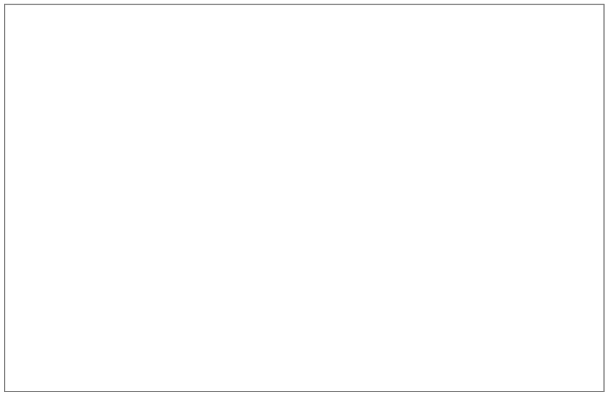
Χρήση της KYX-5461AS 4-ψηφίων 7-τμημάτων(segment) LED οθόνης με το Arduino



Πρόκειται για οθόνη κοινής καθόδου. Αυτό σημαίνει ότι πρέπει να συνδεθούν αντιστάτες στις ακίδες που οδηγούν κάθε ψηφίο (αντί για την ακίδα κάθε τμήματος όπως γίνεται σε οθόνες κοινής ανόδου) για να αποφύγουμε το "κάψιμο" των led.

Κατανομή ακίδων και :

1 Τμήμα E
2 Τμήμα D
3 Δεκαδικό σημείο
4 Τμήμα C
5 Τμήμα G
6 Ψηφίο 4
7 Τμήμα B
8 Ψηφίο 3
9 Ψηφίο 2
10 Τμήμα F
11 Τμήμα A
12 Ψηφίο 1

Αντιστοιχία Τμημάτων :

Για να παρουσιαστεί π.χ ο αριθμός 1 στο τρίτο ψηφίο πρέπει να τροφοδοτήσουμε τις ακίδες 4, 7 και 8.

Για να παρουσιαστούν αριθμοί σε όλα τα ψηφία χρειάζεται να εισάγουμε μια καθυστέρηση κατά την ενεργοποίηση του κάθε ψηφίου στον κώδικα στο loop()

Για κάθε ψηφίο μπορούν να χρησιμοποιηθούν αντιστάτες 220 Ω . Όσο μεγαλώνει η τιμή των αντιστατών τόσο μειώνεται η φωτεινότητα της οθόνης.

