Програмиране с Arduino и eduArdu

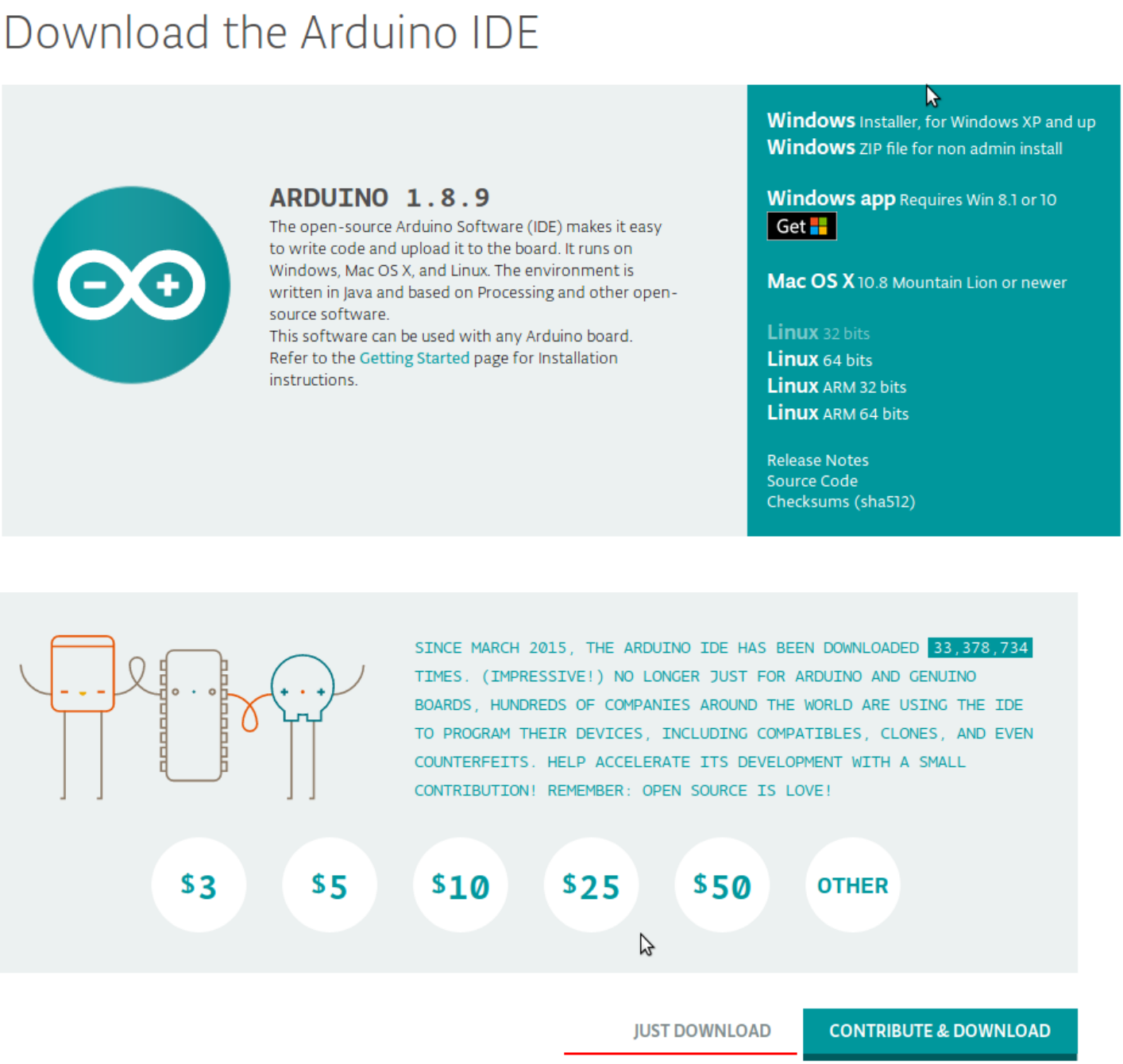
July 2019

Съдържание

1. Инсталиране на Arduino IDE

1.1 Използвайте линка за сваляне на Arduino IDE в зависимост от операционната система :

