Electronic Supplementary Information

Colour sensor integrated into a microcontroller platform as a reliable tool for measuring pH change in biochemistry applications

Ondřej Keresteš and Miroslav Pohanka

University of Defence
Military Faculty of Medicine
Department of Molecular Pathology and Biology
Hradec Kralove
Czech Republic

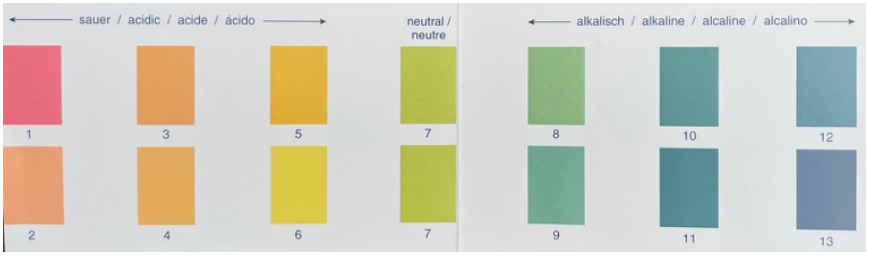


Figure S1: Overview of interval for colorimetry evaluation of pH with UNISOL 113 indicator with naked eye.

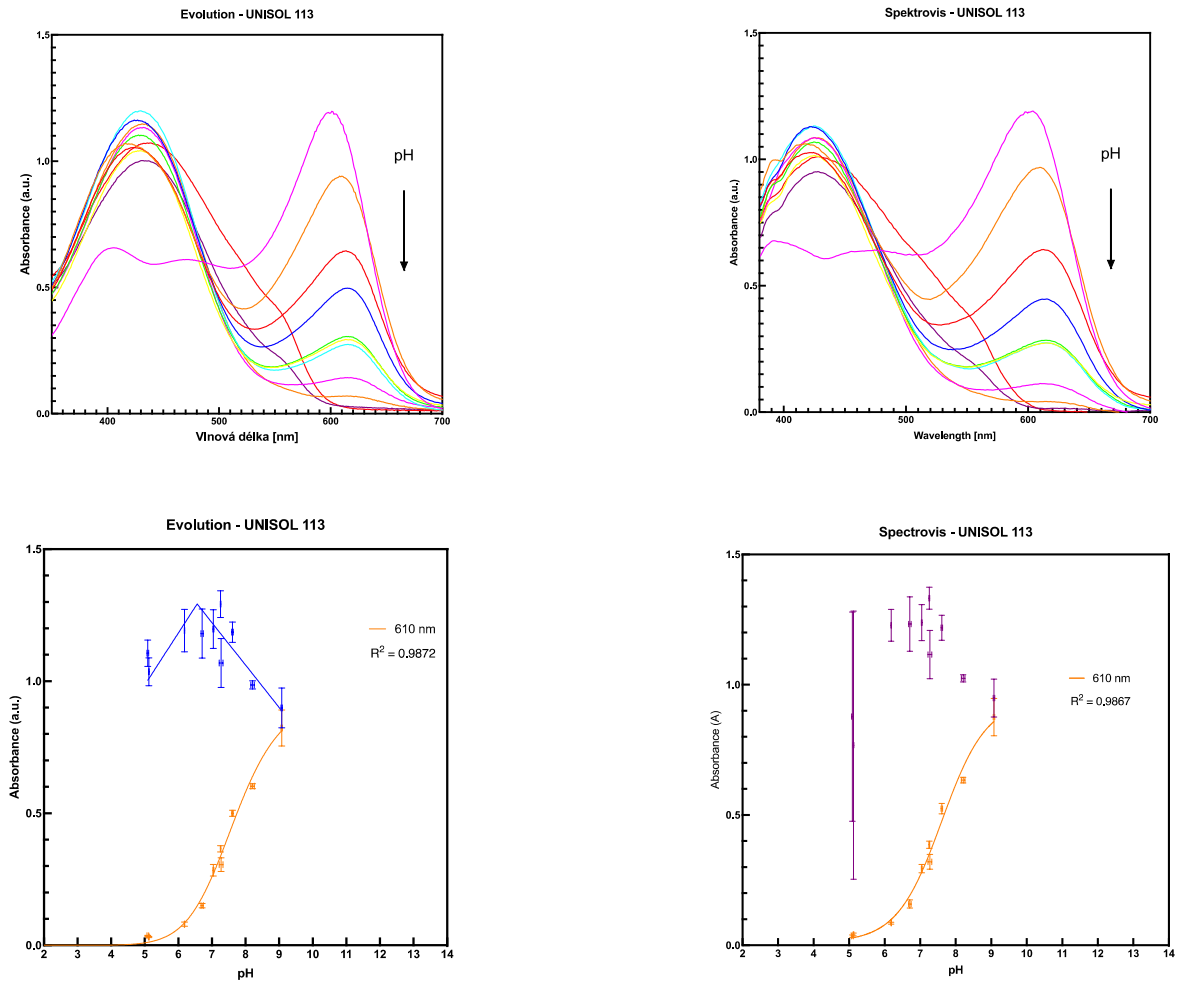
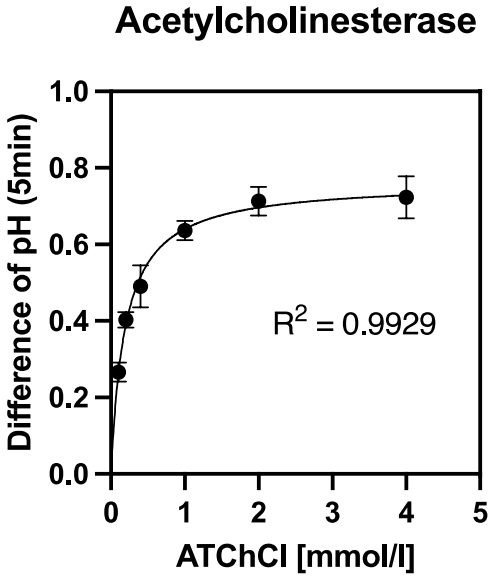
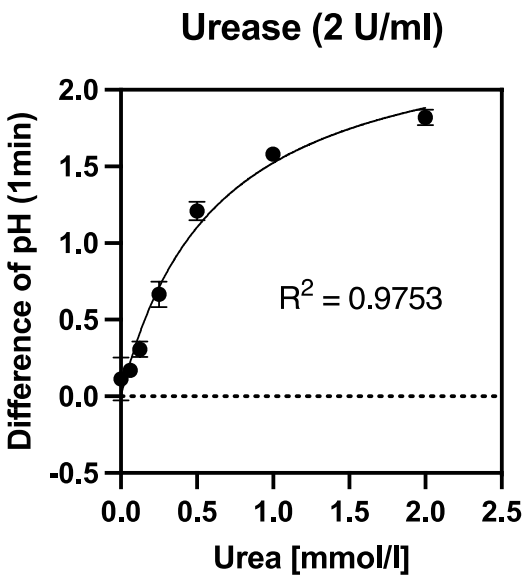


