МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Отчет по аттестационной работе

«Разработка программно-управляемого таймера»

по дисциплине

«Аппаратно-программное обеспечение интегрированных систем»

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ Мартынов Д. С. \_\_

(подпись) (фамилия, и.,о.)

СТУДЕНТЫ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Алексеева Елена

(подпись) (фамилия, и.,о.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Игнаков Константин

(подпись) (фамилия, и.,о.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ковалев Егор

(подпись) (фамилия, и.,о.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Семехин Василий

(подпись) (фамилия, и.,о.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Степушин Сергей

(подпись) (фамилия, и.,о.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сухоруков Валерий

(подпись) (фамилия, и.,о.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Тихонов Виктор

(подпись) (фамилия, и.,о.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Фомин Евгений

(подпись) (фамилия, и.,о.)

\_\_\_\_\_\_19-ВМ\_\_\_\_

(шифр группы)

Работа защищена «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород 2022

Оглавление

[Цель работы 3](#_Toc122992145)

[Ход работы 3](#_Toc122992146)

[1. Список функциональных требований к разрабатываемой системе 3](#_Toc122992147)

[2. Use-Case UML диаграмма 4](#_Toc122992148)

[3. Диаграммы потоков данных – DFD 5](#_Toc122992149)

[4. Диаграммы взаимодействия – WorkFlow 6](#_Toc122992150)

[5. Структурная схема устройства 7](#_Toc122992151)

[6. Дизайн панели управления устройством 7](#_Toc122992152)

[7. Листинг кода с комментариями 7](#_Toc122992153)

[Вывод 10](#_Toc122992154)

# Цель работы

Для устройства, указанного в задании, разработать дискретно-событийную модель функционирования, включающую в себя

* Список функциональных требований;
* Структурную схему устройства при реализации его на базе отладочной платы с микроконтроллером ATMega (Arduino);
* Диаграмму вариантов использования Use-Case, отражающую логику работы устройства;
* UML — диаграмму потоков данных;
* UML — диаграмму последовательности действий.

Вариант 1. Программно-управляемый таймер.

# Ход работы

## Список функциональных требований к разрабатываемой системе

1. Включение и выключению по кнопке.
2. Возможность установки времени таймера.
3. Отображение оставщегося времени на дисплее.
4. Звуковое оповещение по истечении установленного времени.