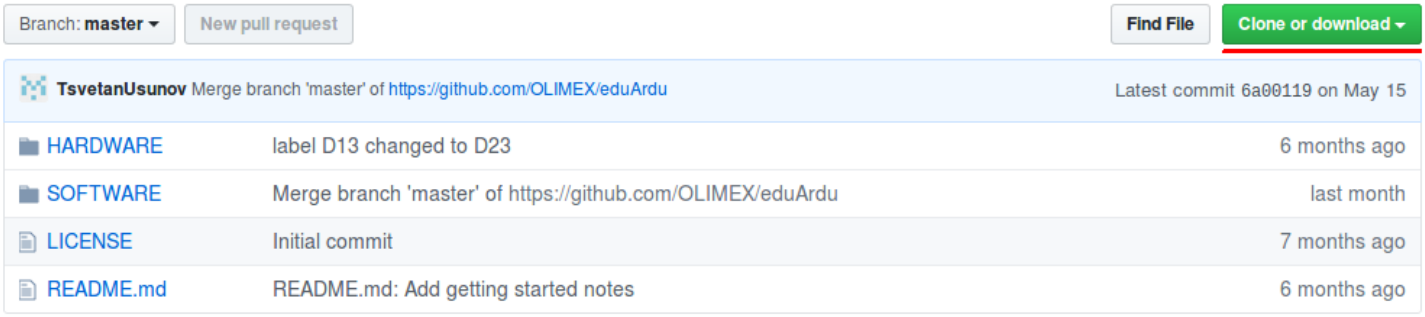
https://www.arduino.cc/en/Main/Software



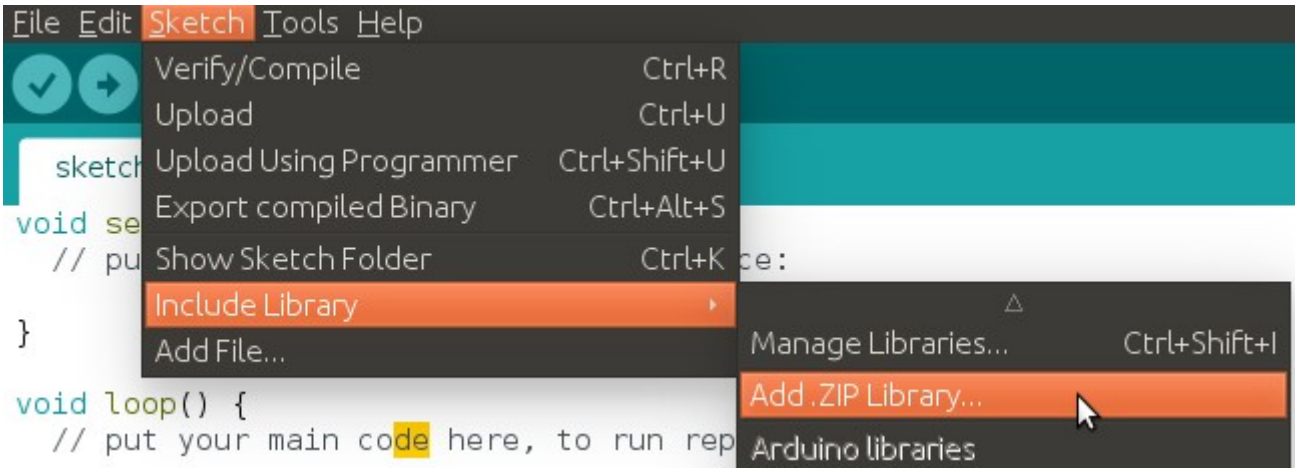
2.1 Инсталирайте Arduino IDE

3.1 Използвайте линка за да свалите всички примери и библиотеки необходими за eduArdu чрез натискането на подчертания бутон :

