

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

**Институт радиотехнических и телекоммуникационных систем**

**Кафедра радиоэлектронных систем и комплексов**

**КУРСОВАЯ РАБОТА**

**по дисциплине**

«**Цифровые устройства и микропроцессоры**»

**Тема курсовой работы «Спортивный секундомер»**

Студент группы РССО-02-15 *Лавренов А. И.*

Руководитель курсовой работы *Богаченков А.Н.*



Работа представлена к защите «11» Мая 2020 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись студента)*

«Допущен к защите» «11» Мая 2020 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись руководителя)*

Москва 2020



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

**Институт радиотехнических и телекоммуникационных систем**

**Кафедра радиоэлектронных систем и комплексов**

**Утверждаю**

Заведующий кафедрой:

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_201\_г. *Замуруев С. Н*

**ЗАДАНИЕ**

**на выполнение курсовой работы**

**по дисциплине** «Цифровые устройства и микропроцессоры»

Студент: Лавренов А. И.

**1. Тема:** «Спортивный секундомер»

**2. Исходные данные:** Устройство обеспечивает отсчет и отображение времени с точностью до 0.1 с по 3–6 каналам. Одновременный запуск всех секундомеров производится кнопкой или внешним логическим сигналом. Остановка секундомеров может осуществляться: а) поочередным нажатием на одну и ту же кнопку; б) отдельной кнопкой для каждого секундомера; в) внешними логическими сигналами. Кроме того, имеются кнопки фиксации промежуточных значений времени, эти значения после нажатия кнопок постоянно выводятся на дисплей, при этом работа секундомеров продолжается. Предусмотреть возможность передачи показаний секундомеров внешнему устройству (компьютеру) посредством одного из стандартных интерфейсов.

**3. Перечень вопросов, подлежащих разработке, и обязательного графического материала:** Анализ технического задания. Внешний вид и работа с устройством. Функциональная схема. Временные диаграммы. Алгоритм устройства. Скриншоты дисплея. Текст программы. Заключение. Список литературы.

**4. Срок представления к защите курсовой работы:** до «\_\_\_»\_\_\_\_\_\_\_\_\_\_201\_г.

|  |  |  |  |
| --- | --- | --- | --- |
| Задание на курсовую работу выдал | «26» Февраля 2018г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись руководителя проекта)* | *Богаченков А.Н.* |
| Задание на курсовую работу получил | «26» Февраля 2018г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись студента исполнителя работы)* | *Лавренов А.И.* |

**Оглавление**

[**Текст задания 4**](#_Toc39774138)

[**Анализ технического задания 4**](#_Toc39774139)

[**Внешний вид и работа с устройством 5**](#_Toc39774140)

[**Функциональная схема 6**](#_Toc39774141)

[**Временные диаграммы 7**](#_Toc39774142)

[**Алгоритм устройства 9**](#_Toc39774143)

[**Скриншоты дисплея 14**](#_Toc39774144)

[**Текст программы 14**](#_Toc39774145)

[**Заключение 19**](#_Toc39774146)

[**Список литературы 20**](#_Toc39774147)

Текст задания

**Спортивный секундомер.**

Устройство обеспечивает отсчет и отображение времени с точностью до 0.1 с по 3–6 каналам. Одновременный запуск всех секундомеров производится кнопкой или внешним логическим сигналом. Остановка секундомеров может осуществляться: а) поочередным нажатием на одну и ту же кнопку; б) отдельной кнопкой для каждого секундомера; в) внешними логическими сигналами. Кроме того, имеются кнопки фиксации промежуточных значений времени, эти значения после нажатия кнопок постоянно выводятся на дисплей, при этом работа секундомеров продолжается. Предусмотреть возможность передачи показаний секундомеров внешнему устройству (компьютеру).

