

Лаборатория электроники. LAB85.RU ^(/)



Ethernet-web мини сервер (телеметрия stm32f100 + enc28j60).

Ethernet UDP

Вход:

User name

Часть 1. Устройство, принцип и реализация.

Часть 2. Добавление авторизации для WEB сервера.

Часть 3. Обновление встроенного ПО (обновление прошивки устройства).

Часть 1. Устройство, принцип и реализация.

Пришел ко мне приятель с просьбой сделать устройство дистанционного измерения и управления

экспериментальным технологическим процессом.

Первоначальные требования были следующие:

- Интерфейс: ethernet web.
- Количество АЦП: 6 шт.
- Количество дискретных выходов: 4 шт.
- Стоимость: как можно меньше (как всегда.....)

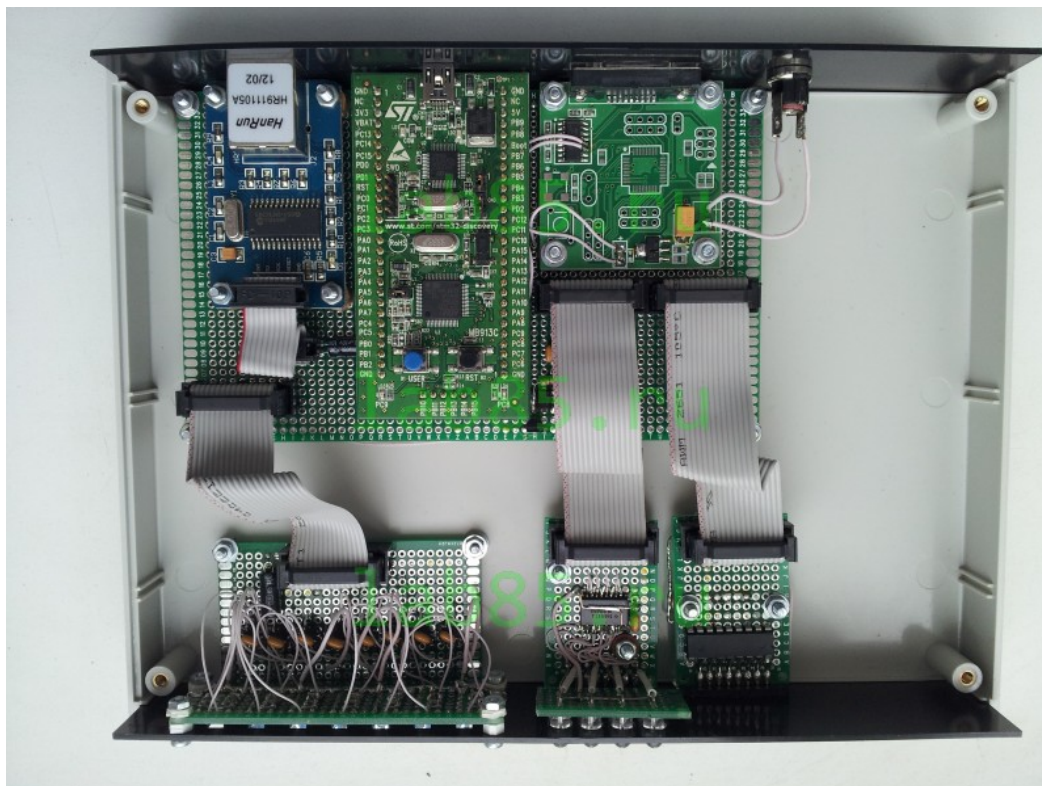
Аппетит как известно приходит во время еды... Процесс пошел, по мере проработки ТЗ добавлялся

новый функционал.

Встал вопрос: **на чем делать ?**

Первоначально хотелось сделать устройство на основе универсальной платы на основе ADSP-BF533, но нечто подобное мне уже довелось делать, решил поднять архивы и применить наработки. Отыскав макет предыдущей версии устройства, было принято решение остановиться на данной реализации.

Макет как он есть...



Основу устройства составляют:

- Контроллер - STM32F100RBT.
- Ethernet трансивер - ENC28J60.
- I2C flash - FM24CL64B.

