СРЕДНО УЧИЛИЩЕ “ЙОРДАН ЙОВКОВ”
гр. Сливен, кв. “Българка”, тел.: 044/66 72 44, 044/66 74 68, факс 044/66 76 84
e-mail: info-2000114@edu.mon.bg

Доклад по проект

„EyeMotions“

(Интерактивни очи с емоции и анимации)

Изготвил: Габриела Стефанова
Клас: 10А / Специалност: „Приложен програмист “
Ръководител: инж. Димитър Желев

Дата: 17.06.2025 г.

Съдържание

1. Въведение

Проектът EyeMotions представлява дисплей, върху който се изобразяват анимирани очи, предаващи различни настроения. Управлението на емоциите става чрез бутони, което позволява взаимодействието между потребителя и устройството. Проектът съчетава знания по програмиране и електроника, демонстрирайки приложението на микроконтролери в създаването на визуални ефекти.

2. Цел на проекта

Целта на проекта е демонстрация на софтуерни и хардуерни умения, чрез създаването на анимирани очи, показващи различни емоции на дисплей. Управлението става с бутони, чрез които се сменят пет основни настроения.

3. Използвани компоненти

Проектът EyeMotions използва следните хардуерни компоненти:

- Arduino Nano – микроконтролерът, който управлява всички функции на устройството и обработва входните сигнали на бутоните.
- 3.66 см 128x128 SPI TFT LCD дисплей – визуализира анимираните очи и промяната на емоциите
- Тактови бутони – използват се за смяна на настроенията на очите, като позволяват управление от потребителя
- Breadboard и жички – свързват всички компоненти
- USB кабел – захранва устройството

Всички компоненти са свързани жично с breadboard, което позволява лесно тестване и промени без необходимост от запояване.

4. Софтуерна реализация

Софтуерната част на проекта е реализирана с помощта на езика **C++ за Arduino**. Основната цел на програмата е да управлява **TFT дисплея**, като реагира на **натискането на бутони**, чрез които се сменят различни емоции на очите. Кодът включва **функции за рисуване на очите**, **анимации при мигане** и **изразителни движения на ирисите**, което придава по-реалистичен ефект на емоциите.

Използвани са библиотеките **Adafruit_GFX** и **Adafruit_ST7735**, които улесняват работата с дисплея и предоставят функции за графични елементи.

В основния цикъл се извършват следните основни действия:

- Следи се състоянието на **бутоните** *чрез digitalRead().* Когато бутон бъде **натиснат (LOW)**, чрез **if-else проверка** се активира съответната емоция.
- Сменят се **анимациите** – радост, ярост, тъга, изненада и обърканост.
- За да се реализира **плавното мигане**, се използва функцията *millis()* вместо *delay().* Това позволява непрекъсната проверка на състоянието на бутоните, дори докато очите мигат.

5. Структура на кода и обяснение

Основни елементи:

- Дисплеят се инициализира чрез *Adafruit_ST7735 tft*, свързан към определени пинове на Arduino.
- Създадени са константи за позициите и радиусите на очите, ирисите и зениците.
- Определени са входовете за бутоните чрез *#define* и *pinMode().*

Главна логика *(в loop()):*

- За всеки бутон е реализирана проверка чрез *digitalRead()*. При натискане, текущото настроение (*currentMood*) се променя и се извиква функцията *drawFace()*, която очертава цялото лице.

Анимации *(в updateAnimation()):*

- Използва се *millis()* за засичане на време, за да се активира мигане на определен интервал (напр. 4 секунди).
- Ирисите се местят леко в произволна посока на всеки 1.5 секунди, за да изглежда така, сакаш очите се "оглеждат".

Рисуване:

- *drawFace()* изчиства екрана и рисува очите, веждите, устата и допълнителни аксесоари в зависимост от емоцията.
- *drawEye()* използва функции като *fillCircle()* за рисуване на очна ябълка, ирис и зеница.
- *drawEyebrows()* и *drawCurvedBrow()* използват *drawLine()* за изразяване на различни емоции чрез вежди (например повдигнати за радост, наклонени за сънливост, зигзагообразни за обърканост).
- *drawMouth()* визуализира различни устни изражения според настроението: усмивка, изненада (кръгла уста), объркване (зигзаг).
- *drawAccessories()* добавя визуални елементи като капка пот, вена или въпросителен знак при нужда.