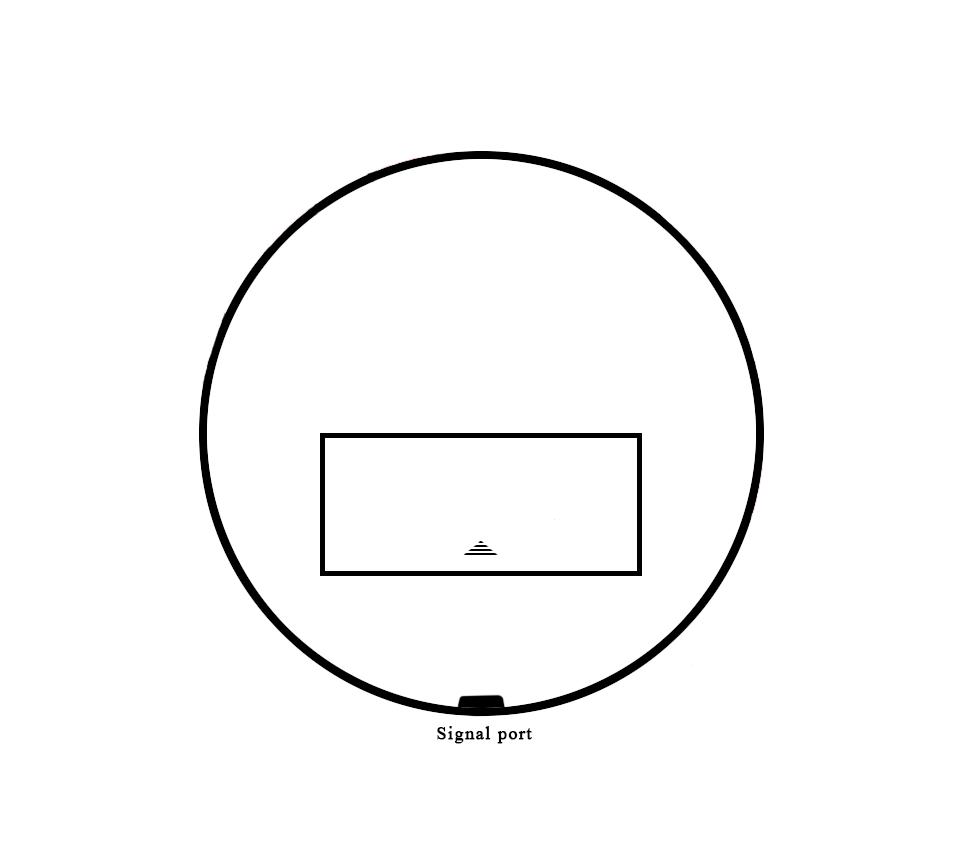
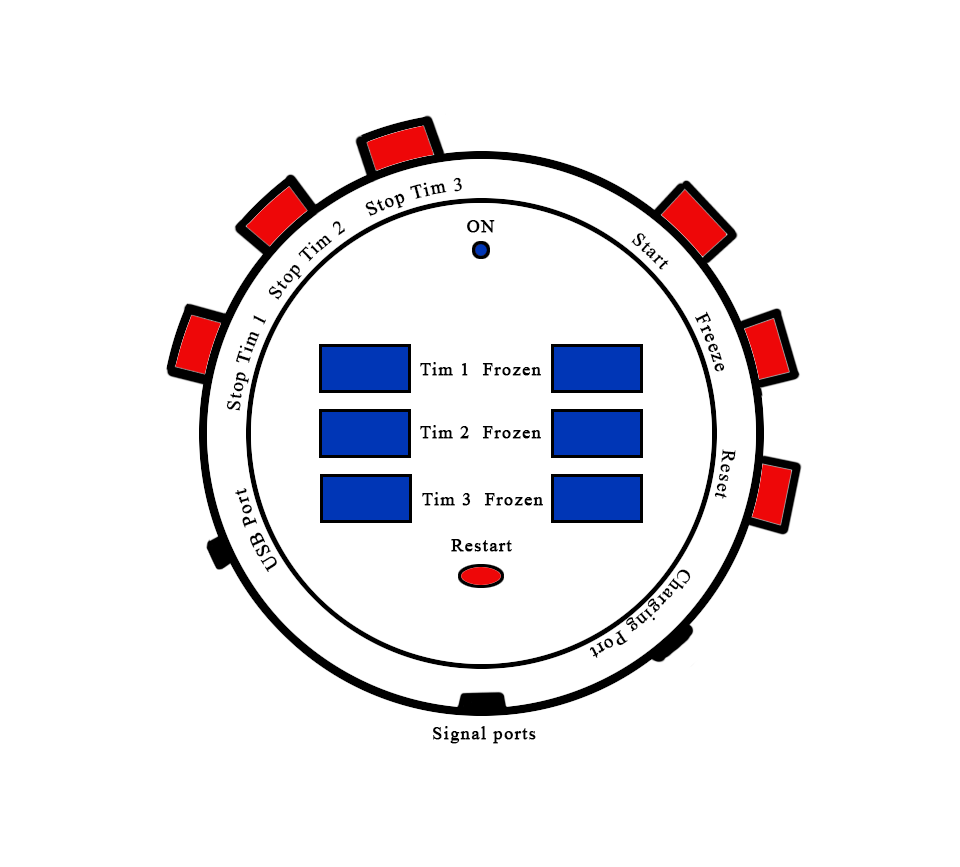
Анализ технического задания

Органы управления:

* Start - Запуск секундомеров.
* Stop Tim 1 - Остановить 1 секундомер.
* Stop Tim 2 - Остановить 2 секундомер.
* Stop Tim 3 - Остановить 3 секундомер.
* Freeze - Скопировать значения секундомеров в TimFrozen.
* Reset - Перезапуск секундомеров.
* Restart - System Reset.

Для выполнения вышеупомянутых критериев нам понадобится таймер с частотой 10 Гц, что соответствует 0.1с (заданной точности). Таймер будет работать в режиме прерываний. ЖК Дисплей, для вывода значений секундомеров. Органы управления, перечисленные в блоке текста задания. А также устройство передачи данных к ПК.

Внешний вид и работа с устройством



**Рис 1. Внешний вид устройства.**

**Описание работы с устройством.**

Для того, чтобы начать работу с секундомерами нужно нажать на боковую кнопку Start. Это запустит счет секундомеров Tim1-3.

Для остановки секундомеров Tim1-3 требуется нажать на боковые кнопки Stop Tim 1-3.

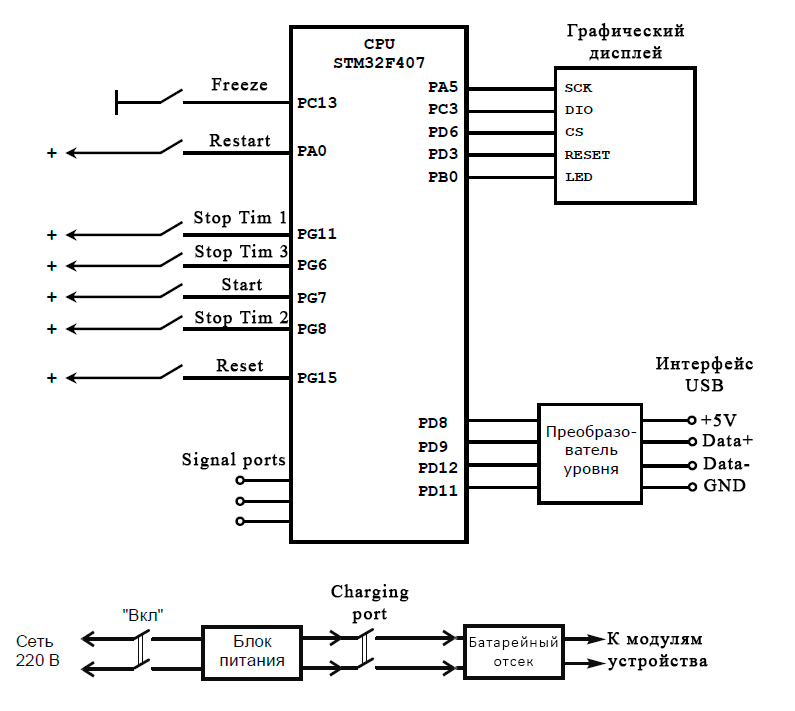
Для фиксации промежуточных значений секундомеров, не останавливая их, используется кнопка Freeze.

Для перезагрузки таймеров используется кнопка Reset.

Для перезагрузки системы используется кнопка Restart.

Signal port предназначен для остановки секундомеров.

