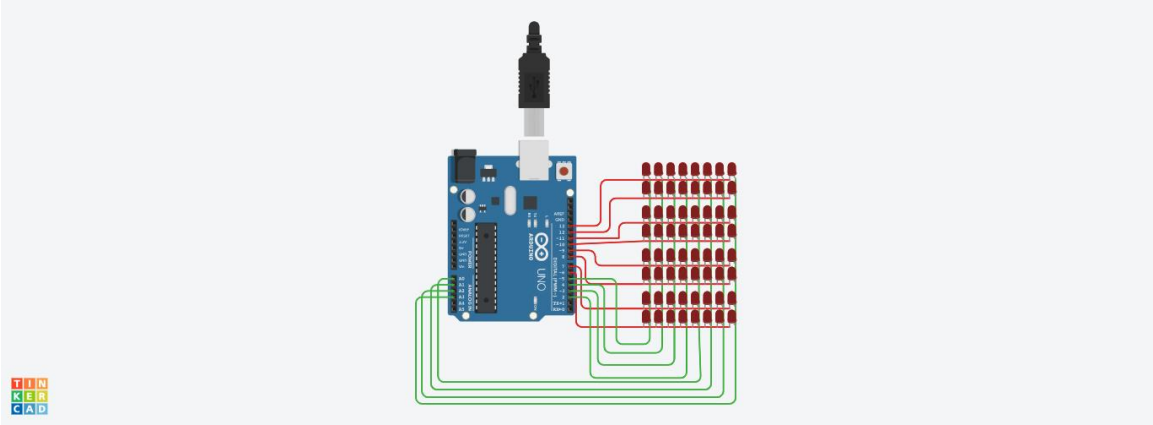
electrical-engineering-1

Electronics and Electrical Engineering track. Robot screen recognition, facial expressions, manufacturing, and screen making experience were done by led matrix's Arduino, the program used Tinkercad.



- Code

```
#define ROW_1 13
#define ROW_2 12
#define ROW_3 11
#define ROW_4 10
#define ROW_5 9
#define ROW_6 8
#define ROW_7 7
#define ROW_8 6

#define COL_1 A3
#define COL_2 A2
#define COL_3 A1
#define COL_4 A0
#define COL_5 2
#define COL_6 3
#define COL_7 4
#define COL_8 5

const byte rows[] = {
    ROW_1, ROW_2, ROW_3, ROW_4, ROW_5, ROW_6, ROW_7, ROW_8
};
const byte col[] = {
  COL_1,COL_2, COL_3, COL_4, COL_5, COL_6, COL_7, COL_8
```

```
};


// It's prefilled with a smiling face (1 = ON, 0 = OFF)
byte a[]=
{B00000000,B00111000,B01000100,B01000100,B01000100,B01000100,B00111000,B00
000000};
byte b[]={
B00001000,B00011000,B00101000,B01001000,B00001000,B00001000,B01111110,B000
00000};


byte ALL[] =
{B11111111,B11111111,B11111111,B11111111,B11111111,B11111111,B11111111,B11
111111};
byte EX[] =
{B00000000,B00000000,B00000000,B00000000,B00000000,B00000000,B00000000,B00
000000};
byte A[] =
{B00000000,B00011000,B00100100,B01000010,B01111110,B01000010,B01000010,B00
000000};
byte B[] =
{B01111000,B01001000,B01001000,B01110000,B01001000,B01000100,B01000100,B01
111100};
byte C[] =
{B00000000,B00111100,B01000010,B01000000,B01000000,B01000010,B00111100,B00
000000};
byte D[] =
{B00000000,B00111000,B00100100,B00100010,B00100010,B00100100,B00111000,B00
000000};
byte E[] =
{B00000000,B00111100,B00100000,B00111000,B00100000,B00100000,B00111100,B00
000000};
byte F[] =
{B00000000,B00111100,B00100000,B00111000,B00100000,B00100000,B00100000,B00
000000};
byte G[] =
{B00000000,B00111110,B00100000,B00100000,B00101110,B00100010,B00111110,B00
000000};
byte H[] =
{B00000000,B00100100,B00100100,B00111100,B00100100,B00100100,B00100100,B00
000000};
byte I[] =
{B00000000,B00111000,B00010000,B00010000,B00010000,B00010000,B00111000,B00
000000};
```

```
byte J[] =
{B00000000,B00011100,B00001000,B00001000,B00001000,B00101000,B00111000,B00
000000};
byte K[] =
{B00000000,B00100100,B00101000,B00110000,B00101000,B00100100,B00100100,B00
000000};
byte L[] =
{B00000000,B00100000,B00100000,B00100000,B00100000,B00100000,B00111100,B00
000000};
byte M[] =
{B00000000,B00000000,B01000100,B10101010,B10010010,B10000010,B10000010,B00
000000};
byte N[] =
{B00000000,B00100010,B00110010,B00101010,B00100110,B00100010,B00000000,B00
000000};
byte O[] =