5. Хардуерна реализация

За хардуерната част на проекта е използван **Arduino Nano**, свързан към **TFT дисплей** и **пет бутона** чрез **breadboard**. Всички компоненти са свързани с жички, като се използват **дигитални пинове на Arduino**, които изпращат управляващи сигнали към дисплея и бутоните.

Дисплеят е свързан чрез **SPI комуникация** с използване на пиновете **CS, DC, RESET, MOSI и SCK**.

SPI (Serial Peripheral Interface) е бърз сериен комуникационен протокол, използван за обмен на данни между микроконтролера и периферни устройства (в случая TFT дисплей). SPI използва няколко линии за комуникация:

- **MOSI (Master Out Slave In) – D11:** линия за предаване на данни от Arduino (master) към дисплея (slave).
- **SCK (Serial Clock) – D13:** синхронизира трансфера на данни, като подава тактови импулси.
- **CS (Chip Select) – D10:** активира конкретното периферно устройство (в случая – дисплея), за да започне комуникация.
- **DC (Data/Command) – D9:** указва дали изпратената информация е данни за показване или команден сигнал.
- **RESET – D8:** използва се за нулиране на дисплея при инициализация.
- **VCC и GND:** за захранване на дисплея.

Използвана е **SPI** комуникация, тъй като:

- Предоставя **по-висока скорост** в сравнение с I²C.
- Позволява **надеждно управление** на графични дисплеи с висока честота на опресняване.
- Използва по-малко ресурси и предоставя по-голяма **гъвкавост в управлението на анимации.**

Захранването на Arduino се осигурява чрез **USB кабел**. Разположението на компонентите върху breadboard позволява лесно свързване, тестване и модификация на хардуера.

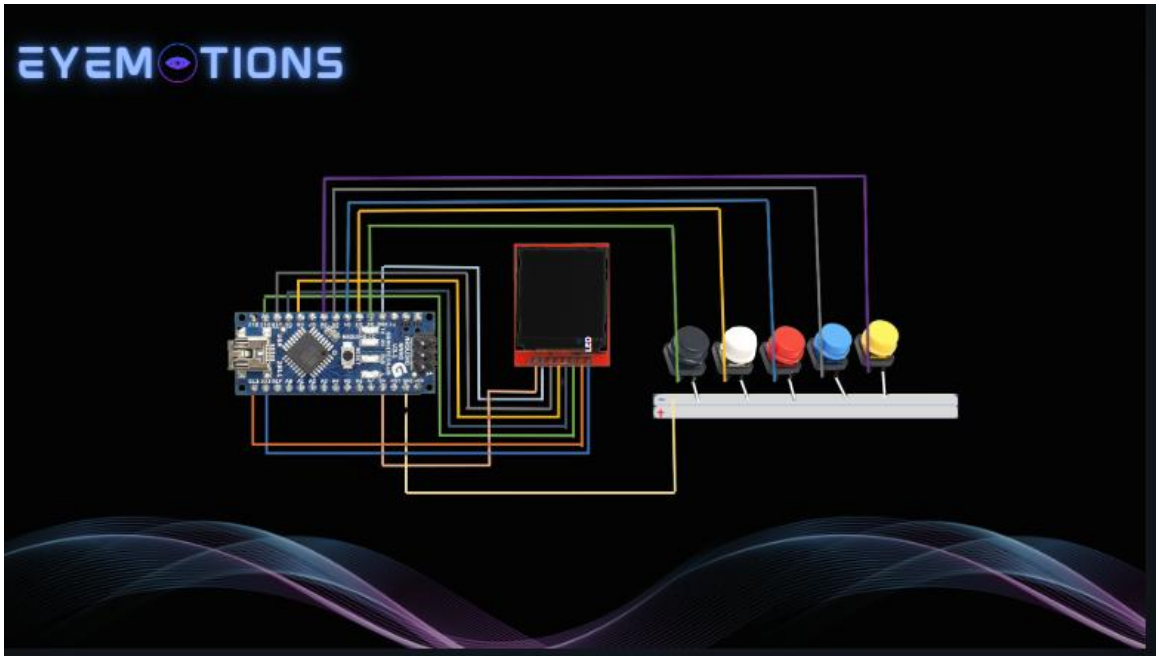
1. Свързване на TFT дисплей към Arduino Nano

