

Assignment 4

ชื่อกลุ่ม : 9 A.M.

สมาชิกกลุ่ม : 64010761 นายวรพล รัชชี

64010757 นายวรโชติ ใจเร็ว

**แนวคิดและวิธีการเล่น :** วิธีการเล่น คือ จะมีผู้เล่น 2 คน โดยแต่ละคนจะต้องควบคุม Bar ในแต่ละฝั่งด้วยปุ่ม 2 ปุ่ม ขึ้น-ลง ตามลำดับ เพื่อทำการสะท้อนลูกบอลที่กำลังลอยมาทางตัวเอง ให้กับอีกฝ่าย หากฝ่ายใดรับลูกบอลไม่ได้ถือว่าแพ้

**อุปกรณ์ที่ใช้ในการทำโครงงาน :**

ชื่ออุปกรณ์	จำนวน
Arduino UNO R3 พร้อมสาย	1
Tact switch 12 mm.	4
Protoboard	1
ลำโพง	1
LED Matrix Driver MAX7219 IC Driver Module พร้อม LED Dot Matrix 8x32 ขนาด40mm. x 40mm.	1
สาย Jump แบบ M-F	5
สาย Jump แบบ M-M	10

**โครงสร้างของโปรแกรม :** ประกอบไปด้วยส่วนหลักๆ ดังนี้

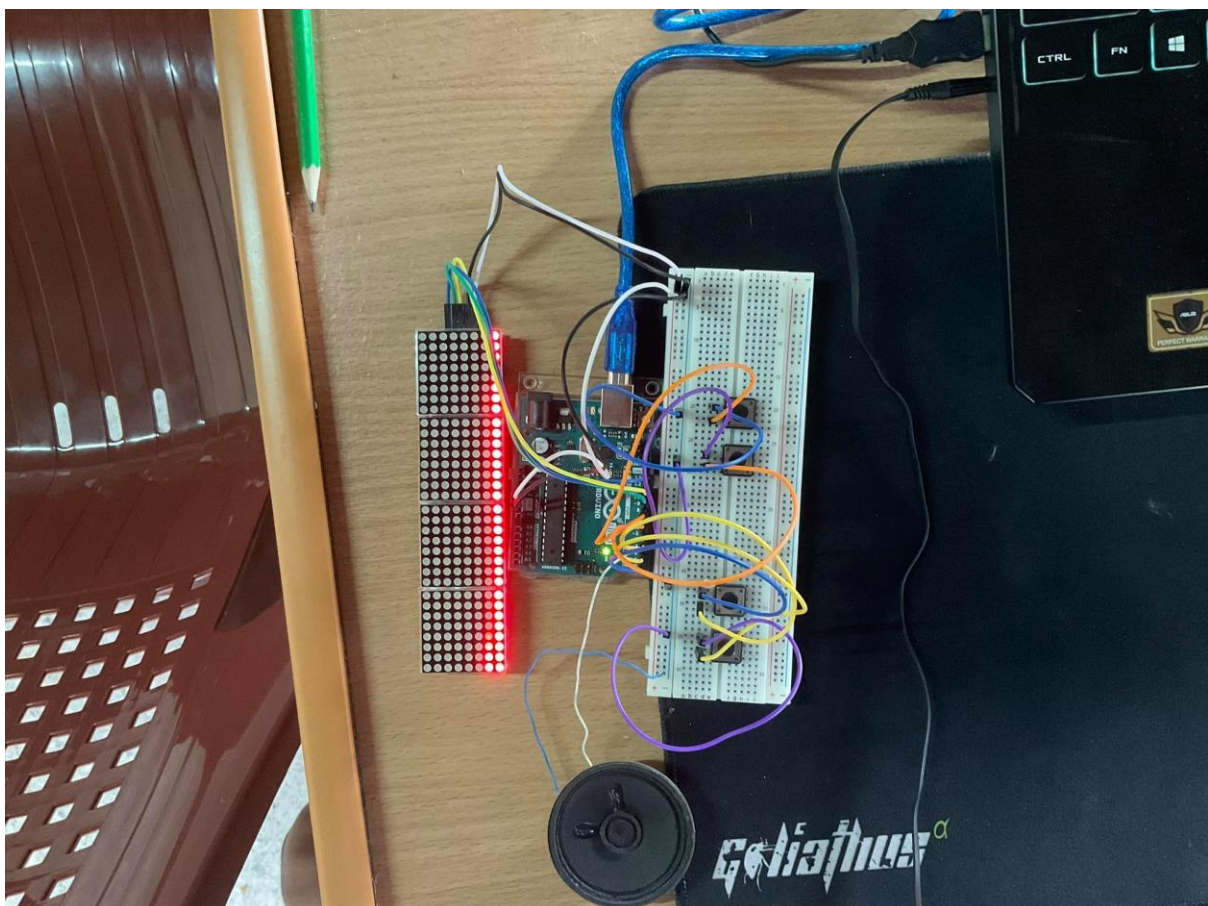
1. ส่วนของที่เก็บตัวแปร Global variable ที่เก็บค่าตัวเลขและข้อมูลต่างๆ เพื่อนำไปดำเนินการ
2. ส่วนของ Function ในส่วนนี้ Function จะคอยรับค่าตัวแปรต่างๆ ที่ถูกส่งเข้ามา และนำมาดำเนินการตามหน้าที่ของแต่ละ Function เช่น Function ที่รับค่าตัวแปร x, y มาเพื่อที่จะแสดง

