Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«Воронежский государственный университет»

Физический факультет

Кафедра электроники

**Разработка системы управления освещением на основе**

**микроконтроллера**

ВКР бакалаврская работа

03.03.03 «Радиофизика»

Профиль «Информационные системы и технологии»

Допущено к защите в ГЭК \_\_.\_\_.\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| Обучающийся |  |  | В. А. Сафонов |
| Руководитель |  |  | А. С. Жабин |
| Зав. кафедрой |  | д.ф.м . н., профессор | А. М. Бобрешов |

Воронеж

2018

**Содержание**

Введение……………………………………………………………..…………….....3

1.Теоретическая часть………………………………………………..……………...4

1.1.Выбор и обоснование компонент системы………………...……….…...4

1.2. Подробное описание компонент системы................................................5

1.2.1. Микроконтроллер STM32F100..............………….………….....5

1.2.2. Фотодатчик KY-018.................………...………………………..6

1.2.3. Wi-fi модуль esp8266-01....………...……………………………7

1.2.4. Датчик движения HC-SR501.....………………………………...8

1.2.5. Радиомодуль CC1101........…...………………………………….9

1.2.6. Широтно-импульсная модуляция..............................................14

1.3. Языки программирования........................................................................15

2. Разработка системы...............................................................................................16

2.1. Функционал системы...............................................................................16

2.2. Описание устройства...............................................................................16

2.3. Алгоритмы работы системы...................................................................17

2.4. Структурные схемы устройств...............................................................18

2.5. Выводы по разделу 2...............................................................................19

3. Экономическая часть.............................................................................................20

3.1. Анализ существующих систем на рынке...............................................20

3.2. Расчет себестоимости...............................................................................20

3.3. Выводы по разделу 3................................................................................21

Заключение.................................................................................................................22

Список литературы....................................................................................................23

Приложение А. Схема...............................................................................................24

Приложение Б.Рисунки.............................................................................................25

Приложение В.Листинг.............................................................................................29

**Введение**

Результатом моей бакалаврской работы стала система управлением освещением на основе микроконтроллер stm32f100. Её плюсами является простота эксплуатации и дешевизна, но в современном мире с его тенденциями к экономичности, экологичности и энергоэффективности этого явно недостаточно. Отсюда и вытекает цель моей текущей работы: модернизировать систему так, чтобы она стала более интеллектуальной и энергоэффективной, и, как следствие, более экологичной. При этом она также должна оставаться простой в эксплуатации и дешевой относительно аналогичных разработок. Достигнуть этого мы попробуем при помощи автоматической регулировки яркости освещения и использования более энергоэффективной элементной базы, а также, собирая дополнительную информацию при помощи беспроводной сенсорной сети (WSN), и дополнительной обработки этой информации. Т.е. от нас потребуется не только модернизация технической части системы, но и разработка новых алгоритмов обработки информации. В результате мы должны получить систему освещения, позволяющую эффективно использовать энергию для освещения помещений. Выгодность данного проекта подтверждается отчетом Navigant Research, согласно которому мировой доход от сетевых средств контроля освещения будет расти с 1,7 млрд. долларов США в год в 2013 году до более чем 5,3 млрд. долларов США к 2020 году

1. **Теоретическая часть**
   1. **Выбор и обоснование компонент системы**

Основой нашей системы станет моя дипломная работа. Для дальнейшей работы нам необходимо провести ревизию данного устройства и установить, что необходимо заменить и добавить для достижения целей, поставленных во введении.

Управлять по-прежнему будем светодиодными лампами т.к. они остаются наиболее безопасными и экономичными вариантами освещения. Также радует простота управления таким освещением.

Управляющим центром системы останется микроконтроллер от компании STMicroelectronics на базе ядра Cortex-M3. Этот микроконтроллер показал себя экономичным, простым в использовании и надежным. К тому же большим плюсом является наличие у меня опыта работы с данным микроконтроллером, а наличие режима низкого энергопотребления делает его незаменимым для наших целей.

Для связи с устройством использовался wi-fi модуль esp8266 – 01. Этот модуль имел на своем борту отдельный встроенный микроконтроллер с 32-bit процессором Tensilica. Этот модуль работал на протоколе IEEE 802.11 b/g/n Wi-Fi. Было принято решение от него отказаться в пользу протокола ZigBee т.к. он надежен и имеет крайне низкое энергопотребление. Большим плюсом является возможность построения самоорганизующейся и самовосстанавливающейся ячеистой топологии сети, что необходимо для построения беспроводной сенсорной сети (WSN). Следовательно, использовать будем другой модуль, с поддержкой протокола ZigBee. Мой выбор пал на СС1101 от фирмы Texas Instruments т.к. это дешевый передатчик с малым энергопотреблением. Трансивер CC1101 отличается улучшенными РЧ-характеристиками (расширенным диапазоном частот, повышенной избирательностью, улучшенным входным уровнем насыщения и возможностью управления выходной мощностью) и характеристиками электропотребления. Благодаря малому потребляемому току, отличным РЧ-характеристикам и высокой степени интеграции он прекрасно подходит для реализации беспроводных компонентов систем. Но в нашей сети найдется и место для esp8266 – 01, что будет следствием компромисса между энергоэффективностью и удобством использования

В роли датчика освещенности использовался фотодатчик KY-018, и я не нашел причин для его замены на что-то другое. Это простое и дешевое устройство, основанное на фоторезисторе: Чем ярче освещен фоторезистор, тем ниже его сопротивление. При подключении к АЦП микроконтроллера мы сможем измерять уровень освещенности и предпринимать какие-либо действия исходя из этих измерений.

Я пришел к выводу, что для более интеллектуального управления освещением нам необходимо добавить датчик движения. Мой взгляд пал на PIR Motion sensor HC-SR501. Подробнее его характеристики будут рассмотрены во второй части. Однако уже сейчас можно сказать, что основными причинами для его выбора были дешевизна, распространенность, небольшой размер и низкое энергопотребление. А это основные качества необходимые нам.

Разработка всего ПО для системы по-прежнему будет вестись на языке C с использованием библиотеки Standard Peripheral Library от компании STMicroelectronics. Это заметно ускорит разработку и упростит чтение кода, а возможное снижение эффективности (в сравнении с ассемблером) будет не заметно на 32-битном процессоре Cortex-M3. Для настройки esp8266-01 будет использоваться комбинация АТ – комманд.