TFT Пин	Arduino Nano Пин	Функция
VCC	5V	Захранване
GND	GND	Земя
CS	D10	Chip Select (SPI)
RESET	D8	Нулиране (Reset)
DC (A0)	D9	Data/Command
SDA (MOSI)	D11	SPI Data
SCK	D13	SPI Clock
LED	3.3V	Подсветка на дисплея

2. Свързване на бутоните към Arduino Nano

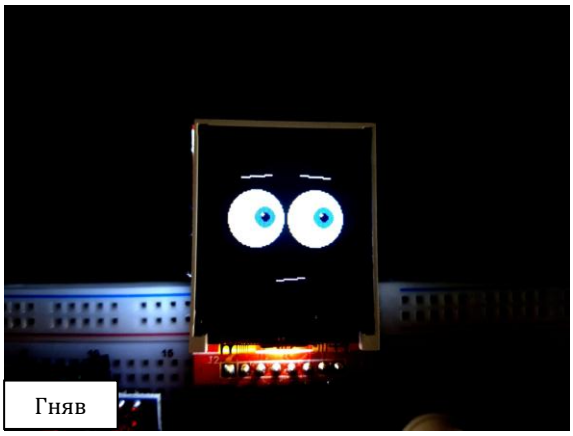
Цвят на бутона	Пин на Arduino	Емоция
Бял	D2	Happy (Щастлив)
Червен	D3	Angry (Ядосан)
Син	D4	Surprised (Изненадан)
Черен	D5	Dreamy (Замислен)
Жълт	D6	Confused (Объркан)