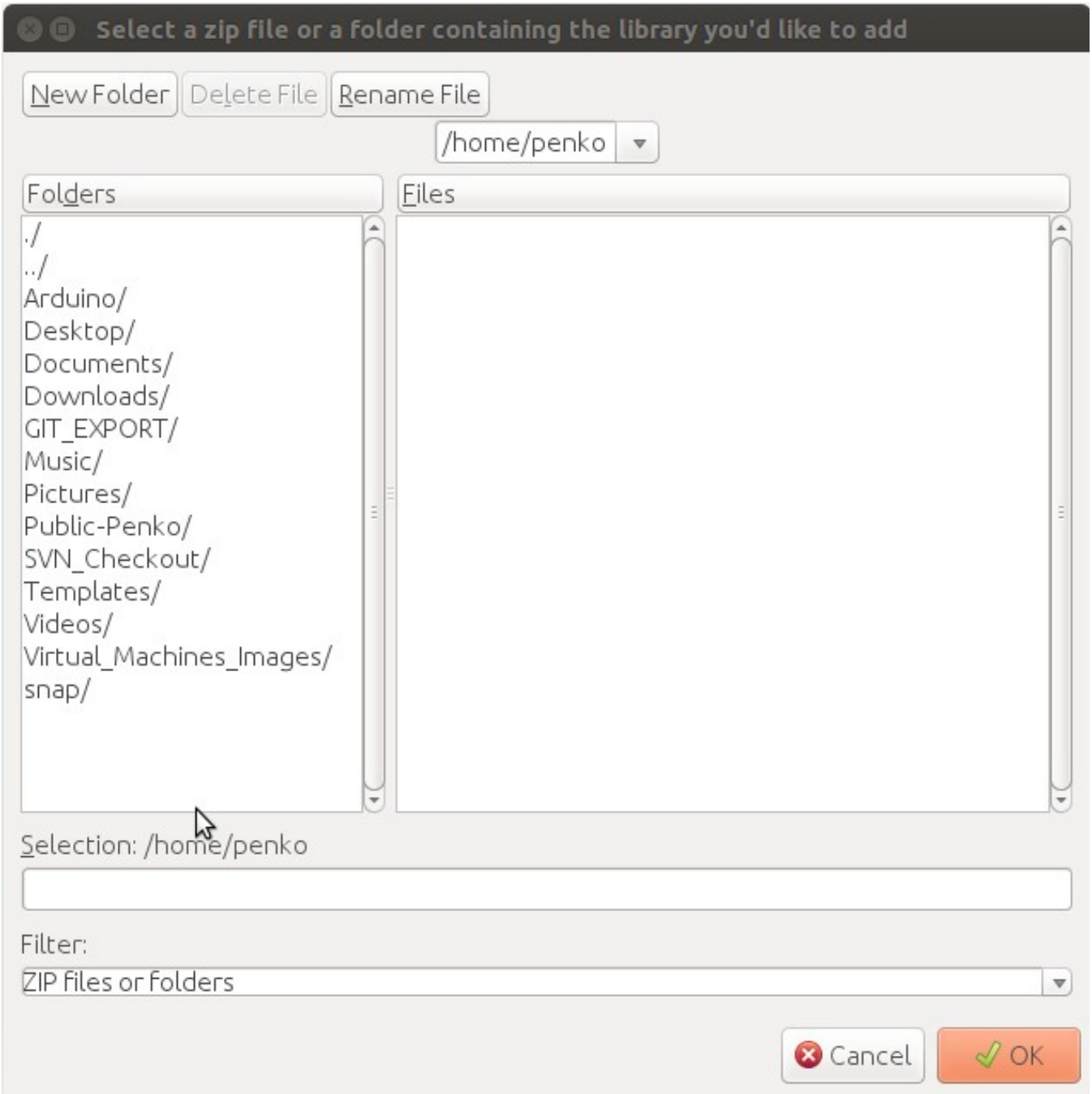
https://github.com/OLIMEX/eduArdu



4.1 Стартирайте Arduino IDE. Влезте в Sketch-include Library-Add .ZIP Library



5.1 Изберете папката в която сте запазили свареното от точка (3.) и започнете едно след друго да добавяте библиотеките, като следвате стъпки 4 и 5 . Файловете, който трябва да добавите ще се намират под Files а папката в която сте запализи файла от точка 3 ще се намира под Folders.



За eduArdu :

Платката EduArdu е предназначена за обучение. Тя е снабдена с множество сензори като например:

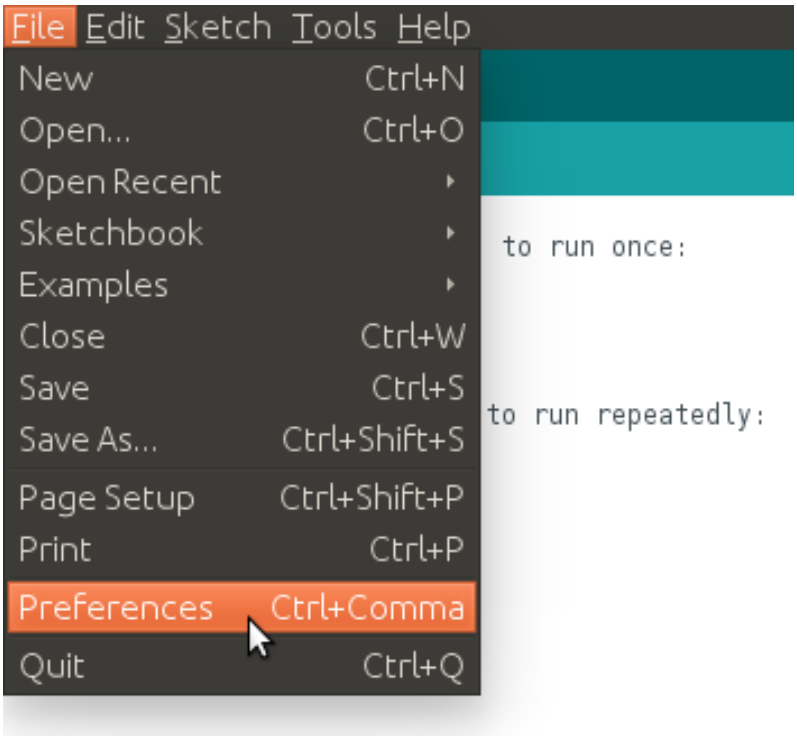
-осветление
-инфрачервен диод
-инфрачервен приемник
-сензор за движение
-сензор за температура

Платката разполага още и с:

-LED матрица
-микрофон
-трицветен светодиод
-два Servo входа
-джойстик
-UEXT вход
-Buzzer

2. Инсталиране на Olimex board support

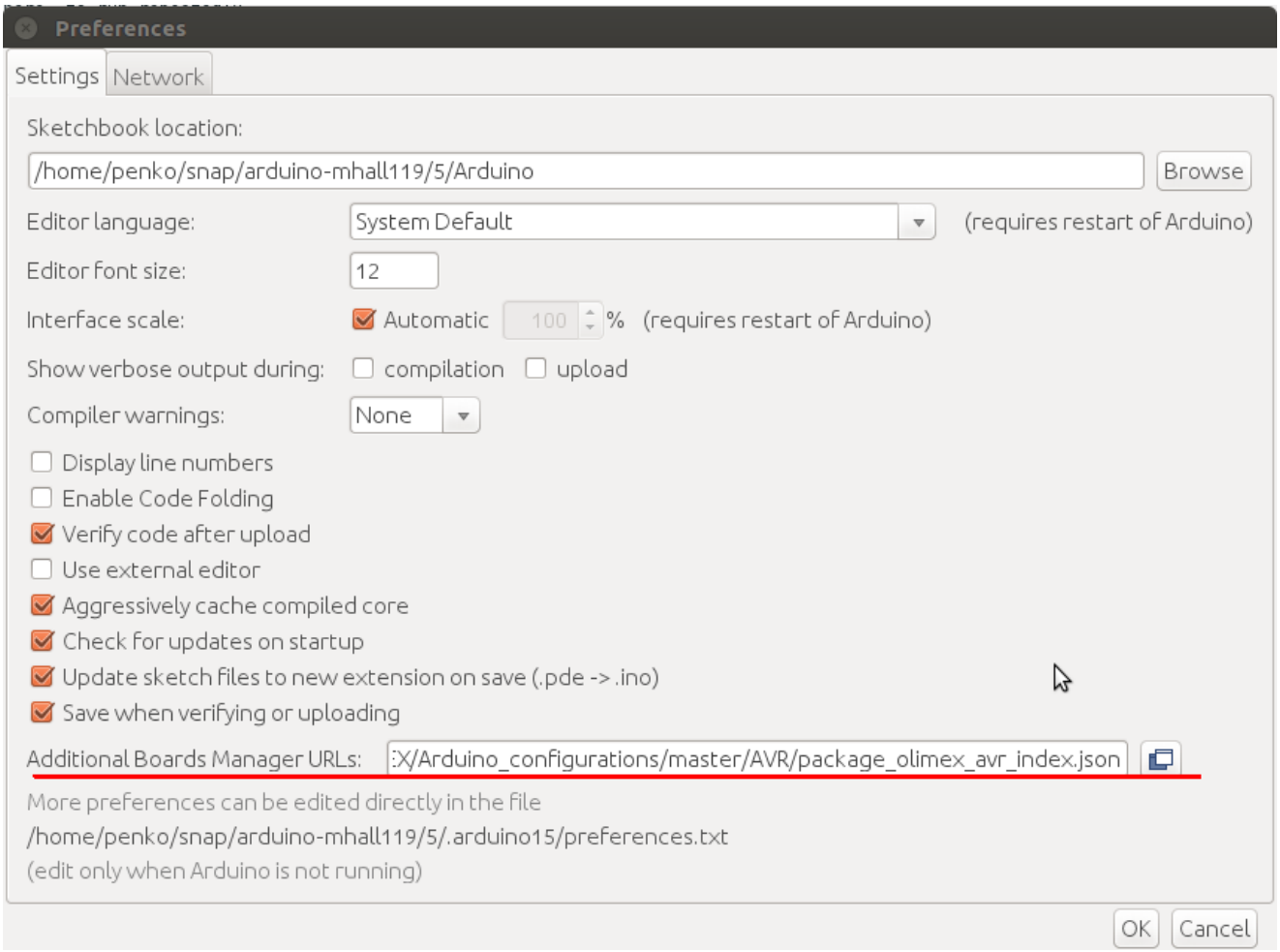
Влизате в file-preference.