## Use-Case UML диаграмма

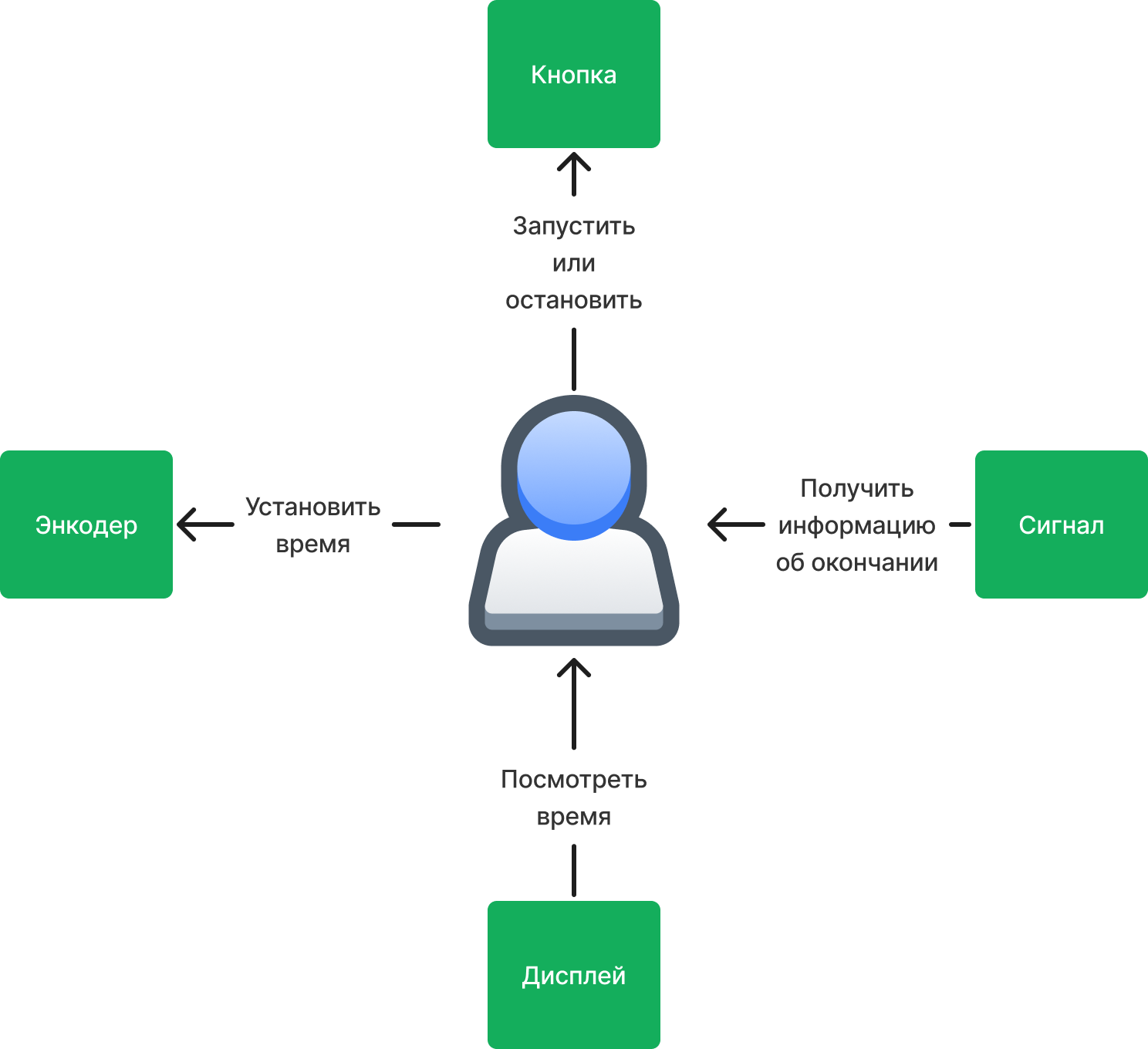


Рис 1. Диаграмма Use-Case

## Диаграммы потоков данных – DFD

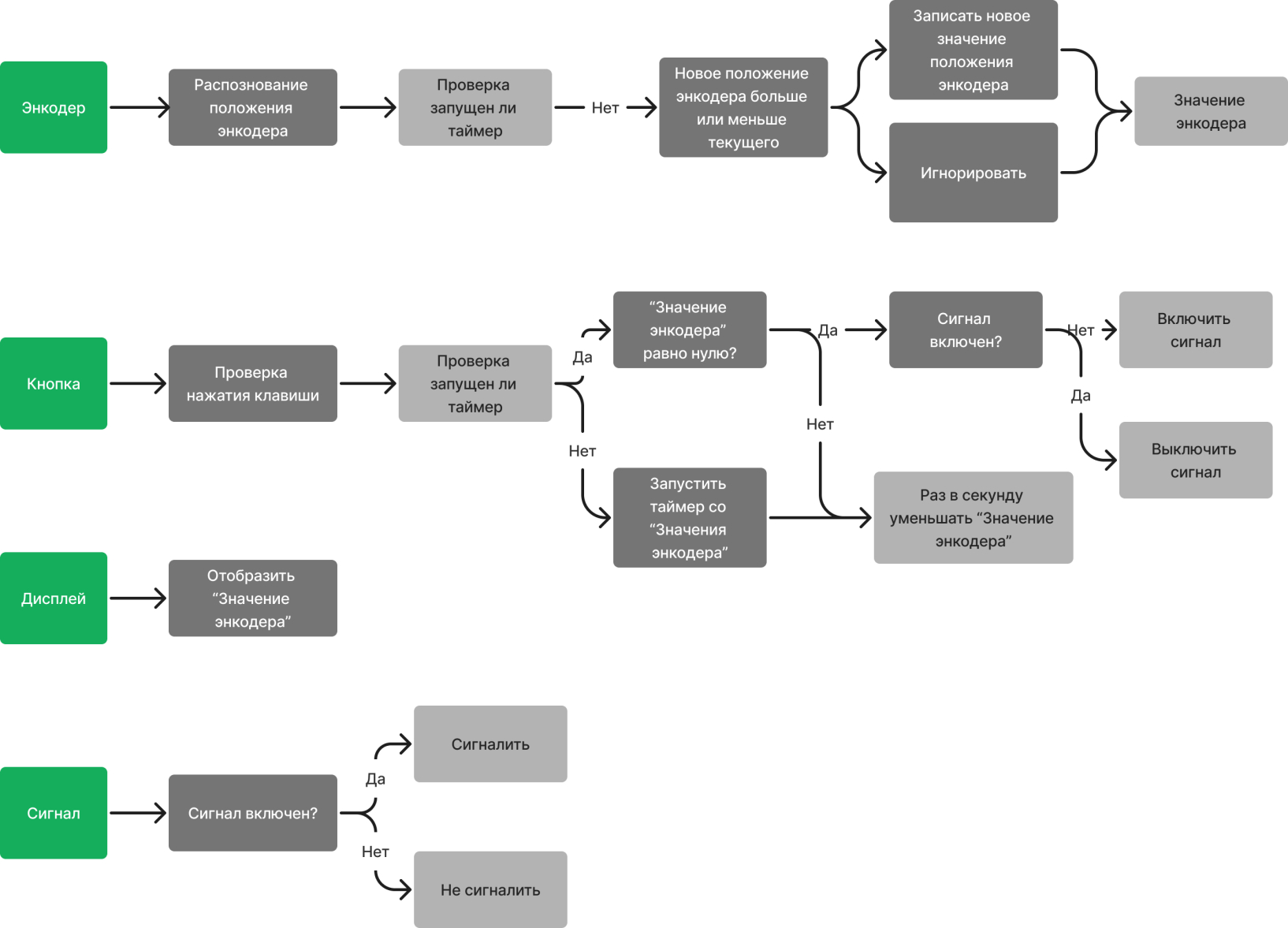


Рис 2. Диаграмма потоков данных

## Диаграммы взаимодействия – WorkFlow



Рис 4. Диаграмма взаимодействия