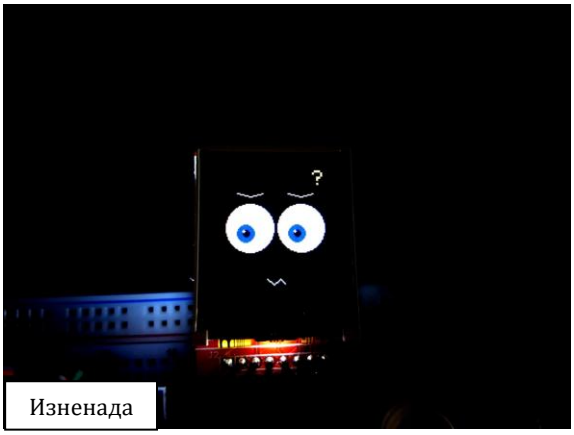
3. Електрическа схема

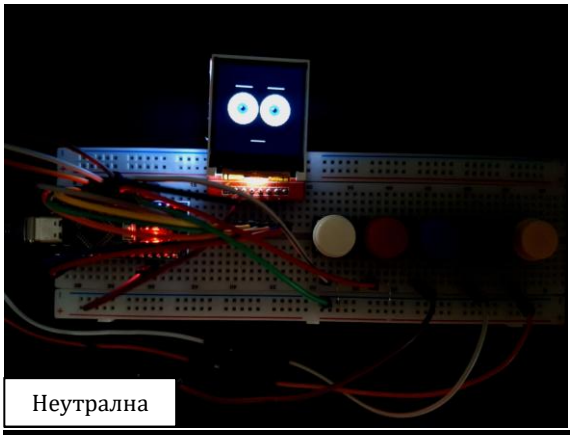


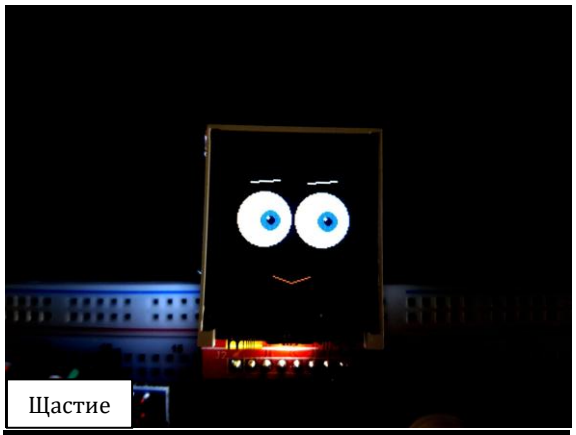
6. Дизайн и конструкция

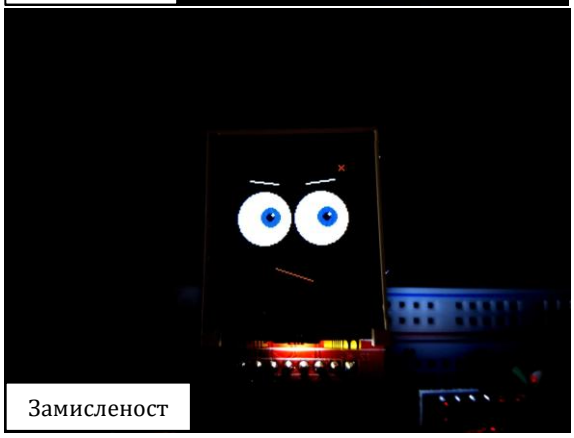
Проектът няма специален корпус, тъй като е създаден за образователна демонстрация и експериментиране. Компонентите са разположени удобно на breadboard, което осигурява лесен достъп до тях. Графичният дизайн на очите е изчистен, с акцент върху ясно различимите емоции, които са показани чрез формата и положението на ирисите, веждите и устата. Анимациите допринасят за по-естествен и динамичен вид.

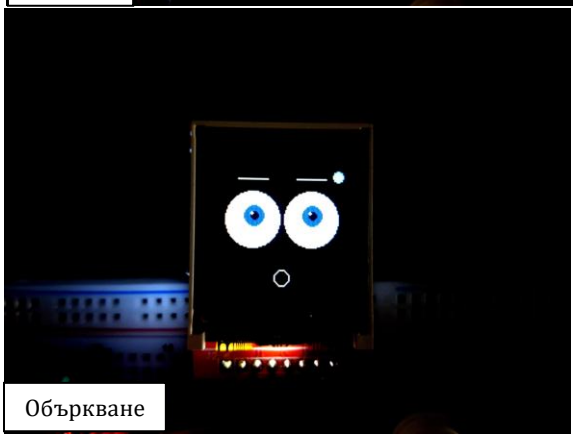

Гняв


Изненада


Неутрална


Щастие


Замисленост


Объркване

7. Предизвикателства и решения

По време на разработката се сблъсках с няколко предизвикателства:

- Плавното мигване на очите без да се спира основният цикъл на програмата, което беше решено чрез използването на функцията millis() за управление на времето.
- Създаването на реалистични изражения чрез прецизно позициониране на елементите на лицето, което изискваше експериментиране с координатите и размерите.

8. Идеи за развитие

В бъдеще проектът може да се разшири по следните начини:

- Добавяне на повече емоции и по-сложни анимации за по-голямо разнообразие.
- Интеграция на сензори за разпознаване на лица или емоции, които автоматично да променят изражението на очите.
- Добавяне на звук или говор, които да съпровождат визуалните изражения.
- Вграждане на Wi-Fi или Bluetooth модул за дистанционно управление и свързване с мобилно приложение.

9. Заключение

Проектът EyeMotions успешно съчетава хардуерни и софтуерни умения за създаване на интерактивно и визуално устройство. Той демонстрира възможностите на микроконтролерите в реализирането на анимации и взаимодействие с потребителя. Работата по този проект ми позволи да затвърдя знанията си по програмиране, електроника и дизайн, като същевременно развих умения за решаване на технически предизвикателства. EyeMotions има потенциал за бъдещо разширяване и внедряване на нови функционалности.

10. Източници

1. Adafruit GFX Library Documentation – https://learn.adafruit.com/adafruit-gfx-graphics-library
2. Adafruit ST7735 Library – https://github.com/adafruit/Adafruit-ST7735-Library
3. Arduino Nano Datasheet – https://store.arduino.cc/products/arduino-nano
4. SPI Communication Protocol – https://www.arduino.cc/en/Reference/SPI
5. Electronics Tutorials: SPI Interface – https://www.electronics-tutorials.ws/io/io_4.html