ตำแหน่งของลูกบอลใน Led Dot Matrix หรือจะเป็น Function ที่คอยเปลี่ยนค่าตัวแปร x, y เพื่อให้ลูกบอลเคลื่อนไหว หรือเพื่อให้ Bar เคลื่อนที่ขึ้นลง และอื่นๆ

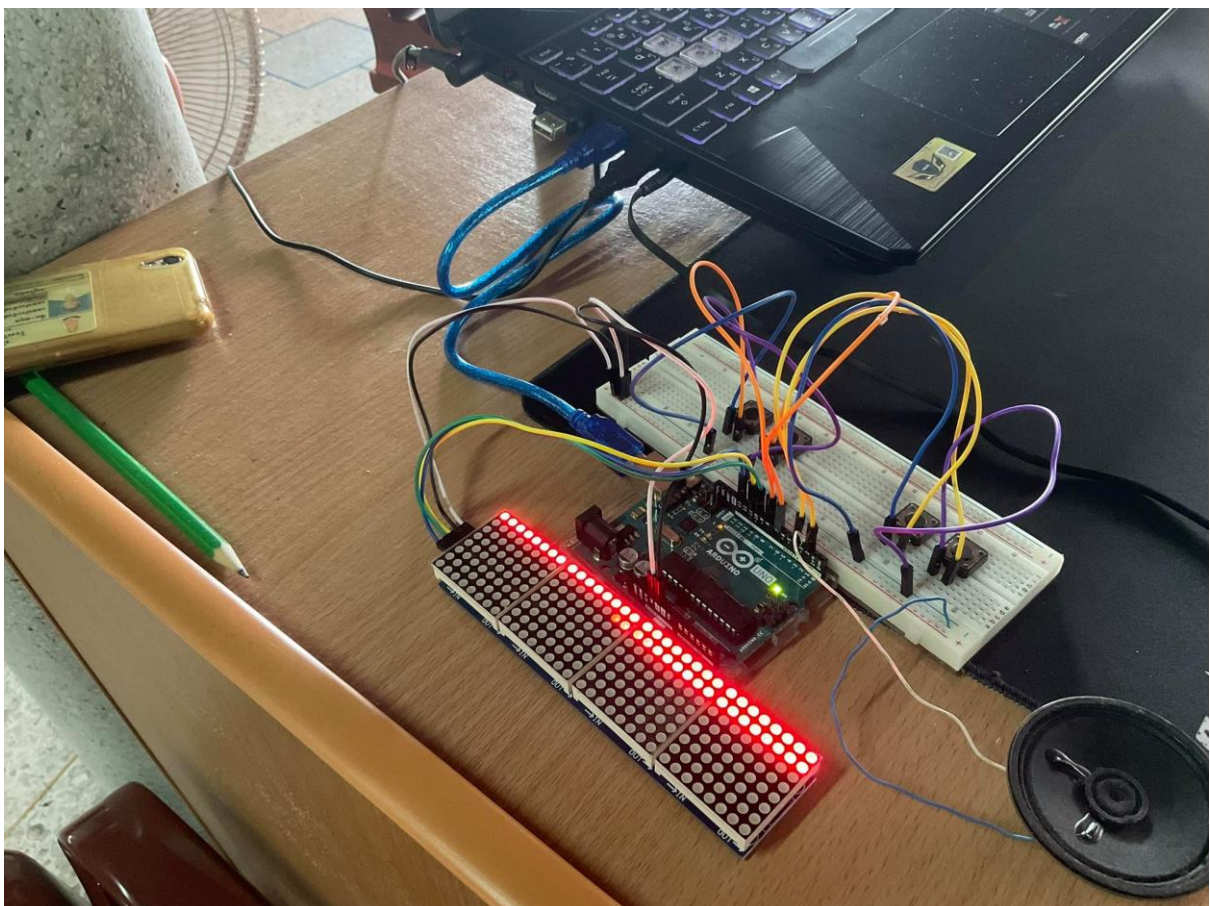
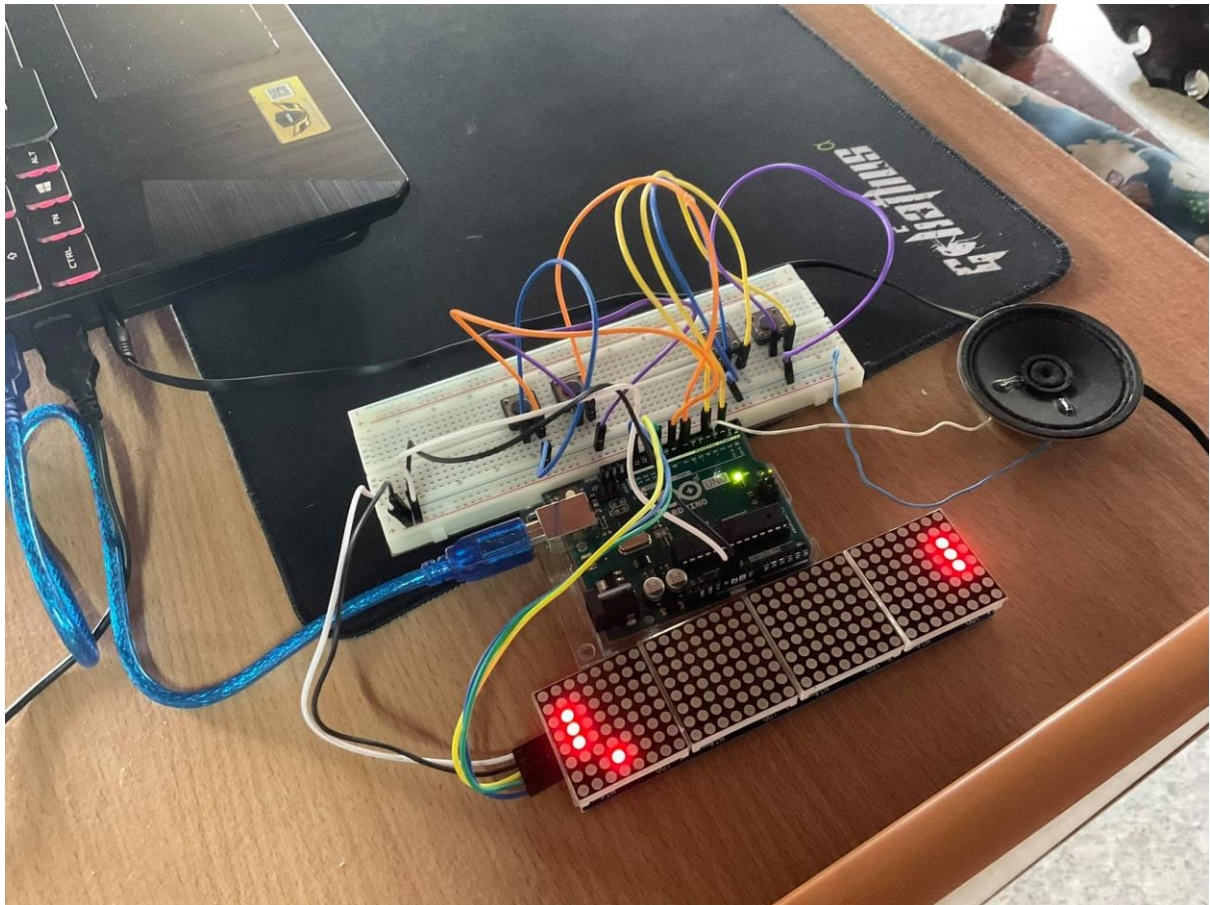
3. ส่วนของ Set up คือ การกำหนดค่าเริ่มต้นการทำงานต่างๆ ทั้งของตัวแปรบางตัวและอุปกรณ์ของ Arduino เช่น pinMode และกำหนดจุดเริ่มต้นของลูกบอล โดย Set up จะทำงานเพียง 1 ครั้งหลังจากได้รับโปรแกรมและจะทำการไปทำงานในฟังก์ชันถัดไป คือ ฟังก์ชัน Loop