Figure S2: Measurement of UNISOL 113 with a benchtop spectrophotometer – absorption spectra of solutions with various pH.

S3) Potentiometric study conducted with a HACH potentiometer

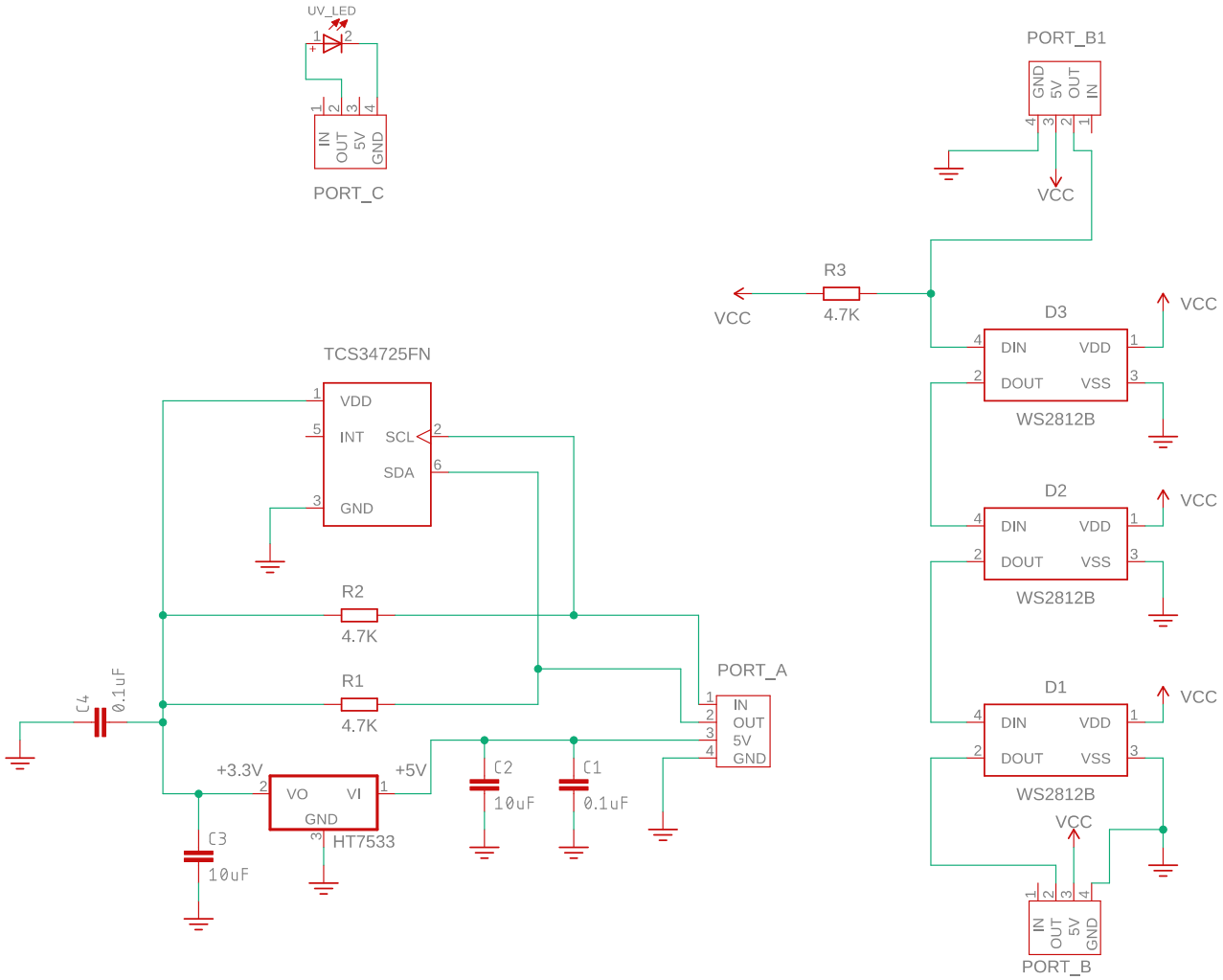
Acetylcholinesterase was mixed with the substrate in a of aqueous solution of natrium chloride (0.8%). The samples are identical to those used in the colorimetry study. The total volume of the mixture was 1500 uL. 37.5 uL of AChE and 50 uL of urease concentrates were mixed with substrate (concentration up to 20 mmol/l in cuvette. Reactions were observed for 5 min. The evaluation was carried out with a difference in pH after 5 min.





S4) Preparation of the sensor platform

a) Electronics



b) 3D printing

There are two parts of the systém which were 3D printed. Both were printed from rPLA with infill 20%, two perimeters and 0.2mm layer height. The temperature used for the process was 215 ° C for the nozzle and 60 ° C for the heatbed.