* 1. **Подробное описание компонент системы**
     1. **Микроконтроллер STM32F100**

STM32F100 младшая линейка дешевых микроконтроллеров от компании STMicroelectronics на базе ядра Cortex-M3. Микроконтроллеры включают в себя широкий набор интерфейсов и большой объем встроенной памяти: ядро Cortex-M3 с частотой процессора до 24 МГц, Flash до 512 кБ, до 32 кБ RAM, большее количество таймеров, часы реального времени (RTC), до 5 UART, до 2 I2C, до 3 SPI, 12-битный АЦП и 12-битный ЦАП, встроенный температурный датчик, а также контроллер внешней памяти (EMC). Микроконтроллеры семейства STM32F100х («Value Line») предназначены для различных крайне чувствительных к стоимости применений, где возможностей 16-битного микроконтроллера уже недостаточно, а функциональность обычных 32-битных микроконтроллеров избыточна. Выпускаются в корпусах: LQFP48, LQFP64, TFBGA64, LQFP100, LQFP144.

Основные характеристики линейки STM32F100х («Value Line»):

* Максимальная тактовая частота 24 МГц (30 DMIPS)
* Умножение и деление за 1 такт
* Напряжения питания 2.0 – 3.6 В
* От 4 до 8 Кб ОЗУ
* От 16 до 128 Кб флэш-памяти
* Два встроенных и откалиброванных тактовых генератора на 40 КГц и 8 МГц
* 7-канальный DMA контроллер
* 16-канальный 12-битный АЦП (1.2 мкс) с датчиком температуры
* Два 12–битных ЦАП
* До 80 быстрых портов ввода – вывода (есть совместимость с 5 В)
* 16 внешних прерываний
* Два сторожевых таймера (IWDG и WWDG)
* До 10 таймеров общего и расширенного назначений
* До 2х I2C(SMBus/PMBus), до 3х USART (Lin, IrDa, modem control), до 2 SPI(2 Мбит/с), HDMI (CEC), RTC
* Управление питанием и сбросом (3 режима низкого потребления, PVD, BOR)
* Аппаратный расчет CRC
* 96–битный уникальный идентификатор (ID)
  + 1. **Фотодатчик KY-018**

Принцип работы данного датчика довольно прост: Чем ярче освещен фоторезистор, тем ниже его сопротивление. Сопротивление фоторезистора при изменении освещенности меняется в широких пределах от единиц кОм и до сотен кОм или МОм. Точно выяснить изменение сопротивление фоторезистора следует экспериментально с помощью омметра. Контакты и схема модуля KY-018 позволяют использовать только фоторезистор или фоторезистор в составе делителя напряжения. Схема устройства на рисунке 2 в приложении Б.

Для этого на плате установлен резистор 10 кОм.  
Питание модуля подают на контакт +5 В. С увеличением освещенности на выходе модуля фоторезистора напряжение будет падать, при ярком свете напряжение выхода будет около половины напряжения питания. Величина напряжения на выходе зависит от типа фоторезистора. В темноте напряжение выхода будет близко к напряжению контакта +5 В.

При работе совместно с МК выход модуля фоторезистора соединяют с входом АЦП микроконтроллера. Так как изменение сопротивления фоторезистора при освещении значительно, то с помощью АЦП можно легко фиксировать наступление темноты или включение освещения. Схема АЦП МК STM32F100 представлена на рисунке 5 в приложении Б.

* + 1. **Wi-fi модуль esp8266 - 01**

ESP8266 — микроконтроллер китайского производителя Espressif с интерфейсом Wi-Fi. Помимо Wi-Fi микроконтроллер отличается возможностью исполнять программы из внешней флеш-памяти с интерфейсом SPI.

Основное применение ESP8266 находит в управлении разнообразными бытовыми приборами через беспроводные сети. Концепцию такого управления часто называют «Internet of Things» (IoT, «интернет вещей»). Верхний уровень IoT представлен разнообразными приложениями под популярные платформы (Android, iOS, Windows). Эти приложения позволяют разработчику прибора адаптировать приложение под управление его прибором и передать пользователю готовое решение.

Всего существует 13 модификаций данного модуля от esp8266-01 до esp8266-13. Они отличаются размерами, антеннами, наличием или отсутствием экрана, а также количеством и назначением ножек на плате.

На плате имеется свой встроенный микроконтроллер с параметрами:

* 80 MHz 32-bit процессор Tensilica L106. Возможен негарантированный разгон до 160 МГц.
* IEEE 802.11 b/g/n Wi-Fi. Поддерживается WEP и WPA/WPA2.
* 14 портов ввода-вывода(из них возможно использовать 11), SPI, I²C, I²S, UART, 10-bit АЦП.
* Питание 2,2…3,6 В. Потребление до 215 мА в режиме передачи, 100 мА в режиме приема, 70 мА в режиме ожидания. Поддерживаются три режима пониженного потребления, все без сохранения соединения с точкой доступа: Modem sleep (15 мА), Light sleep (0.4 мА), Deep sleep (15 мкА).

Но в ходе данной работы использовать непосредственно этот микроконтроллер мы не будем, так как у нас уже есть внешний программируемый микроконтроллер stm32f100.

Модуль esp8266-01 является простейшей модификацией платы esp8266. Она отличается простотой в использовании, полосковой антенной (PCB antenna) с радиусом действия до 400 метров на открытой местности. Модуль управляется АТ командами.

* + 1. **Датчик движения HC-SR501**

Модуль представляет собой датчик движения человека. При вхождении человека в зону обзора датчика фиксируется присутствие. Принцип работы модуля HC-SR501 заключается в регистрации инфракрасного излучения от подвижного объекта. Чувствительный элемент – пироэлектрический датчик 500BP. Он состоит из двух элементов заключенных в одном  
корпусе. Чувствительный элемент закрыт белым куполом – линзой Френеля. Особенности линзы Френеля таковы, что инфракрасное излучение от подвижного объекта попадает сначала на один элемент датчика 500BP, затем на другой. Электроника модуля HC-SR501 регистрирует поочередное поступление сигналов от двух элементов из состава 500BP и при фиксации движения выходная цепь модуля формирует логический сигнал.

Технические характеристики “HC-SR505”:

* Напряжение питания постоянного тока, В: 4.5-20
* Потребляемый ток (в покое), мкА: < 60
* Потребляемый ток (при срабатывании), мкА: < 80
* Выходной уровень, В: высокий 3.3 / низкий 0
* Длительность высокого уровня на выходе после срабатывания, С: 8 ± 30%
* Угол зоны обнаружения, градусов: < 100
* Дальность обнаружения, м: 2-3
* Рабочая температура: -20…+80
* Диаметр линзы датчика, мм: 10
* Размеры платы, мм: 10\*23
  + 1. **Радиомодуль СС1101**

Малогабаритный и малопотребляющий модуль, основанный на базе приёмопередатчика CC1101 от Texas Instruments. Модуль предназначен для установки в изделия.