В окончательном виде устройство приобрело следующий функционал:

- Интерфейс: ethernet web (просмотр значений + управление).
- Количество АЦП: 8 шт (АЦП встроенное в микроконтроллер), также задаются умножитель и делитель для каждого входа.

- Диапазон измеряемых напряжений: от 1 вольта до 100 вольт. Задается внешним резистивным делителем.
- Количество дискретных Выходов: 8 шт.
- Количество дискретных Входов: 8 шт.
- Консоль устройства: RS232 115200 8N1 (настройка устройства).
- I2C flash FM24CL64B: для хранения параметров.

Далее в макет были внесены изменения и частично переписан функционал, как говорится: версия - версии рознь.

Исходники выложены на github.com/lab85-ru/ethernet_tel_server (https://github.com/lab85-ru/ethernet_tel_server), компилятор IAR 6.50.

Устройство работает под управлением ос FreeRTOS и TCP/IP стека uIP.
На устройстве работают две задачи: Консоль и сам TCP стек.

Схема устройства представлена в директории github.com/lab85-ru/ethernet_tel_server/tree/master/sch (https://github.com/lab85-ru/ethernet_tel_server/tree/master/sch):

в формате PCAD2004 (https://github.com/lab85-ru/ethernet_tel_server/blob/master/sch/server_dat.sch), в формате PDF (https://github.com/lab85-ru/ethernet_tel_server/blob/master/sch/server_mini.pdf).

При подаче питания устройство печатает в консоль: Версию софта, железа, имя устройства, а также версию софта по GIT-у и название микросхемы Ethernet контроллера (т.к. были разные вариации) и запускает web интерфейс.

Device start.

Hard Reset From WatchDog....

=====
Soft Version = SV:1.3.18

Hard Version = HV:1.0.12

Device Name = SERVER_DAT

GIT commit = (2018-03-07 b60b2c3)
=====

Ethernet CHIP driver ENC28J60.

Load K for ADC.

DEVICE>ETH init.

Консоль позволяет выполнять команды, список команд печатается при передаче команды ? или **help**.

DEVICE>?

----- MENU -----
help or ? - This help.
view - Views all variables.
version - View version hard+soft.
ipdevice - Set IP this device.
ipgw - Set IP default gateway.
ipmask - Set IP MASK.

DEVICE>

При подаче команды **root**, устройство печатает дополнительные (скрытые) команды.

DEVICE>root

---root command-----
sn - Set serial numer.
txt - Set text.
reboot - reboot device.

Краткое описание команд:

help or ? - Помощь, производится печать доступного списка команд
view - Просмотр состояния устройства
version - печать версий софта и железа
ipdevice - Задать IP адреса устройства
ipgw - Задать IP default gateway
ipmask - Задать маску сети
sn - Задать серийный номер, четыре знака
txt - Строка описание устройства
reboot - Перезагрузка устройства (необходима после изменения параметров сети)

Через консоль производится предварительная настройка устройства. Задаются параметры сети, описание устройства, а также осуществляется просмотр состояния устройства.

Просмотр состояния устройства (команда **view**):

DEVICE>view

```
=====
System uptime: 00 Years, 000 Days, 00:01:05
=====
MAC : 00-01-02-03-04-05
sn = 1212
txt = Макет на основе STM32 Discovery.
IP Device = 192.168.13.200
IP gateway = 192.168.13.1
IP MASK = 255.255.255.0
-----GPIO IN-----
INPUT = 0x00
7 6 5 4 3 2 1 0
0 0 0 0 0 0 0 0
-----GPIO OUT-----
OUTPUT = 0x00
7 6 5 4 3 2 1 0
0 0 0 0 0 0 0 0
-----ADC 12 bit-----
| ADC code: | U adc input: | K= mul/div | Uinput = Uadc * K |
+-----+-----+-----+-----+
| ADC(0x0000) | U adc(0) = 0,0 v | K = 5,0 | U rdiv = 0,0 v |
| ADC(0x0000) | U adc(1) = 0,0 v | K = 5,0 | U rdiv = 0,0 v |
| ADC(0x0000) | U adc(2) = 0,0 v | K = 5,0 | U rdiv = 0,0 v |
| ADC(0x0000) | U adc(3) = 0,0 v | K = 5,0 | U rdiv = 0,0 v |
| ADC(0x0000) | U adc(4) = 0,0 v | K = 5,0 | U rdiv = 0,0 v |
| ADC(0x0000) | U adc(5) = 0,0 v | K = 5,0 | U rdiv = 0,0 v |
| ADC(0x0000) | U adc(6) = 0,0 v | K = 5,0 | U rdiv = 0,0 v |
| ADC(0x0000) | U adc(7) = 0,0 v | K = 5,0 | U rdiv = 0,0 v |
+-----+-----+-----+-----+
```

