Workshop 1

1 Task 1

1. What voltage levels are used for logic on an Arduino Uno?

   Logic High: 5 V. Logic Low: 0 V.

2. What is the maximum current draw from each I/O pin can supply on an Arduino Uno?

   20 mA

3. What is the maximum current draw of the whole board?

   ~200 mA

4. What is the 5V voltage regulator IC part name on the Arduino Uno and what is the typical current it can regulate? (Hint: U1) https://content.arduino.cc/assets/UNO-TH_Rev3e_sch.pdf

   NCP1117ST50T3G. typical current limit is 1500 mA

5. What pins exist on the Arduino Uno and which are capable of Digital Read, Digital Write, Analog Read, Analog Write / PWM. https://content.arduino.cc/assets/Pinout-UNOrev3_latest.pdf

   Digital Read/Write: D0-D19

   Analog Read : A0-A5, or, D14 – D19

   Analog Write : D9-D11,D3,D5-D6

2 Task 2

1. In your own words how would you describe the elements of the bare minimum Arduino code? List any parts that you are uncertain of.

   Void setup(): runs once since start up. Used to setup things, set pins, serial port

   Void loop() : runs after setup and once it finished, runs again.

2. In your own words what does the function digitalWrite do and what inputs does it take?

   hint: try digitalWrite(led_pin, 123);

   make the output 5V if logic true, output 0V if logical False.

3. What units are delay in? Hint: just google

   Milliseconds

4. Briefly investigate and summarise in your own words what the following functions do:
   delayMicroseconds() : delays by microseconds
   millis() : returns number of milliseconds after program starts
   micros() : return number of micro second after program starts

3 Task 3

Push button to light up led

```
int data1 = 0;
int data2 = 0;

const int led_pin = 13;
const int btn1_pin = 12;
const int btn2_pin = 11;
void setup()
{
 pinMode(led_pin, OUTPUT);
 pinMode(btn1_pin, INPUT);
 pinMode(btn2_pin, INPUT_PULLUP);
}
void loop()
{
 data1= digitalRead(btn1_pin);
 data2= !digitalRead(btn2_pin);

 digitalWrite(led_pin, count);
 delay(10);
}
```

4 Task 4

Implement the system and then modify the code so that the LED will stay on (after releasing the button) for the duration that the button was pressed for.

```
int data1 = 0;
int data2 = 0;
int count=0;

const int led_pin = 13;
const int btn1_pin = 12;
const int btn2_pin = 11;
void setup()
{
 pinMode(led_pin, OUTPUT);
 pinMode(btn1_pin, INPUT);
 pinMode(btn2_pin, INPUT_PULLUP);
}
void loop()
{
 data1= digitalRead(btn1_pin);
 data2= !digitalRead(btn2_pin);

 if(data1|data2){
  count++;
 }

 digitalWrite(led_pin, count);
 if(!data1&!data2&&count>0){
  count--;
 }
 delay(10);
}
```

5 Task 5

1. What is the difference between INPUT and INPUT_PULLUP ?

   INPUT_PULLUP connects an internal resistor to the output

2. List 4 logical operators can be used in C/C++?

   &&, ||, !

3. What is the data size and range of values int, double, bool, and long can take?

int	: 16 bits
double	: 64 bits
bool	: 8 bits
long	: 32 bits

6 Task 6

Implement the following design using the code and schematic using a 1K potentiometer. Then modify the code so that the LED is brightest in the middle and a linear mapping to off at the far left and right of the potentiometer. Hint: the abs() function may be helpful.

```
int data_in = 0;
int data_out = 0;
const int led_pin = 11;
const int pot_pin = A0;
void setup()
{
 pinMode(led_pin, OUTPUT);
 pinMode(pot_pin, INPUT);
}
void loop()
{
 data_in = analogRead(pot_pin);
 data_out = map(data_in, 0, 1023, -255, 255);
analogWrite(led_pin, 255-abs(data_out));
 delay(10);
}
```

7 Task 7

1. How many bits is the Arduino Uno's ADC?

   The ADC on the Arduino is a 10-bit ADC meaning it has the ability to detect 1,024 (2^10) discrete analog levels.

2. What frequency is the PWM output on the Arduino Uno?

   490 Hz, pins 5 and 6 : 980Hz

3. How does the map function work?

   Re-maps a number from one range to another. That is, a value of fromLow would get mapped to toLow, a value of fromHigh to toHigh, values in-between to values in-between, etc. LINEAR MAPPING

   ```
   map(value, fromLow, fromHigh, toLow, toHigh)
   ```

8 Task 8

```
int data_in = 0;
int data_out = 0;
const int led_pin = 13;
const int pot_pin = A0;
void setup()
{
 pinMode(led_pin, OUTPUT);
 pinMode(pot_pin, INPUT);
 Serial.begin(9600);
}
void loop()
{
 data_in = analogRead(pot_pin);
 data_out = map(data_in, 0, 1023, -255, 255);
 analogWrite(led_pin, 255-abs(data_out));
 Serial.print("led 13: ");
 Serial.print(255-abs(data_out));
 Serial.print("\n");
 delay(10);
}
```