Основные возможности:

* Рабочая частота 868 MHz
* Регулируемая мощность до 10 dBm.
* Типы модуляции: GFSK/2-FSK/ASK/OOK/MSK
* Скорость передачи до 500 Kbps.
* Интерфейс SPI
* Напряжение питания 1.8В - 3.6В
* Коррекция ошибок FEC.
* Цифровой измеритель RSSI.
* «Пробуждение» по радио (Wake-on-radio)
* Встроенный термодатчик.

Технические характеристики при температуре Ta=25°C, VCC=3.3В,

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование параметра | Минимальное значение | Номинальное значение | Максимальное значение |
| Частотный диапазон, МГц | 300 387 779 | — — — | 348 464 928 |
| Рабочий температурный диапазон, °С | -40 | — | 85 |
| Рабочее напряжение питания, В | 1,8 | — | 3,6 |
| Скорость передачи (программируемая), кбод | 1,2 | — | 500 |
| Выходная мощность (программируемая), дБм | -30 | — | +12 |
| Чувствительность приемника1), дБм | — | -113 | — |
| Потребляемый ток: – режим приема2), мА – режим передачи (0 дБм), мА – режим передачи (12 дБм), мА – режим отключения, мкА | — — — — | 14,7 15,0 30,0 <1 | — — — — |

Из таблицы видно, что потребляемый ток у СС1101 в режиме передачи меньше более чем в 14 раз (15мА против 215мА) и в более чем в 6 раз при приеме (14.7мА против 100мА). Схема устройства на рисунке 4 в приложении Б.

* + - 1. **Описание протокола связи ZigBee**

Основой протокола ZigBee является стандарт IEEE 802.15.4. В отличие от других беспроводных технологий, где ставится задача обеспечить высокую скорость передачи, большую дальность или высокое качество обслуживания, ZigBee/IEEE 802.15.4 создавался изначально по критериям малой дальности действия, низкой цены, низкой потребляемой мощности, низкой скорости передачи и малых габаритов. Эти свойства идеально соответствуют требованиям к большинству промышленных датчиков. Поэтому ZigBee часто отождествляют с промышленными беспроводными сенсорными сетями WSN. А это именно то, что нам и требуется.

Данный протокол использует не лицензируемый диапазон 2.4ГГц, а также частоту 868МГц в Европе и 915 в США. Указанный диапазон был разделен на 11...26 каналов с шириной в 5МГц каждый. Хотя в свойствах и указана низкая скорость передачи, но максимальная скорость составляет 250Кб/сек. Средняя скорость передачи полезных данных, в зависимости от загрузки сети и числа ретрансляций, составляет от 5 до 40 кбит/с.

Данный протокол позиционируется, как протокол для использования в ячеистой топологии, где он может самовосстанавливаться и гарантировать доставку пакета в случае обрыва связи между отдельными узлами. При создании ячеистой топологии в ZigBee используются три типа устройств:

* координатор - формирует топологию сети и может устанавливать мосты с другими сетями. В каждой ZigBee сети имеется только один координатор;
* маршрутизатор - работает как промежуточное звено, передавая в нужном направлении данные от других устройств;
* конечное устройство - передает данные координатору или маршрутизатору  и не может связываться с аналогичными ему устройствами.

Топология Zigbee-сети поддерживается уровнем NWK и может иметь форму звезды, дерева или ячеистой сети. В топологии типа звезды сеть контролируется координатором. Координатор отвечает за инициализацию и обслуживание сетевых устройств и всех конечных устройств, непосредственно взаимодействующих с координатором. В ячеистой и древовидной структуре сети координатор отвечает за организацию сети и выбор некоторых ключевых параметров, но сеть может быть расширена с помощью ZigBee маршрутизаторов. В сети с древовидной топологией маршрутизаторы перемещают данные и управляющие сообщения по сети, используя иерархическую стратегию маршрутизации. Древовидные сети могут использовать маячковую стратегию маршрутизации.

Ячеистая сеть должна обеспечить полную одноранговую коммуникацию устройств, т.е. в ячеистой сети нет устройств разных рангов (координаторов, маршрутизаторов и т.п. - все устройства равноправны).

Также он является криптографически защищенным. Устройства, работающие с данным протоколом, имеют крайне низкое энергопотребление. Например, используемый мной СС1011 имеет режим сна, при котором энергопотребление менее 1мА.

Сеть ZigBee – самоорганизующаяся, и ее работа начинается с формирования. Устройство, назначенное при проектировании координатором персональной сети (PAN координатор), определяет канал, свободный от помех, и ожидает запросов на подключение.   
Устройства, пытающиеся присоединиться к сети, рассылают широковещательный запрос. Пока PAN координатор – единственное устройство в сети, отвечает на запрос и предоставляет присоединение к сети только он. В дальнейшем присоединение к сети могут предоставлять также присоединившиеся к сети маршрутизаторы.   
 Устройство, получившее ответ на широковещательный запрос, обменивается с присоединяющим устройством сообщениями, чтобы определить возможность присоединения. Возможность определяется способностью присоединяющего маршрутизатора обслужить новые устройства в дополнение к ранее подключенным.  
 Вступление в сеть (присоединение)   
Существует два способа присоединения: МАС ассоциация и повторное сетевое присоединение (NWK rejoin).  
МАС ассоциация  
МАС ассоциация доступна любому устройству ZigBee и осуществляется на МАС уровне. Механизм МАС ассоциации следующий:  
Устройство, позволяющее присоединиться к нему, выставляет на МАС уровне разрешение на присоединение.  
Устройство, вступающее в сеть, выставляет на МАС уровне запрос на присоединение и передает широковещательный запрос маячка.  
Получив маячок от устройств, готовых подключить присоединяемое устройство, последнее определяет, в какую сеть и к какому устройству оно желает присоединиться, и выставляет на МАС уровне требование о вступлении с флажком «повторное присоединение» в значении FALSE.  
Затем вступающее устройство направляет на выбранное для присоединения устройство запрос присоединения и получает ответ с присвоенным ему сетевым адресом.  
 При МАС ассоциации данные передаются не зашифрованными, поэтому МАС ассоциация не является безопасной.  
Повторное сетевое присоединение Повторное сетевое присоединение вопреки названию может применяться и при первичном присоединении. Оно выполняется на сетевом уровне. При этом, если вступающее устройство знает текущий сетевой ключ, обмен пакетами может быть безопасным. Ключ может быть получен, например, при настройке.   
При повторном подключении присоединяющееся устройство выставляет на сетевом уровне запрос присоединения и обменивается с подключающим устройством пакетами «запрос присоединения» – «ответ на запрос присоединения».