1. Colorimeter base

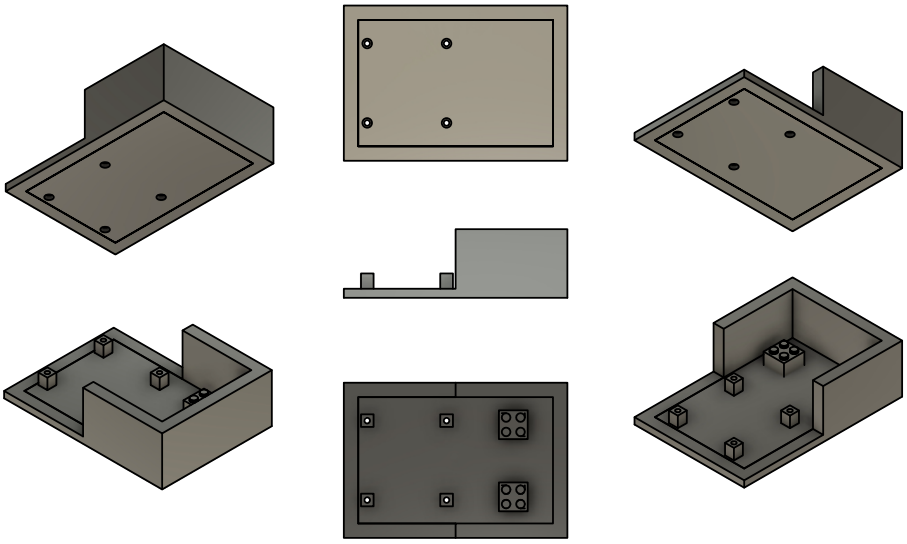


Figure S3-1: Overall Perspective View for Colorimeter Base