Для работы устройству необходимо задать настройки сети !

После того как было произведена установка значений параметров TCP/IP сети, и произведена перезагрузка устройства, становится доступен WEB интерфейс.

Web интерфейс представлен четырьмя страницами.

Главная страница:

Содержит меню с переходом на другие страницы (просмотр состояние входов выходов, изменение состояние выходов, установка параметров: текстовое описание входов/выходов и умножителей и делителей).



Страница **Текущее состояние входов/выходов** - На данной странице можно просмотреть текущее значение состояний входов и выходов.

Состояние Входов/ВЫходов.

< Назад в главное меню

Изменение состояния выходов >

Аналоговые Входы

№	U вх.АЦП	* К	U дел.	Описание
0	0,0	5,0	0,0	adc 0
1	0,0	5,0	0,0	adc 1
2	0,0	5,0	0,0	adc 2
3	0,0	5,0	0,0	adc 3
4	0,0	5,0	0,0	adc 4
5	0,0	5,0	0,0	adc 5
6	0,0	5,0	0,0	adc 6
7	0,0	5,0	0,0	adc 7

Цифровые Входы

№	Знач.	Описание
0	0	input
1	0	input
2	0	input
3	0	input
4	0	input
5	0	input
6	0	input
7	0	input
1 - Вх.= +Vin, 0 - Вх.= 0 v		

Цифровые ВЫходы

№	Знач.	Описание
0	0	output
1	0	output
2	0	output
3	0	output
4	0	output
5	0	output
6	0	output
7	0	output
1-Замкнут, 0-Разомкнут		

Страница **Изменение состояния выходов** - На данной странице можно произвести изменения

состояний ВЫходов.

Также для удобства на данной странице выводятся таблицы с значениями всех входов (АЦП и дискретных). Это позволяет посмотреть значения входов без перехода на предыдущую страницу.

Состояние Входов/ВЫходов.

< Назад в главное меню

Аналоговые Входы

№	U вх.АЦП	* К	U дел.	Описание
0	0,0	5,0	0,0	adc 0
1	0,0	5,0	0,0	adc 1
2	0,0	5,0	0,0	adc 2
3	0,0	5,0	0,0	adc 3
4	0,0	5,0	0,0	adc 4
5	0,0	5,0	0,0	adc 5
6	0,0	5,0	0,0	adc 6
7	0,0	5,0	0,0	adc 7

Цифровые Входы

№	Знач.	Описание
0	0	input
1	0	input
2	0	input
3	0	input
4	0	input
5	0	input
6	0	input
7	0	input

Цифровые ВЫходы

№	Знач.	Описание	Уст.
0	0	output	<input type="checkbox"/>
1	0	output	<input type="checkbox"/>
2	0	output	<input type="checkbox"/>
3	0	output	<input type="checkbox"/>
4	0	output	<input type="checkbox"/>
5	0	output	<input type="checkbox"/>
6	0	output	<input type="checkbox"/>
7	0	output	<input type="checkbox"/>
Очистка, сброс в состояние 0 всех выходов 1-Замкнут, 0-Разомкнут			<input type="checkbox"/>

Изменить состояние выходов

Страница **Конфигурация устройства** - На данной странице производится установка значений

множителя и делителя для каждого из входов АЦП, а также задается текстовое описание входов

и выходов.

Конфигурация устройства.

< Назад в главное меню

MAC: 00-01-02-03-04-05

IP адрес устройства: 192.168.13.200

Маска сети: 255.255.255.0

IP адрес роутера(gateway): 192.168.13.1

Описание устройства: Макет на основе STM32 Discovery

Сохранить

Описание назначения/подключений для аналоговых Входов.