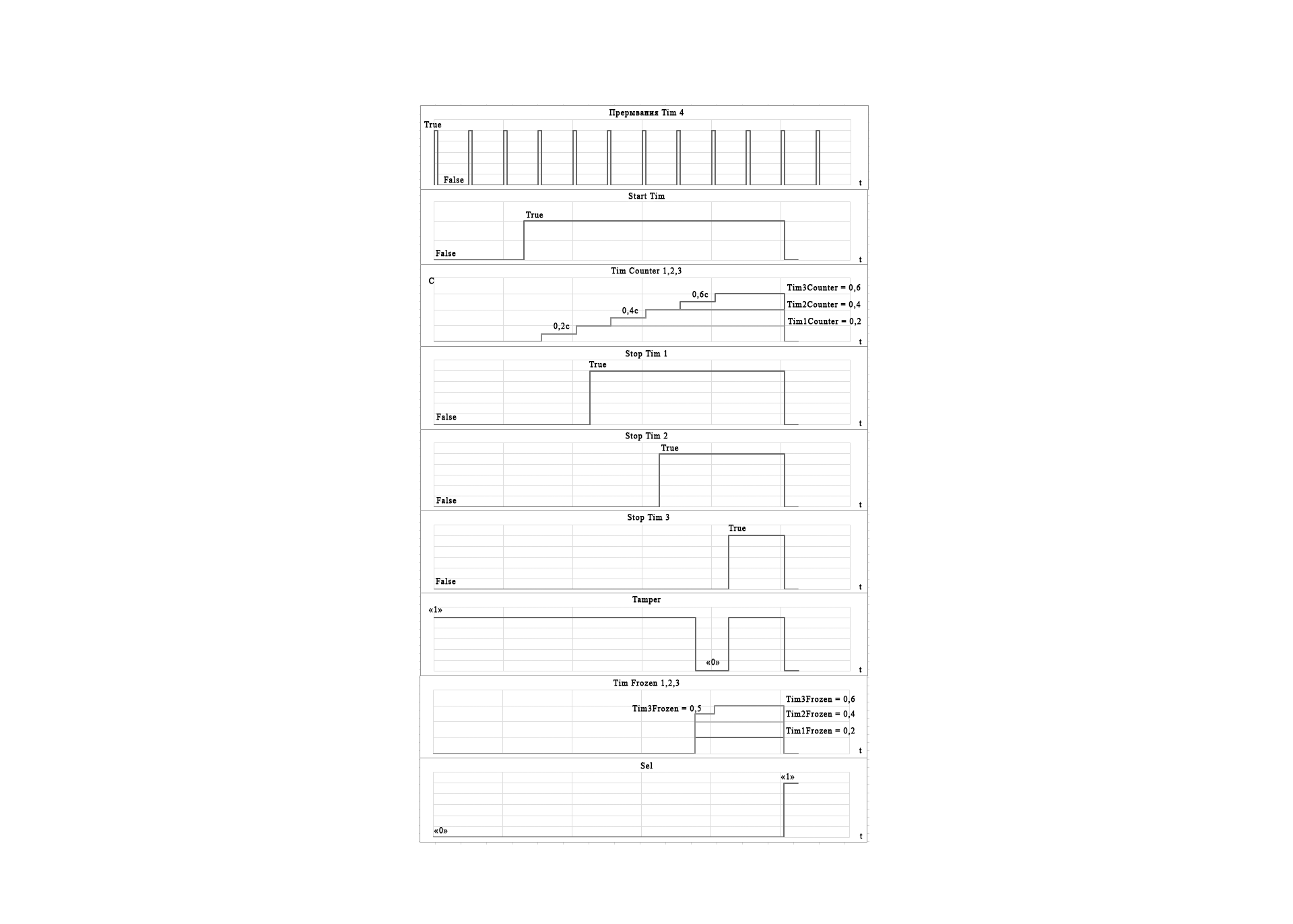
Функциональная схема



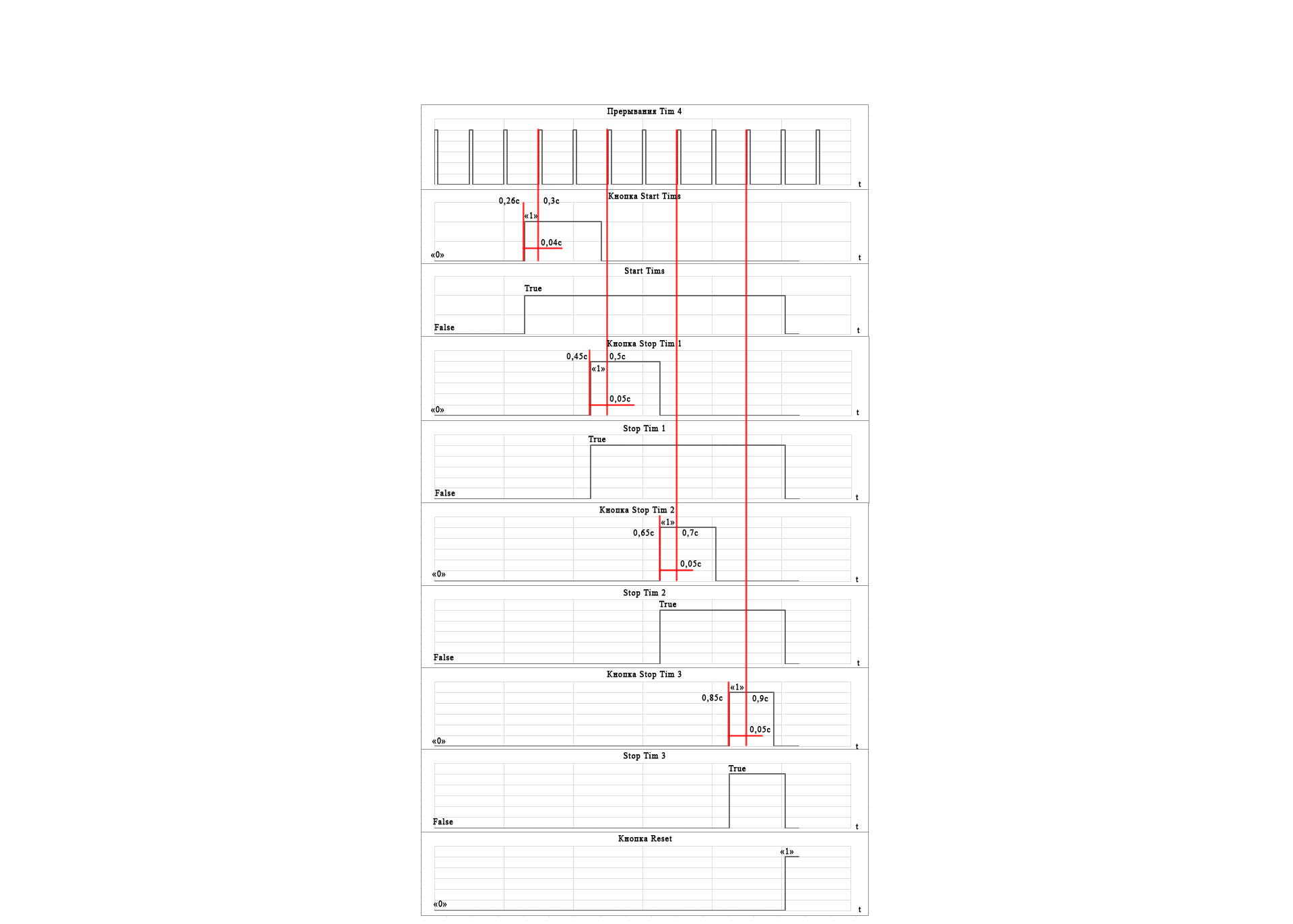
**Рис 2. Функциональная схема**

Все кнопки, кроме кнопки Tamper работают однотипно и выдают логическую единицу при замыкании. Кнопка Tamper же все время выдает логическую единицу, и замыкается на общий провод (землю). Все кнопки могут быть выполнены в виде утапливаемых в корпус тактовых кнопок с 2 контактами, к примеру тактовая кнопка SWT 6x6-5 (напряжение питания от 3.3 до 12 В). К микроконтроллеру подключен цветной ЖК дисплей 128 на 128 пикселей с светодиодной подсветкой работающий посредством четырехпроводного интерфейса (к примеру реплика дисплея Nokia 6610, как на плате лабораторного макета). Так же в устройстве имеется интерфейс USB который подключен к компьютеру посредством кабеля для передачи данных.

