

|   |    |
|---|----|
| Содержание  |    |
| Содержание.....   | 1  |
| Введение.....   | 3  |
| Глава 1. Семейство STM-32.....  | 4  |
| 1.1 Стандартная библиотека периферии STM32.....   | 5  |
| 1.2 Выбор программного инструментария для разработки.....                                       | 6  |
| 1.3 Работа с блоками микроконтроллера STM32.....  | 7  |
| Глава 2. Работа спектрометра с двойной модуляцией.....  | 9  |
| 2.1 Двойная модуляция.....  | 9  |
| 2.2 Синхронный детектор.....  | 11 |
| Глава 3. Основы построения Web-интерфейса.....  | 13 |
| 3.1 HTML.....   | 13 |
| 3.2 JavaScript.....   | 15 |
| 3.3 PHP.....  | 19 |
| Глава 4. Построение программы управления спектрометра на основе микроконтроллера STM32F407..... | 22 |
| 4.1 Конфигурация режимов ЦАП.....   | 22 |
| 4.2 Конфигурация режимов АЦП.....   | 23 |
| 4.3 Конфигурация режимов ТТЛ выходов.....   | 24 |
| 4.4 Пример работы с микроконтроллером в режиме включения- выключения светодиодов.....           | 25 |
| Глава 5. Удаленное управление при помощи Web-интерфейса.....                                    | 27 |

|                                 |    |
|---------------------------------|----|
| 5.1 Web-интерфейс.....          | 27 |
| 5.2 Примеры Web-интерфейса..... | 27 |
| Заключение.....                 | 30 |
| Список литературы.....          | 31 |

## Введение

На сегодняшний день актуально развитие цифровой техники для необходимости перехода на новые модели микроконтроллеров.

Использование микроконтроллеров позволит автоматизировать работу с ЯМР и ЭПР установками, в том числе производить удаленное управление установкой через интернет.

Разработанное оборудование предполагается использовать на практике в будущем для модернизации лабораторных установок ЯМР и создания компактных ЭПР спектрометров.

Цель работы:

Разработка модуля управления спектрометром на основе микроконтроллера STM32F407.

В соответствии с целью работы были поставлены следующие задачи: Изучение основ программирования микроконтроллеров семейства STM32.

Конфигурирование микроконтроллера под задачи стационарного магнитного резонанса; разработка протокола обмена с помощью виртуального СОМ-порта на основе библиотеки “USBD CDC”; Разработка Web-интерфейса управления микроконтроллером.

## Глава 1. Семейство STM-32

Компания STM первая вывела на рынок класс микроконтроллеров на ядре ARM Cortex M3 и до сих пор по праву занимает лидирующее место среди производителей класса микроконтроллеров. Введение началось в 2007 году с двух компаний – Performance Line (STM32F103) и Access Line (STM32F101). Они постоянно работают как над увеличением разновидностей класса, так и над улучшением параметров пополняя аппаратную составляющую микроконтроллера. На сегодняшний день микроконтроллеры семейства STM32 уже состоит из 10 линеек (рисунок 1) для всевозможных применений микроконтроллера и постановки задачи для использования – высокой мощнось, производительность, дешевизна микроконтроллеров для общего применения, с очень низким энергопотреблением, со встроенным радиомодулем для беспроводного подклчюения. Все это на одном ядре ARM Cortex-M3.

