**ГЛАВА ЧЕТВЪРТА**

**Проектиране и софтуер за управление на автоматизирана система за сортиране на хранителни продукти по цвят**

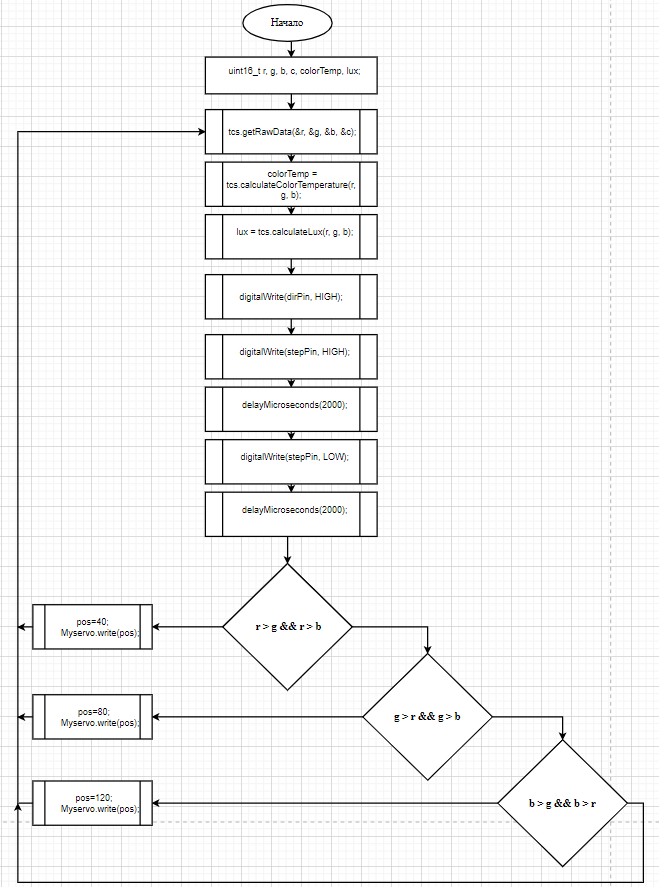
**4. Алгоритмизация и Блок схема на софтуера**

**4.1 Алгоритъм на програмата**

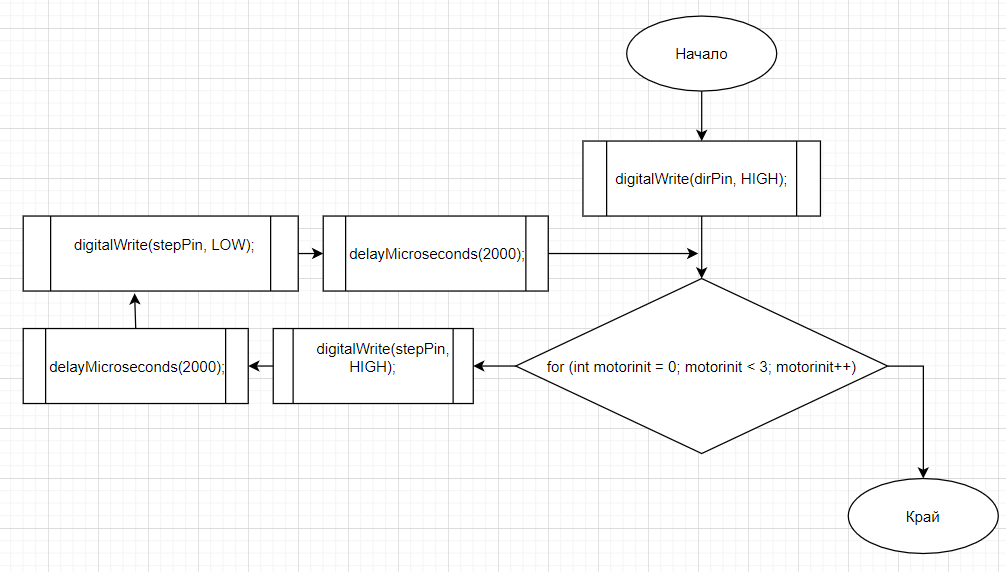
За да се пусне макета е нужно да се включи ключа за захранването и ардуиното да бъде свързано. При подаване на първоначално захранващо напрежение към контролера той извършва инициализация. Това означава, че се инициализират променливите. След като се инициализират променливите стъпковият мотор прави три стъпки. Това позволява на елементите да паднат на местата и да приготвят работата за сензора. Със това инициализацията приключва.

