



MTU

Ollscoil Teicneolaíochta na Mumhan
Munster Technological University

Lab 5

COMP6043: Physical Computing

Student ID: R00201303

Date: 7.03.2022

Group: COMP1D-Y

Task1:

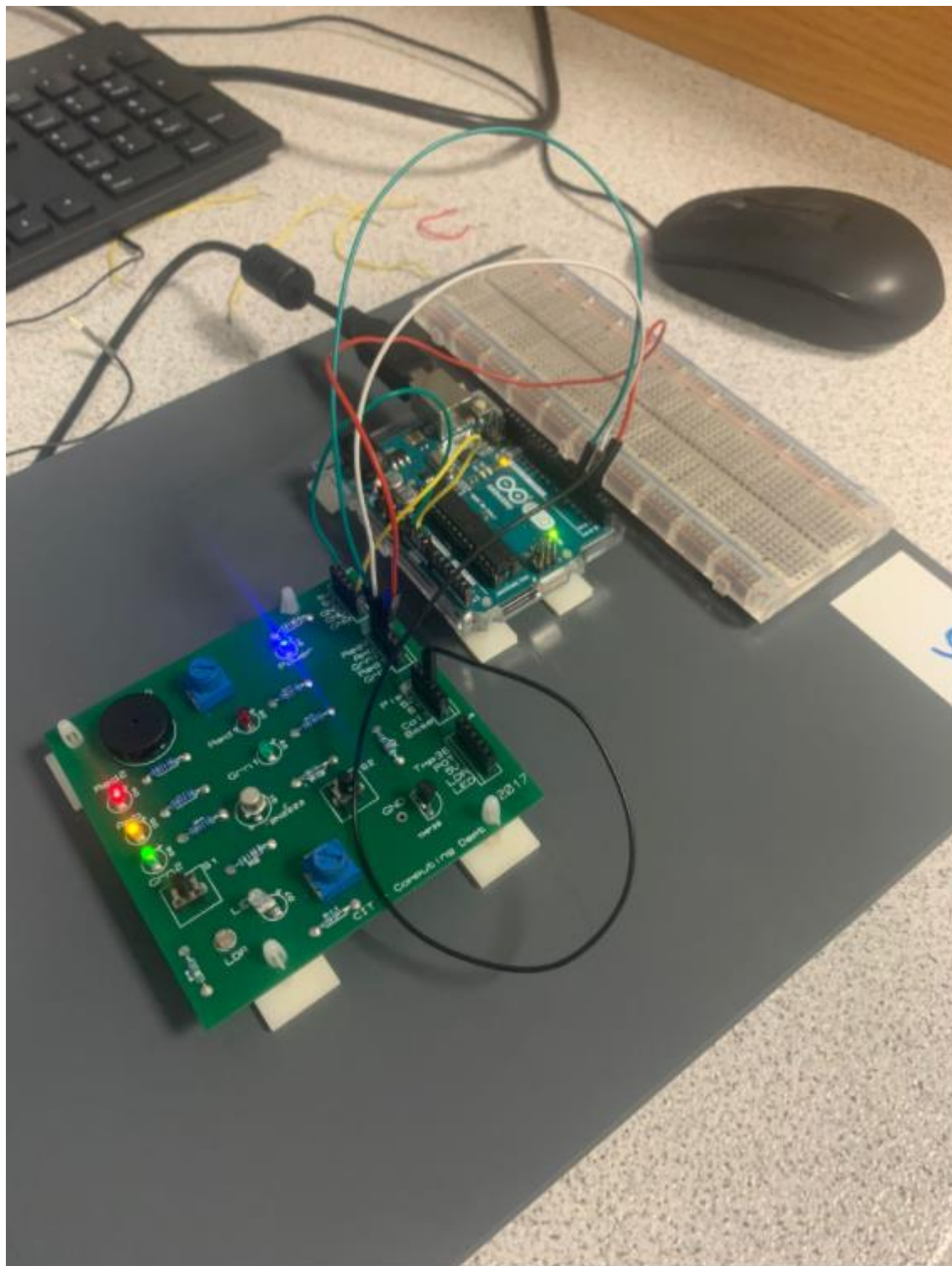
```

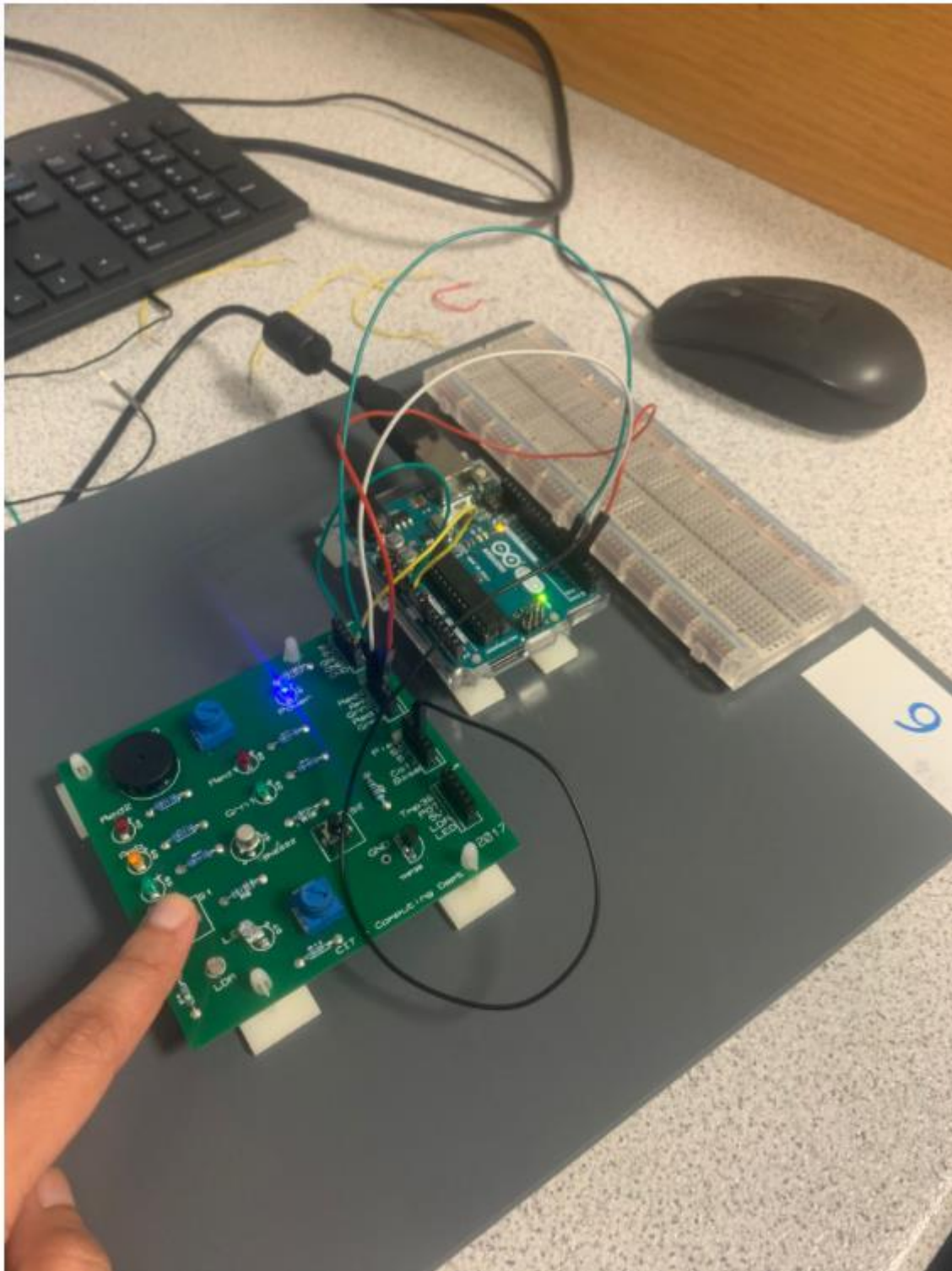
const int switchPin = 2; // named constant for the switch pin
unsigned long previousTime = 0; // store the last time an LED was updated
int switchState = 0; // the current switch state
int prevSwitchState = 0; // the previous switch state
int led = 3; // a variable to refer to the LEDs
int interval = 5000; // interval at which to light the next LED (5 s)

void setup() {
  // set the LED pins as outputs and turn LEDs off
  for(int i = 3; i < 6; i++){
    pinMode(i, OUTPUT);
    digitalWrite(i, LOW);
  }
  // set the switch pin as input
  pinMode(switchPin, INPUT);
}

void loop(){
  // store the time since the Arduino started running in a variable
  unsigned long currentTime = millis();
  // compare the current time to the previous time an LED turned on
  // if it is greater than your interval, turn next LED on
  if(currentTime - previousTime > interval && led < 6) {
    // save the current time as the last time you changed an LED
    previousTime = currentTime;
    // Turn the LED on
    digitalWrite(led, HIGH);
    // increment the led variable
    // in 5 s the next LED will light up
    led++;
  }
  // read the switch value
  switchState = digitalRead(switchPin);
  // if the switch has changed
  if(switchState != prevSwitchState){
    // turn all the LEDs low
    for(int i = 3; i < 6; i++){
      digitalWrite(i, LOW);
    }
    // reset the LED variable to the first one
    led = 3;
    //reset the timer
    previousTime = currentTime;
  }
  // set the previous switch state to the current state
  prevSwitchState = switchState;
}