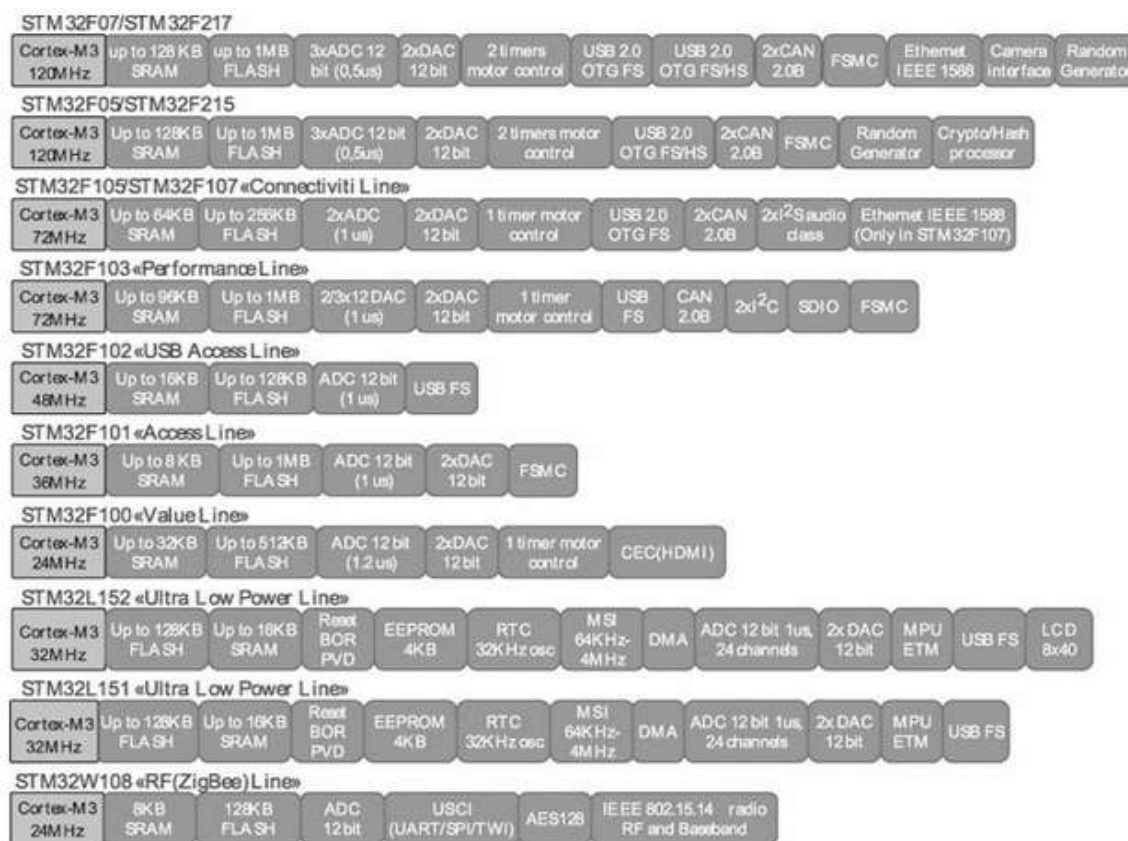


Рисунок 1 – Линейка микроконтроллеров семейства STM32.

Для проведения работ и дальнейшего изучения и программирования микроконтроллера разработчику необходимо 3 инструмента :

- 1) программная среда разработки (например Сооох),
- 2) программатор-отладчик и плата от производителя или от других производителей микроконтроллеров.

### 1.1 Стандартная библиотека периферии STM32

STM для более легкого труда для разработчиков предоставляет бесплатные библиотеки периферии для линейки микроконтроллеров и в том числе для линейки микроконтроллеров семейства STM32.

Стандартная библиотека периферии написана в соответствии со стандартом ANSI C и может использоваться с любым компилятором. Структура библиотеки не так сложна, как кажется на первый взгляд, и состоит из двух взаимодополняющих составляющих.

1 –ая из составляющих – заголовки к файлам и файлы всей периферии микроконтроллеров линейки STM32: STM32F10x\_StdPeriph\_Driver. Весь набор функций периферийных модулей имеет описание в заголовках файлов и в файлах реализации – для портов ввода-вывода это 2 файла:

- stm32f10x\_gpio.h;

- stm32f10x\_gpio.c.

Для дальнейшего пользования библиотекой периферии нужно в файл основной программы (main.c) включить файл - # include “stm32f10x.h” и указать определенные переменные в свойствах работы.

Библиотекой периферии предоставляются 3 файла для модификации разработчиком:

- 1) файл конфигурации библиотеки stm32f10x\_conf.h

Также файлы прерываний:

- 2) stm32f10x\_it.h

- 3) stm32f10x\_it.c.

Для использования определенных модулей периферии в работу необходимо добавить файлы реализации и проделать конфигурирование файла - stm32f10x\_conf.h. Под конфигурацией файла stm32f10x\_conf.h подразумеваем:

- раскомментирование строчек с названием периферийного модуля, предполагаемого для использования.

## 1.2 Выбор программного инструментария для разработки

Под ARM-архитектуре существует довольно широкий выбор программных средств разработки. Приведем лишь основные и самые популярные программные пакеты на российском рынке в таблице.

Таблица 1.

## Программные пакеты на российском рынке.

| Инструментарий | Среда разработки   | C/C++ компилятор | Ограничение Си-инструментария                 | Программатор-отладчик | Ссылка   |
|----------------|--------------------|------------------|---|-----------------------|--|
| IAR Systems    | Embedded Workbench | IAR C/C++        | 32 Кбайт или полная с ограничением на 30 дней | J-Link ST-Link        | <a href="http://www.iar.com">www.iar.com</a>               |
| Keil           | uVision (MDK-ARM)  | Keil C/C++       | 32 Кбайт                                      | ULink ST-Link         | <a href="http://www.keil.com">www.keil.com</a>             |
| Raisonance     | Ride7 + RKIT-ARM   | GCC C/C++        | Нет, ограничения по отладке                   | RLink                 | <a href="http://www.raisonance.com">www.raisonance.com</a> |
| Atollic        | TrueStudio         | GCC C/C++        | Нет, ограничения по функционалу               | ST-Link STICE         | <a href="http://www.atollic.com">www.atollic.com</a>       |
| Open source    | Eclipse            | GCC C/C++        | Без ограничений                               | ARM-Link              | <a href="http://www.eclipse.org">www.eclipse.org</a> [2]   |

Наиболее популярными и самыми дорогостоящими среди производителей для разработки Программного Обеспечения по ARM архитектуре, которые являются конфигурациями от компаний Keil и IAR Systems. Это обуславливается наиболее современными C-инструментариями с точки зрения оптимизации и минимизации кода. Помимо лидирующих позиций в C-инструментариях, данные компании имеют широкие наборы дополнительного Программного Обеспечения – операционные системы имеющиеся на сегодняшний, USB стеки, TCP/IP стеки и другое ПО, но за дополнительную плату.