4. ส่วนของ Loop ทำหน้าที่เรียกใช้งาน Function ต่างๆเพื่อให้ Function ทั้งหมดทำงานร่วมกัน และแสดงตัวเกม Pong ขึ้นมาผ่าน LED Dot Matrix และยังทำหน้าที่ตรวจสอบการกดปุ่มเพื่อเปลี่ยนค่า y ของ Bar ทำให้ Bar สามารถเคลื่อนที่ขึ้น-ลงได้

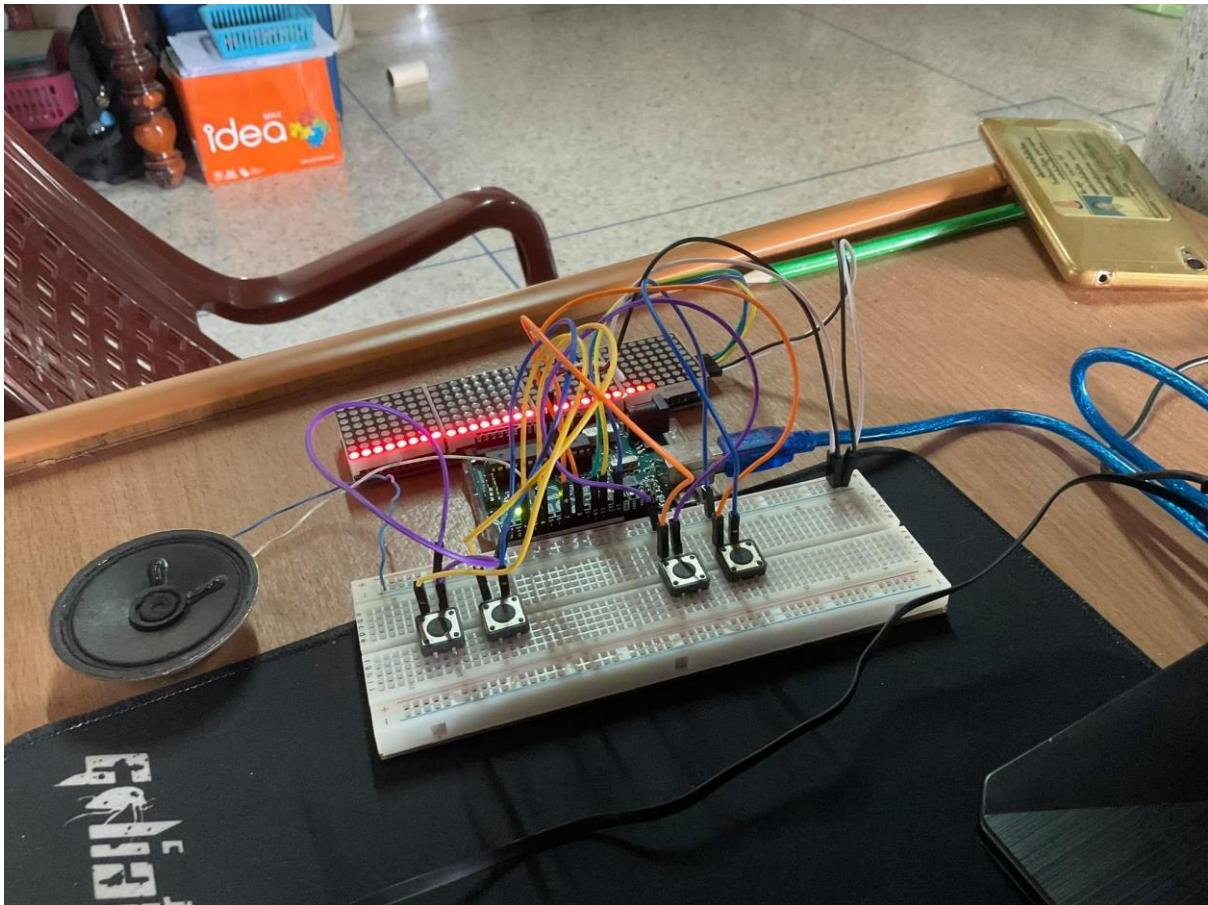
รูปถ่ายชิ้นงาน :