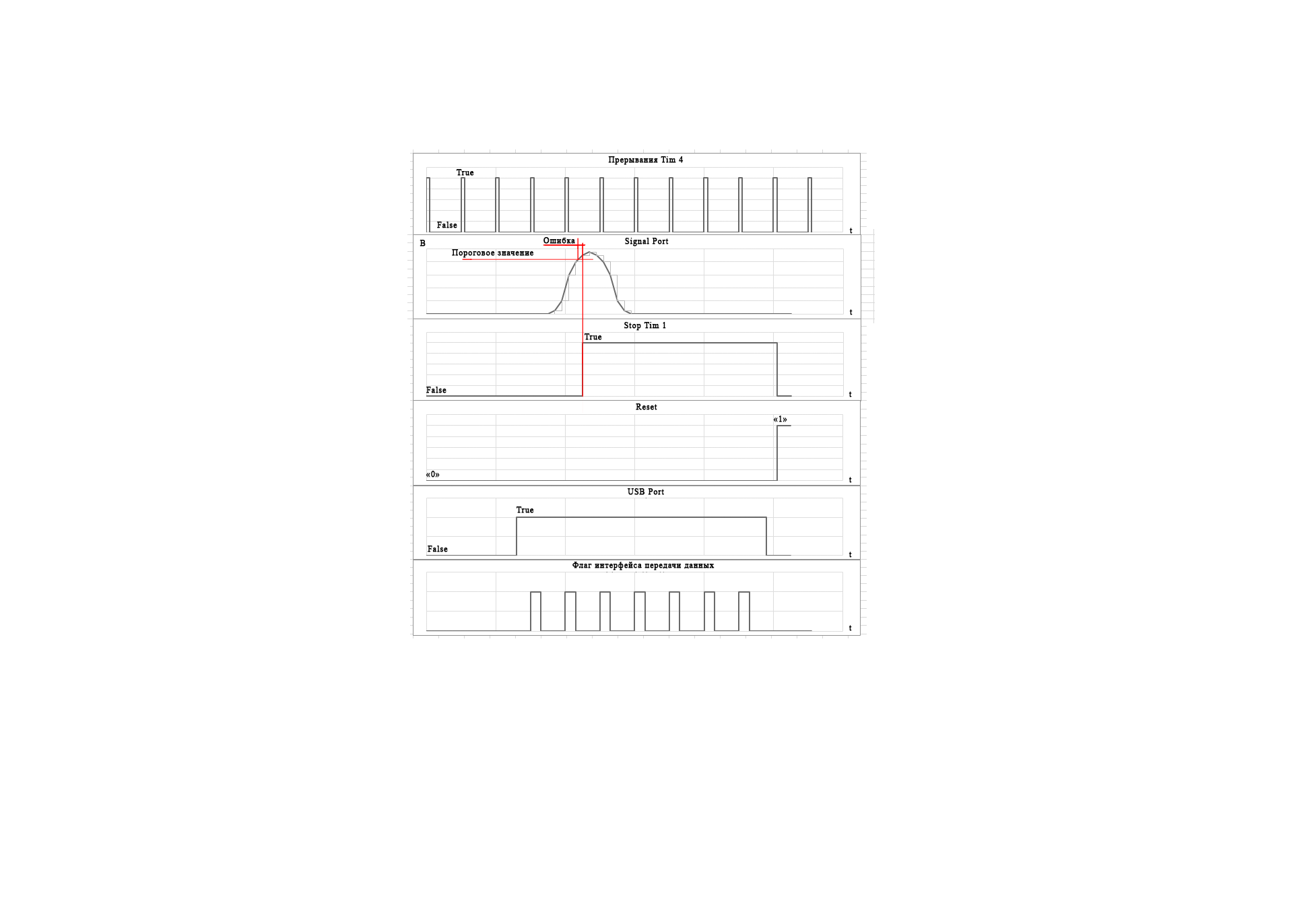
Временные диаграммы



**Рис 3. Временные диаграммы переменных программы.**



**Рис 4. Временные диаграммы реакции системы на кнопки.**

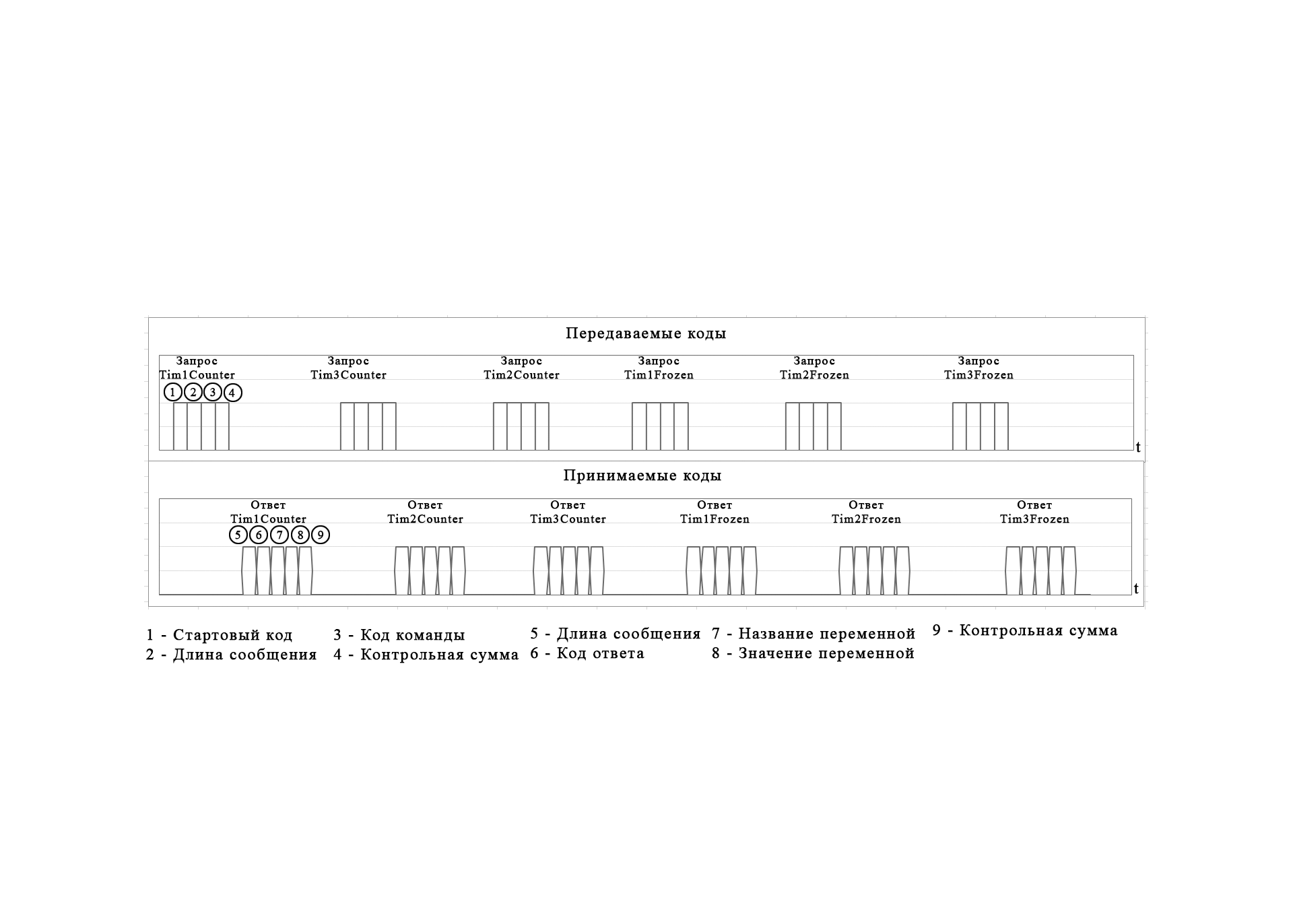


**Рис 5. Временные диаграммы реакции системы на сигнал остановки Tim 1.**

Точность секундомера в 0.1 секунду обусловлена следующей причиной:

Наращивание счетчика каждого секундомера происходит в моменты прерывания при перезагрузке таймера. Это означает, что время после прерывания таймера и до изменения булевой переменной отвечающей за фиксирование остановки секундомера нигде не учитывается. Аналогичная история происходит и при включении секундомеров и остановке Tim 1 с помощью сигнального порта.

Точность же остановки Tim 1 с помощью сигнального порта зависит не только от выше приведенной причины, а также и от частоты дискретизации входного сигнала. И тогда точность остановки по сигналу будет равна сумме точности «таймера» и интервала дискретизации.



**Рис 6. Временные диаграммы процесса запрос-ответ, для получения значений таймеров.**

****

**Рис 7. Форматы сообщений при передаче переменных.**

Стартовый код сообщения служит для синхронизации. Длина сообщений для TimCounter и TimFrozen присутствует, тк при передаче значений переменных указывается название переменно, а размеры названий TimCounter и TimFrozen отличаются. Код команды служит для того, чтобы запросить у устройства данные. Код ответа для того, чтобы различать данные при потерях пакетов подтверждения, несовпадении контрольных сумм и последующей повторной передачи. Контрольная сумма нужна для идентификации возможный ошибок в массиве.

Номер устройства имеет смысл указывать в формате сообщений, но тк мы рассматриваем изолированный случай, когда есть только один хост и одно устройство, то можно его не указывать.