9 Task 9

Tic tac toe

```
#define BOARD_WIDTH 3
#define BOARD_LENGTH pow(BOARD_WIDTH,2)

int board[9];

const char blank_tile = '*';
const char user_tile = 'X';
const char comp_tile = 'O';

const int blank_id = 0;
const int user_id = 1;
const int comp_id = 2;

const int win = 8;
const int loose = 7;
const int game = 12;
int player_move_text;

bool valid_move;
bool game_on = true;

// Array for board
// 0 1 2
// 3 4 5
// 6 7 8

void setup() {
    Serial.begin(9600);
  pinMode(win,OUTPUT);
  pinMode(loose,OUTPUT);
  pinMode(game,OUTPUT);
  digitalWrite(game,HIGH);
    startGame();
```

```
}

void loop() {

    if (game_on) {

            if (Serial.available()&(remainingTurns()>0)) {
                    player_move_text = Serial.parseInt() - 1;
                    Serial.println(player_move_text);
                    if ((remainingTurns() % 2) == (user_id % 2)) {

                            valid_move = turn(user_id, player_move_text);
                            if (!valid_move) {
                                    Serial.println("invalid move...");
                                    printBoard();
                            }
                    }
                    checkWinner(user_id);
            }

            if (((remainingTurns() % 2) == (comp_id % 2)&(remainingTurns()>0))) {
                    valid_move = false;
                    while (!valid_move) {
                            valid_move = turn(comp_id, random(BOARD_LENGTH));
                    }
                    printBoard();
                    checkWinner(comp_id);


            }
            if ((remainingTurns() == 0) & (game_on)) {
                    Serial.println("It was a tie!?!?!");
            }
    }
    else {
   digitalWrite(game,LOW);
   digitalWrite(win,LOW);
   digitalWrite(loose,LOW);
   delay(100);
        startGame();
        game_on = true;
    }
    delay(100);
}



void printBoard() {
    Serial.println();
    for (int i = 0; i < BOARD_LENGTH; i++) {
        Serial.print((board[i] == 0) ? blank_tile : (board[i] == 1) ? user_tile : (board[i] == 2) ? comp_tile : '?');
        Serial.print(" ");
        if ((i % BOARD_WIDTH) == (BOARD_WIDTH - 1)) {
            Serial.print("\n");
        }
    }
    Serial.println();
}

void clearBoard() {
    for (int i = 0; i < BOARD_LENGTH; i++) {
        board[i] = 0;
    }
}

bool checkWinner(int player_id) {
    //for every for any row , col or diagonal and the value is eqaual to player_id
    bool is_winner = false;
    // Array for board
    // 0 1 2
    // 3 4 5
    // 6 7 8
    //Rows
    is_winner |= (player_id == board[0]) & (board[0] == board[1]) & (board[1] == board[2]);
    is_winner |= (player_id == board[3]) & (board[3] == board[4]) & (board[4] == board[5]);
    is_winner |= (player_id == board[6]) & (board[6] == board[7]) & (board[7] == board[8]);
    //Cols
    is_winner |= (player_id == board[0]) & (board[0] == board[3]) & (board[3] == board[6]);
    is_winner |= (player_id == board[1]) & (board[1] == board[4]) & (board[4] == board[7]);
    is_winner |= (player_id == board[2]) & (board[2] == board[5]) & (board[5] == board[8]);
    //Diag
    is_winner |= (player_id == board[0]) & (board[0] == board[4]) & (board[4] == board[8]);
    is_winner |= (player_id == board[2]) & (board[2] == board[4]) & (board[4] == board[6]);

    if (is_winner) {
        if (player_id == user_id) {
            Serial.println("YOU WIN!!!!!!!!!!!!!!!!!!!!!!!");
     digitalWrite(win,is_winner);
     delay(1000);
        }
        else {
            Serial.println("YOU LOSE..............");
    digitalWrite(loose,is_winner);
    delay(1000);
        }
        game_on = false;
    }
    return is_winner;
}

int remainingTurns() {
    int turns_rem = 0;
    for (int i = 0; i < BOARD_LENGTH; i++) {
        turns_rem += (board[i] == 0);
    }
    return turns_rem;
}

bool turn(int player_id, int player_move) {
    bool is_valid_turn = false;
    if ( (player_move >= 0) & (player_move < BOARD_LENGTH)) {
        if (board[player_move] == blank_id) {
                is_valid_turn = true;
                board[player_move] = player_id;
            }
    }
    return is_valid_turn;
}

void startGame() {
    clearBoard();
    Serial.println("------------------------");
  digitalWrite(game,HIGH);
    Serial.println("TicTacToe Game starting...");
    Serial.print("You will be playing as ");
    Serial.print(user_tile);
    Serial.println(" against the Arduino!\nInput a number to make your move.");
    Serial.println("1 2 3\n4 5 6\n7 8 9");
    printBoard();
}
```