Code : [https://github.com/NineWP/My-ardulTC/tree/main/64010761\\_Assignment\\_4](https://github.com/NineWP/My-ardulTC/tree/main/64010761_Assignment_4)

```
#include <LedControl.h>
#include <time.h>
#define Speaker_Pin 3
LedControl lc = LedControl(11, 13, 10, 4); // DIN,CLK,CS,Modul

//////////////////////// Global Variable //////////////////////////
int ball_x = 7;
int ball_y = 0;
int address = 0;
int c = 1;
bool bounce_x = 0;
int ButtonA_Up, ButtonA_Down, ButtonB_Up, ButtonB_Down;

class Bar {
public:
    int address ;
    int x ;
    int y ;
};
```

```

Bar barA = {3, 1, 3}, barB = {0, 6, 3};

////////// LED_Matrix Setup
void ledMatrixSetup() {

    int device = lc.getDeviceCount();
    for (int address = 0; address < device ; address++) {
        lc.shutdown(address, false);
        lc.setIntensity(address, 1);
        lc.clearDisplay(address);
    }
}

void drawPoint(int address, int a, int b) {////////// DrawBall //////////
    , b, a, true);
}

void deletePoint(int address, int a, int b) {////////// DeleteBall ////

    lc.setLed(address, b, a, false);
}

void drawBar(int address, int a, int b) { ////////// DrawBar //////////

    for (int i = 0; i < 3; i++)
        lc.setLed(address, b + i, a, true);
}

void deleteBar(int address, int a, int b) { ////////// DeleteBar //////////
    for (int i = 0; i < 3; i++)
        lc.setLed(address, b + i, a, false);
}

////////// Change ball position from Right to Left //////////

void RightToLeft() {
    if (ball_x == 7 && address == 0)
        ball_x = 6;
    deletePoint(address, ball_x + 1, ball_y);

    Change_Y();

    if (ball_x < 0 && address != 3) { // Go to next modul
        deletePoint(address, ball_x, ball_y);
        address += 1;
        ball_x = 7;
    }
    else if (ball_y + c >= barA.y && ball_y + c <= barA.y + 2 && ball_x - 1 == b
arA.x && address == 3)

```

```

    bounce_x = 1;

    drawPoint(address, ball_x, ball_y); //Serial.println(ball_x);
    ball_x += -1;

    if (ball_x == barA.x && address == 3)
        deletePoint(address, ball_x + 1, ball_y);
}
//////////Change ball position from Left to Right //////////

void LeftToRight() {
    if (ball_x == 0 && address == 3)
        ball_x = 1;

    deletePoint(address, ball_x - 1, ball_y);

    Change_Y();

    if (ball_x > 7 && address != 0) { // Go to next modul
        deletePoint(address, ball_x, ball_y);
        address += -1;
        ball_x = 0;
    }
    else if (ball_y + c >= barB.y && ball_y + c <= barB.y + 2 && ball_x + 1 == barB.x && address == 0)
        bounce_x = 0;

    drawPoint(address, ball_x, ball_y); //Serial.println(ball_x);
    ball_x += 1;

    if (ball_x == barB.x && address == 0)
        deletePoint(address, ball_x - 1, ball_y);
}
////////// Change Y axis of ball //////////

void Change_Y() {
    ball_y += c; //Serial.println(ball_y);

    if (ball_y + 1 > 7 || ball_y - 1 < 0) {
        c = -1 * c;
        tone(Speaker_Pin, 370);
        delay(20);
        noTone(Speaker_Pin);
    }
}

void Close() {