2. Colorimeter cuevette cell

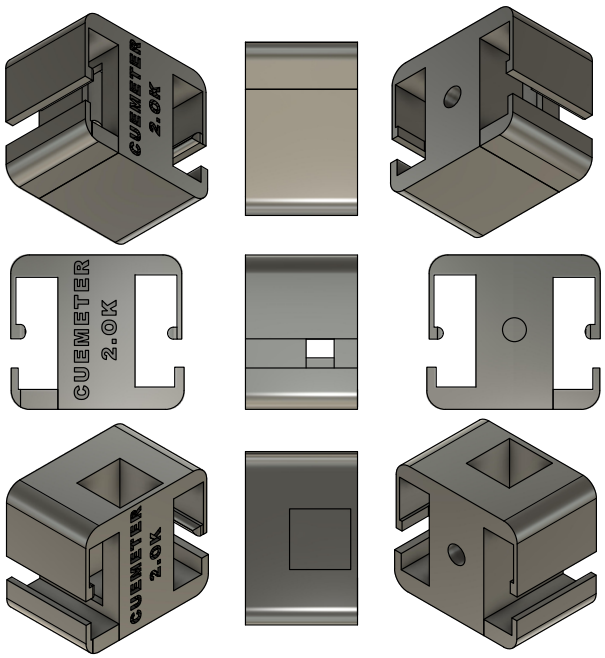
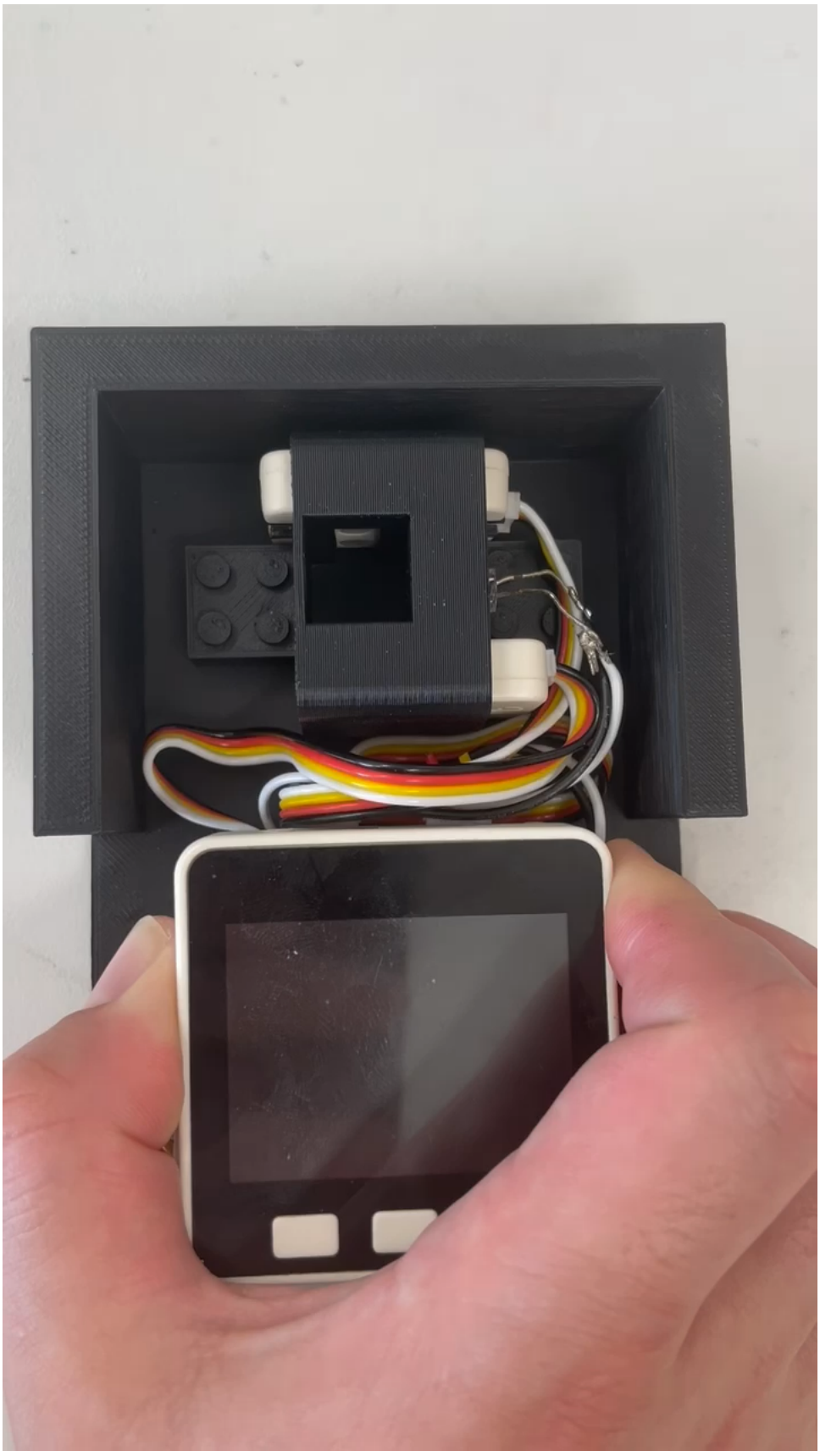