* + 1. **Широтно-импульсная модуляция**

Этот принцип является исключительно важным для нашей системы т.к. именно он отвечает за корректное программное изменение яркости освещения. Именно поэтому требуется рассмотреть его отдельно.

Человеческий глаз - сложная вещь. В нем есть палочки и колбочки, а яркость, с которой мы наблюдаем объекты, зависит от количества фотонов, которые упали на них. Особенность заключается в том, что «оцифровка» числа фотонов не происходит сразу. Они работают как сумматор, то есть накапливают «заряд», и через определенный промежуток времени показания снимаются. Это называется инерцией человеческого глаза. Другими словами, если объект мерцает быстрее, чем происходит чтение, мы просто не замечаем мерцание. Время «регистрации» фиксировано (варьируется от человека к человеку), что означает, что фиксированный количество фотонов (если мы возьмем конкретный источник света) может упасть на колбу / палочку. Если половина этого времени светодиод будет гореть, а другую - нет - тогда колба / палочка будет получать вдвое больше фотонов, чем если бы светодиод горел. Приблизительная частота, с которой средний человек не замечает мерцание, составляет 50 Гц.

Частота (период) фиксированна, а вот заполнение (англ. duty) может меняться. Пример разных уровней заполнения на рисунке Б.4 в приложении Б.

В микроконтроллерах широтно-импульсная модуляция реализуется с помощью таймеров. В режиме ШИМ для нас критичны две величины: одна отвечает за период (TIM\_Period), а вторая за заполнение (TIM\_Pulse). Счетчик стартует с «0» (на выходе низкий уровень) и считает до TIM\_Period, затем перезагружается и начинает всё сначала. До момента, когда таймер досчитает до потолка, он пройдет промежуточное значение, которое мы задали, указав TIM\_Pulse. Как только таймер это сделает, он может переключить состояние ножки на противоположное состояние. Т.е. нам необходимо выполнить несколько действий:

* Включаем тактирование порта и таймера.
* Настраиваем ножку.
* Настраиваем таймер.
* Включаем и настраиваем режим ШИМ выбранного таймера.

Непосредственно уровень яркости будет регулироваться значение переменной TIM\_Pulse. Она может принимать значения от 0 до 255(0x00-0xFF), где 0 соответствует минимальной яркости, а 255, в свою очередь, максимальную. Можно разделить этот диапазон на 5 режимов с разницей в 64. Т.е. Нам потребуется всего лишь 5 сигналов для включения определенного уровня яркости.

* 0 – 0x00
* 64 – 0x40
* 128 – 0x80
* 192 – 0xC0
* 255 – 0xFF

Когда микроконтроллер получит какой-то из этих сигналов, то он присвоит [соответствующ](https://ru.wiktionary.org/wiki/%D1%81%D0%BE%D0%BE%D1%82%D0%B2%D0%B5%D1%82%D1%81%D1%82%D0%B2%D1%83%D1%8E%D1%89%D0%B8%D0%B9)ее значение переменной TIM\_Pulse и уровень яркости измениться. Изображение ШИМ на рисунке 3 в приложении Б.

* 1. **Выводы по разделу 1**

К концу данного раздела уже можно представить общий вид устройства. Это микроконтроллер STM32F100 с радиомодулем СС1101 и периферией в виде нескольких датчиков движения и освещенности. Также на некоторых устройствах будет wi-fi модуль для связи с пользователем.

1. **Разработка системы**
   1. **Описание функционала системы**

Для начала опишем функционал, который мы имеем к началу создания системы. Во-первых, это просто удаленное включение\выключение, во-вторых, возможность настройки уровня яркости освещения, в-третьих, автоматическое включение\выключение освещение при переходе через какую-то грань освещенности, определяемую датчиком освещенности KY-018.

Как видно, функционал системы делает её не слишком-то интеллектуальной. Попробуем изменить это. Самое очевидное, что можно добавить, так это автоматическое включение света при входе человека в комнату и его отключение при выходе. За это и будет отвечать новый датчик движения HC-SR501. Также добавим возможность авторегулировки яркости освещения. Для этого единую границу включения\выключения освещения разделим на несколько подграниц, при переходе которых яркость будет увеличиваться или уменьшаться. Для этого потребуется лишь более точная и долгая калибровка датчика KY-018 при подключении ко входу ацп микроконтроллера.

Теперь можно сказать, что данная система может полностью автоматически управлять освещением в каком-либо помещении, но и возможность ручного управления также останется благодаря возможно связи человека с системой через телефон посредством wi-fi. Именно это и имелось в виду под компромиссом между удобство пользователя и энергоэффективностью.

* 1. **Описание устройства**

Опишем систему и её общее устройство: вся система состоит из нескольких одинаковых устройств, отвечающих за сбор и обработку информации, исходя из которой они и принимают решения, но есть еще и аналогичное устройство с модулем esp8266-01, которое отвечает за связь с человеком посредством wi-fi связи с мобильным устройство. Назовем его master, а остальные устройства slaves. По сути, master – это координатор сети, slaves – маршрутизаторы.

Через этот модуль можно передавать предпочитаемые настройки остальным частям системы. Т.е. возможен не только автоматический режим работы, но и точная ручная настройка освещения в соответствии с личными предпочтениями. При получении таких настроек master передаст информацию на следующее устройство, которое в свою очередь ретранслирует его дальше и т.д. Т.е. будет использоваться одно из свойств ZigBee сетей: автоматическая шифровка, маршрутизация и обеспечение доставки пакета. Всё это будет происходить с невысокой скоростью от 5 до 40 кбит/с., но и объем передаваемой информации у нас довольно мал. Расстояние между рабочими станциями сети составляет десятки метров внутри помещений и сотни метров на открытом воздухе, что нас вполне устраивает.

В конечном виде у нас получается mesh сеть на основе протокола ZigBee с устройствами двух видов, способных принимать решения на основе данных от датчиков или работать в полной зависимости от желаний человека.