```

```

int frequency = 0;
for (int a = 0; a < 8; a++) {
    for (int i = 0; i < 4; i++) {
        for (int j = 7; j >= 0; j--) {
            lc.setLed(i, a, j, true);
            delay(5);
        }
        if (frequency <= 25) { /////////////////////////////////// Sound effect

            tone(Speaker_Pin, frequency += 5);
            //Serial.println(frequency);
        }
        else
            noTone(Speaker_Pin);
    }
}
for (int a = 0; a < 8; a++) {
    for (int i = 0; i < 4; i++) {
        for (int j = 7; j >= 0; j--) {
            lc.setLed(i, a, j, false);
            delay(5);
        }
        if (frequency >= 0) { /////////////////////////////////// Sound effect
            tone(Speaker_Pin, frequency -= 5);
            //Serial.println(frequency);
        }
        else
            noTone(Speaker_Pin);
    }
}
}

void CheckRespawn() { /////////////////////////////////// Respawn Ball ///////////////////////////////////

    if (ball_x < 0 && address == 3 ) {
        Close();
        deletePoint(address, ball_x + 1, ball_y);
        delay(500);
        address = 2;
        bounce_x = 1;
        ball_x = 1 + rand() % 6;
        ball_y = 1 + rand() % 6;
    }
    else if (ball_x > 7 && address == 0) {
        Close();
        deletePoint(address, ball_x - 1, ball_y);
        delay(500);
        address = 1;
    }
}

```

```

    bounce_x = 0;
    ball_x = 1 + rand() % 6;
    ball_y = 1 + rand() % 6;
}
}

//////////////////////////////// Set up //////////////////////////////////
void setup() {
    srand(time(NULL));
    ball_x = 1 + rand() % 6;
    ball_y = 1 + rand() % 6;
    Serial.begin(9600);
    ledMatrixSetup();
    pinMode(Speaker_Pin, OUTPUT);
    pinMode(2, INPUT_PULLUP);
    pinMode(4, INPUT_PULLUP);
    pinMode(7, INPUT_PULLUP);
    pinMode(8, INPUT_PULLUP);
}

//////////////////////////////// LOOP //////////////////////////////////

void loop() {

    ButtonA_Up = digitalRead(2);
    ButtonA_Down = digitalRead(4);
    ButtonB_Up = digitalRead(7);
    ButtonB_Down = digitalRead(8);
    ////////////////////////////////// A bar button //////////////////////////////////
    if (ButtonA_Up == 0 && barA.y != 0)
    {
        deleteBar(barA.address, barA.x, barA.y);
        barA.y += -1;
    }
    else if (ButtonA_Down == 0 && barA.y != 5)
    {
        deleteBar(barA.address, barA.x, barA.y);
        barA.y += 1;
    } //Serial.println(barA.y);
    //Serial.println(barB.y);

    ////////////////////////////////// B bar button //////////////////////////////////
    if (ButtonB_Up == 0 && barB.y != 0)
    {
        deleteBar(barB.address, barB.x, barB.y);
        barB.y += -1;
    }
    else if (ButtonB_Down == 0 && barB.y != 5)

```



```
{
    deleteBar(barB.address, barB.x, barB.y);
    barB.y += 1;
}

CheckRespawn();
drawBar(barA.address, barA.x, barA.y);
drawBar(barB.address, barB.x, barB.y);

if (1) {
    if (bounce_x == 0)
        RightToLeft();
    else
        LeftToRight();
}
delay(100);
}
```