Алгоритм устройства

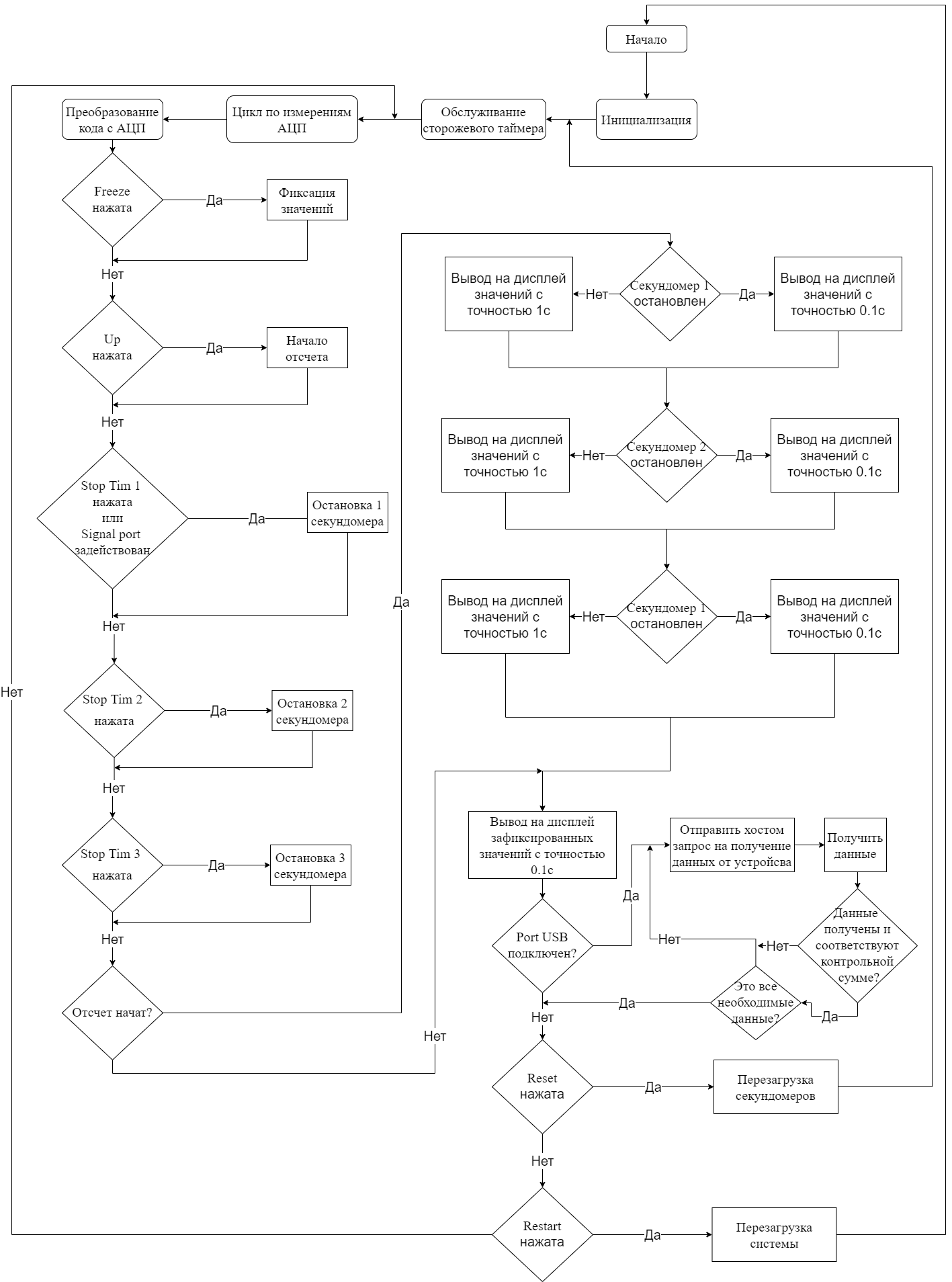
В программе задан таймер с частотой 10 Гц и периодом 0.1с. В моменты перезагрузки происходит прерывание. В Обработчике прерываний заданы 3 условия для каждого секундомера. Каждое из этих условий проверяет были ли запущены секундомеры (StartTims == true) и не остановлен ли конкретный секундомер (StopTim1 == false). При выполнении этих условий при каждом прерывании будет происходит наращивания переменной счетчика для каждого таймера на 0.

В программе используются булевые переменные, в цикле while(1) при воздействии на определенный орган они меняют значения.

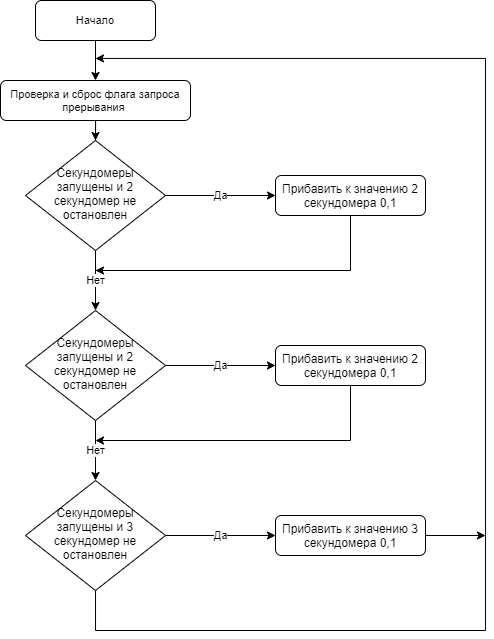
Так же в цикле while(1) присутствуют условия для вывода на дисплей значений секундомеров. Если секундомер работает, на дисплей выводится значение Тim1 с точностью 1с.

Если же секундомер остановлен, то на экран в Тim1 выводится значение с точностью 0.1с.

Далее выводятся снятые с секундомеров значения в Tim1-3 Frozen.

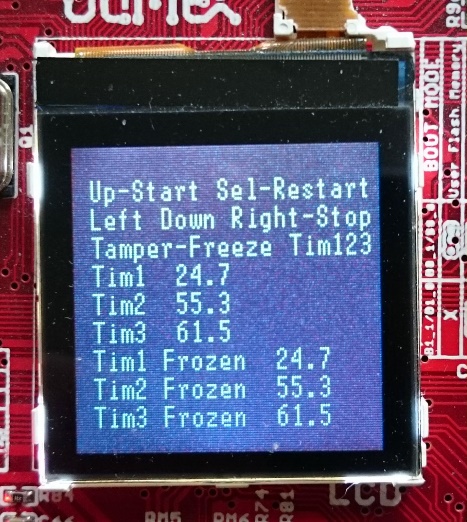


**Рис 8. Основной цикл программы.**



**Рис 9. Обработчик прерывания от таймера.**

Скриншоты дисплея

** **

Скриншот 1: Начало работы Скриншот 2: Конец работы

Текст программы

**Модуль main.c**

#include "stm32\_p407.h" //Файл конфигурации отладочной платы STM32-P407

#include "adc.h" //Функции для работы с АЦП

#include "lcd.h" //Функции для работы с дисплеем

//---------------------------------------------------------------------------

/\*

////////Органы управления:////////

UP - Запуск секундомеров.