* 1. **Алгоритмы работы системы**

Рассмотрим алгоритм работы одиночного устройства. Всё освещение выключено до момента, когда в помещение входит человек т.е. срабатывает датчик движения. Если у нас включено полностью автоматическое управление, то датчик освещения производит замер окружающего света. Полученное значение он сравнивает с некими, определенным нами заранее, константами. Далее, исходя из полученного значения приходит к решению включать или не включать свет. Если свет включается, то уровень яркости выставляется также исходя из сравнения с набором констант. Пока человек находится в комнате периодически будет повторятся замер уровня освещенности и сравнение с константами. При изменении количества света будет и изменятся уровень яркости вплоть до максимума или отключения света. При выходе человека из комнаты освещение выключается.

* 1. **Структурные схемы устройств**

Зная требования к функционалу системы, и учитывая условия из предыдущих пунктов мы можем представить структурную схему master и slave (рисунок Б.1.1. и Б.1.2. соотвественно в приложении Б.)

Структурная схема состоит из следующих элементов:

* МК – микроконтроллер
* WF – wi-fi модуль
* ДО – датчик освещенности(фоторезистор)
* ИП – источник питания
* ОУ – объект управления
* ДД – датчик движения
* РМ - радиомодуль

Данные с датчиков и команды с телефона (через wi-fi модуль) передаются на master, который их обрабатывает и передает ретранслирует их далее до самого объекта управления, в результате чего и достигается необходимый уровень освещения в помещении. Это для ручного управления или задания настроек. В автоматическом режиме каждый slave модуль является микроконтроллером с радиомодулем и периферией в виде датчиков, собирающих информацию. На основе этой информации, и полученных настроек, микроконтроллер принимает решение о включении или выключении света, уровне яркости и т.д.

* 1. **Выводы по разделу 2**

Во втором разделе дано словесное описание устройство, разработана структурная схема устройства и описаны принципы mesh сети на основе протокола ZigBee.

1. **Экономическая часть**
   1. **Анализ существующих на рынке систем**

Новый функционал системы позволяет перейти из разряда систем управления для умного дома в разряд промышленных систем управления освещением. Даже поверхностный поиск позволяет найти множество подобных предложений. Например, КУЛОН-РД от компании Sundrax, АСУО от Arman, СУС от Duray. У всех этих предложений разные характеристики с разным функционалом, но объединяет их одно: закрытый ценник. Что позволяет предположить высокую цену, а большое количество предложений позволяет предположить высокий спрос, ибо спрос рождает предложение. Но если мы попробуем вернуться обратно в лигу домашних предложений, то особой конкуренции мы там не найдем т.к. функционал намного превышает системы, описанные в моей предыдущей работе, а цена не слишком выросла.

* 1. **Расчет себестоимости**

Перечислим все узлы и детали, используемые в нашем готовом продукте:

* Wi-fi модуль esp8266-01
* Микроконтроллер STM32F100
* Светочувствительный модуль KY-018
* Светодиодная лента
* Полевой транзистор IRF520
* СС1011

Соберем все данные о стоимости данных элементов в одной таблице:

Таблица 3.1 Стоимость деталей

|  |  |  |  |
| --- | --- | --- | --- |
| Деталь | Кол-во | Цена(руб.) | Место покупки |
| Esp8266-01 | 1 | 144.45 | aliexpress.com. |
| KY-018 | 1 | 20.30 | aliexpress.com |
| STM32F100 | 1 | 69.16 | aliexpress.com |
| PC817 | 1 | 8 | chipdip.ru |
| LM1117DT | 1 | 68 | chipdip.ru |
| СС1011 | 2 | 133.72 | aliexpress.com |
| HC SR501 | 1 | 21.25 | aliexpress.com |

Итого 464.58 рублей. Несомненно, эта цена меньше цены любого промышленного модуля, и по-прежнему меньше цены модулей умного дома.

* 1. **Выводы по разделу 3**

Проведя подсчеты себестоимости, мы выяснили, что за дополнительный функционал, позволяющий более эффективно управлять освещением, что, в свою очередь, позволит экономить на затратах электроэнергии, увеличило стоимость на 50%. Даже при этом цена нашего устройства не приблизилась к цене устройств умного дома.

Также мы выяснили, что с новым функционалом наша система может выйти на рынок систем промышленного управления освещением.

1. **Заключение**

В процессе выполнения дипломной работы мы модифицировали систему управления освещением так, что она приобрела новый качества. Качества такие как:

* Энергоэффективность (благодаря интеллектуальному управлению)
* Экологичность (следует из энергоэффективности т.к. позволяет беречь энергоресурсы)
* Интеллектуальность (способна к полностью автоматическому управлению
* Конкурентоспособность на рынке домашних и промышленных решений

Таким образом, спроектированная система удовлетворяет всем начальным условиям, и обладает при этом дешевизной, простотой реализации и эксплуатации присущими предыдущей реализации системы. Т.е. она приобрела новые полезные качества, не потеряв старых.

**Список литературы**

Основная литература

1. Бройдо, В. Л. Вычислительные системы, сети и телекоммуникации: Учебник для вузов / В. Л. Бройдо. – Санкт-Петербург : ПИТЕР, 2004. – 702 c.
2. Предко, М. Руководство по микроконтроллерам. Том 1 / М. Предко. – Москва : Постмаркет, 2001. – 416 c.
3. Предко, М. Руководство по микроконтроллерам. Том 2 / М. Предко. – Москва : Постмаркет, 2001. – 418 c.

Дополнительная литература

1. Joseph, Yiu. the Definitive Guide to the ARM Cortex-M3 / Yiu. Joseph. – Burlington : Elsevier Inc, 2007. – 531 c.
2. Trevor, Martin. The Insider’s Guide To The STM32 ARM®Based Microcontroller / Martin. Trevor. – Coventry : Hitex (UK) Ltd., 2008. – 96 c.