```





Task2: What difference would you observe if you replaced the `loop()` function in the above code with the following? In particular, why is the code from task 1 a better implementation than the below?

Ans: The `delay` function prevents the further execution of LEDs. It is difficult to deal with inputs and outputs at the same time while

delay() is called. As a result, we are avoiding this function in this project, where time is critical. In this, the delay function is an example of a blocking code.

Task3: The code from task 1 uses an int datatype for the interval variable but a long datatype for currentTime and previousTime.

1. What would be the effect of declaring currentTime and previousTime as type int?
2. What change to your code would be required to change the interval from 5 seconds to 10 minutes?

Ans:

1. Overflow occurs when signed variables are forced to exceed their maximum or minimum capacity; the effect is unknown, hence it should be avoided. The variable "rolling over" from its maximum capacity to its minimum capacity or vice versa is a common symptom of an overflow, however, this is not always the case. Use unsigned int if you want this behavior.
2. `int interval = 600000; (600000 milliseconds).`

Task4: Given that digitalWrite() sets the voltage on an output pin to 0 V (LOW) or 5 V (HIGH), and that the forward voltage of the LEDs is 1.8 V, what is the current that flows through an LED when:

1. The LED is turned on.
2. The LED is turned off.

Ans:

1. 0.0032
2. Zero

Task5:

lab_5_code | Arduino 1.8.19

File Edit Sketch Tools Help

lab_5_code

```
const int switchPin = 7; // named constant for the switch pin
unsigned long previousTime = 0; // store the last time an LED was updated
int switchState = 0; // the current switch state
int prevSwitchState = 0; // the previous switch state
int led = 3; // a variable to refer to the LEDs
int interval = 5000; // interval at which to light the next LED (5 s)

void setup() {
  // set the LED pins as outputs and turn LEDs off
  for(int i = 3; i < 6; i++){
    pinMode(i, OUTPUT);
    digitalWrite(i, LOW);
  }
  // set the switch pin as input
  pinMode(switchPin, INPUT);
}

void loop() {
  // store the time since the Arduino started running in a variable
  unsigned long currentTime = millis();
  // compare the current time to the previous time an LED turned on
  // if it is greater than your interval, turn next LED on
  if(currentTime - previousTime > interval & led < 6) {
    // save the current time as the last time you changed an LED
    previousTime = currentTime;
    // Turn the LED on
    digitalWrite(led, HIGH);
    // increment the led variable
    // in 5 s the next LED will light up
    led++;
  }
  // read the switch value
  switchState = digitalRead(switchPin);
  // if the switch has changed
  if(switchState != prevSwitchState){
    // turn all the LEDs low
    for(int i = 3; i < 6; i++){
      digitalWrite(i, LOW);
    }
    // reset the LED variable to the first one
    led = 3;
    //reset the timer
    previousTime = currentTime;
  }
  // set the previous switch state to the current state
  prevSwitchState = switchState;
}
}
```

Done uploading

Sketch uses 1156 bytes (3%) of program storage space. Maximum is 32256 bytes.
Global variables use 19 bytes (0%) of dynamic memory, leaving 2029 bytes for local variables. Maximum is 2048 bytes.

Activate Windows
Go to Settings to activate Windows.

Arduino Uno as COM5