Figure S3-2: Overall perspective view for Colorimeter cuevette cell

S5) Video record of the operation procedure



S6) Code to programme the M5stack microcontroller through Arduino IDE

Firstly, the WavelenghtToOtherRGB must be placed in Arduino/libraries folder (usually in /Documents/Arduino folder)

```
/*Colorimeter based on M5stack GO, RGB LED module and Color sensing module
It offers the ability to change the color source mode, reset the time, switch the sample.

Programmed by Ondrej Kerestes
*/


#include <Adafruit_NeoPixel.h>
#include <M5Stack.h>
#include <WavelengthToOtherRGB.h>         //  functions to calculate the RGB values for the LED module
#include <Adafruit_TCS34725.h>

#include "SPI.h"


// Libraries for SD card
#include "FS.h"
#include "SD.h"




#define BUT_A   39
#define BUT_B   38
#define BUT_C   37
#define RGB_PIN 26
// počet RGB diod
#define NUMPIXELS 3



#define commonAnode true  //set to false if using a common cathode LED.  //如果使用普通阴极LED，则设置为false

byte gammatable[256]; // our RGB -> eye-recognized gamma color

WavelengthToOtherRGB rgb;

float zeroTime;
unsigned long zero = millis()-millis();

String dataMessage;
String startingMessage("time_ms,regime,Sample,imitated_wavelength,red,green,blue,clear");
int Fluoro_Source = 17;
int sample = 0;
int UV_Value;
```

```
int regime = 0;
const int chipSelect = 4;

static uint16_t color16(uint16_t r, uint16_t g, uint16_t b) {
  uint16_t _color;
  _color = (uint16_t)(r & 0xF8) << 8;
  _color |= (uint16_t)(g & 0xFC) << 3;
  _color |= (uint16_t)(b & 0xF8) >> 3;
  return _color;
}

unsigned long previousMillis=0;

Adafruit_TCS34725 tcs = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_154MS, TCS34725_GAIN_16X);

//supportive values for functions in libraries
double Gamma = 0.80;
int MaxIntensity;

//initial values of the wavelenghts
int wavelenghtAngle;


// vytvoření objekt pixels z knihovny
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, RGB_PIN, NEO_GRB + NEO_KHZ800);
// proměnná pro uložení délky zpoždění


void setup() {
  // zahájení komunikace s M5 stack



  M5.begin();
  // zahájení komunikace s napájecím čipem
  M5.Power.begin();

    // vypnutí reproduktoru
  dacDisable(25);
  // nastavení tlačítek jako vstupů
  pinMode(BUT_A, INPUT_PULLUP);
  pinMode(BUT_B, INPUT_PULLUP);
  pinMode(BUT_C, INPUT_PULLUP);


  pinMode(17, OUTPUT);

  tcs.setIntegrationTime(TCS34725_INTEGRATIONTIME_154MS); //Sets the integration time for the TC34725.
设置TC34725的集成时间
  tcs.setGain(TCS34725_GAIN_16X);  //Adjusts the gain on the TCS34725.  调整TCS34725上的增益
  // zahájení komunikace s RGB diodami
  pixels.begin();
```

```
  // výpis informace na displej
  Serial.begin(115200);

  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }
 if (!SD.begin(chipSelect)) {
    Serial.println("Card failed, or not present");
    // don't do anything more:
    M5.Lcd.setCursor(50, 150, 2);
    M5.Lcd.println("SD not present");

    while (1);
}
  Serial.println("card initialized.");
  M5.Lcd.setCursor(50, 150, 2);
  M5.Lcd.println("SD is present");


  M5.Lcd.setTextFont(3);
  M5.lcd.setTextSize(1);
  M5.Lcd.setCursor(40, 200, 2);
  M5.Lcd.println("(Regime)");

  M5.Lcd.setTextFont(3);
  M5.Lcd.setCursor(130, 200, 2);
  M5.Lcd.println("(Sample)");

  M5.Lcd.setTextFont(3);
  M5.Lcd.setCursor(220, 200, 2);
  M5.Lcd.println("(Time = 0)");

  File dataFile = SD.open("data.txt");
      appendFile(SD, "/data.txt", startingMessage.c_str());
      dataFile.close();




}

void loop() {


float currentTime = millis();


float sampleTime =  currentTime - zeroTime;


  uint16_t clear, red, green, blue;
```

```
 tcs.getRawData(&red, &green, &blue, &clear);  //Reads the raw red, green, blue and clear channel values.
读取原始的红、绿、蓝和清晰的通道值



 // Figure out some basic hex code for visualization.  生成对应的十六进制代码
 uint32_t sum = clear;
 float r, g, b;
 r = red; r /= sum;
 g = green; g /= sum;
 b = blue; b /= sum;
 r *= 256; g *= 256; b *= 256;
 uint16_t _color = color16((int)r, (int)g, (int)b);

// pomocí čtení stavu tlačítek
 if (digitalRead(BUT_A) == LOW) {
   regime += 1;

   MaxIntensity = 255;
   wavelenghtAngle = constrain(wavelenghtAngle,380,780);                          //  limits