**Приложение А. Схема**

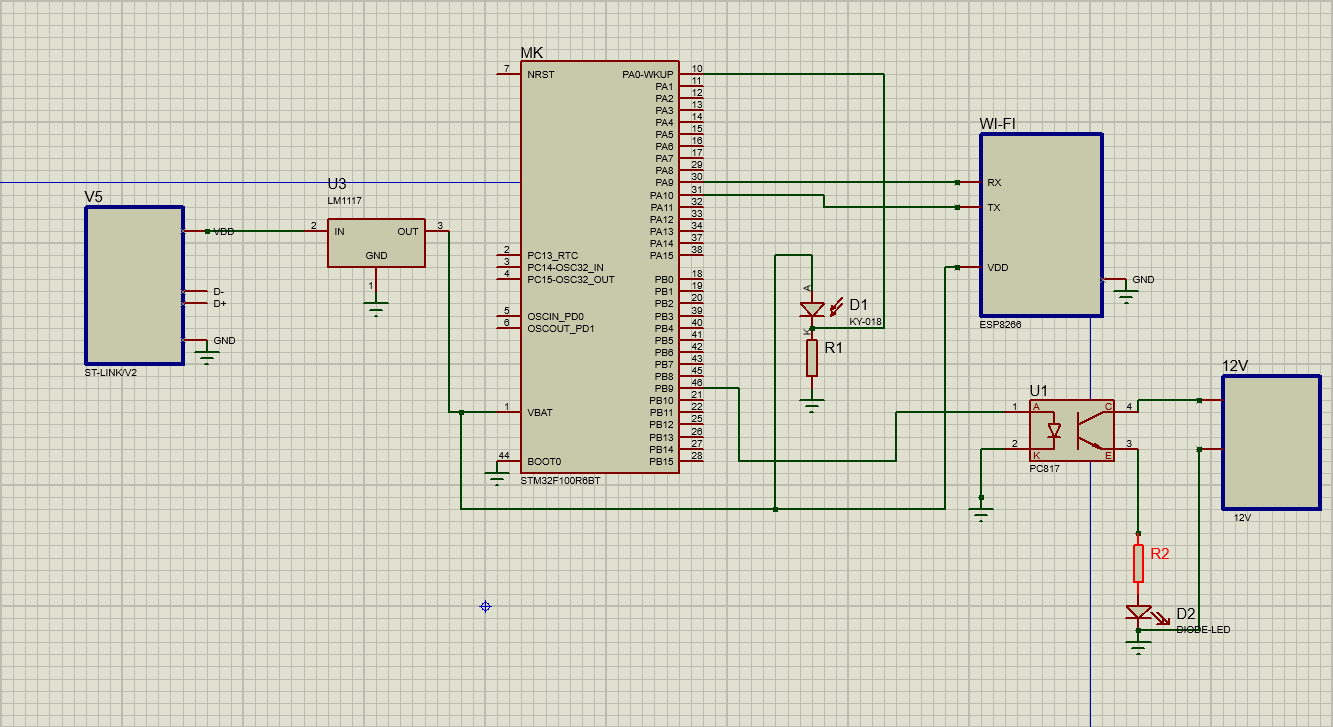


Схема А.1. Общая схема устройства

**Приложение Б. Рисунки**

Рисунок Б.1.1. Структурная схема MASTER

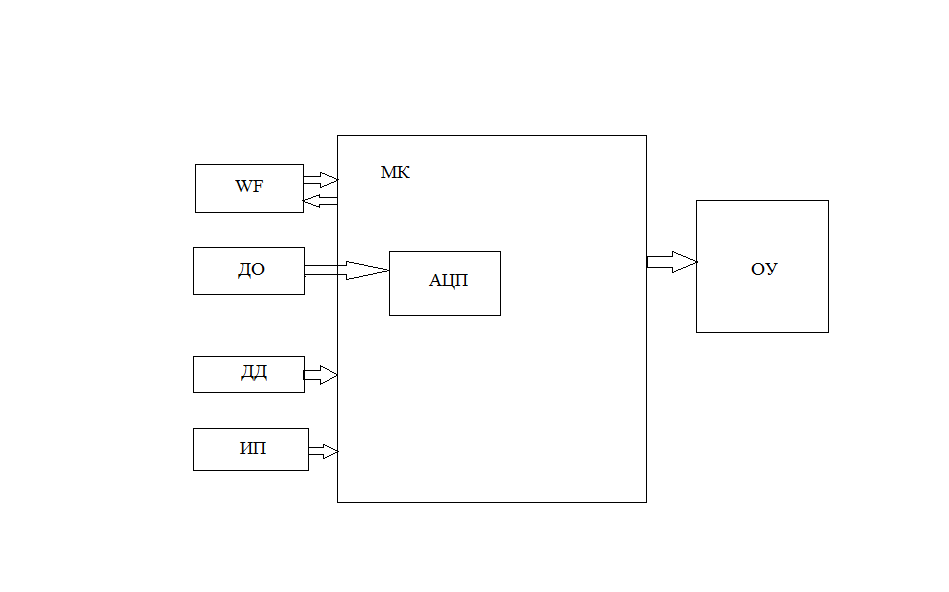
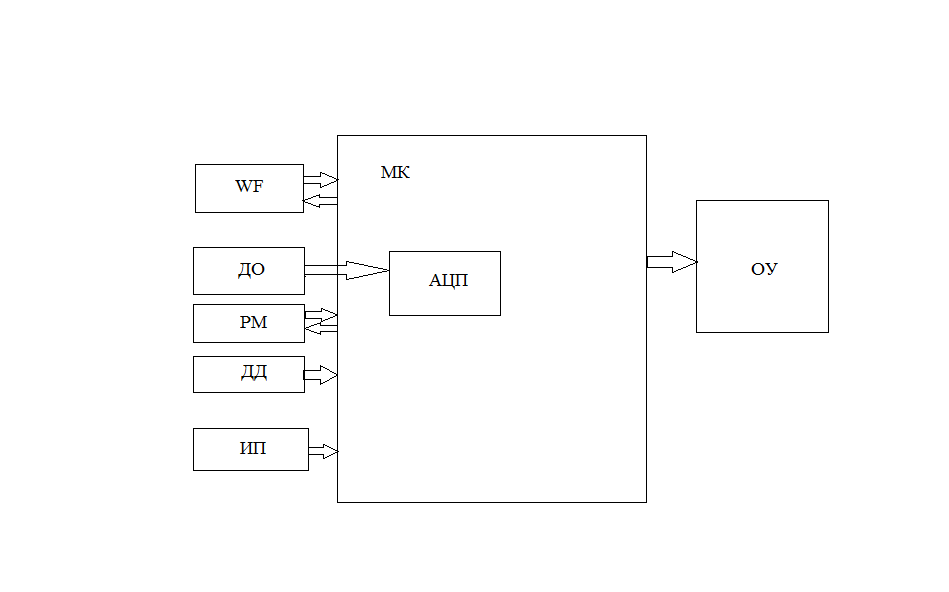