## Структурная схема устройства

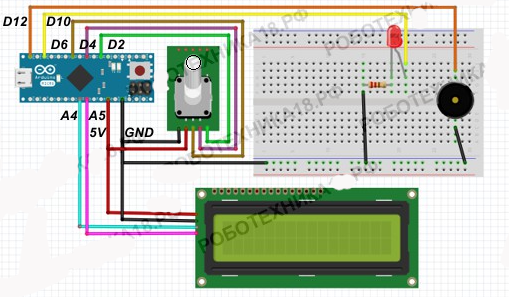


Рис 5. Структурная схема

## Дизайн панели управления устройством

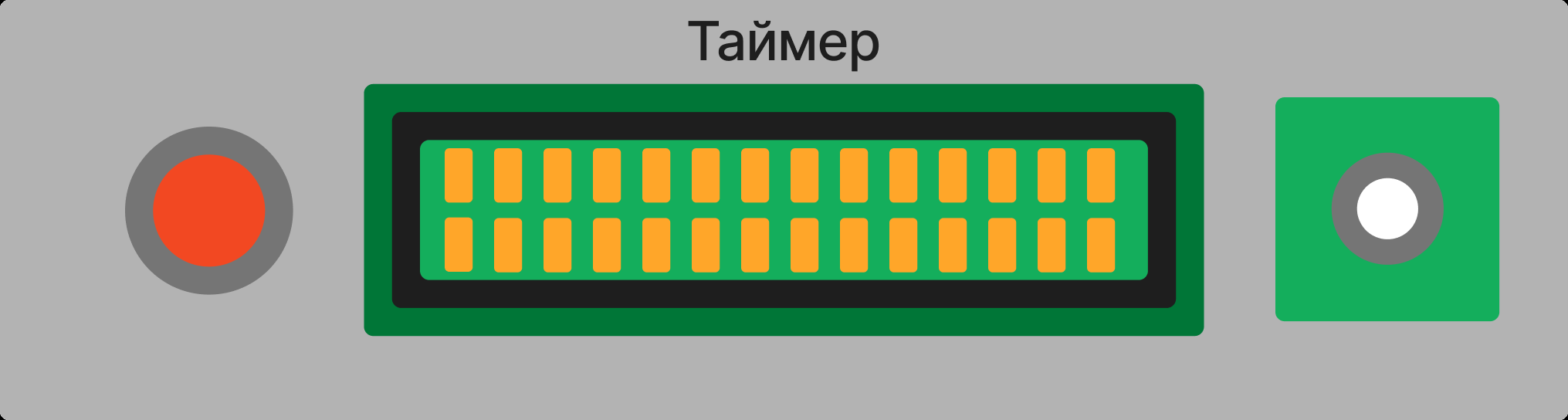


Рис 6. Дизайн панели управления

Высота корпуса – 2 см.

Длина корпуса – 20 см.

Ширина корпуса – 30 см.