След иницилизация макета влиза в работен режим. Сензора за светлина взема данните за елемента над него. След като се вземат данните стъпковият мотор прави една стъпка. След като направи стъпката има малко изчакване от 1 секунда. В зависимост от данните снети от сензора серво мотора завърта рампата към контейнера за който трябва да се отдели елемента. Със това завършва алгоритъма и се връща към стъпката със сканирането на сензора. Системата е направена да работи непрекъснато до ръчно изключване от оператор.

. **4.2 Блок схема на алгоритъма**

****

**Фиг. 56** Блок схема на loop алгоритъма

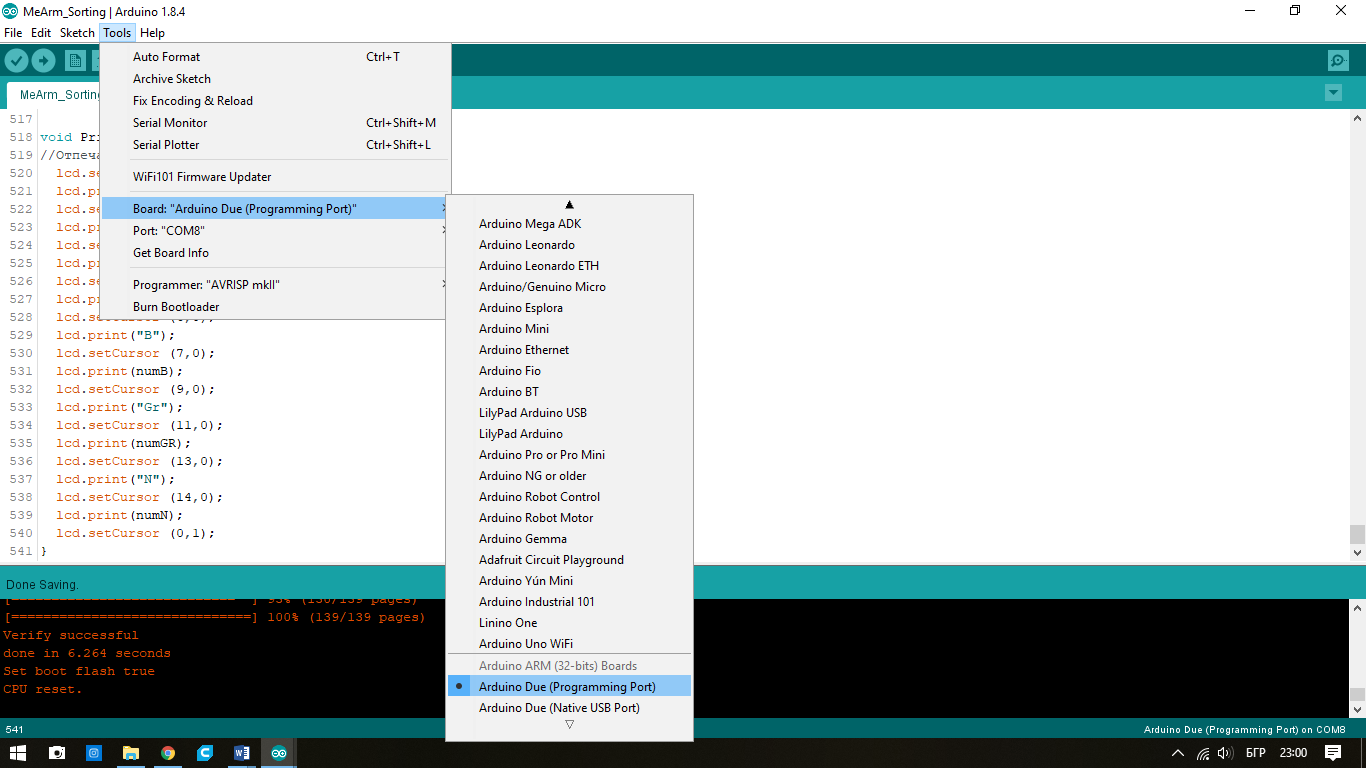


**Фиг. 57** Блок схема на initialization алгоритъма

**4.3 Програмиране на софтуер за управление в среда ардуино**

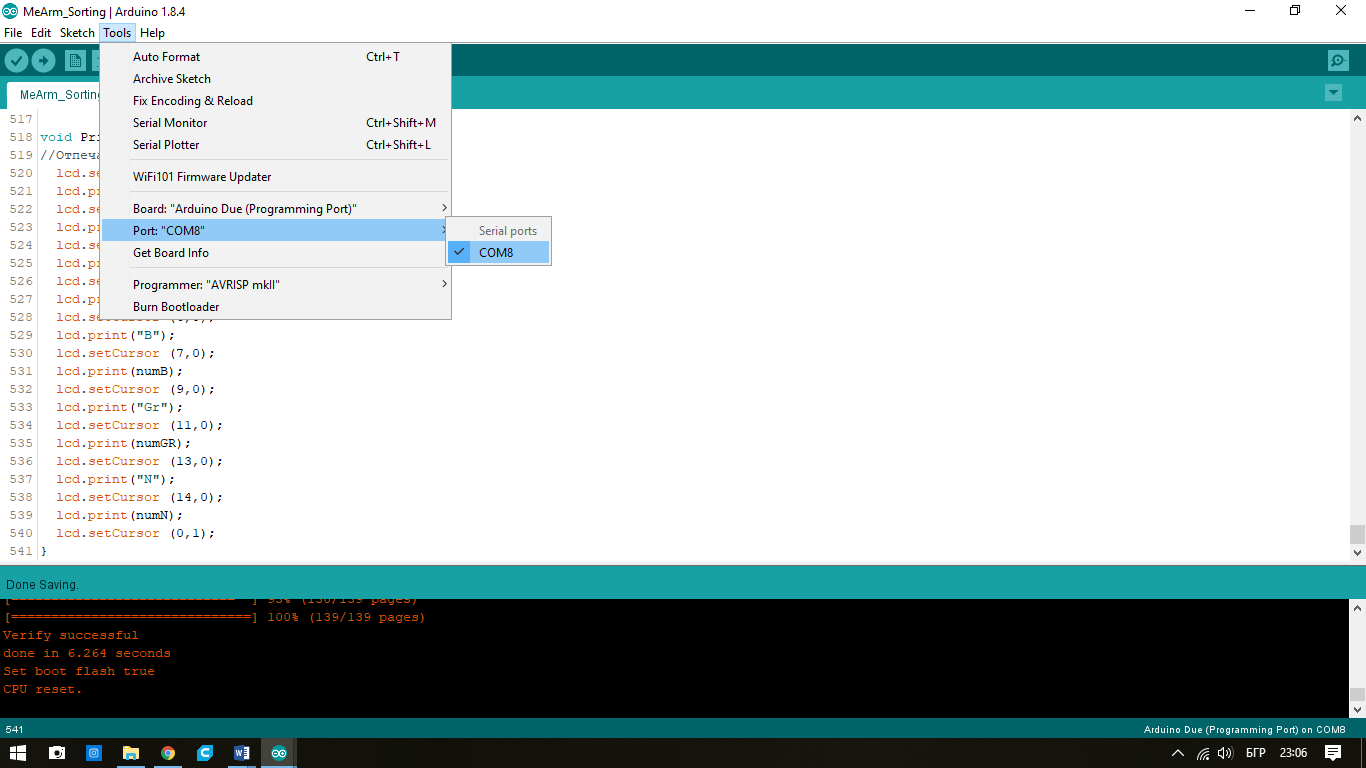
**4.3.1 Конфигурация на среда за писане и записване на код в** **контролера**