LEFT - Остановить 1 секундомер.

DOWN - Остановить 2 секундомер.

RIGHT- Остановить 3 секундомер.

TAMPER - Скопировать значения секундомеров в TimFrozen.

SEL - Перезапуск секундомеров.

WAKEUP - System Reset.

////////Описание программы////////

В программе задан таймер с частотой 10 Гц и перодом 0.1с .

В моменты перезапуски происходит прерывание. В Обработчике прерываний заданы 3 условия для каждого секундомера.

Каждое из этих условий проверяет были ли запущены секундомеры (StartTims == true) и не остановлен ли конкретный секундомер (StopTim1 == false).

При выполнении этих условий при каждом прерывании будет происходит наращивания переменной счетчика для каждого таймера на 0.1 .

В программе используются булевые переменные, в цикле while(1) при воздействии на определенный орган они меняют значения.

Так же в цикле while(1) присутствуют условия для вывода на дисплей значений секундомеров. Если секундомер работае, на дисплей выводится значение Тim1 с точностью 1с.

Если же секундомер остановлен, то на экран в Тim1 выводится значение с точностью 0.1с.

Далее выводятся снятые с секундомеров значения в Tim1-3 Frozen.

\*/

//---------------------------------------------------------------------------

// ГЛАВНАЯ ФУНКЦИЯ

bool StartTims = false;

bool StopTim1 = false;

bool StopTim2 = false;

bool StopTim3 = false;

float Tim1Counter = 0;

float Tim2Counter = 0;

float Tim3Counter = 0;

void TimerConfig(void);

int main()

{

long AdcData; //Суммированные данные с АЦП

unsigned short UData16; //Приведенный смещенный (беззнаковый) код

int Average = 1; //Текущее число усреднений

int i; //Текущая переменная цикла

NVIC\_InitTypeDef NVIC\_InitStructure; //Структура конфигурации прерываний

float P;

float Tim1Frozen = 0;

float Tim2Frozen = 0;

float Tim3Frozen = 0;

//Инициализация кнопок и джойстика

STM\_PBInit(BUTTON\_TAMPER, BUTTON\_MODE\_GPIO); //Кнопка TAMPER

STM\_PBInit(BUTTON\_WAKEUP, BUTTON\_MODE\_GPIO); //Кнопка WAKEUP

STM\_PBInit(BUTTON\_RIGHT, BUTTON\_MODE\_GPIO); //Позиции джойстика

STM\_PBInit(BUTTON\_LEFT, BUTTON\_MODE\_GPIO); // - " -

STM\_PBInit(BUTTON\_UP, BUTTON\_MODE\_GPIO); // - " -

STM\_PBInit(BUTTON\_DOWN, BUTTON\_MODE\_GPIO); // - " -

STM\_PBInit(BUTTON\_SEL, BUTTON\_MODE\_GPIO); // - " -

//Инициализация дисплея

LCD\_Init(); LCD\_Clear(1);

LCD\_FontSize(9);

LCD\_FontColor(0x0F0);

LCD\_TextPos(1,1); LCD\_print("Up-Start Sel-Restart");

LCD\_TextPos(1,2); LCD\_print("Left Down Right-Stop");

LCD\_TextPos(1,3); LCD\_print("Tamper-Freeze Tim123");

LCD\_TextPos(1,4); LCD\_print("Tim1 ");

LCD\_TextPos(1,5); LCD\_print("Tim2 ");

LCD\_TextPos(1,6); LCD\_print("Tim3 ");

LCD\_TextPos(1,7); LCD\_print("Tim1 Frozen ");

LCD\_TextPos(1,8); LCD\_print("Tim2 Frozen ");

LCD\_TextPos(1,9); LCD\_print("Tim3 Frozen ");

LCD\_TextPos(7,4); LCD\_print("%.0f", Tim1Counter);

LCD\_TextPos(7,5); LCD\_print("%.0f", Tim2Counter);

LCD\_TextPos(7,6); LCD\_print("%.0f", Tim3Counter);

LCD\_TextPos(14,7); LCD\_print("%.0f", Tim1Frozen);

LCD\_TextPos(14,8); LCD\_print("%.0f", Tim2Frozen);

LCD\_TextPos(14,9); LCD\_print("%.0f", Tim3Frozen);

//Инициализация и первичный запуск АЦП

ADC\_Initialize(ADC2, ADC\_Channel\_10, 0);

ADC\_SoftwareStartConv(ADC2);

//Конфигурирование прерываний от АЦП

NVIC\_InitStructure.NVIC\_IRQChannel = ADC\_IRQn; //Номер (линия) прерывания

NVIC\_InitStructure.NVIC\_IRQChannelPreemptionPriority = 0; //Приоритет группы

NVIC\_InitStructure.NVIC\_IRQChannelSubPriority = 0; //Приоритет внутри группы

NVIC\_InitStructure.NVIC\_IRQChannelCmd = ENABLE; //Разрешение/запрет прерывания

NVIC\_Init(&NVIC\_InitStructure); //Функция конфигурирования

//Конфигурирование таймера (см. файл tim)

TimerConfig();

while (1)

{

//Цикл по измерениям

for (i = Average, AdcData = 0; i > 0; i--)

{

while (!ADC\_GetFlagStatus(ADC2, ADC\_FLAG\_EOC)); //Ожидание окончания преобразования

AdcData += ADC\_GetConversionValue(ADC2); //Ввод и суммирование показаний

ADC\_SoftwareStartConv(ADC2); //Новый запуск преобразования

Delay\_ms(1);

}

//Преобразование кода с АЦП

UData16 = AdcData / Average; //Вычисление среднего

P = ((float)UData16/65520)\*3.3f;

if (STM\_PBGetState(BUTTON\_TAMPER)){} //Снятие значений с секундомеров.

else

{

Tim1Frozen = Tim1Counter;

Tim2Frozen = Tim2Counter;

Tim3Frozen = Tim3Counter;