Рисунок Б.1.2. Структурная схема SLAVE

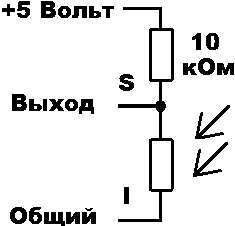


Рисунок Б.2. Модуль KY-018

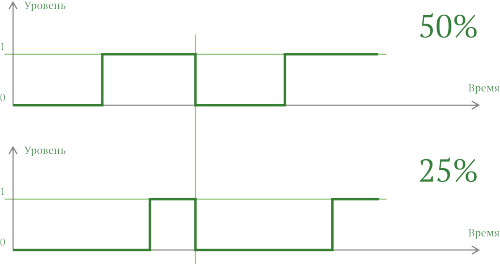


Рисунок Б.3. ШИМ

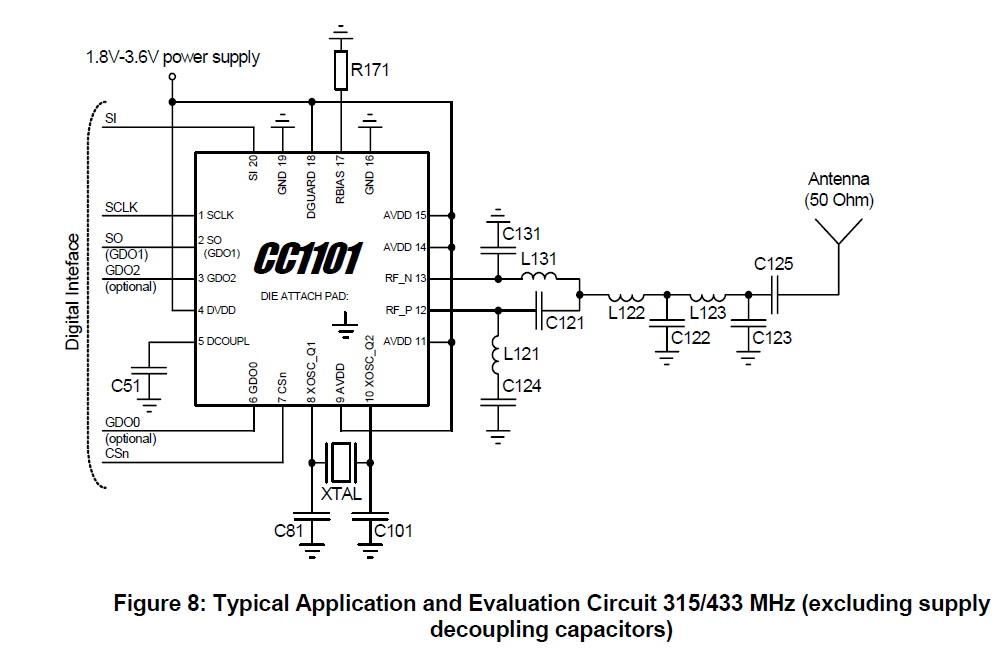


Рисунок Б.4. СС1101

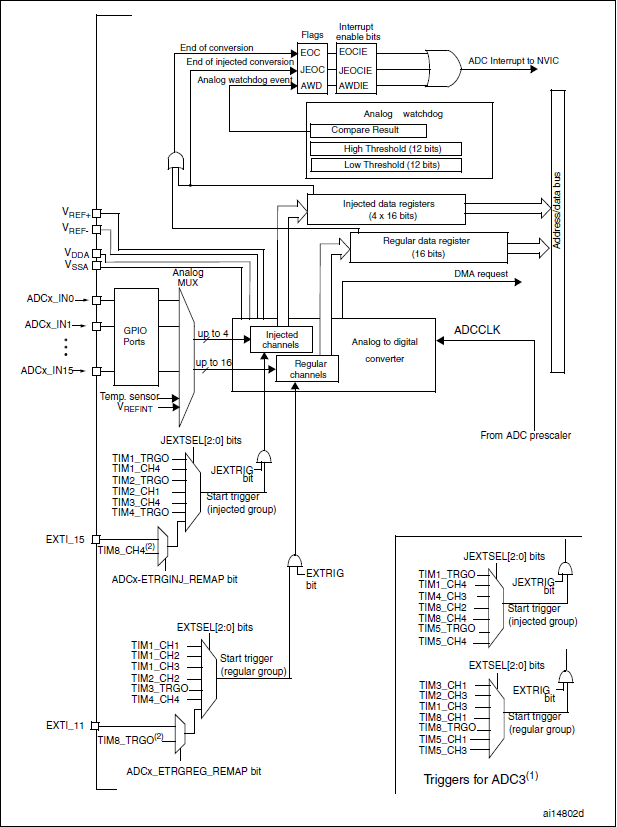


Рисунок Б.5. Схема АЦП в stm32 (http://st.com/)

**Приложение В. Листинг**

Листинг 1. Код прошивки микроконтроллера на языке С

#include <stm32f10x.h>

#include <stm32f10x\_gpio.h>

#include <stm32f10x\_rcc.h>

#include <stm32f10x\_tim.h>

#include <misc.h>

void init\_pwm();

void init\_USART1();

void init\_Usart1\_Nvic();

void adc\_init();

uint16\_t get\_adc\_value();

//n-длина команды

void Pause()

{

for(int i=0;i<100000;i++)

{}

}

void send\_command(uint8\_t command[], int n)

{

for(int i=0;i<n;i++)

{

USART\_SendData(USART1,command[i]);

Pause();

}

}

enum {TX=GPIO\_Pin\_9,RX=GPIO\_Pin\_10};

/\*41 54 2B 43 49 50 4D 55 58 3D 31 0D 0A = AT+CIPMUX=1 Режим множественных подключений \*/

/\*41 54 2B 43 49 50 53 45 52 56 45 52 3D 31 2C 38 38 38 38 0D 0A = AT+CIPSERVER=1,8888 Запустить сервер на порту 8888 \*/

//41 54 2B 43 49 46 53 52 0D 0A = AT+CIFSR Получить ip

uint8\_t cipmux[] = {0x41,0x54,0x2B,0x43,0x49,0x50,0x4D,0x55,0x58,0x3D,0x31,0x0D,0x0A};

uint8\_t cipserver[] = {0x41,0x54,0x2B,0x43,0x49,0x50,0x53,0x45,0x52,0x56,0x45, 0x52,0x3D,0x31,0x2C,0x38,0x38,0x38,0x38,0x0D,0x0A};

TIM\_OCInitTypeDef pwm;

int main(void)

{

init\_pwm();

init\_USART1();

init\_Usart1\_Nvic();

adc\_init();

//первичная настройка esp8266

send\_command(cipmux,sizeof(cipmux));