11. Приложение – Изходен код

По-долу е представен изходният код, използван за реализиране на устройството с TFT дисплей и бутони, което визуализира анимирани очи с различни емоции. Кодът е написан на езика C++ за платформата Arduino и използва библиотеките Adafruit GFX и Adafruit ST7735 за управление на дисплея.

```
#include <Adafruit_GFX.h>
#include <Adafruit_ST7735.h>
#include <SPI.h>

#define TFT_CS     10
#define TFT_RST    8
#define TFT_DC     9

Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_RST);

// Очни параметри
int eyeLX = 40, eyeRX = 88, eyeY = 64;
#define EYE_RADIUS 22
#define IRIS_RADIUS 8
#define PUPIL_RADIUS 3

#define BUTTON_HAPPY     2
#define BUTTON_ANGRY     3
#define BUTTON_SURPRISED 4
#define BUTTON_DREAMY    5
#define BUTTON_CONFUSED  6
#define BUTTON_NEUTRAL   7

String currentMood = "neutral";

// Анимации
unsigned long lastBlinkTime = 0;
unsigned long lastEyeMoveTime = 0;
bool isBlinking = false;
int blinkStep = 0;
int irisOffsetX = 0, irisOffsetY = 0;

void setup() {
  tft.initR(INITR_144GREENTAB);
  tft.fillScreen(ST77XX_BLACK);
```

```
  randomSeed(analogRead(A0));

  pinMode(BUTTON_HAPPY, INPUT_PULLUP);
  pinMode(BUTTON_ANGRY, INPUT_PULLUP);
  pinMode(BUTTON_SURPRISED, INPUT_PULLUP);
  pinMode(BUTTON_DREAMY, INPUT_PULLUP);
  pinMode(BUTTON_CONFUSED, INPUT_PULLUP);
  pinMode(BUTTON_NEUTRAL, INPUT_PULLUP);

  drawFace(currentMood);
}

void loop() {
  if (digitalRead(BUTTON_HAPPY) == LOW) {
    currentMood = "happy";
    drawFace(currentMood);
    delay(300);
  }
  else if (digitalRead(BUTTON_ANGRY) == LOW) {
    currentMood = "angry";
    drawFace(currentMood);
    delay(300);
  }
  else if (digitalRead(BUTTON_SURPRISED) == LOW) {
    currentMood = "surprised";
    drawFace(currentMood);
    delay(300);
  }
  else if (digitalRead(BUTTON_DREAMY) == LOW) {
    currentMood = "dreamy";
    drawFace(currentMood);
    delay(300);
  }
  else if (digitalRead(BUTTON_CONFUSED) == LOW) {
    currentMood = "confused";
    drawFace(currentMood);
    delay(300);
  }
  else if (digitalRead(BUTTON_NEUTRAL) == LOW) {
    currentMood = "neutral";
    drawFace(currentMood);
    delay(300);
  }

  updateAnimation();
```

```
}

// --------------------- Анимации -------------------------

void updateAnimation() {
  unsigned long now = millis();

  // Мигане на всеки 4 сек
  if (now - lastBlinkTime > 4000 && !isBlinking) {
    isBlinking = true;
    blinkStep = 0;
    lastBlinkTime = now;
  }

  if (isBlinking) blinkEyes();

  // Движение на ирисите
  if (now - lastEyeMoveTime > 1500) {
    irisOffsetX = random(-2, 3);
    irisOffsetY = random(-2, 3);
    drawFace(currentMood);
    lastEyeMoveTime = now;
  }
}

void blinkEyes() {
  if (blinkStep == 0) {
    // Затваряне
    tft.fillRect(eyeLX - EYE_RADIUS, eyeY - EYE_RADIUS, EYE_RADIUS *
2, EYE_RADIUS * 2, ST77XX_BLACK);
    tft.fillRect(eyeRX - EYE_RADIUS, eyeY - EYE_RADIUS, EYE_RADIUS *
2, EYE_RADIUS * 2, ST77XX_BLACK);
    blinkStep = 1;
    delay(100);
  } else if (blinkStep == 1) {
    // Отваряне
    drawFace(currentMood);
    isBlinking = false;
  }
}

// --------------------- Рисуване -------------------------

void drawFace(String mood) {