### 1.3 Работа с блоками микроконтроллера STM32

В микроконтроллере STM32 имеется четыре типа блоков ТТЛ, АЦП, ЦАП, таймер.



ТТЛ (Time to live) — предельный период времени или число итераций или переходов, в пределах которого набранные данные (пакет данных) может существовать до своего окончания.

Таймер/счетчик представляет собой устройство на основе 8-разрядного регистра TL0 (в T/C0) или TL1 (в T/C1). При каждом заполнении данных регистров в процессе счета, помимо установки в регистре TCON флагов TF0 или TF1, происходит автоматический перезапуск этих регистров данными из регистров TH0 или TH1. Назначение bit-ов управления в данном режиме аналогичное, как и в режимах 0 и 1.

## Глава 2. Работа спектрометра с двойной модуляцией

Наиболее распространенный метод Радиоспектроскопии является метод Ядерного Магнитного резонанса (ЯМР) и Электронный парамагнитный резонанс (ЭПР). Такой интерес этих методов основан на эффектах поглощения электромагнитной энергии веществом, в РЧ-диапазоне 1 МГц– 100 GHz. Резонанс наблюдается когда квант электромагнитной энергии « $h\nu$ » равен расщеплению уровней энергии ядра (ЯМР)  $h\gamma H$  или электронной оболочки  $g\beta H$  (ЭПР), где  $h$  есть постоянная Планка,  $\nu$  – частота,  $g$ - « $g$ -фактор» Ланде,  $\beta$  -магнетон Бора (магнетон Бора),  $\gamma$  - «ядерное» гиромагнитное отношение, а  $H$  – напряженность магнитного поля. Спектр ЭПР или ЯМР в данном примере будет состоять из отдельных линий, соответствующие разным состояниям электронных оболочек атома или ядер.

### 2.1 Двойная модуляция

Большие сигналы ЯМР (на протонах воды) мы можем наблюдать на осциллографе, когда используем низкочастотную модуляцию магнитного поля. Мощность  $P$  шумов в данной схеме, пропорциональна ширине линии пропускания  $\Delta F$  усилителя низкой частоты (УНЧ), довольно высокая. Для наблюдения слабых сигналов необходимо уменьшить мощность шумов, т.е. сократить полосу пропускания УНЧ. В случае когда сигнал очень слабый, ширина полосы пропускания должна быть меньше 0,1 Гц.

Наряду с медленным изменением магнитного поля по линейному закону (первая модуляция) осуществляем быструю модуляцию магнитного поля с частотой  $\Omega$  (меньше ширины линии Ядерного Магнитного Резонанса) (вторая модуляция). В итоге имеем изменение магнитного поля по закону:

$$H = at + H_M \sin\Omega t;$$

(где  $a = const.$ ,  $H_M < \Delta H$  – глубина модуляции) которое приводит к тому, что сигнал Ядерного Магнитного Резонанса переходит на определенную фиксированную частоту  $\Omega$ , причём амплитуда этого сигнала пропорциональна производной поглощения  $d\chi''/dH$ , а фаза колебаний с частотой  $\Omega$  на разных склонах кривой поглощения  $\chi''(H)$  отличаются на  $\pi$  ( $180^\circ$ ).

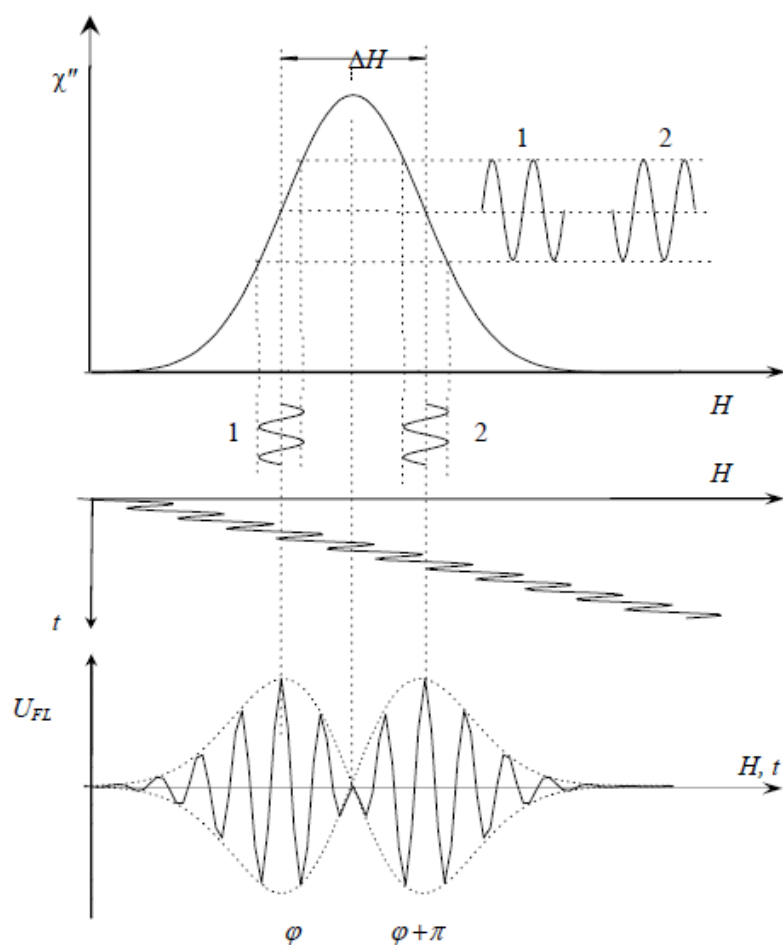


Рисунок 2- Изменение фазы колебаний ( на  $180^\circ$ ).