Вход	Множ.	Делит.	К=Множ./Делит.	Описание	Сохранить
АЦП 0	5	1	5,0	adc 0	Сохранить
АЦП 1	5	1	5,0	adc 1	Сохранить
АЦП 2	5	1	5,0	adc 2	Сохранить
АЦП 3	5	1	5,0	adc 3	Сохранить
АЦП 4	5	1	5,0	adc 4	Сохранить
АЦП 5	5	1	5,0	adc 5	Сохранить
АЦП 6	5	1	5,0	adc 6	Сохранить
АЦП 7	5	1	5,0	adc 7	Сохранить

Описание назначения/подключений для цифровых Входов.

Вход	Описание	Сохранить
Вх. 0	input	Сохранить
Вх. 1	input	Сохранить
Вх. 2	input	Сохранить
Вх. 3	input	Сохранить
Вх. 4	input	Сохранить
Вх. 5	input	Сохранить
Вх. 6	input	Сохранить
Вх. 7	input	Сохранить

Описание назначения/подключений для цифровых Выходов.

Выход	Описание	Сохранить
Вых. 0	output	Сохранить
Вых. 1	output	Сохранить
Вых. 2	output	Сохранить
Вых. 3	output	Сохранить
Вых. 4	output	Сохранить
Вых. 5	output	Сохранить
Вых. 6	output	Сохранить
Вых. 7	output	Сохранить

Поля: **Множ.** и **Делит.**

Значения **Множ.** и **Делит.** задаются по следующему принципу:

ВАРИАНТ 1. Измерение больших величин напряжения (более 3 вольт).

При измерении высокого напряжения (более 3 В), необходимо применять на входе **резистивный**

делитель, так как вход микроконтроллера рассчитан на работу с входным напряжением не более 3 вольт.

Резистивный делитель имеет коэффициент деления **К**. Устройству значение коэффициента **К** задается в виде двух коэффициентов **Множитель** и **Делитель** на странице конфигурации устройства.

Значение коэффициента **К** рассчитывается по формуле:

$$K = \text{Множ} / \text{Делит}$$

Таким образом величина входное напряжение (на входе резистивного делителя) с учетом резистивного делителя, определяется по формуле:

$$U_{вх} = U_{АЦП} * K$$

или

$$U_{вх} = U_{АЦП} * \text{Множ} / \text{Делит}$$

Итого, в случаи применения резистивного делителя соотношения величин:

Множ < Делит, соответствует $K < 1$.

ВАРИАНТ 2. Измерение малых величин напряжений (менее 1 вольта и т.п.).

При измерении малых величин напряжений, необходимо применять на входе **усилитель**.

Усилитель имеет коэффициент умножения **К**. Расчетные формулы остаются те же, меняется только значения для **Множ** и **Делит**.

Итого, в случаи применения усилителя получаем следующее соотношение:

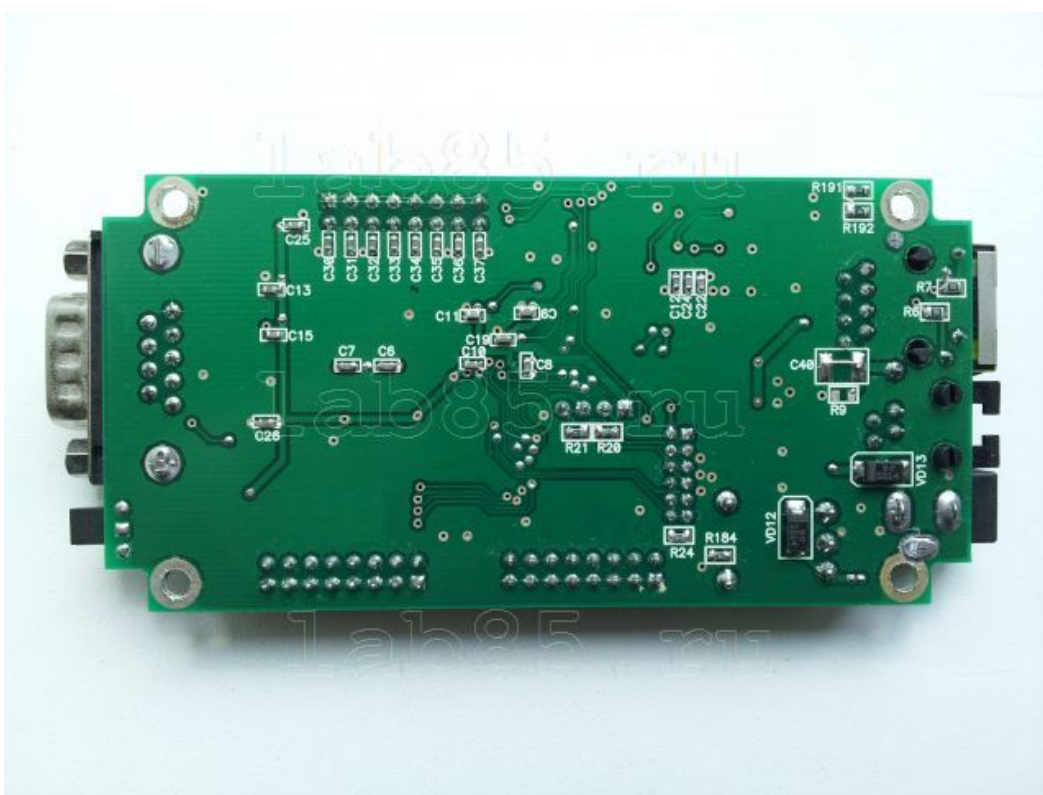
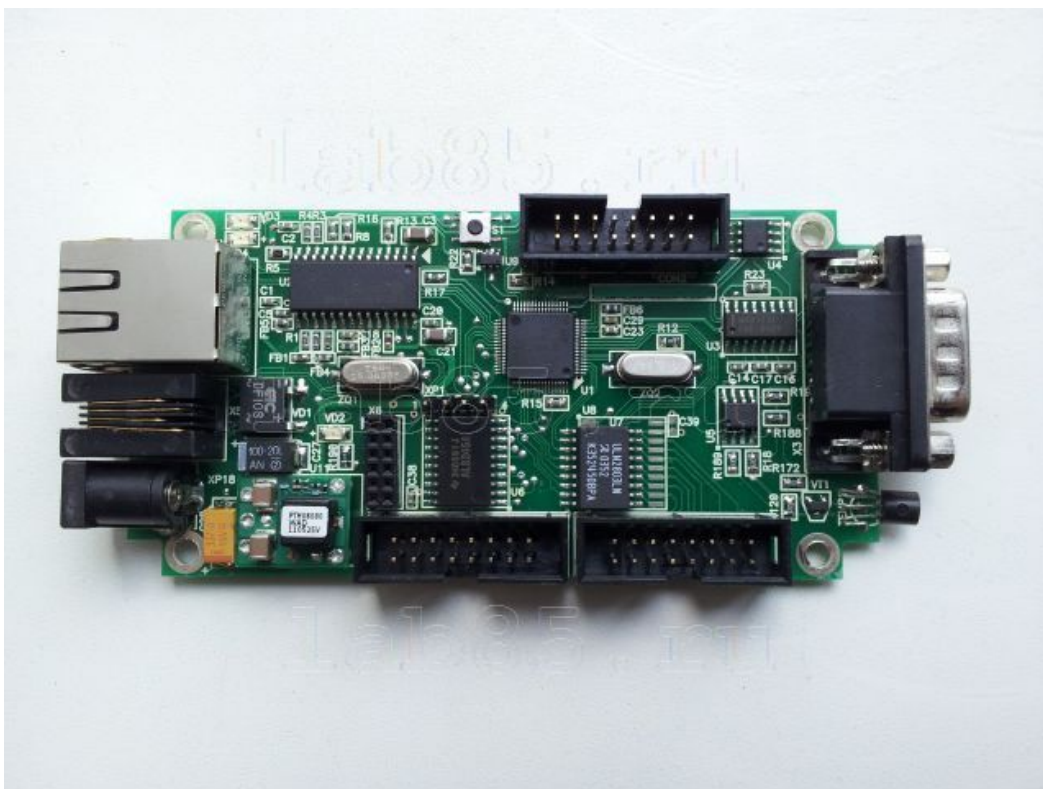
Множ > Делит, соответствует $K > 1$.