  tft.fillScreen(ST77XX_BLACK);
```

```
  drawEye(eyeLX, eyeY, mood);
  drawEye(eyeRX, eyeY, mood);
  drawEyebrows(mood);
  drawMouth(mood);
  drawAccessories(mood);
}

void drawEye(int cx, int cy, String mood) {
  int irisX = cx + irisOffsetX, irisY = cy + irisOffsetY;
  int eyelidOffset = 0;

  if (mood == "happy")      { irisX += 3; irisY += 3; eyelidOffset =
-2; }
  else if (mood == "angry") { irisX -= 4; irisY -= 2; }
  else if (mood == "surprised") { irisY -= 3; }
  else if (mood == "dreamy") { irisX += 5; irisY -= 3; }
  else if (mood == "confused") { irisX -= 4; irisY += 4; }

  tft.fillCircle(cx, cy, EYE_RADIUS, ST77XX_WHITE);
  tft.fillCircle(irisX, irisY, IRIS_RADIUS, ST77XX_BLUE);
  tft.fillCircle(irisX, irisY, PUPIL_RADIUS, ST77XX_BLACK);
  tft.fillCircle(irisX - 2, irisY - 2, 1, ST77XX_WHITE); // отблясък
}

void drawEyebrows(String mood) {
  int h = eyeY - EYE_RADIUS - 10;
  drawCurvedBrow(eyeLX, h, getBrowType(mood, true));
  drawCurvedBrow(eyeRX, h, getBrowType(mood, false));
}

String getBrowType(String mood, bool left) {
  if (mood == "happy") return "up";
  if (mood == "angry") return left ? "downL" : "downR";
  if (mood == "dreamy") return left ? "tiltL" : "tiltR";
  if (mood == "confused") return "zigzag";
  return "flat";
}

void drawCurvedBrow(int cx, int cy, String type) {
  int o = 12;
  if (type == "up")        tft.drawLine(cx - o, cy + 2, cx + o, cy,
ST77XX_WHITE);
  else if (type == "downL")tft.drawLine(cx - o, cy, cx + o, cy + 3,
ST77XX_WHITE);
```

```
  else if (type == "downR")tft.drawLine(cx - o, cy + 3, cx + o, cy,
ST77XX_WHITE);
  else if (type == "tiltL")tft.drawLine(cx - o, cy, cx + o, cy - 2,
ST77XX_WHITE);
  else if (type == "tiltR")tft.drawLine(cx - o, cy - 2, cx + o, cy,
ST77XX_WHITE);
  else if (type == "zigzag") {
    tft.drawLine(cx - o, cy, cx, cy + 3, ST77XX_WHITE);
    tft.drawLine(cx, cy + 3, cx + o, cy, ST77XX_WHITE);
  }
  else tft.drawLine(cx - o, cy, cx + o, cy, ST77XX_WHITE); // flat
}

void drawMouth(String mood) {
  int mx = 64, my = 110, w = 30;

  if (mood == "happy") {
    tft.drawLine(mx - w / 2, my, mx, my + 5, ST77XX_RED);
    tft.drawLine(mx, my + 5, mx + w / 2, my, ST77XX_RED);
  }
  else if (mood == "angry") {
    tft.drawLine(mx - w / 2, my + 5, mx + w / 2, my - 5,
ST77XX_RED);
  }
  else if (mood == "surprised") {
    tft.drawCircle(mx, my, 6, ST77XX_WHITE);
  }
  else if (mood == "dreamy") {
    tft.drawLine(mx - w / 4, my + 2, mx + w / 2, my,
ST77XX_MAGENTA);
  }
  else if (mood == "confused") {
    tft.drawLine(mx - 8, my, mx - 2, my + 5, ST77XX_WHITE);
    tft.drawLine(mx - 2, my + 5, mx + 4, my, ST77XX_WHITE);
    tft.drawLine(mx + 4, my, mx + 8, my + 4, ST77XX_WHITE);
  }
  else {
    tft.drawLine(mx - 10, my, mx + 10, my, ST77XX_WHITE);
  }
}

void drawAccessories(String mood) {
  if (mood == "surprised") {
    tft.fillCircle(110, 30, 4, ST77XX_CYAN); // пот
  } else if (mood == "angry") {
```

```
    tft.drawLine(20, 20, 24, 24, ST77XX_RED);
    tft.drawLine(24, 20, 20, 24, ST77XX_RED); // жилка
  } else if (mood == "confused") {
    tft.setCursor(100, 10);
    tft.setTextColor(ST77XX_YELLOW);
    tft.setTextSize(2);
    tft.print("?");
  }
}
```