{B00000000,B00111100,B01000010,B01000010,B01000010,B01000010,B00111100,B00
000000};
byte P[] =
{B00000000,B00111000,B00100100,B00100100,B00111000,B00100000,B00100000,B00
000000};
byte Q[] =
{B00000000,B00111100,B01000010,B01000010,B01000010,B01000110,B00111110,B00
000001};
byte R[] =
{B00000000,B00111000,B00100100,B00100100,B00111000,B00100100,B00100100,B00
000000};
byte S[] =
{B00000000,B00111100,B00100000,B00111100,B00000100,B00000100,B00111100,B00
000000};
byte T[] =
{B00000000,B01111100,B00010000,B00010000,B00010000,B00010000,B00010000,B00
000000};
byte U[] =
{B00000000,B01000010,B01000010,B01000010,B01000010,B00100100,B00011000,B00
000000};
byte V[] =
{B00000000,B00100010,B00100010,B00100010,B00010100,B00010100,B00001000,B00
000000};
byte W[] =
{B00000000,B10000010,B10010010,B01010100,B01010100,B00101000,B00000000,B00
000000};
byte X[] =
{B00000000,B01000010,B00100100,B00011000,B00011000,B00100100,B01000010,B00
000000};
```

```
byte Y[] =
{B00000000,B01000100,B00101000,B00010000,B00010000,B00010000,B00010000,B00
000000};
byte Z[] =
{B00000000,B00111100,B00000100,B00001000,B00010000,B00100000,B00111100,B00
000000};

float timeCount = 0;

void setup()
{

    Serial.begin(9600);

    for (byte i = 2; i <= 13; i++)
        pinMode(i, OUTPUT);
    pinMode(A0, OUTPUT);
    pinMode(A1, OUTPUT);
    pinMode(A2, OUTPUT);
    pinMode(A3, OUTPUT);
}

void loop() {

delay(10);
timeCount += 1;

if(timeCount <  40)
{
drawScreen(H);
}
else if (timeCount <  80)
{
drawScreen(E);
}
else if (timeCount <  120)
{
drawScreen(L);
}
else if (timeCount <  160)
{
drawScreen(L);
}
else if (timeCount <  200)
{
```

```
drawScreen(O);
}
else if (timeCount <  220)
{
drawScreen(ALL);
}
else if (timeCount <  240)
{
  drawScreen(ALL);
}
else if (timeCount <  280)
{
drawScreen(T);
}
else if (timeCount <  320)
{
drawScreen(E);
}
else if (timeCount <  360)
{
drawScreen(A);
}
else if (timeCount <  400)
{
drawScreen(C);
}
else if (timeCount <  440)
{
drawScreen(H);
}
else if (timeCount <  480)
{
drawScreen(E);
}
else if (timeCount <  520)
{
drawScreen(R);
}
else if (timeCount <  540)
{
drawScreen(ALL);
}
else if (timeCount <  560)
{
drawScreen(ALL);
```

```
}
  else {

  timeCount = 0;
  }
}


 void  drawScreen(byte buffer2[])
 {

    for (byte i = 0; i < 8; i++)
     {
        digitalWrite(rows[i], LOW);
        for (byte a = 0; a < 8; a++)
        {

          digitalWrite(col[a], (buffer2[i] >> a) & 0x01);

          delayMicroseconds(100);

          digitalWrite(col[a], 0);
        }
        digitalWrite(rows[i], HIGH);
    }
}
```