## 2.2 Синхронный детектор

В синхронном детекторе выходное напряжение  $u(t)$  селективного усилителя:

$$u(t) = u_0 \sin \Omega t + e(t),$$

которое представляет собой сумму полезного сигнала и сигнала шума  $e(t)$ , и умножается на периодическую функцию  $F(t)$  (с периодом  $2\pi\Omega$ ) - опорное напряжение, и затем берем интеграл по времени:  $\tau \gg 2\pi / \Delta\Omega$ ;

здесь  $\Delta\Omega$  – ширина частотного спектра сигнала выходного напряжения  $u(t)$ , ограниченный шириной полосы пропускания селективного усилителя.

На рисунке 3 изображен синхронный детектор на полевом транзисторе с 2 изолирующими затворами. Время интегрирования по данной схеме определяется постоянной времени цепи:

$$\tau = 2\pi * RC;$$

Увеличивая значения сопротивления « $R$ » и емкости « $C$ », добьемся эквивалентной полосы пропускания синхронного детектора, размером нескольких десятков герца. Синхронный детектор называется также фазовым детектором, т.к. напряжение на его выходе зависит от опорного напряжения и соотношения фаз сигнала.

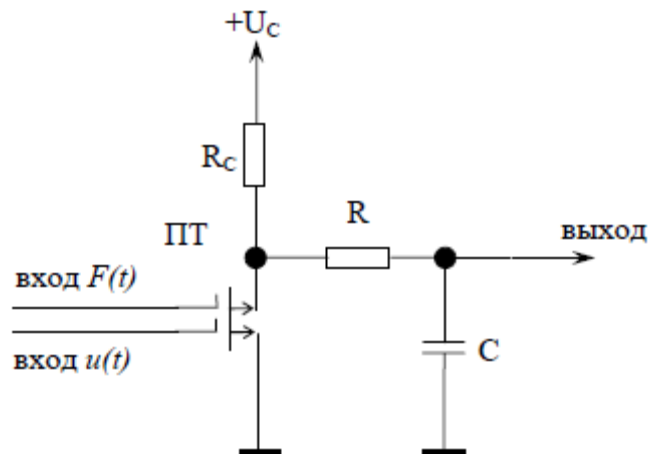


Рисунок 3 - Схема синхронного детектора на полевом транзисторе с двумя изолированными затворами.

В итоге получаем, что если опорное напряжение отстает по фазе относительно сигнала на угол  $\varphi$ , то на выходе синхронного детектора имеем (поскольку  $\Omega\tau \gg \Delta\Omega\tau \gg 1$ ):

$$U(\tau, \varphi) = \frac{u_0 t}{2} \cos\varphi.$$

Простой способ произведения вх. сигнала  $u(t)$  на опорное напряжение  $F(t)$  это применение диодного переключателя.

В итоге опорное напряжение  $u(t)$  представляется в виде:

$$F(t) = -1, -\frac{T}{2} < t < 0,$$

$$+1, 0 < t < +\frac{T}{2}, T = \frac{2\pi}{\Omega}$$

Если  $\Delta\Omega \ll \Omega$ , то спектр шумов на входе синхронного детектора не имеет общих составляющих с частотами  $3\Omega$ ,  $5\Omega$  и т.д., и поэтому части  $F(t)$  с  $n \neq 0$  не относятся к результирующему сигналу.

## Глава 3. Основы построения Web-интерфейса

Web-интерфейсы получили широкое распространение в связи с ростом популярности всемирной паутины и соответственно — повсеместного распространения Web-браузеров.

Web-интерфейс — это совокупность средств, при помощи которых пользователь взаимодействует с веб-сайтом или любым другим приложением через браузер.

### 3.1 HTML

HTML является языком программирования разработанный, чтобы позволять создавать сайты.

Язык HTML (Hyper TextMarkup Language) на сегодняшний день является основным стандартом для написания Веб-страниц во Всемирной сети. HTML использует для формирования страниц специальные команды — теги.

Теги HTML состоят из следующих элементов:

- угловой скобки «<»;
- слэша «/» (тег — конечный тег, который закрывает некую структуру);
- name;
- угловой скобки «>»

Для каждого тега определяются возможные параметры. Параметры состоят из следующих элементов:

- name параметра;
- знака равно «=»;

- значение параметра, которое задается строкой символов;

Многие теги являются парными (имеют начальный - открывающий и конечный - закрывающий тег). Отличие закрывающего тега от открывающего в том, что в начале закрывающего тега пишется символ «/»

Обязательная структура HTML документа:

<HTML>

<HEAD>

<TITLE>Заголовок < /TITLE >

<HEAD>

<BODY>

Тело документа

</BODY>

</HTML>

HTML тег верхнего уровня, определяющий тип документа.