Тестовое описание входов/выходов.

Устройство создавалось как универсальное, поэтому для удобства были введены текстовые поля для

описания назначения каждого входа/выхода отдельно.

После того как проект устаканился было принято решение о выпуске печатной платы, что и было сделано.



Достоинства:

Применялись наработки от предыдущего проекта, что позволило сэкономить время и силы.
Минимализм: микроконтроллер, стек.

Недостатки:

Стек uIP рассчитан на применение в устройствах с минимальным количеством памяти...
Как следствие из-за особенностей реализации стек достаточно медленный.

Решение данной проблемы перейти на более продвинутый контроллер (с большим количеством памяти и аппаратным MAC) и использовать другой стек (Lwip например).

На будущее есть, что улучшить. Прогресс не стоит на месте, микроконтроллеры дешевеют

при этом количество периферии на борту растет... это приятно.

Данным устройством заказчик доволен, а это главное ! :)

Часть 2. Добавление авторизации для WEB сервера.

ВНИМАНИЕ: Исходников для новой версии в открытом доступе не будет, только теоретическая часть.

Заказчик попросил добавить авторизацию для WEB сервера.

Есть такой недостаток, все кому не лень могут войти на сервер и изменять состояние входов/выходов

устройства и т.п. Так не должно быть.

Первоначально решение проблемы отсутствия авторизации решалось путем выделения группы

устройств своей подсети, но как говориться это костыль и только.

Еще:

В микроконтроллере STM32F100RB мало ОЗУ (всего 8К), а хотелось бы развернуться, но для

этого необходимо больше ОЗУ.

На момент создания первой версии проекта это был 2010 год, микроконтроллер STM32F100RB

был достаточно дешевый поэтому и был задействован, но время идет появляются новые микроконтроллеры и за ту же стоимость уже можно приобрести микроконтроллер с большим

количеством ОЗУ и Flash памяти.

Из ближайших pin-to-pin совместимых аналогов был выбран микроконтроллер STM32F103RE.

Данный микроконтроллер имеет на борту 64К ОЗУ и 512К flash (против 8к-128к у STM32F100RB).

При этом стоимость STM32F103RE составляет меньше или как у STM32F100RB.

Так как STM32F103RE и STM32F100RB pin-to-pin совместимы нет необходимости вносить изменения в печатную плату. Достаточно просто перепаять микроконтроллер !

Поэтому пришла идея заменить микроконтроллер на STM32F103RE и переписать софт с применением стека LwIP.

Добавление авторизации.

Принцип:

- Авторизация будет производиться, по логину и паролю.
- При успешном входе на сервер, каждой странице сервера будет передаваться строка (в виде CGI параметра) с кодовой комбинацией.
- Кодовая комбинация будет уникальна для каждой сессии.

- Кодовая комбинация будет проверяться на каждой странице, в случае отсутствия или не совпадения кода со значением, которое хранится в памяти устройства будет, производится автоматический редирект на страницу авторизации.

Реализация:

Вводим два дополнительные параметра, логин и пароль для входа на сервер.

Логин и пароль будут храниться в памяти устройства. С консоли задаем значение логина и пароля.

По паролю рассчитывается хеш по алгоритму MD5. Затем сохраняем MD5 хеш в виде строки в конфигурационной памяти устройства.

Создаем новую страницу index.shtml (прежнюю переименовываем), на которой добавляем форму

авторизации для ввода логина, пароля. Также добавляем на страницу java-script, который будет производить расчет хеша MD5 для введенного пользователем пароля.

После ввода логина и пароля, по средствам CGI (в строке браузера) устройству пересылаются

введенный пользователем логин и рассчитанный хеш MD5 от пароля. Устройство проверяет на совпадение логина и хеш MD5(пароль), если НЕ сошлось, то выводит сообщение об ошибке и ожидает повторного ввода логина и пароля.