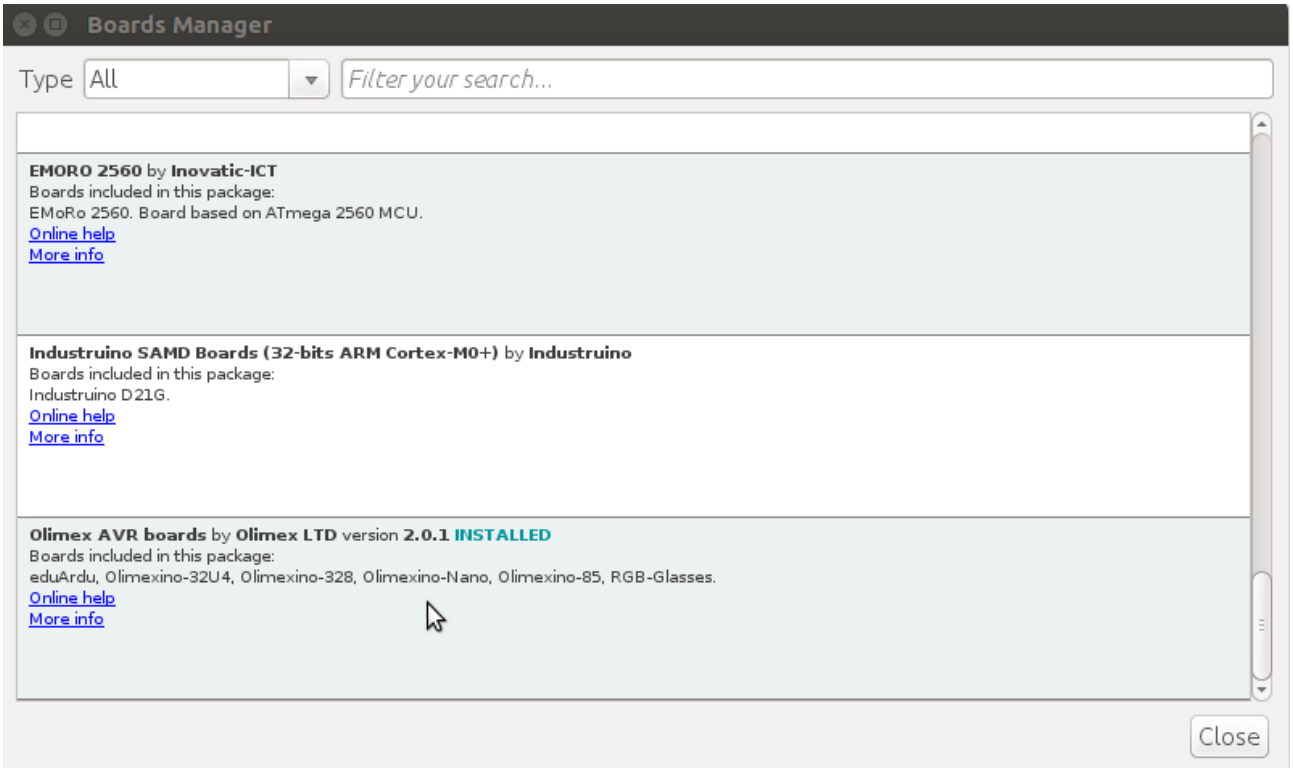
Копирате линка**:**

https://raw.githubusercontent.com/OLIMEX/Arduino_configurations/master/AVR/package_olimex_avr_index.json