Pause();

Pause();

send\_command(cipserver,sizeof(cipserver));

uint16\_t adc;

while(1)

{

adc=get\_adc\_value();

}

}

//Инициализация шим

void init\_pwm(void)

{

RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOC|RCC\_APB2Periph\_AFIO,ENABLE);

RCC\_APB1PeriphClockCmd(RCC\_APB1Periph\_TIM3,ENABLE);

GPIO\_PinRemapConfig(GPIO\_FullRemap\_TIM3,ENABLE);

//инициализируем ножку как выход с альтернативной функцией

GPIO\_InitTypeDef gpio;

gpio.GPIO\_Pin = GPIO\_Pin\_8;

gpio.GPIO\_Mode = GPIO\_Mode\_AF\_PP;

gpio.GPIO\_Speed = GPIO\_Speed\_2MHz;

GPIO\_Init(GPIOC,&gpio);

//задаем параметры таймера 3ch3 на этой ножке

TIM\_TimeBaseInitTypeDef t3;

TIM\_TimeBaseStructInit(&t3);

t3.TIM\_Period = 200;

t3.TIM\_Prescaler = 2400;

t3.TIM\_ClockDivision = 0;

t3.TIM\_CounterMode = TIM\_CounterMode\_Up;

TIM\_TimeBaseInit(TIM3,&t3);

// задаем параметры шим

pwm.TIM\_OCMode =T IM\_OCMode\_PWM1;

pwm.TIM\_OutputState = TIM\_OutputState\_Enable;

pwm.TIM\_Pulse = 0;

pwm.TIM\_OCPolarity = TIM\_OCPolarity\_High;

TIM\_OC3Init(TIM3,&pwm);

TIM\_OC3PreloadConfig(TIM3,TIM\_OCPreload\_Enable);

TIM\_ARRPreloadConfig(TIM3,ENABLE);

TIM\_Cmd(TIM3,ENABLE);

}

void init\_USART1()

{

//инициализация приема с USART(PA10)

RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOA | RCC\_APB2Periph\_USART1 | RCC\_APB2Periph\_AFIO,ENABLE);

GPIO\_InitTypeDef gpio;

gpio.GPIO\_Mode = GPIO\_Mode\_IN\_FLOATING;

gpio.GPIO\_Pin = RX;

GPIO\_Init(GPIOA,&gpio);

//инициализация выхода на USART(PA9)

gpio.GPIO\_Mode = GPIO\_Mode\_AF\_PP;

gpio.GPIO\_Speed = GPIO\_Speed\_50MHz;

gpio.GPIO\_Pin = TX;

GPIO\_Init(GPIOA,&gpio);

//инициализация USART

USART\_InitTypeDef usart;

usart.USART\_BaudRate = 115200;

usart.USART\_WordLength = USART\_WordLength\_8b;

usart.USART\_StopBits = USART\_StopBits\_1;

usart.USART\_Parity = USART\_Parity\_No;

usart.USART\_HardwareFlowControl =

USART\_HardwareFlowControl\_None;

usart.USART\_Mode = USART\_Mode\_Rx | USART\_Mode\_Tx;

USART\_Init(USART1,&usart);

USART\_Cmd(USART1,ENABLE);

//прерывание

USART\_ITConfig(USART1,USART\_IT\_RXNE,ENABLE);

NVIC\_EnableIRQ(USART1\_IRQn);

}

//разрешаем и задаем приоритет прерыванию по прибытию //информации в USART1

void init\_Usart1\_Nvic()

{

NVIC\_InitTypeDef nvic;

nvic.NVIC\_IRQChannel = USART1\_IRQn;

nvic.NVIC\_IRQChannelPreemptionPriority = 1;

nvic.NVIC\_IRQChannelSubPriority = 1;

nvic.NVIC\_IRQChannelCmd = ENABLE;

NVIC\_Init(&nvic);

}

//обработчик прерывания USART1

void USART1\_IRQHandler()

{

uint8\_t data;

//Проверяем, действительно ли прерывание вызвано приемом нового байта

if (USART\_GetITStatus(USART1, USART\_IT\_RXNE) != RESET)

{

USART\_ClearITPendingBit(USART1,USART\_IT\_RXNE);

data=USART\_ReceiveData(USART1);

//обработка внешних сигналов

if(data == 0x00) pwm.TIM\_Pulse = 0;

if (data == 0x40) pwm.TIM\_Pulse = 64;

if(data == 0x80) pwm.TIM\_Pulse = 128;

if(data == 0xC0) pwm.TIM\_Pulse = 192;

if(data == 0xFF) pwm.TIM\_Pulse = 255;

if(data == 0x01) GPIO\_ResetBits(PORTC,GPIO\_Pin\_9);

if(data == 0x02) GPIO\_SetBits(PORTC,GPIO\_Pin\_9);

}

}

void adc\_init()

{

RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_ADC1, ENABLE);

// настройки ADC

ADC\_InitTypeDef ADC\_InitStructure;

ADC\_StructInit(&ADC\_InitStructure);

ADC\_InitStructure.ADC\_Mode = ADC\_Mode\_Independent;

// режим работы - одиночный, независимый

ADC\_InitStructure.ADC\_ScanConvMode = DISABLE;

// просто измерить один канал

ADC\_InitStructure.ADC\_ContinuousConvMode = DISABLE;

ADC\_InitStructure.ADC\_ExternalTrigConv = ADC\_ExternalTrigConv\_None;

// без внешнего триггера

ADC\_InitStructure.ADC\_DataAlign = ADC\_DataAlign\_Right; //выравнивание битов результат - прижать вправо

ADC\_InitStructure.ADC\_NbrOfChannel = 1;

//количество каналов - одна штука

ADC\_Init(ADC1, &ADC\_InitStructure);

ADC\_Cmd(ADC1, ENABLE);

// настройка канала

ADC\_RegularChannelConfig(ADC1, ADC\_Channel\_8, 1, ADC\_SampleTime\_55Cycles5);

// калибровка АЦП

ADC\_ResetCalibration(ADC1);

while (ADC\_GetResetCalibrationStatus(ADC1));

ADC\_StartCalibration(ADC1);

while (ADC\_GetCalibrationStatus(ADC1));

}

//получение измерения с ацп

uint16\_t get\_adc\_value()

{

ADC\_SoftwareStartConvCmd(ADC1, ENABLE);

while(ADC\_GetFlagStatus(ADC1, ADC\_FLAG\_EOC) == RESET);

return ADC\_GetConversionValue(ADC1);

}