## Листинг кода с комментариями

#include "Wire.h" // библиотека для протокола I2C

#include "LiquidCrystal\_I2C.h" // библиотека для LCD 1602

LiquidCrystal\_I2C LCD(0x27, 20, 2); // присваиваем имя дисплею

//библиотека для энкодера

#include <RotaryEncoder.h>

//пины подключение энкодера (DT, CLK)

RotaryEncoder encoder(4, 2);

// задаем шаг энкодера, максимальное и минимальное значение

#define STEPS 1

#define POSMIN 0

#define POSMAX 30

int lastPos, newPos;

boolean buttonWasUp = true;

byte w = 0;

int SEC = 0;

int MIN = 0;

unsigned long timer;

void setup() {

pinMode(6, INPUT\_PULLUP); // пин для кнопки энкодера

encoder.setPosition(0 / STEPS);

pinMode(10, OUTPUT); // подключаем светодиод и зуммер

pinMode(12, OUTPUT);

digitalWrite(10, HIGH);

LCD.init(); // инициализация дисплея

LCD.backlight(); // включение подсветки

LCD.setCursor(2, 0);

LCD.print("TIMER STOP");

LCD.setCursor(5, 1);

LCD.print(MIN);

LCD.print(" : ");

LCD.print(SEC);

}

void loop() {

// проверяем положение ручки энкодера

encoder.tick();

newPos = encoder.getPosition() \* STEPS;

if (newPos < POSMIN) {

encoder.setPosition(POSMIN / STEPS);

newPos = POSMIN;

}

else if (newPos > POSMAX) {

encoder.setPosition(POSMAX / STEPS);

newPos = POSMAX;

}

// если положение изменилось - меняем

//переменную MIN и выводим на дисплей

if (lastPos != newPos) {

MIN = newPos;

lastPos = newPos;

LCD.clear();

LCD.setCursor(2, 0);

LCD.print("TIMER STOP");

LCD.setCursor(5, 1);

LCD.print(MIN);

LCD.print(" : ");

LCD.print(SEC);

}

// если была нажата кнопка энкодера запускаем отсчет времени

boolean buttonIsUp = digitalRead(6);

if (buttonWasUp && !buttonIsUp && MIN > 0) {

delay(10);

buttonIsUp = digitalRead(6);

if (!buttonIsUp) {

if (SEC == 0) { SEC = 60; MIN = MIN - 1; }

if (MIN < 0 ) { MIN = 0; }

digitalWrite(10, LOW);

w = 1;

}

}

// запоминаем состояние кнопки

buttonWasUp = buttonIsUp;

while (w == 1 ) {

// если прошло 1000 мс - вычитаем одну

// секунду от переменной SEC

if (millis() - timer > 1000) {

timer = millis();

SEC = SEC - 1;

// если отсчет закончился - обнуляемся,

//включаем сигнал и выходим из цикла

if (SEC == 0 && MIN == 0) {

lastPos = 0; newPos = 0; MIN = 0; SEC = 0;

LCD.clear();

LCD.setCursor(2, 0);

LCD.print("TIMER STOP");

LCD.setCursor(5, 1);

LCD.print(MIN);

LCD.print(" : ");

LCD.print(SEC);

digitalWrite(10, HIGH);

tone(12, 100);

delay(500);

noTone(12);

w = 0;

}

// если секунды дошли до нуля - вычитаем одну минуту

if (SEC == 0 && w==1) {

SEC = 59; MIN = MIN - 1;

if (MIN < 0 ) { MIN = 0; }

}

// если из цикла while еще не вышли –

//выводим информацию на дисплей

if (w == 1) {

LCD.clear();

LCD.setCursor(2, 0);

LCD.print("TIMER START");

LCD.setCursor(5, 1);

LCD.print(MIN);

LCD.print(" : ");

LCD.print(SEC);

}

}

// если была нажата кнопка - обнуляем переменные

// и выходим из цикла

buttonIsUp = digitalRead(6);

if (buttonWasUp && !buttonIsUp) {

delay(10);

buttonIsUp = digitalRead(6);

if (!buttonIsUp) {

lastPos = 0; newPos = 0; MIN = 0; SEC = 0;

LCD.clear();

LCD.setCursor(2, 0);

LCD.print("TIMER STOP");

LCD.setCursor(5, 1);

LCD.print(MIN);

LCD.print(" : ");

LCD.print(SEC);

digitalWrite(10, HIGH);

w = 0;

}

}

buttonWasUp = buttonIsUp; // запоминаем состояние кнопки

}

}

# Вывод

В ходе работы была разработана и практически реализована модель программно-управляемого цифрового таймера, созданы UML-диаграммы. Были закреплены знания и умения по программированию с использованием микроконтроллера ATMega.