След това го поставяте в Additional board manager URLs:

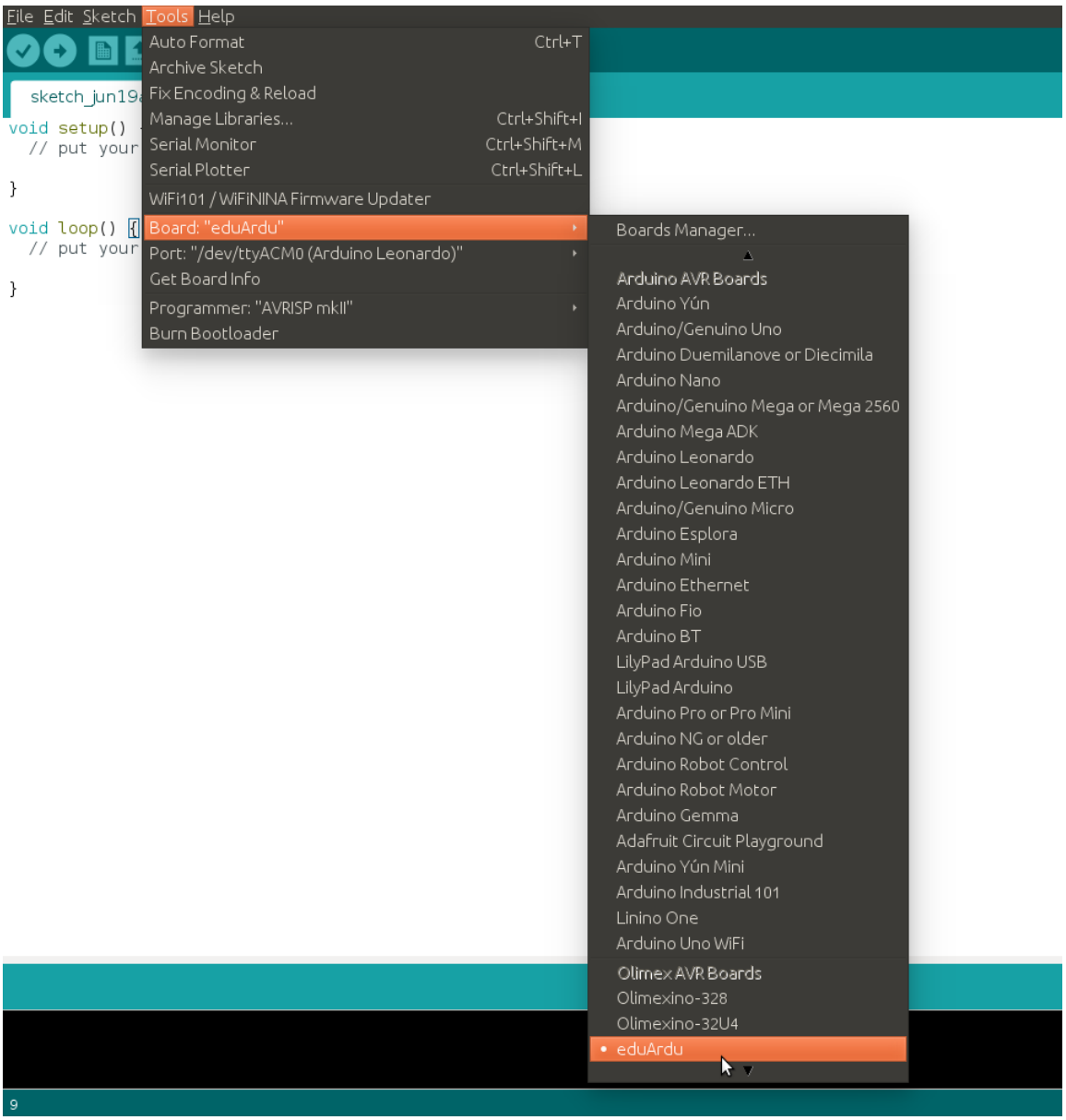


Влезте в tools-board-boards manager : и инсталирайте Olimex AVR boards.



3. Как да конфигурираме Arduino IDE

Първо: Свържете платката с USB към компютър. Втора стъпка:конфигурирайте платката като за целта трябва да влезнете в Tools-Board и изберете eduArdu. След това Port и изберете вашата платка.