Если комбинация логин + MD5(пароль) совпали, то устройство генерит код, который будет передаваться всем страницам устройства в виде CGI параметра, (Пример: ?log_c=a50e0121b). Затем выполняется редирект на страницу меню устройства.

Итого: при переходе на любую страницу, устройство(сервер) получает помимо запроса страницы

еще и CGI параметры, в которых находится текущий код авторизации. Сервер производит сравнение

кода пришедшего в CGI с кодом, который храниться во внутренней переменной устройства.

Если кода совпадают, то продолжает работу. Если кода не совпадают то, делает редирект на страницу

авторизации index.shtml.

Еще на все страницы устройства добавляется кнопка ВЫХОДА(logout-та).

В устройстве реализована функция выхода(logout-та) по таймауту равному 10 мин, в случае если

пользователь бездействует или покинул сервер не выполнив logout.

В результате:

Нельзя просто взять и подменить строку с ссылкой на другую страницу в браузере (прямой переход на

страницу устройства, если известно имя страницы перехода), чтобы обойти авторизацию.

Нельзя подменить код. Так как при каждом новом входе код генериться заново, т.е. код подобрать и

угадать нельзя.

Достоинства:

- Простая реализация, (подходит как для uip так и для lwip стеков)

- Пароль посылается в закрытом виде (хотя на сегодняшний день считается, что MD5 не стойкий алгоритм, и не рекомендуют его применять, и тем не менее для данного устройства это вполне допустимо).
- Каждая сессия защищена уникальным кодом.

Недостатки:

- Данная реализация поддерживает только одного пользователя (при желании можно расширить на несколько пользователей).
- При открытии второй, третьей и т.д. сессии произойдет сброс текущей сессии и автоматический logout текущего пользователя.

Часть 3. Обновление встроенного ПО (обновление прошивки устройства).

Теперь необходимо обзавестись функцией дистанционного обновления ПО.

Обновление прошивки будет производиться для удобства через WEB интерфейс.

Прошивка перед загрузкой в устройство предварительно подготавливается, т.е. при помощи специальной программы на ПК в начало прошивки вставляется специальный заголовок в котором

прописана сигнатура (для загрузчика), длинна и контрольная сумма прошивки.

Заголовок необходим для того, чтобы нельзя было обновить ПО любым файлом и тем самым "испортить" устройство.

Прошивка будет загружаться во внутреннюю flash память микроконтроллера, в специальную область

предназначенную для промежуточного хранения прошивки.

Затем устройство перезагружается, стартует загрузчик, который проверяет корректность прошивки и

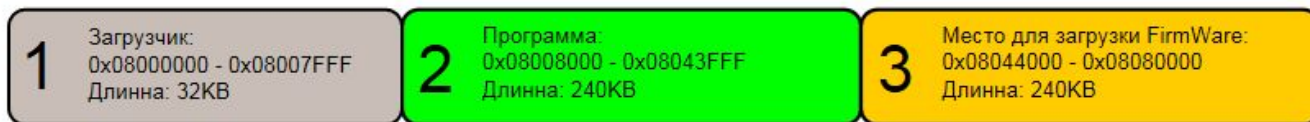
далее загрузчик обновляет прошивку и затем передает управление основной программе.

Все просто....

Для реализации обновления ПО необходимо внести следующие дополнения в проект.

Разделим flash память микроконтроллера на 3 области.

Карта Flash памяти микроконтроллера STM32F103RE: 0x08000000 - 512KB



1 - Начало flash памяти, место расположения загрузчика.

2 - Место для основной программы, эту часть будет обновлять загрузчик.

3 - Промежуточный буфер для хранения файла прошивки перед обновлением.

Загрузчик.

Загрузчик располагается в начале памяти 1.

Загрузчик отвечает за обновление ПО (копирование из 3 в 2) и запуск основной программы 2.

Принцип работы.

При старте системы загрузчик получает управление и проверяет наличие прошивки в области 3.

Наличие прошивки проверяется по следующим признакам:

1. Наличие сигнатуры.
2. Корректность длинны прошивки.
3. Рассчитывается контрольная сумма загруженной прошивки и сравнивается с контрольной суммой из заголовка.