Ακολουθεί ο κώδικας για ένα αισθητήρα θερμοκρασίας LM35 και την παρουσίαση των τιμών θερμοκρασίας στην οθόνη του KYX-5461AS.

```
//reading temperature and outputting to display Ανάγνωση θερμοκρασίας και εμφάνιση στην οθόνη
int sensorPin = 0;

//display pins
int segA = 5;
int segB = 13;
int segC = 10;

int segD = 8;
int segE = 7;
int segF = 4;
int segG = 11;
int segPt = 9;

int d1 = 6;

int d2 = 3;
int d3 = 2;
int d4 = 12;

int delayTime = 900;

int counter = 0;

float temperature = 77.7;

void setup() {
// put your setup code here, to run once: Κώδικας που "τρέχει" κατά την εκκίνηση
//start serial communications Εκκίνηση των σειριακών επικοινωνιών
Serial.begin(9600);

//set up outputs ορισμός εξόδων
pinMode(12, OUTPUT);
```

```
pinMode(11, OUTPUT);
pinMode(10, OUTPUT);
pinMode(9, OUTPUT);
pinMode(8, OUTPUT);
pinMode(7, OUTPUT);
pinMode(6, OUTPUT);
pinMode(5, OUTPUT);
pinMode(4, OUTPUT);
pinMode(3, OUTPUT);
pinMode(2, OUTPUT);
pinMode(13, OUTPUT);
pinMode(0,INPUT);
delay(1000);
}

void loop() {

// put your main code here, to run repeatedly: Κώδικας που "τρέχει" συνεχώς

//only read temp every 100 cycles "Διάβασε" τη θερμοκρασία κάθε 100 κύκλους
if(counter%500 == 0)
{
// read the pin
int reading = analogRead(sensorPin);
//convert reading to volts
float volts = (reading * 5.0);
volts /= 1024.0;
// Serial.print(volts);
// Serial.println(" v");
temperature = volts * 100.0;
Serial.print(temperature);
Serial.println(" degrees Celsius");

//test output to display
// allHigh();
//reset our counter
counter = 0;
}

counter ++;

selectDigit(1);
sendDigit(tens(temperature));
delayMicroseconds(delayTime);

digitalWrite(d1, HIGH);
selectDigit(2);
```

```
sendDigit(ones(temperature));
point();
delayMicroseconds(delayTime);

digitalWrite(d2, HIGH);
selectDigit(3);
sendDigit(points(temperature));
delayMicroseconds(delayTime);
//turn point off

digitalWrite(d3, HIGH);
digitalWrite(segPt,HIGH);
selectDigit(4);
cee();
delayMicroseconds(delayTime);
digitalWrite(d4, HIGH);



}

void allLow() {

digitalWrite( 13, LOW); // A
digitalWrite( 2, LOW); // B
digitalWrite( 3, LOW); // C
digitalWrite(4, LOW); // D
digitalWrite(5, LOW); // E
digitalWrite( 6, LOW); // F
digitalWrite( 7, LOW); // G
digitalWrite(8, LOW); //point
digitalWrite(9, LOW);
digitalWrite(10, LOW);
digitalWrite(11, LOW);
digitalWrite(12, LOW);

}

void allHigh() {

digitalWrite( 13, HIGH); // A
digitalWrite( 2, HIGH); // B
digitalWrite( 3, HIGH); // C
digitalWrite(4, HIGH); // D
digitalWrite(5, HIGH); // E
digitalWrite( 6, HIGH); // F
digitalWrite( 7, HIGH); // G
digitalWrite(8, HIGH); //point
```

```
digitalWrite(9, HIGH);
digitalWrite(10, HIGH);
digitalWrite(11, HIGH);
digitalWrite(12, HIGH);

}

void one()
{
digitalWrite(segA, LOW);
digitalWrite(segB, HIGH);
digitalWrite(segC, HIGH);
digitalWrite(segD, LOW);
digitalWrite(segE, LOW);
digitalWrite(segF, LOW);
digitalWrite(segG, LOW);
digitalWrite(segPt, LOW);

}
void two()
{
digitalWrite(segA, HIGH);
digitalWrite(segB, HIGH);
digitalWrite(segC, LOW);
digitalWrite(segD, HIGH);
digitalWrite(segE, HIGH);
digitalWrite(segF, LOW);
digitalWrite(segG, HIGH);
digitalWrite(segPt, LOW);

}
void three()
{
digitalWrite(segA, HIGH);
digitalWrite(segB, HIGH);
digitalWrite(segC, HIGH);
digitalWrite(segD, HIGH);
digitalWrite(segE, LOW);
digitalWrite(segF, LOW);
digitalWrite(segG, HIGH);
digitalWrite(segPt, LOW);

}
void four()
{
digitalWrite(segA, LOW);
```

```
digitalWrite(segB, HIGH);
digitalWrite(segC, HIGH);
digitalWrite(segD, LOW);
digitalWrite(segE, LOW);
digitalWrite(segF, HIGH);
digitalWrite(segG, HIGH);
digitalWrite(segPt, LOW);

}
void five()
{
digitalWrite(segA, HIGH);
digitalWrite(segB, LOW);
digitalWrite(segC, HIGH);
digitalWrite(segD, HIGH);
digitalWrite(segE, LOW);
digitalWrite(segF, HIGH);
digitalWrite(segG, HIGH);
digitalWrite(segPt, LOW);

}
void six()
{
digitalWrite(segA, HIGH);
digitalWrite(segB, LOW);
digitalWrite(segC, HIGH);
digitalWrite(segD, HIGH);
digitalWrite(segE, HIGH);
digitalWrite(segF, HIGH);
digitalWrite(segG, HIGH);
digitalWrite(segPt, LOW);

}
void seven()
{
digitalWrite(segA, HIGH);
digitalWrite(segB, HIGH);
digitalWrite(segC, HIGH);
digitalWrite(segD, LOW);
digitalWrite(segE, LOW);
digitalWrite(segF, LOW);
digitalWrite(segG, LOW);
digitalWrite(segPt, LOW);

}
void eight()
```

```
{
digitalWrite(segA, HIGH);
digitalWrite(segB, HIGH);
digitalWrite(segC, HIGH);
digitalWrite(segD, HIGH);
digitalWrite(segE, HIGH);
digitalWrite(segF, HIGH);
digitalWrite(segG, HIGH);
digitalWrite(segPt, LOW);

}
void nine()
{
digitalWrite(segA, HIGH);
digitalWrite(segB, HIGH);
digitalWrite(segC, HIGH);
digitalWrite(segD, HIGH);
digitalWrite(segE, LOW);
digitalWrite(segF, HIGH);
digitalWrite(segG, HIGH);
digitalWrite(segPt, LOW);

}
void zero()
{
digitalWrite(segA, HIGH);
digitalWrite(segB, HIGH);
digitalWrite(segC, HIGH);
digitalWrite(segD, HIGH);
digitalWrite(segE, HIGH);
digitalWrite(segF, HIGH);
digitalWrite(segG, LOW);
digitalWrite(segPt, LOW);

}

void cee()
{
digitalWrite(segA, HIGH);
digitalWrite(segB, LOW);
digitalWrite(segC, LOW);
digitalWrite(segD, HIGH);
digitalWrite(segE, HIGH);
digitalWrite(segF, HIGH);
digitalWrite(segG, LOW);
digitalWrite(segPt, LOW);
```

```
}

void point()
{
digitalWrite(segPt, HIGH);
}

void selectDigit(int d)
{
/*
digitalWrite(d1,HIGH);

digitalWrite(d2,HIGH);
digitalWrite(d3,HIGH);
digitalWrite(d4,HIGH);
*/

switch (d)
{
case 1:
digitalWrite(d1, LOW);
break;
case 2:
digitalWrite(d2, LOW);
break;
case 3:
digitalWrite(d3, LOW);
break;
default:
digitalWrite(d4, LOW);
break;
}
}

void sendDigit(int x)
{
switch(x)
{
case 1:
one();
break;
case 2:
two();
break;
case 3:
three();
```

```
break;
case 4:
four();
break;
case 5:
five();
break;
case 6:
six();
break;
case 7:
seven();
break;
case 8:
eight();
break;
case 9:
nine();
break;
case 10:
cee();
break;
default:
zero();
break;
}
}

int tens(float x)
{
float divided = x/10.0;
return (int)divided;
}

int ones(float x)
{
float divided = x - (10.0 * tens(x));
// Serial.print(divided);
// Serial.println(" ***ones***");
return (int)divided;
}

int points(float x)
{
float divided = x - (10.0 * tens(x)) - ones(x);
divided *= 10;
```

```
return (int)divided;
}
```