   rgb.Convert(wavelenghtAngle, Gamma, MaxIntensity);
   UV_Value = 0;
   analogWrite(17,UV_Value);
   if (regime == 7) {
     regime = 1;
    }
 }

 else if (digitalRead(BUT_B) == LOW) {
   sample += 1;

 }

 else if (digitalRead(BUT_C) == LOW) {
   zeroTime = millis();

   }



 analogWrite(17,UV_Value);                          //  limits
 M5.lcd.setCursor(0,0);  //Place the cursor at (0,20).  将光标固定在(0,20)
 M5.lcd.fillRect(0,0,150,100,BLACK); //Fill the screen with a black rectangle.  将屏幕填充黑色矩形

 M5.Lcd.print("Sample: ");
 M5.Lcd.println(sample);
 M5.Lcd.print("C:"); M5.Lcd.println(clear);
 M5.Lcd.print("R:"); M5.Lcd.println(red);
 M5.Lcd.print("G:"); M5.Lcd.println(green);
 M5.Lcd.print("B:"); M5.Lcd.println(blue);
 M5.Lcd.print("Wavelenght [nm] = ");
```

```
  M5.Lcd.println(wavelenghtAngle);
  M5.Lcd.print("Time: ");
  M5.Lcd.println(sampleTime/1000);
  M5.Lcd.print("Regime: ");
  M5.Lcd.println(regime);


dataMessage = String(sampleTime/1000) + "," + String(regime) + "," + String(sample) + "," + String(wavelenghtAngle) + ","
+ String(red) + "," + String(green) + "," + String(blue) + "," + String(clear) + "\r\n";

Serial.print(dataMessage);
appendFile(SD, "/data.txt", dataMessage.c_str());









if (regime == 1) {
  wavelenghtAngle = 412;
  wavelenghtAngle = constrain(wavelenghtAngle,380,780);                    //  limits
  rgb.Convert(wavelenghtAngle, Gamma, MaxIntensity);
  pixels.setPixelColor(0, pixels.Color(rgb.GetOtherRed(), rgb.GetOtherGreen(), rgb.GetOtherBlue()));
  pixels.setPixelColor(1, pixels.Color(rgb.GetOtherRed(), rgb.GetOtherGreen(), rgb.GetOtherBlue()));
  pixels.setPixelColor(2, pixels.Color(rgb.GetOtherRed(), rgb.GetOtherGreen(), rgb.GetOtherBlue()));
  pixels.show();
  UV_Value = 0;
}

if (regime == 2) {
  wavelenghtAngle = 435;
  wavelenghtAngle = constrain(wavelenghtAngle,380,780);                    //  limits
  rgb.Convert(wavelenghtAngle, Gamma, MaxIntensity);
  pixels.setPixelColor(0, pixels.Color(rgb.GetOtherRed(), rgb.GetOtherGreen(), rgb.GetOtherBlue()));
  pixels.setPixelColor(1, pixels.Color(rgb.GetOtherRed(), rgb.GetOtherGreen(), rgb.GetOtherBlue()));
  pixels.setPixelColor(2, pixels.Color(rgb.GetOtherRed(), rgb.GetOtherGreen(), rgb.GetOtherBlue()));
  pixels.show();
  UV_Value = 0;
}

if (regime == 3) {
  wavelenghtAngle = 535;
  wavelenghtAngle = constrain(wavelenghtAngle,380,780);                    //  limits
  rgb.Convert(wavelenghtAngle, Gamma, MaxIntensity);
  pixels.setPixelColor(0, pixels.Color(rgb.GetOtherRed(), rgb.GetOtherGreen(), rgb.GetOtherBlue()));
  pixels.setPixelColor(1, pixels.Color(rgb.GetOtherRed(), rgb.GetOtherGreen(), rgb.GetOtherBlue()));
  pixels.setPixelColor(2, pixels.Color(rgb.GetOtherRed(), rgb.GetOtherGreen(), rgb.GetOtherBlue()));
  pixels.show();
  UV_Value = 0;
```

```
}

if (regime == 4) {
  wavelenghtAngle = 610;
  wavelenghtAngle = constrain(wavelenghtAngle,380,780);
  analogWrite(17,UV_Value);                          //  limits
  rgb.Convert(wavelenghtAngle, Gamma, MaxIntensity);
  pixels.setPixelColor(0, pixels.Color(rgb.GetOtherRed(), rgb.GetOtherGreen(), rgb.GetOtherBlue()));
  pixels.setPixelColor(1, pixels.Color(rgb.GetOtherRed(), rgb.GetOtherGreen(), rgb.GetOtherBlue()));
  pixels.setPixelColor(2, pixels.Color(rgb.GetOtherRed(), rgb.GetOtherGreen(), rgb.GetOtherBlue()));
  pixels.show();
  UV_Value = 0;
}
if (regime == 5) {
  pixels.setPixelColor(0, pixels.Color(255,255,255));
  pixels.setPixelColor(1, pixels.Color(255,255,255));
  pixels.setPixelColor(2, pixels.Color(255,255,255));
  pixels.show();
  UV_Value = 0;
}
if (regime == 6) {
    UV_Value = 255;
    pixels.setPixelColor(0, pixels.Color(0,0,0));
    pixels.setPixelColor(1, pixels.Color(0,0,0));
    pixels.setPixelColor(2, pixels.Color(0,0,0));
    pixels.show();







}

}
// Write to the SD card (DON'T MODIFY THIS FUNCTION)
void writeFile(fs::FS &fs, const char * path, const char * message) {


  File dataFile = fs.open(path, FILE_WRITE);

  dataFile.close();
}
void appendFile(fs::FS &fs, const char * path, const char * message) {

  File dataFile = fs.open(path, FILE_APPEND);
  dataFile.close();
}
```