Голямо предимство при контролерите “ардуино” е, че средата за програмиране е напълно безплатна и може да бъде изтеглена от официалния сайт свободно. За да започне писането на кода след стартирането на програмата първо се задава кой е контролера, който ще бъде програмиран. Това става в меню Tools->Board-> . За целта избираме контролера Arduino Due(Programing Port).



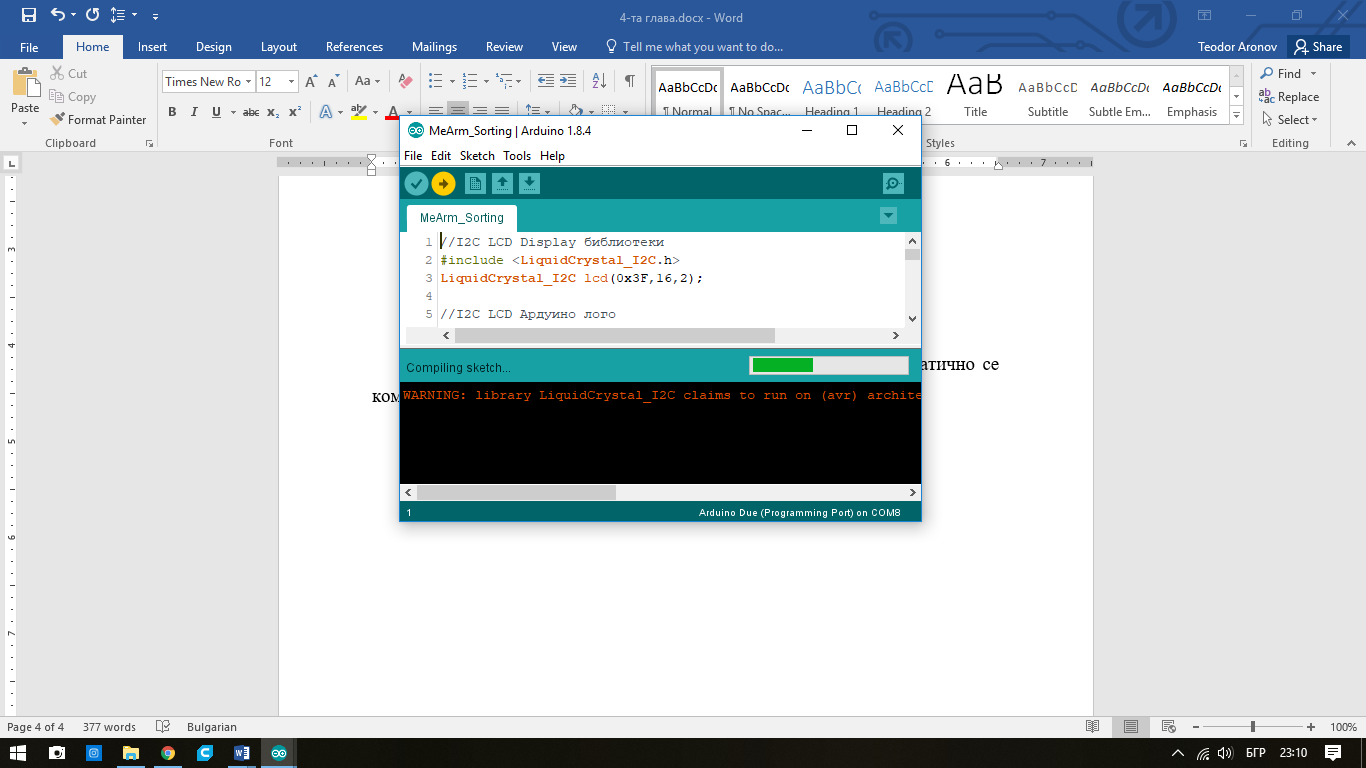
**Фиг. 58** Конфигуриране на контролера в средата на ардуино.