4. LED Matrix

```
#include <string.h>

#include "LED_Matrix.h"         // Използваме LED_Matrix.h за да имаме достъп до SlideRight()
                                // и SlideLeft().
#include "font.h"               // Чрез  font.h инициализираме азбуката и всички знаци.
#include "Joystick.h"           // Инициализира джойстика.

#define JOYSTICK_X  A0          // Където срещне OYSTICK_X го заменя с аналогов вход A0.
#define JOYSTICK_Y  A1          // Където срещне OYSTICK_Y го заменя с аналогов вход A1.
#define JOYSTICK_BUTTON 31      // Където срещне  JOYSTICK_BUTTON го заменя с 31
#define SPEED 110               // Където срещне  SPEED го заменя с 110

#define LED_LATCH 11            // Инициализираме LED матрицата.
#define LED_DATA 16
#define LED_CLOCK 15

LED_Matrix Matrix(LED_LATCH,LED_DATA,LED_CLOCK);    // Задаваме стойност на функция Matrix
                                                    // която приема като параметри LED
                                                    // Matrix pin.

Joystick Joy(JOYSTICK_X, JOYSTICK_Y);               // Използваме функцията за да взимаме
                                                    // стойностите на X(лява/дясна граница) и на
                                                    // Y(горна/долна граница) на джойстика.

unsigned char Text[STRING_MAX_CHAR] = "Olimex eduArdu LED matrix example!";

/*
      Използва се за изписването на текста върху LED matrix.Пример ако искаме да изпишем
      “Hello Olimex” трябва да го променим така :
      unsigned char Text[STRING_MAX_CHAR] = "Hello Olimex";
*/

char Terminal_Output[256];                  // Използва се за изписване на резултатът в конзолата
                                            // като дължината на символите зависи от цифрата
                                            // която е в скобите “[ ]”.
void setup()
{
  Serial.begin (115200);                    // Използва се за инициализиране.Т.е скоростта на
                                            // трансфера на данните.
 Matrix.DisplayText (Text, 0);              // Използва се за инициализиране на текста като
                                            // втория параметър е за да може  текста да започне
}                                           // отначало.

static unsigned long Time=0, PrevTime=0;    // Задаваме стойности за променливи Time,PrevTim.

  Time = millis();                          // Записваме в Time, изтеклото време в милисекунди от
                                            // пускането на платката.

  if (Time-PrevTime > 110)                  // Изважда от Time променливата  PrevTime и провер
   {                                        // дали е по малко от 110.
    PrevTime = Time;                        // PrevTime взема стойността на  Time.

    if (Joy.X () < 20)                      // Проверява дали джойстика е натиснат наляво.
      Matrix.SlideLeft (1);                 // Ако е натиснат наляво. Функцията премества текста
                                            // наляво.

    if (Joy.X () > 80)                      // Проверява дали джойстика е натиснат надясно.
       Matrix.SlideRight (1);               // Ако е натиснат надясно.Функцията премества текста
                                            // надясно.

    if (Joy.Y () < 20)                      // Проверява дали джойстика е натиснат надолу. Ако е
                                            // натиснат преминава на другия ред и извиква
```

```
                                        // функцията която намалява светлината.
Matrix.ChangeBrightness (-10);

if (Joy.Y () > 80)                      // Проверява дали джойстика е натиснат нагоре. Ако е
                                        // натиснат преминава на другия ред и извиква
                                        // функцията която увеличава светлината.
Matrix.ChangeBrightness (10);
 Serial.println ("--------------------------------");      // Изписва на конзолава намиращите
                                                          // се символи между кавичките в
                                                          // скобите.

sprintf (Terminal_Output, "Joystick: X = %d%%; Y = %d%%; Button: %d", (int)Joy.X(), (int)Joy.Y(),
Joy.But());                             // Взима стойностите X,Y,Button и ги записва в
                                        // Terminal_Output

Serial.println (Terminal_Output);       //Изписва съдържанието на Terminal_Output.
   }
  Matrix.UpdateText ();                 //Актуализира матрицата.
}
```

5. Масиви

Това са последователни клетки от паметта в който можем да запазваме различни данни от различен тип. Пример за това е в проекта Led Matrix,използването на :

```
Unsigned char Text[STRING_MAX_CHAR] = "Olimex eduArdu LED matrix example!";
char Terminal_Output[256];
```

При тези масиви искаме да запишем думи. Пример ако искаме да сменим текста който се изписва на матрицата ще променим текста в кавичките :

```
unsigned char Text[STRING_MAX_CHAR] = "Hello Olimex";
```

така текста на матрицата ще се промени.