HEAD –заголовок документа. Внутри этого тега обычно указывается другой тег –TITLE –имя документа.

BODY –тело документа. Помещая тело в HTML-страницы, можно задать параметры, влияющие на представление документа в целом (установить цвет или фон).

## Параметры тега BODY.

| Имя параметра | Значение  |
|---------------|---|
| BGCOLOR       | фоновый цвет документа  |
| TEXT          | цвет для текста документа   |
| LINK          | цвет для <i>непосещенной</i> гипертекстовой связи   |
| VLINK         | цвет для <i>посещенной</i> гипертекстовой ссылки  |
| ALINK         | цвет для <i>активной</i> гипертекстовой ссылки; используется для выделения текста связи на момент, когда пользователь нажал на ссылку |
| BACKGROUND    | URL фонового рисунка  |

## 3.2 JavaScript

JavaScript — это легко интерпретируемый язык с функциями как объект первого класса. Он более известен как язык script-ов для Web-страниц, но также используемый и в других программных продуктах, например, node.js или Apache. Внутри среды исполнения JavaScript может быть связан с объектами данной среды и предоставлять программный контроль над ними.

Java — это язык программирования, основанный на классах и предназначенный для быстрого выполнения и безопасности типов.

По сравнению с Java, JavaScript является языком с очень свободной формой. Вам не надо объявлять переменные, классы и методы. Вам не надо беспокоиться о том, являются ли методы публичными (public), приватными (private) или защищенными (protected), а также вам не надо реализовывать интерфейсы.



Документ ECMAScript не предназначен для помощи в создании скриптов. Чтобы получить информацию об написании скриптов, используется документация JavaScript.

СMAScript — это набор требований, касающихся реализации ECMAScript.

Приложения, написанные на JavaScript, могут исполняться на серверах, использующих Java 6 и более поздних версий. Это обстоятельство используется для построения серверных приложений, позволяющих обрабатывать JavaScript на стороне сервера.

#### Область применения

Помимо Java 6, существует ряд платформ, использующих существующие движки (интерпретаторы) JavaScript для исполнения серверных приложений. (Как правило, речь идёт о повторном использовании движков, ранее созданных для исполнения кода JavaScript в браузерах WWW.

Платформы исполнения серверных приложений на  
JavaScript.

| Название                   | Используемый движок<br><u>JavaScript</u> | Языки,<br>на которых<br>написан движок<br>и платформа | Лицензия                              |
|----------------------------|--|---|---------------------------------------|
| <u>Jaxer</u>               | <u>SpiderMonkey</u>                      | C++, C  | GPL 3                                 |
| <u>persevere-framework</u> | <u>Rhino</u>                             | Java  | Модифицированная лицензия BSD         |
| <u>Helma</u>               | <u>Rhino</u>                             | Java, <u>JavaScript</u>                               | BSD-подобная <u>Helma License 2.0</u> |
| <u>v8cgi</u>               | V8                                       | C++, <u>JavaScript</u>                                | Лицензия BSD                          |
| <u>node.js</u>             | V8                                       | C++   | Лицензия MIT                          |

### Реализация JavaScript

На JavaScript реализованы интерпретаторы ряда языков программирования, позволяющие использовать для них как среду выполнения веб-браузера. Они могут найти свое применение в образовательных целях.

Таблица 4.

## Реализации языков программирования на JavaScript

| Язык  | Название реализации               | Основные авторы   |
|---|-----------------------------------|---|
| <a href="#">JavaScript</a>                        | <a href="#">s-mr</a>              | <a href="#">Andrei Formiga</a>  |
| <a href="#">PostScript</a>                        | WPS                               | <a href="#">Tom Hlavaty</a>   |
| <a href="#">PDF</a>                               |                                   |   |
| <a href="#">Ассемблер для MOS Technology 6502</a> | <a href="#">6502asm</a>           | <a href="#">Stian Soreng</a>  |
| <a href="#">Objective-J</a>                       | <a href="#">Cappuccino</a>        | <a href="#">Ross Boucher</a>  |
| <a href="#">Haskell</a>                           | <a href="#">ycr2js</a>            | <a href="#">Tom Shackell</a> , <a href="#">Neil Mitchell</a> ,<br><a href="#">Andrew Wilkinson</a> , <a href="#">Mike Dodds</a> ,<br><a href="#">Bob Davie</a> , <a href="#">Dimitry Golubovsky</a> |
| <a href="#">Prolog</a>                            | <a href="#">Monash Toy Prolog</a> | <a href="#">Lloyd Allison</a>   |
|   | <a href="#">Ioctl</a>             | <a href="#">Jan Grant</a>   |
| <a href="#">Cat</a>                               | <a href="#">Cat Interpreter</a>   | <a href="#">Christopher Diggins</a>   |
| <a href="#">Scheme</a>                            | <a href="#">BiwaScheme</a>        | <a href="#">Yutaka Hara</a>   |
| <a href="#">BASIC</a>                             | <a href="#">Quite BASIC</a>       | <a href="#">Nikko Strom</a>   |
| <a href="#">Lily (англ.)</a>                      | <a href="#">Lily</a>              | <a href="#">Bill Orcutt</a>   |
| <a href="#">Forth</a>                             | <a href="#">wForth</a>            | <a href="#">K Jacobson</a>  |
| <a href="#">PHP</a>                               | <a href="#">Phype</a>             | <a href="#">casperbp</a> , <a href="#">hjelmen0</a>   |

## 3.3 PHP

**PHP** — скриптовый язык программирования общего назначения, который применяется для разработки веб-приложений. В настоящее время этот язык программирования поддерживается подавляющим большинством Hosting-

Providers и является одним из лидеров среди существующих языков программирования, применяемые для создания веб-сайтов.