В случаи наличия прошивки:

- Загрузчик производит копирование прошивки из области 3 в область 2.
- Выполняется стирание области 3, (защитная мера от повторного обновления ПО).
- Производится передача управления в область 2.

В случаи отсутствия прошивки:

Загрузчик передает управление на основную программу в область 2.

ВНИМАНИЕ: Выбор места расположения загрузчика.

Первоначально загрузчик было решено размещать в **старших адресах** и переходить на него только при обновлении ПО, позже было принято решение разместить загрузчик в начале памяти.

Для обновления ПО необходимо загрузить прошивку и осуществить переход на загрузчик.

Перед тем как передавать управление загрузчику необходимо выполнить следующие действия:

1. Отключить прерывания. В загрузчике своя таблица прерываний и свои обработчики.
2. Переключить таблицу векторов из области ПО 2 в область расположения загрузчика 1.
3. Загрузить указатель стека значением из таблицы векторов загрузчика.
4. Сбросить(отключить - Delnit) используемую периферию. Так как загрузчик отдельная программа(проект), не зависящая от основной, то загрузчик производит инициализацию необходимой периферии под себя.
5. Нельзя допустить бесконтрольную работу периферии, это может привести к краху системы.
5. Выполнить переход на загрузчик.

Тонкости работы ядра микроконтроллера и передача управления загрузчику.

Ядро микроконтроллера Cortex-M3 может находиться в следующих состояниях: привилегированном(main), непривилегированном(process) и исключении.

При старте микроконтроллер находится в привилегированном состоянии main.

В привилегированном состоянии, у ядра есть доступ ко всем ресурсам.

В непривилегированном состоянии (process) нет доступа к части ресурсов и таким образом производится разграничения доступа и защита от не преднамеренного доступа к ресурсам.

В каждом состоянии используется свой указатель стека SP, т.е получается два указателя

стека SP_main и SP_process.

Теперь о работе программы под управлением FreeRTOS.

В программе производится настройка периферии, запуск ядра ОС и запуск задач на исполнение.

Первоначально на этапе инициализации ядро микроконтроллера работает в режиме main, затем ОС производит переключение режима работы ядра в process и все задачи работают в состоянии process.

Итого получается, загрузить новое значение в указатель стека SP_main нельзя, т.к. ядро микроконтроллера находится в состоянии process. Из состояния process нет доступа к SP_main, и в данный момент используется указатель стека SP_process.

Переключить таблицу векторов тоже нельзя т.к. нет доступа к регистру SCB->VTOR из состояния process.

Для того чтобы получить доступ к данным регистрам необходимо переключиться в привилегированный режим, и для этого необходимо вызвать исключение и уже от туда производить все манипуляции.

Программное прерывание занято ОС, и модифицировать его нет желания. Вызывать фиктивное

(свое) прерывание или встраиваться в уже имеющиеся, как-то сложно и не красиво..... плюс надо

будет сбросить контроллер прерываний.....

Сложно получается, проще поступить по другому. А именно: загрузчик располагается в начале flash памяти.

При старте системы первым стартует загрузчик, который в зависимости от необходимых действий передает управление на основную программу если нет необходимости обновлять ПО или производится обновление ПО.

В основной программе: Принимается файл и записывается во flash, затем производится перезапуск микроконтроллера. Все ! Дальше будет работать загрузчик.

Осуществить переход из загрузчика на основную программу проще. Так как ядро находится в состоянии main и нечто не мешает выполнить все необходимые подготовительные манипуляции,

с переходом на основную программу. Процесс перехода становится тривиальной задачей.

Для перехода из основной программы на загрузчик достаточно произвести программный сброс

микроконтроллера, при этом вся периферия будет автоматически переведена в первоначальное

состояние, то есть состояние сброса. После этого микроконтроллер автоматически попадает в загрузчик.

Изменения и дополнения в проекте.

1. Добавлена страница с формой для загрузки файла прошивки при помощи метода POST.

2. В алгоритм обработки http запросов добавить обработку метода POST, а также прием и запись файла во внутреннюю flash память.
 3. Дополнительный проект Загрузчика.
 4. Дополнительное ПО для ПК, генератор заголовка для прошивки.
-