6. Функции и Проверки

Тук ще разгледаме някой от функциите и проверките в примера за Led Matrix.

6.1 Функции

```
void setup()
{
  Serial.begin (115200);
  Matrix.DisplayText (Text, 0);
}
```

За пример можем да вземем `Void Setup()` тази функция не връща като резултат нищо.Използва се за инициализиране на входове/изходи. В тялото(т.е. намиращото се в { }) на тази функция имаме `Serial.begin (115200);` което инициализира скоростта на трансфер на данните.
Функцията `Matrix.DisplayText (Text, 0);` изписва на екрана записания текст в масива Text и откъде да започне текста като ако приложим втория параметър `(Text, 0)` да не е `0` ще видим че текста се измества.

6.2 Проверки

Нека разгледаме

```
if (Time-PrevTime > SPEED)
  {
    PrevTime = Time;
    if (Joy.X () < 20)
      Matrix.SlideLeft (1);
    if (Joy.X () > 80)
      Matrix.SlideRight (1);

    if (Joy.Y () < 20)
      Matrix.ChangeBrightness (-10);
    if (Joy.Y () > 80)
      Matrix.ChangeBrightness (10);
}
```

Тук имаме проверки “ if () ” при нея можем да сложим условие в ( ), проверяваме,ако условието е изпълнено(вярно е) програмата преминава на следващия ред и го изпълнява.При първият `if (Time-PrevTime > SPEED)` проверяваме `Time-PrevTime` дали е по-голямо от SPEED където SPEED е зададено като статична стойност чрез `#define SPEED 110` намиращо се в началото на програмата.Пример за променянето на SPEED е :

```
#define SPEED 40
```

така след промяната на текста чрез натискане на джойстика,той ще се движи по-бързо.

Когато програмата навлезе в тялото на функцията :

```
{
     PrevTime = Time;
    if (Joy.X () < 20)
      Matrix.SlideLeft (1);
    if (Joy.X () > 80)
      Matrix.SlideRight (1);
    if (Joy.Y () < 20)
      Matrix.ChangeBrightness (-10);
    if (Joy.Y () > 80)
      Matrix.ChangeBrightness (10);
}
```

Тук отново имаме проверки от тип if ( ) при тях обаче проверяваме дали сме натиснали джойстика `if (Joy.X () < 20)` наляво ако условието в скобите ( ) е изпълнено, т.е. джойстика е натиснат,наляво програмата изпълнява `Matrix.SlideLeft (1);` .При изпълнението на тази функция можем да променим стойността с която текста преминава през матрицата :

`Matrix.SlideLeft (3)` стойността може да е произволна.

Същото при `Matrix.ChangeBrightness (-10);` може да променяме колко да се увеличава осветителността.

7. Цикли

Тук ще се запознаем с основните цикли. `For () , While() ,do While.`

7.1 Цикъл For ( )

```
for(int i=0; i<50; i++)
{
//тяло на цикъла
}
```

В този пример операндите, който приема цикъла са нулиране на применливата “`i`” `int i=0,`

приверка дали “`i`” е по-малко от 50 и увеличаване на “`i`” с едно. Програмата преминава през цикъла по следния начин :

1. Нулирането на “`i`” (`int i=0)`

2. Проверка дали “`i`” е по-малко от 50

3.Навлизане и изпълнение на тялото на цикъла

4.Увеличаване на “`i`” с едно

Пример за такъв цикъл ще дадем от програмата с използване на микрофона:

```
for(int i=0; i<50; i++)
  {
    int Sample;
    Sample = analogRead(MICROPHONE);
    if (Sample > Max)
      Max = Sample;
    if (Sample < Min)
      Min = Sample;
    if((Max-Min) > THRESHOLD)
    {
      digitalWrite (LED, !digitalRead(LED));
      delay(100);
      break;
    }
  }
```

За тялото на функцията и проверките в него, може да прочетете в глава 6.

7.2 Цикъл While ( )

Тук цикъла има само един параметър и това е условието :

```
While (“условие”)
{
//тяло на цикъла
}
```

При този цикъл докато условието е вярно ще се повтаря кода в тялото.

7.3 Цикъл do While ( )

Тук цикъла има пак един параметър и това е условието но тук тялото се изпълнява и след това се прави проверката :

```
do
{
//тяло на цикъла
}While (“условие”)
```

!! Внимание цикъла може да се изпълни поне един път.