След това е нужно да се избере серийния порт през, който ще бъде свързан кабела за програмиране.



**Фиг. 59** Конфигуриране на серийния порт в средата на ардуино.

Когато кода е написан се записва на контролера като автоматично се компилира преди това с бутона “Upload”.



**Фиг. 60** Компилиране и записване на програмата в контролера.

**4.2.2. Управляваща програма**

Програмите за ардуино се пишат на език модифициран от C++. Модифициран е заради спецификата на контролерите.

Използвани библиотеки:

- Wire.h – Библиотека която помага за работата със сензора за цвят.

-Adafruit\_TCS34725.h – Библиотека която помага за работата със сензора за цвят.

- Servo.h – С тази Библиотека се управлява серво мотора.

#include <Wire.h>

#include "Adafruit\_TCS34725.h"

#include<Servo.h>

#define dirPin 2

#define stepPin 3

#define stepsPerRevolution 8

Servo Myservo;

int pos;

int motorinit;

/\* Инициализиране на сензора със специфично време и стойности \*/

Adafruit\_TCS34725 tcs = Adafruit\_TCS34725(TCS34725\_INTEGRATIONTIME\_700MS, TCS34725\_GAIN\_1X);

void setup()

{

//Задаване режима на пиновете

Myservo.attach(9);

pinMode(stepPin, OUTPUT);

pinMode(dirPin, OUTPUT);

//Проверка за сензора

Serial.begin(9600);

if (tcs.begin())

{

Serial.println("Found sensor");

} else

{

Serial.println("No TCS34725 found ... check your connections");

while (1);

delay(100);

}

}

//Иницилизация на алгоритама

void initialization()

{

// Задаваме посока на въртене

digitalWrite(dirPin, HIGH);

// Стъпковият мотор прави 3 стъпки

for (int motorinit = 0; motorinit < 3; motorinit++)

{

digitalWrite(stepPin, HIGH);

delayMicroseconds(2000);

digitalWrite(stepPin, LOW);

delayMicroseconds(2000);

}

}

void loop()

{

//Снемаме стойностите на сензора

uint16\_t r, g, b, c, colorTemp, lux;

tcs.getRawData(&r, &g, &b, &c);

colorTemp = tcs.calculateColorTemperature(r, g, b);

lux = tcs.calculateLux(r, g, b);

//Следващата част от кода е за оператор да може да провери дали сезора работи правилно

Serial.print("Color Temp: "); Serial.print(colorTemp, DEC); Serial.print(" K - ");

Serial.print("Lux: "); Serial.print(lux, DEC); Serial.print(" - ");

Serial.print("R: "); Serial.print(r, DEC); Serial.print(" ");

Serial.print("G: "); Serial.print(g, DEC); Serial.print(" ");

Serial.print("B: "); Serial.print(b, DEC); Serial.print(" ");

Serial.print("C: "); Serial.print(c, DEC); Serial.print(" ");

Serial.println(" ");

delay(1000);

//Мотора прави 1 стъпка

digitalWrite(dirPin, HIGH);

digitalWrite(stepPin, HIGH);

delayMicroseconds(2000);

digitalWrite(stepPin, LOW);

delayMicroseconds(2000);

delay(1000);

//В зависимост от стойностите снимани от сензора завъртаме сервото

if (r > g && r > b)

{

pos=40;

Myservo.write(pos);

}

else if (g > r && g > b)

{

pos=80;

Myservo.write(pos);

}

else if (b > g && b > r)

{

pos=120;

Myservo.write(pos);

}

delay(1000);

}