Workshop 1

# Task 1

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

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

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

20 mA

1. **What is the maximum current draw of the whole board?**

~200 mA

1. **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

1. **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

# 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.

1. **What units are delay in? Hint: just google**

Milliseconds

1. **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

# 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);

}

# 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);

}

# Task 5

1. **What is the difference between INPUT and INPUT\_PULLUP ?**

INPUT\_PULLUP connects an internal resistor to the output

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

&&, ||, !

1. **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

# 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);

}

# 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.

1. **What frequency is the PWM output on the Arduino Uno?**

490 Hz, pins 5 and 6 : 980Hz

1. **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)

# 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);

}

# 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();

}