Популярность в области построения веб-сайтов определяется наличием большого набора встроенных средств для разработки веб-приложений.

*Основные из них:*

- автоматическое извлечение POST и GET-параметров, а также переменных окружения веб-сервера в предопределённые массивы;
- взаимодействие с большим количеством различных систем управления базами данных (MySQL, MySQLi, SQLite, PostgreSQL, Oracle (OCI8), Oracle, Microsoft SQL Server, Sybase, ODBC, mSQL, IBM DB2, Cloudscape и Apache Derby, Informix, Ovrimos SQL, Lotus Notes, DB++, DBM, dBase, DBX, FrontBase, FilePro, Ingres II, SESAM, Firebird / InterBase, Paradox File Access, MaxDB, Интерфейс PDO);
- автоматизированная отправка HTTP-заголовков;
- работа с HTTP-авторизацией;
- работа с cookies и сессиями;
- работа с локальными и удалёнными файлами, сокетами;
- обработка файлов, загружаемых на сервер;
- работа с XForms;

Типы данных

PHP является языком программирования с динамической типизацией, не требующим указания типа при объявлении переменных, равно как и самого

объявления переменных. Преобразования между скалярными типами зачастую осуществляются неявно без дополнительных усилий (впрочем, PHP предоставляет широкие возможности и для явного преобразования типов).

К скалярным типам данных относятся:

- целый тип (`integer`),
- вещественный тип данных (`float`, `double`),
- логический тип (`boolean`),
- строковый тип (`string`), и специальный тип `NULL`.

К нескаларным типам относятся:

- «ресурс» (`resource`),
- массив (`array`),
- объект (`object`),

К псевдотипам относятся:

- `mixed` один или несколько необязательных параметров,
- `number` число (`integer` либо `float`)
- `callback` (`string` или анонимная функция)
- `void` отсутствие параметров

## Глава 4. Построение программы управления спектрометра на основе микроконтроллера STM32F407

В микроконтроллере STM32 имеется четыре типа блоков ТТЛ, АЦП, ЦАП, таймер.

### 4.1 Конфигурация режимов ЦАП

Цифро-аналоговый преобразователь (ЦАП) - это устройство для преобразования цифрового кода в аналоговый сигнал по величине, пропорциональной значению кода.

ЦАП применяются для связи цифровых управляющих систем с устройствами, которые управляются уровнем аналогового сигнала.

*Вводим команды в окно Command Window в MatLab:*

*Настраиваем на 4 ножку ЦАПа*

```
gpio.GPIO_Pin = GPIO_Pin_4;//ЦАП 1
```

*Включаем аналоговый режим*

```
gpio.GPIO_Mode = GPIO_Mode_AN;
```

*Подтяжка выключена*

```
gpio.GPIO_PuPd = GPIO_PuPd_NOPULL;
```

*Устанавливаем состояние порта A*

```
GPIO_Init(GPIOA, &gpio);
```

*Включаем ЦАП*

```
RCC_APB1PeriphClockCmd(RCC_APB1Periph_DAC, ENABLE)
```

*Инициализируем структуру значениями по умолчанию*

```
DAC_StructInit(&dac);
```

*Конфигурируем ЦАП*

```
DAC_Init(DAC_Channel_1,&dac);
```

*Разрешаем работу ЦАП*

```
DAC_Cmd(DAC_Channel_1,ENABLE);
```

*Устанавливаем напряжение на входе  $5V/4096*2048=2.5V$*

```
DAC_SetChannel1Data(DAC_Align_12b_R,2048);
```

## 4.2 Конфигурация режимов АЦП

Аналого-цифровой преобразователь(АЦП) — устройство, преобразующее входной аналоговый сигнал в дискретный код (цифровой сигнал). АЦП — электронное устройство, преобразующее напряжение в двоичный цифровой код.

