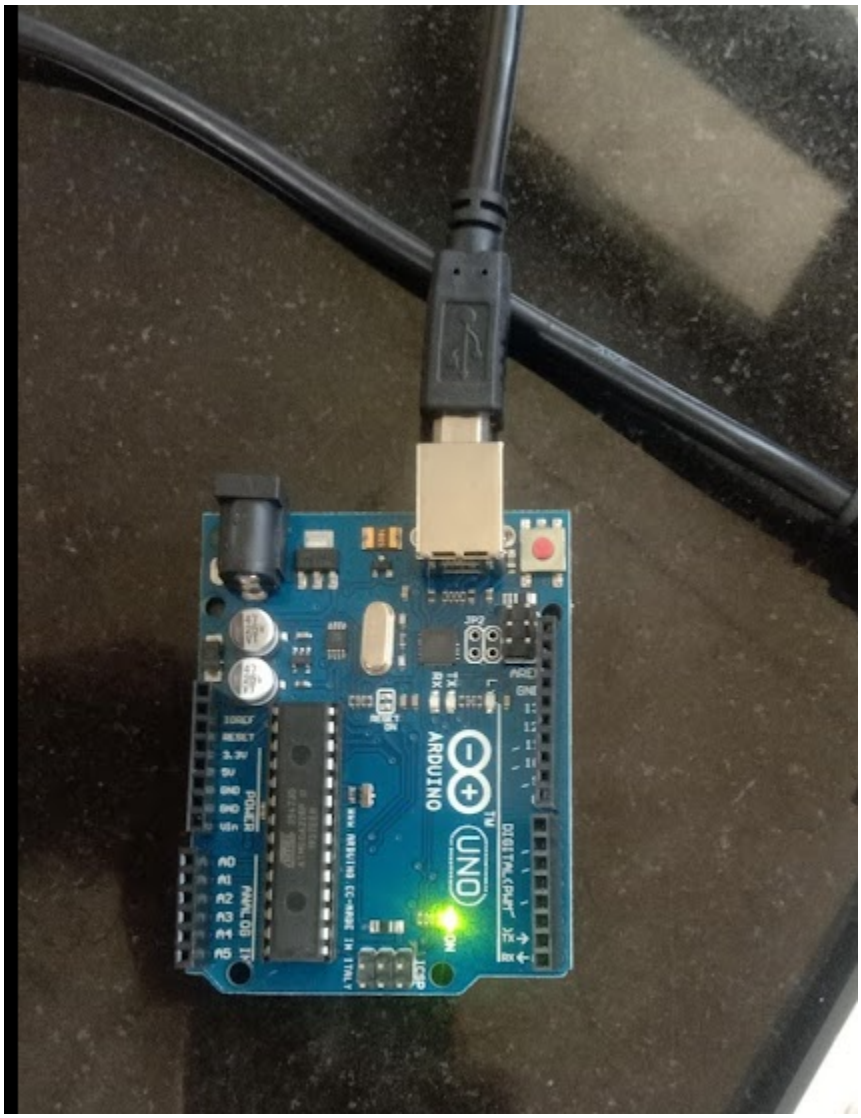
Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244
Date : 05-04-2022

TASK 1:

For 1s
The diagram and the snippet of the code have been pasted below :



The code is as follows :

Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244

Date : 05-04-2022

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

We can change the delay() factor depending on the given glow times given to us for the LEDs like for example, delay(5000) would make it glow for 5s etc.

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

Done uploading.

Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244
Date : 05-04-2022

For 0.1s

Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244

Date : 05-04-2022

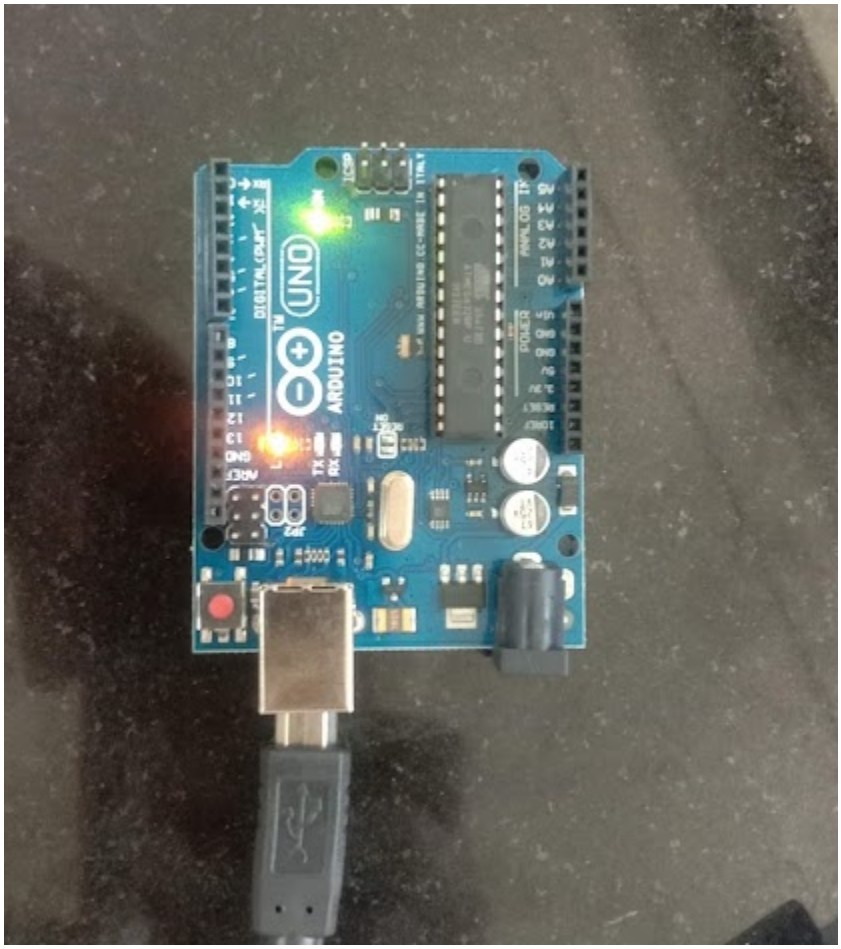
```
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(100);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(100);                       // wait for a second
}
```

Done uploading.

Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244

Date : 05-04-2022



For 0.5s

Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244

Date : 05-04-2022

```
 */

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(500);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(500);                       // wait for a second
}
```

```
Done uploading.

Sketch uses 924 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.
```

5s

Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244

Date : 05-04-2022

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(5000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(5000);                       // wait for a second
}
```

```
Done uploading.

Sketch uses 924 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.
```

For 10s

Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244

Date : 05-04-2022

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(10000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(10000);                       // wait for a second
}
```

```
Done uploading.

Sketch uses 924 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.
```

Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244
Date : 05-04-2022

Task 2: To blink an external LED

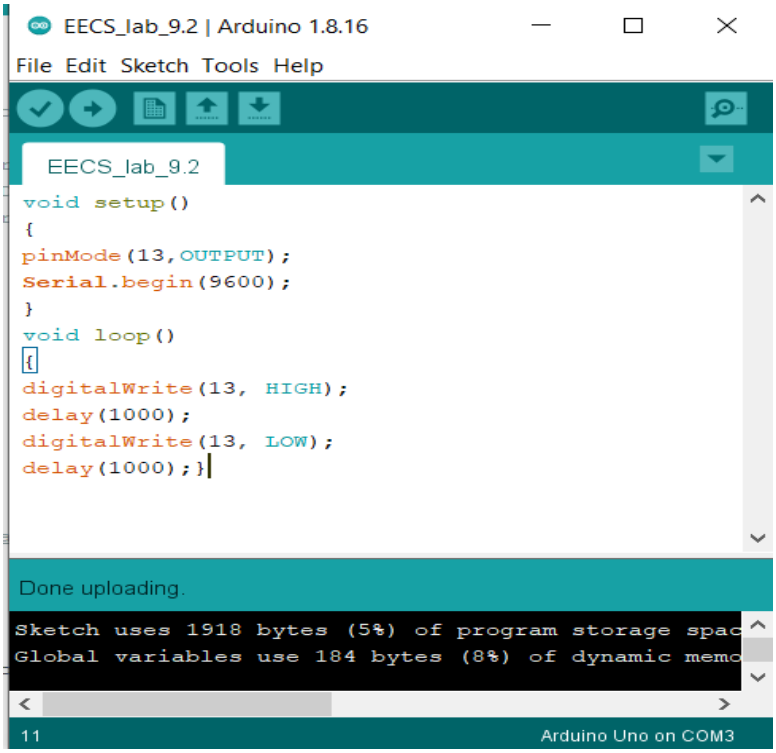
Setup:

Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244
Date : 05-04-2022

Code for the above:

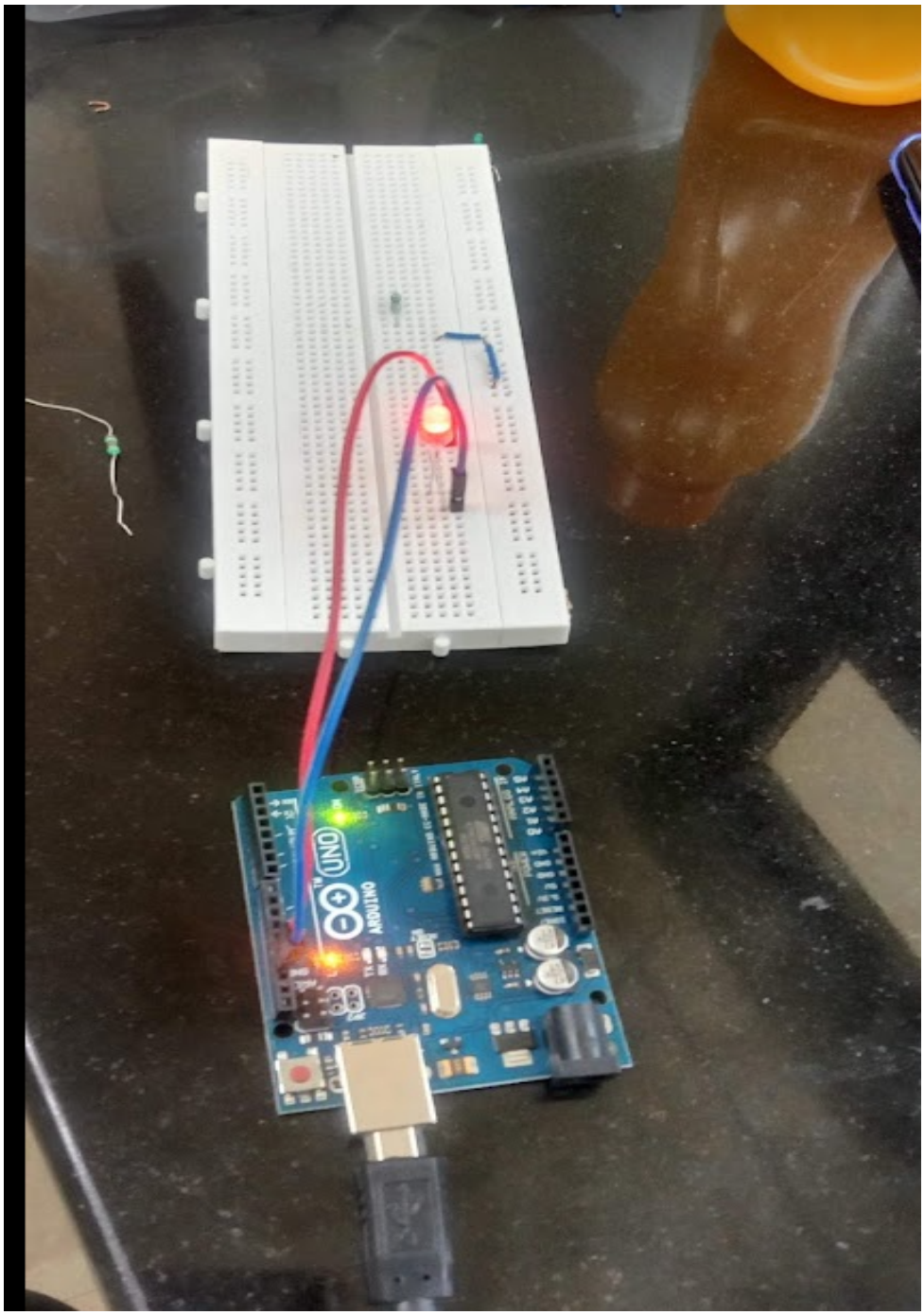
```
void setup()
{
pinMode(13,OUTPUT);
Serial.begin(9600);
}
void loop()
{
digitalWrite(13, HIGH);
delay(1000);
digitalWrite(13, LOW);
delay(1000);}
```



Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244
Date : 05-04-2022

TASK 3 : BLINK LED WITHOUT TIME DELAY



Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244
Date : 05-04-2022

Blink Without Delay code :

```
const int ledPin =  LED_BUILTIN;// the number of the LED pin

// Variables will change:
int ledState = LOW;             // ledState used to set the LED

// Generally, you should use "unsigned long" for variables that hold time
// The value will quickly become too large for an int to store
unsigned long previousMillis = 0;        // will store last time LED was updated

// constants won't change:
const long interval = 1000;           // interval at which to blink (milliseconds)

void setup() {
  // set the digital pin as output:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // here is where you'd put code that needs to be running all the time.

  // check to see if it's time to blink the LED; that is, if the difference
  // between the current time and last time you blinked the LED is bigger than
  // the interval at which you want to blink the LED.
  unsigned long currentMillis = millis();

  if (currentMillis - previousMillis >= interval) {
    // save the last time you blinked the LED
    previousMillis = currentMillis;

    // if the LED is off turn it on and vice-versa:
    if (ledState == LOW) {
      ledState = HIGH;
    } else {
      ledState = LOW;
    }

    // set the LED with the ledState of the variable:
    digitalWrite(ledPin, ledState);
```

Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244

Date : 05-04-2022

```
 }
}
```

BlinkWithoutDelay §

```
const int ledPin =  LED_BUILTIN;// the number of the LED pin
int ledState = LOW;

unsigned long previousMillis = 0;

const long interval = 1000;           // interval at which to blink (milliseconds)

void setup() {
  // set the digital pin as output:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // here is where you'd put code that needs to be running all the time.

  // check to see if it's time to blink the LED; that is, if the difference
  // between the current time and last time you blinked the LED is bigger than
  // the interval at which you want to blink the LED.
  unsigned long currentMillis = millis();

  if (currentMillis - previousMillis >= interval) {
    // save the last time you blinked the LED
    previousMillis = currentMillis;

    // if the LED is off turn it on and vice-versa:
    if (ledState == LOW) {
      ledState = HIGH;
    } else {
      ledState = LOW;
    }

    // set the LED with the ledState of the variable:
    digitalWrite(ledPin, ledState);
  }
}
```

Done uploading.

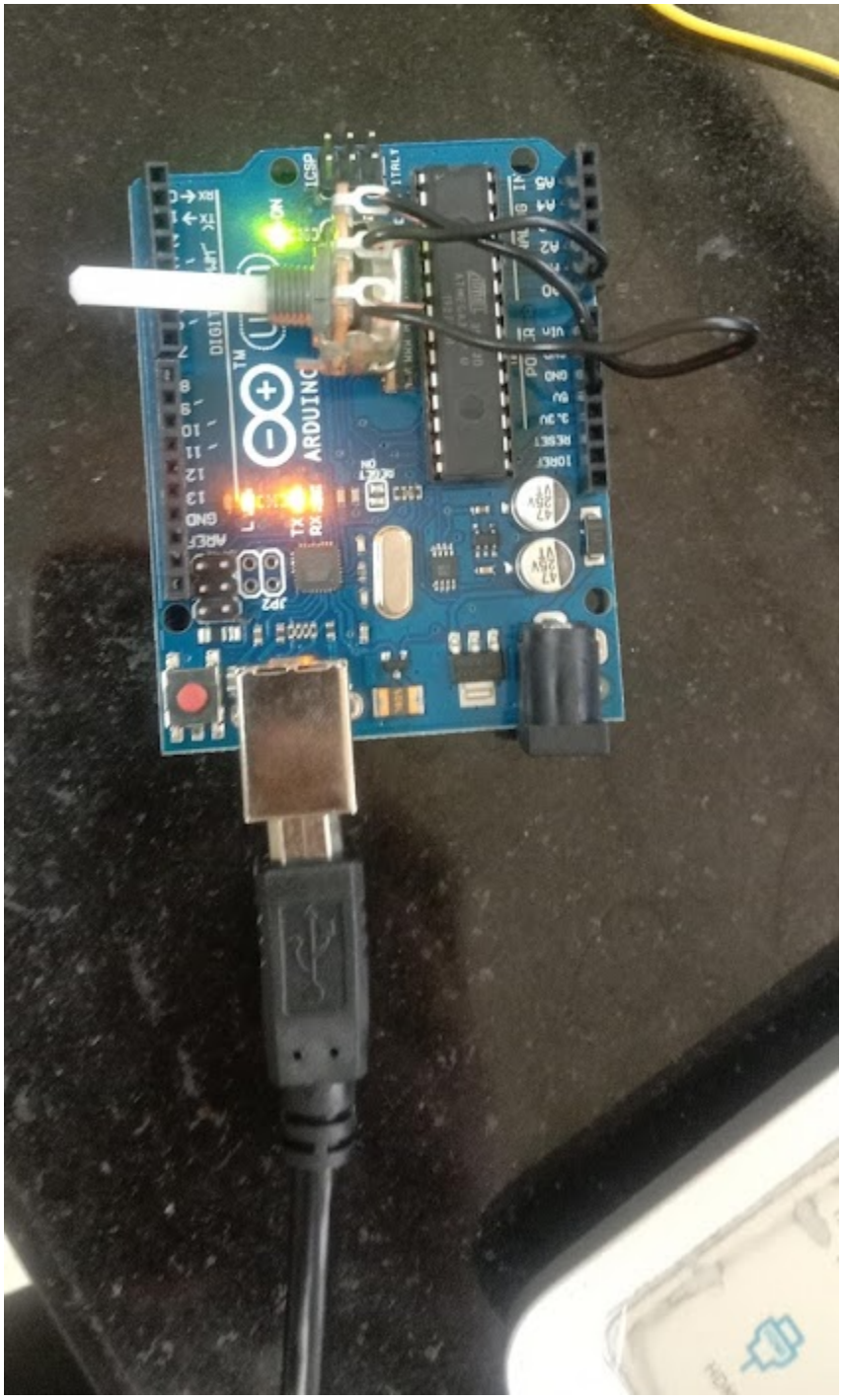
```
Sketch uses 860 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 15 bytes (0%) of dynamic memory, leaving 2033 bytes for local variables. Maximum is 2048 bytes.
```

Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244
Date : 05-04-2022

TASK 4: Connect the 10 K potentiometer and vary the resistance

Potentiometer set-up:



Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244

Date : 05-04-2022

Code:

```
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // 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(voltage);
}
```

Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244

Date : 05-04-2022

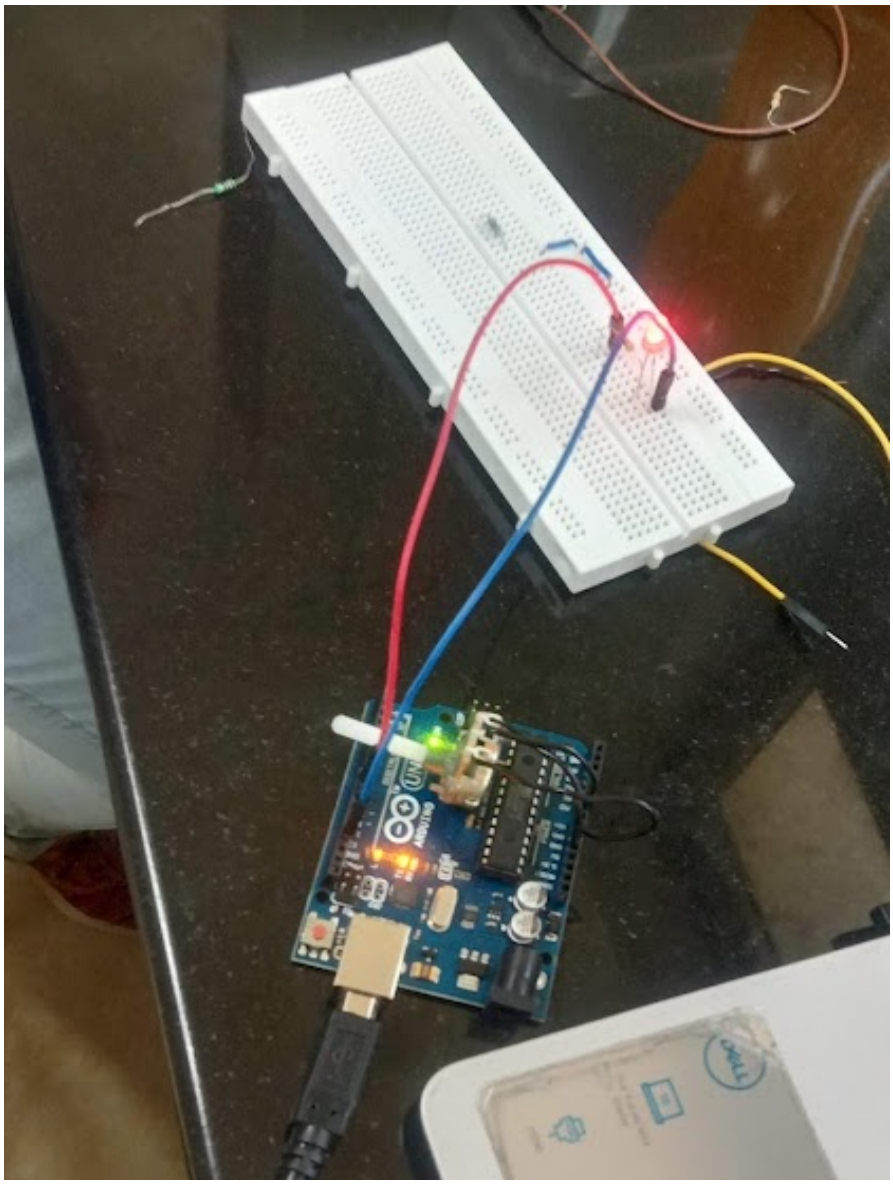
```
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // 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(voltage);
}
```

```
Done uploading.

Sketch uses 2992 bytes (9%) of program storage space. Maximum is 32256 bytes.
Global variables use 200 bytes (9%) of dynamic memory, leaving 1848 bytes for local variables. Maximum is 2048 bytes.
```

Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244
Date : 05-04-2022

TASK-5: LIGHT THE LED WITH THE POTENTIOMETER



Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244
Date : 05-04-2022

Code for the above set-up:

```
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:
  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
  // converter to settle after the last reading:
  delay(2);
}
```

Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244

Date : 05-04-2022

Code execution snippet :

```
// These constants won't change. They're used to give names to the pins used:
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:
  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
  // converter to settle after the last reading:
  delay(2);
}
```

```
Done uploading.

Sketch uses 2612 bytes (8%) of program storage space. Maximum is 32256 bytes.
Global variables use 214 bytes (10%) of dynamic memory, leaving 1834 bytes for local variables. Maximum is 2048 bytes.
```

Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244
Date : 05-04-2022

ADDITIONAL TASK :

Build a Traffic control system using 2 LEDs where one LED glows for 15 seconds, the other glows for 10s subsequently.

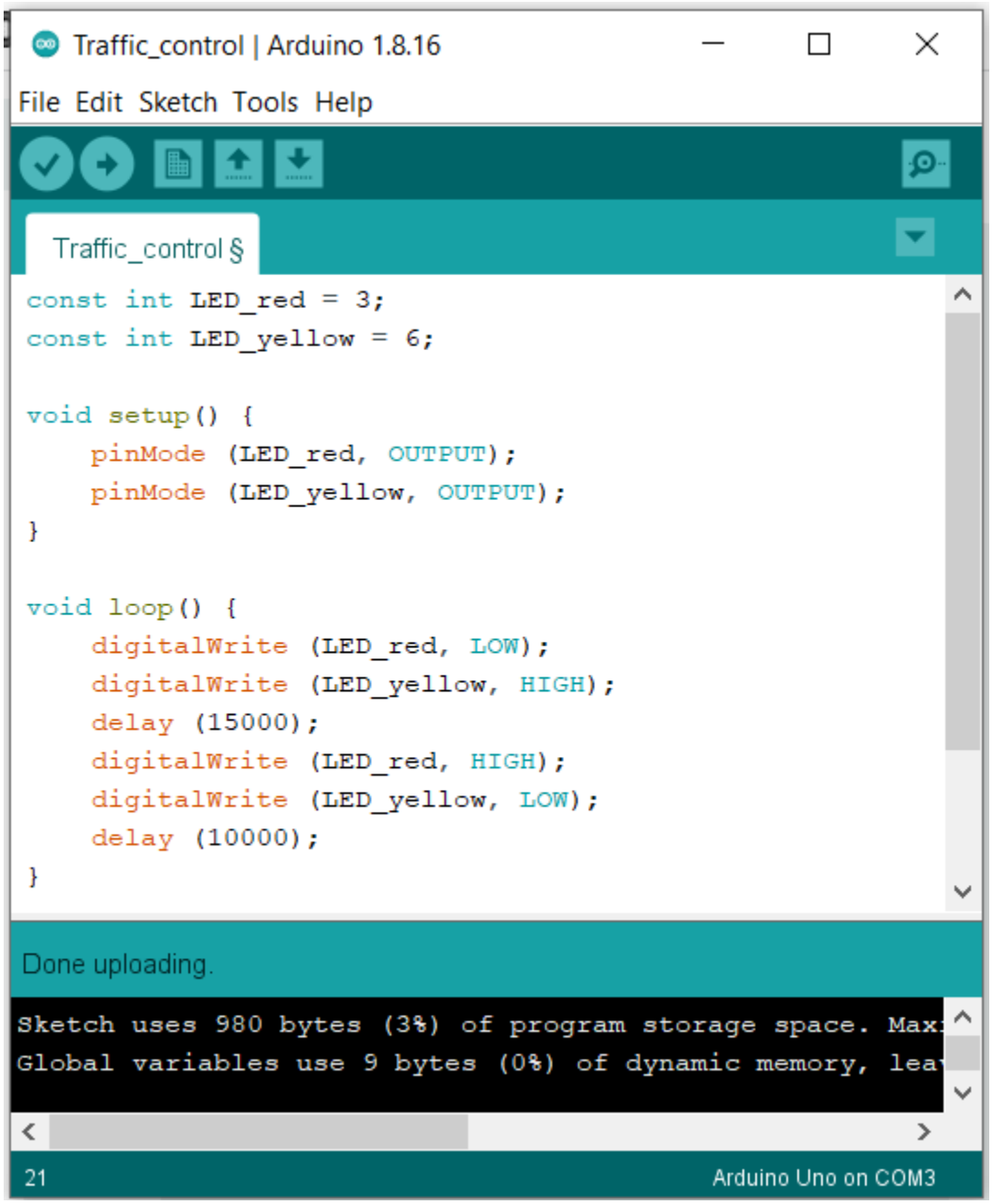
Code for the above:

```
const int LED_red = 3;
const int LED_yellow = 6;

void setup() {
   pinMode (LED_red, OUTPUT);
   pinMode (LED_yellow, OUTPUT);
}

void loop() {
   digitalWrite (LED_red, LOW);
   digitalWrite (LED_yellow, HIGH);
   delay (15000);
   digitalWrite (LED_red, HIGH);
   digitalWrite (LED_yellow, LOW);
   delay (10000);
}
```

Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244

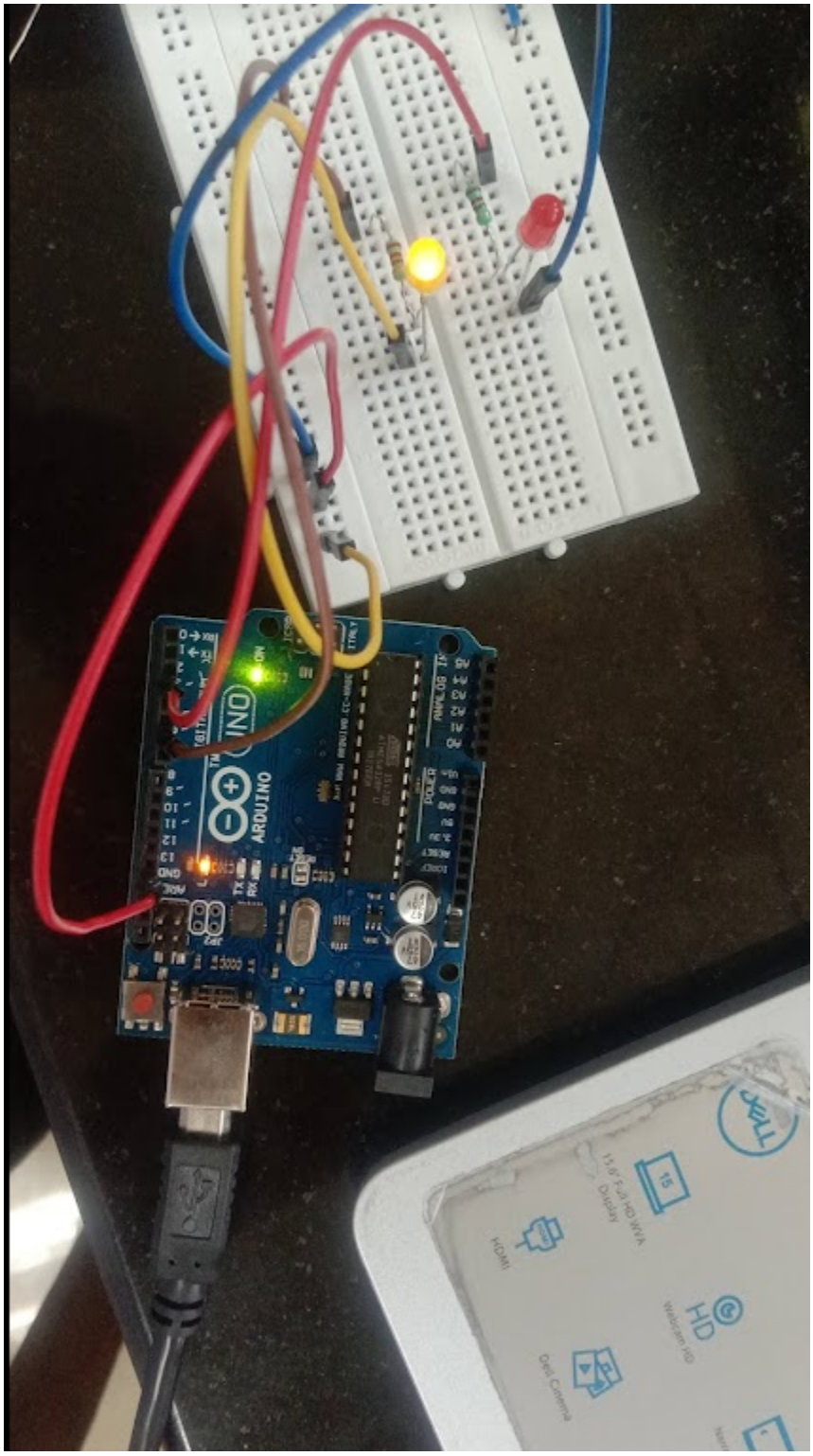
Date : 05-04-2022



Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244

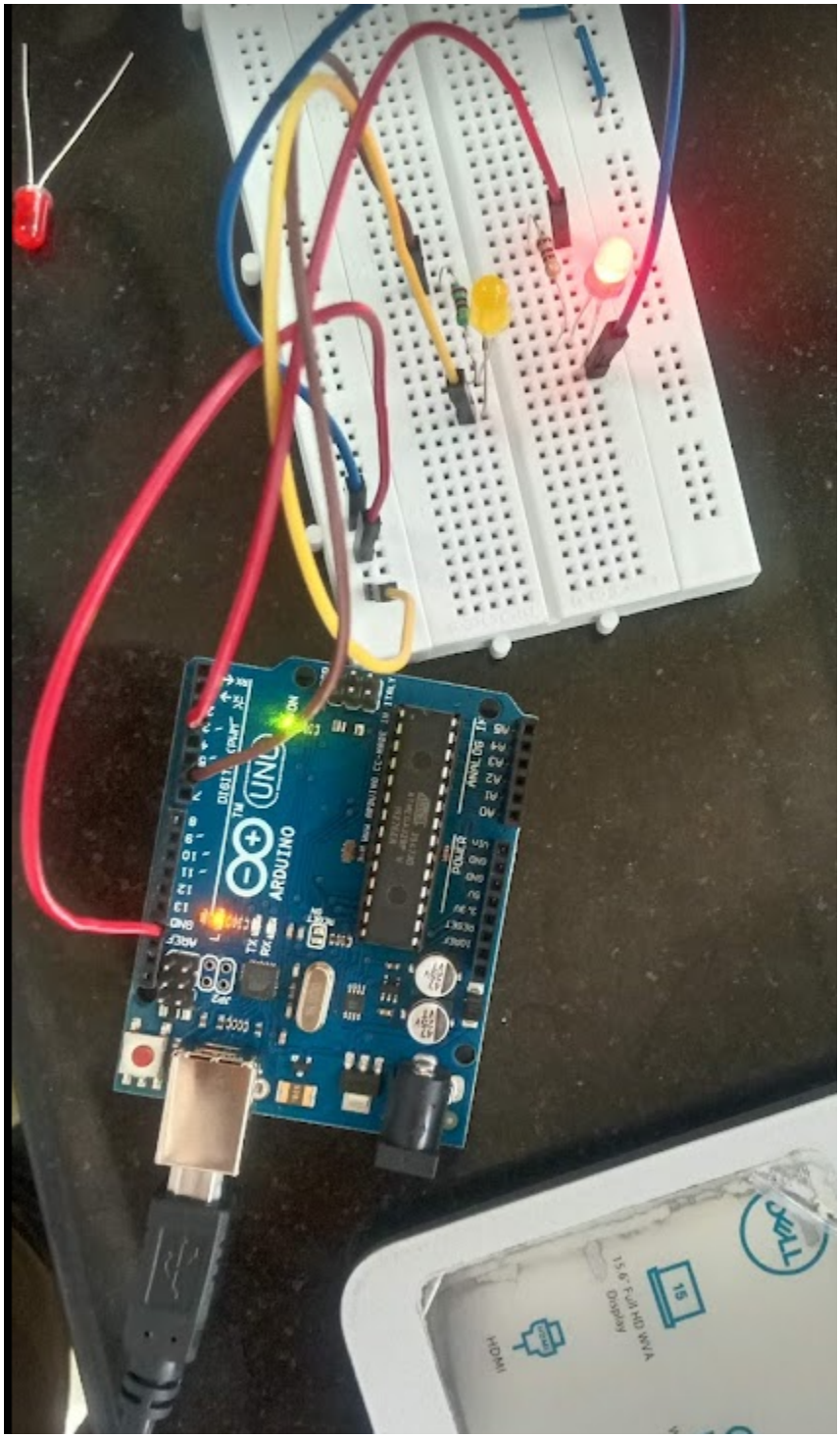
Date : 05-04-2022



Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244

Date : 05-04-2022



Experiment 9 : EECS Laboratory 1 - Working with Arduino

Rita Abani 19244
Date : 05-04-2022

Conclusion :

As seen from the above experiments, Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs like light on a sensor, a - and turn it into an output - turning on an LED etc. One can tell the board what to do by sending a set of instructions to the microcontroller on the board. To do so we use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.
The traffic control code made above using the to LEDs is an elementary example of some of the most useful applications that an Arduino board can be used to make with the right logic. Several smart sensing devices like Temperature or humidity controlled sensors can also be made using similar logic.