}

if (STM\_PBGetState(BUTTON\_UP)) StartTims = true; //Запуск секундомеров

if (STM\_PBGetState(BUTTON\_LEFT) || P > 1.6f) StopTim1 = true; //Остановка 1 Секундомера

if (STM\_PBGetState(BUTTON\_DOWN)) StopTim2 = true; //Остановка 2 Секундомера

if (STM\_PBGetState(BUTTON\_RIGHT)) StopTim3 = true; //Остановка 3 Секундомера

if (StartTims == true)

{

if (StopTim1 == true)

{

LCD\_TextPos(7,4); LCD\_print("%.1f", Tim1Counter); //Вывод окончательного значения с точностью

//0.1с при остановленном секундомере

}

else

{

LCD\_TextPos(7,4); LCD\_print("%.0f", Tim1Counter); //Вывод значения таймер с точностью 1с

}

if (StopTim2 == true)

{

LCD\_TextPos(7,5); LCD\_print("%.1f", Tim2Counter);

}

else

{

LCD\_TextPos(7,5); LCD\_print("%.0f", Tim2Counter);

}

if (StopTim3 == true)

{

LCD\_TextPos(7,6); LCD\_print("%.1f", Tim3Counter);

}

else

{

LCD\_TextPos(7,6); LCD\_print("%.0f", Tim3Counter);

}

LCD\_TextPos(14,7); LCD\_print("%.1f", Tim1Frozen); //Зафиксированные значения секундомеров

LCD\_TextPos(14,8); LCD\_print("%.1f", Tim2Frozen);

LCD\_TextPos(14,9); LCD\_print("%.1f", Tim3Frozen);

}

if (STM\_PBGetState(BUTTON\_SEL)) //Перезагрузка секундомеров.

{

StartTims = false;

StopTim1 = false;

StopTim2 = false;

StopTim3 = false;

Tim1Counter = 0;

Tim2Counter = 0;

Tim3Counter = 0;

Tim1Frozen = 0;

Tim2Frozen = 0;

Tim3Frozen = 0;

LCD\_Clear(1);

LCD\_TextPos(1,1); LCD\_print("Up-Start Sel-Restart");

LCD\_TextPos(1,2); LCD\_print("Left Down Right-Stop");

LCD\_TextPos(1,3); LCD\_print("Tamper-Freeze Tim123");

LCD\_TextPos(1,4); LCD\_print("Tim1 ");

LCD\_TextPos(1,5); LCD\_print("Tim2 ");

LCD\_TextPos(1,6); LCD\_print("Tim3 ");

LCD\_TextPos(1,7); LCD\_print("Tim1 Frozen ");

LCD\_TextPos(1,8); LCD\_print("Tim2 Frozen ");

LCD\_TextPos(1,9); LCD\_print("Tim3 Frozen ");

LCD\_TextPos(7,4); LCD\_print("%.0f", Tim1Counter);

LCD\_TextPos(7,5); LCD\_print("%.0f", Tim2Counter);

LCD\_TextPos(7,6); LCD\_print("%.0f", Tim3Counter);

LCD\_TextPos(14,7); LCD\_print("%.0f", Tim1Frozen);

LCD\_TextPos(14,8); LCD\_print("%.0f", Tim2Frozen);

LCD\_TextPos(14,9); LCD\_print("%.0f", Tim3Frozen);

}

if (STM\_PBGetState(BUTTON\_WAKEUP)) NVIC\_SystemReset(); //Сброс Системы

}

}

// ОБРАБОТЧИК ПРЕРЫВАНИЯ ОТ ТАЙМЕРА 4

void TIM4\_IRQHandler(void)

{

//Проверка установленного флага (источника прерывания)

if (TIM\_GetITStatus(TIM4, TIM\_IT\_Update) != RESET)

{

TIM\_ClearITPendingBit(TIM4, TIM\_IT\_Update); //Очистка флага

if (StartTims == true && StopTim1 == false) // Условие фиксирования значений

{

Tim1Counter+=0.1f; //Счет времени Tim1

}

if (StartTims == true && StopTim2 == false) // Условие фиксирования значений

{

Tim2Counter+=0.1f; //Счет времени Tim2

}

if (StartTims == true && StopTim3 == false) // Условие фиксирования значений

{

Tim3Counter+=0.1f; //Счет времени Tim3

}

}

}

//---------------------------------------------------------------------------

// ОБРАБОТЧИК ПРЕРЫВАНИЯ ОТ АЦП

void ADC\_IRQHandler(void)

{

if (ADC\_GetFlagStatus(ADC2, ADC\_FLAG\_AWD) != RESET) //Проверка флага "аналогового сторожа"

{

ADC\_ClearFlag(ADC2, ADC\_FLAG\_AWD); //Сброс флага

STM\_LEDOn(LED3); //Сигнализация о выходе уровня за пределы

}

}

Заключение

В курсовой работе был разработан трех канальный спортивный секундомер с функциями остановки каждого канала раздельно, в том числе одного посредством аналогового сигнала. Так же была разработана система фиксации значений без остановки таймеров. В процессе выполнения курсовой работы были приобретены теоретические знания по использованию таймеров на линейке микроконтроллеров STM32. Практически была осознана ценность форматирования текста программ для дальнейшего удобства пользования, изменения и демонстрирования его на различных ЭВМ. В дальнейшем при улучшении и модернизации разработанного устройства возможны: Увеличение количество каналов секундомеров, фиксирование значений секундомеров по каждому каналу в отдельности, остановка каждого канала секундомеров различными аналоговыми и цифровыми сигналами, а также автоматизация в целях использования на спортивных соревнованиях (бег на разные дистанции, биатлон, различные марафоны).

Список литературы

* Datasheet STM32-P407 User Manual
* А.Н. Богаченков «ПРОЦЕССОРЫ С ЯДРОМ ARM В ЛАБОРАТОРНОМ ПРАКТИКУМЕ И КУРСОВОМ ПРОЕКТИРОВАНИИ» Учебное пособие

по выполнению лабораторных и курсовых работ, Москва, 2016 г.

* А.Н. Богаченков ЦИФРОВЫЕ УСТРОЙСТВА И МИКРОПРОЦЕССОРЫ Методические указания по выполнению курсовой работы, Москва, 2018 г.
* А.Н. Богаченков ЦУМП2\_ARM\_Lab1-3\_Тексты\_программ