*Вводим команды в окно Command window в MatLab:*

*Канал -1 выведен на 1 ножку*

```
gpio.GPIO_Pin = GPIO_Pin_1;//АЦП 1
```

*Включаем аналоговый режим*

```
gpio.GPIO_Mode = GPIO_Mode_AN;
```

*Подложка выключена*

```
gpio.GPIO_PuPd = GPIO_PuPd_NOPULL;
```

```
GPIO_Init(GPIOA, &gpio);
```

*Включаем АЦП*

```
RCC_APB2PeriphClockCmd(RCC_APB2Periph_ADC1, ENABLE);
```

*Сбрасываем работу АЦП*

```
ADC_DeInit();
```

*Инициализируем структуру значениями по умолчанию*

```
ADC_StructInit(&adc);
```

*Конфигурируем АЦП*

```
ADC_Init(ADC1, &adc);
```

*Инициализируем структуру значениями по умолчанию*

```
ADC_CommonStructInit(&comadc);
```

*Конфигурируем общие настройки АЦП (сразу для всех 3)*

```
ADC_CommonInit(&comadc);
```

*Возвращаем работу АЦП*

```
ADC_Cmd(ADC1, ENABLE);
```

#### 4.3 Конфигурация режимов ТТЛ выходов

Рассмотрим работу блока микроконтроллера с ТТЛ выходами на примере включения и выключения светодиода.

Сперва необходимо сконфигурировать выходы для светодиодов

*Первым делом следующей командой подаем тактовую частоту на соответствующий блок:*

```
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA, ENABLE);
```

*Затем выбираем в описании микроконтроллера соответствующие ножки - общий выход ножки №12, 13, 14, 15*

```
gpio.GPIO_Pin = GPIO_Pin_12 | GPIO_Pin_13 | GPIO_Pin_14 | GPIO_Pin_15;
```



*Для выбранных 4 ножек включаем следующие настройки:*

*Настраиваем ножки на выход*

```
gpio.GPIO_Mode = GPIO_Mode_OUT;
```

*Устанавливаем тактовую частоту как низкую – 2 МГц*

```
gpio.GPIO_Speed = GPIO_Speed_2MHz;
```

*Тип выхода обычный Push-Pull*

```
gpio.GPIO_OType = GPIO_OType_PP;
```

*Выключаем резистор подтягивающий ножку к напряжению питания*

```
gpio.GPIO_PuPd = GPIO_PuPd_NOPULL;
```

*После того, как структура с настройками сформирована посылаем ее в блок ввода-вывода D.*

```
GPIO_Init(GPIOD, &gpio);
```

*Рассмотрим как пример, включение 1 светодиода, необходимо использовать следующую команду*

```
GPIO_WriteBit(GPIOD,GPIO_Pin_12,Bit_SET); //1 светодиод вкл.
```

*Для его выключения команда изменяется на*

```
GPIO_WriteBit(GPIOD,GPIO_Pin_12,Bit_RESET); //1 светодиод выкл.
```

Таким образом используя приведенную выше последовательность команд можно управлять портами ввода-вывода.

#### 4.4 Пример работы с микроконтроллером в режиме включения- выключения светодиодов

```
% --- Executes on button press in LED1. // Нажатие кнопки LED1
```

Значение COM-порта хранится в глобальной структуре handles

*Возврат в состояние переключателя LED1*

```
comp=handles.comport;
```

*Посылает команду на включение первого диода*

```
if(get(hObject,'Value'))
```

*Посылает команду на выключение первого диода*

```
fprintf(comp,'SET L 1Y \r');
```

```
else
```

*Считывает информацию с COM-порта об подтверждении команды*

```
fprintf(comp,'SET L 1N \r');
```

```
end
```

*Выводит полученную информацию в окно статуса*

```
str=fgets(comp);
```

```
set(handles.Status,'String',str);
```

## Глава 5. Удаленное управление при помощи Web-интерфейса

### 5.1 Web-интерфейс

Web-интерфейсы получили широкое распространение в связи с ростом популярности всемирной паутины и соответственно — повсеместного распространения Web-браузеров.

Web-интерфейс — это совокупность средств, при помощи которых пользователь взаимодействует с веб-сайтом или любым другим приложением через браузер.

### 5.2 Примеры Web-интерфейса

Для управления спектрометрами магнитного резонанса были разработаны 2 вида интерфейса. Один от другого отличается оформлением и количеством изменяемых параметров. Изменение параметров, добавление новых полей и конфигураций можно осуществить без изменения структуры самого кода. Иными словами при изменении интерфейса предыдущие параметры остаются неизменными.

Первый тип разработанного Web-интерфейса выглядит следующим образом:

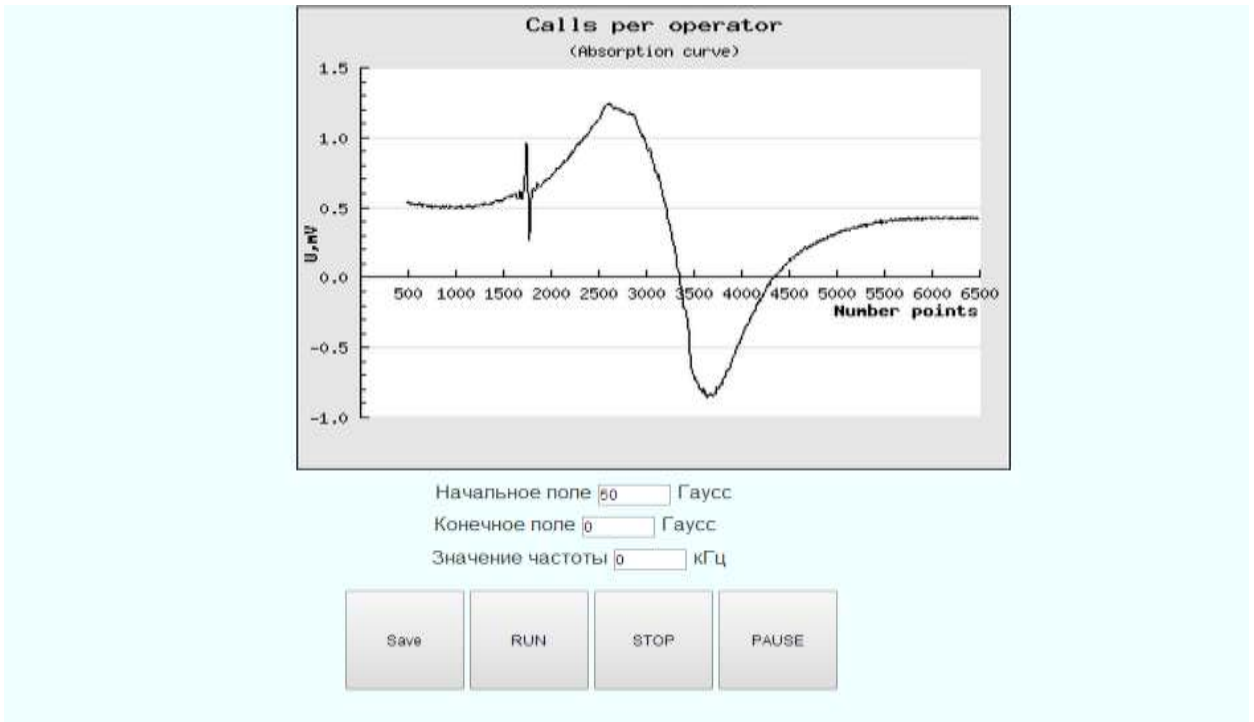


Рисунок 1 – Первый тип Web-интерфейса.

Второй тип Web-интерфейса.

### Будущий спектрометр

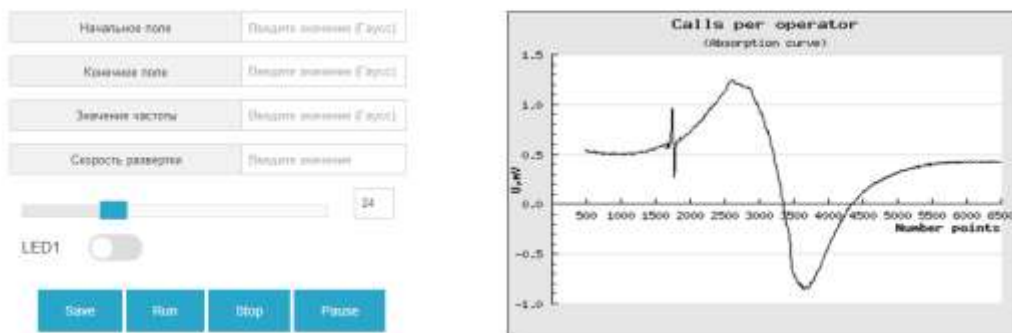


Рисунок 1 – Второй тип Web-интерфейса.

## Заключение

Написана программа микроконтроллера для управления спектрометром магнитного резонанса.

Разработан протокол обмена данными с помощью виртуального COM-порта.

Разработана система удаленного управления спектрометрами магнитного резонанса при помощи Web-интерфейса.

Использование микроконтроллеров семейства STM32 позволяет создать гибкую аппаратно-программную платформу для разработки спектрометров магнитного резонанса.

## Список литературы

1. Описание микроконтроллера STM32F407/417 [Электронный ресурс]. – сайт фирмы STM, 2014:  
[www.st.com/web/en/catalog/mmc/FM141/SC1169/SS1577/LN11](http://www.st.com/web/en/catalog/mmc/FM141/SC1169/SS1577/LN11).
2. Описание работы в MatLab [Электронный ресурс]. – Сайт работы с программой MatLab,2014: [www.mathworks.com](http://www.mathworks.com)
3. Кетков, Ю., Шульц М., Кетков, А. MATLAB 7 программирование, численные методы [Текст]/ Кетков, Ю., Шульц М., Кетков / Санкт-Петербург.- БХВ-Петербург .-2005.
4. Теплов, М. А., Тагиров, М.С., Егоров, А.В., Кудряшов, А.А. Мамин, Г.В. Лабораторная работа. Стационарный ядерный магнитный резонанс в твердых телах. [Текст. А., Тагиров, М.С., Егоров, А.В., Кудряшов, А.А. Мамин, Г.В / Казанский (Приволжский) Государственный университет. -2008.
5. Форум пользователей MatLab [Электронный ресурс]. – Сайт работы с программой MatLab,2014: Описание работы в MatLab [Электронный ресурс]. – Сайт работы с программой MatLab,2014:  
[matlab.exponenta.ru/forum/](http://matlab.exponenta.ru/forum/)
6. Приложения с GUI и дескрипторная графика [Электронный ресурс]. – Сайт работы с приложением GUIDE,2014.  
[matlab.exponenta.ru/gui/book2/1.php](http://matlab.exponenta.ru/gui/book2/1.php)