

УДК 681.5

АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ НА БАЗЕ МИКРОКОНТРОЛЛЕРА STM32



Г.Р. Ипатов
Студент кафедры
информационных
радиотехнологий БГУИР,
germanipatov6@gmail.com



Т.Н. Дворникова
Старший преподаватель
кафедры информационных
радиотехнологий БГУИР,
магистр технических наук
dvornikova@bsuir.by

Г.Р. Ипатов

Студент кафедры информационных радиотехнологий БГУИР. Область научных интересов связана с программированием микроконтроллеров.

Т.Н. Дворникова

Окончила Белорусский государственный университет информатики и радиоэлектроники. Область научных интересов связана с программированием встраиваемых систем, исследованием быстрого преобразования корректирующих кодов.

Аннотация. Передовые технологии встраиваемых систем, которые позволяют автоматизировать бытовые задачи, стремительно развиваются с каждым днем. Автоматизированная система управления на базе микроконтроллера *STM32* реализована в виде лабораторного макета для дисциплины «Встраиваемые системы», а также разработан алгоритм функционирования системы. Разработка актуального и востребованного лабораторного оборудования дает возможность познакомиться студентам с новейшими технологиями встраиваемых и киберфизических систем, а также является важным аспектом обеспечения качественной подготовки высококвалифицированных специалистов.

Разработанная Автоматизированная система управления в виде лабораторного макета может быть использована в учебном процессе для изучения дисциплины «Встраиваемые системы», а кроме того, как решение в системах умного дома.

Ключевые слова: *MQTT*-протокол, встраиваемые системы, технические требования, схема электрическая структурная, схема электрическая функциональная, схема электрическая принципиальная, микроконтроллер.

Введение. Передовые технологии встраиваемых систем, которые позволяют автоматизировать бытовые задачи, стремительно развиваются с каждым днем. В связи с этим у пользователей появилась возможность управлять и контролировать электрические устройства дистанционно.

В настоящее время наиболее актуальным и перспективным направлением развития технологий во встраиваемых системах является технология *IoT (Internet of Things)*, которая нашла свое применение в комплексных системах управления, как «умный дом».

В связи с быстрым расширением функций и областей применения встраиваемых систем, использование широкой номенклатуры современных микропроцессоров и микроконтроллеров, а также возрастание роли и сложности программного обеспечения для встраиваемых систем, разработка автоматизированной системы управления является актуальной задачей. В работе предлагается лабораторный макет автоматизированной системы управления на базе микроконтроллера семейства *STM32* производства *STMicroelectronics*.

Разработка дает возможность сделать процесс обучения более интересным, повысить практические навыки программирования, а также познакомиться с новейшими технологиями автоматизированных встраиваемых систем.

Описание принципа работы автоматизированной системы управления на базе микроконтроллера *STM32*. Основой автоматизированной системы управления, выполненной в виде макета, являются отладочная плата на базе микроконтроллера *STM32F103C8T6* и модуль *Wi-Fi ESP8266*.

Отладочная плата – макетная плата, которая предоставляет широкий спектр экспериментов по созданию встраиваемых устройств на основе микроконтроллера *STM32*.

К основным достоинствам отладочной платы на базе микроконтроллера *STM32F103C8T6* относятся:

- обеспечение требуемой эффективности;
- широкий спектр реализуемых задач;
- возможности подключения различных устройств ввода/вывода;
- стандартные интерфейсы для подключения вспомогательных модулей.

На рисунке 1 представлено изображение отладочной платы на базе микроконтроллера *STM32F103C8T6*.

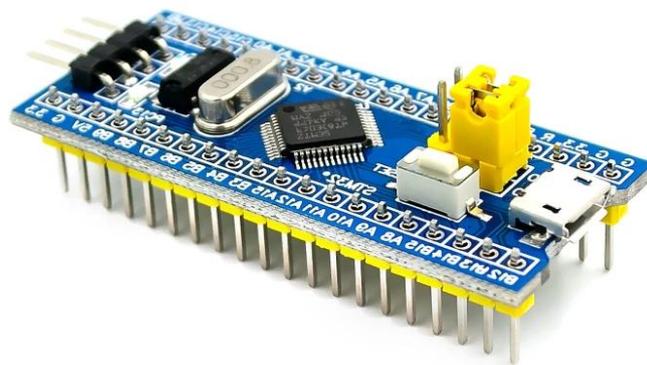


Рисунок 1. Отладочной плата на базе микроконтроллера *STM32F103C8T6*

В автоматизированной системе управления модуль *Wi-Fi ESP8266* организует беспроводную связь по интерфейсу *Wi-Fi*, обеспечивая возможность дистанционного управления и сбора данных о текущем состоянии системы.

К основным достоинствам модуля *ESP8266* относятся:

- встроенные возможности для обработки и хранения данных;
- порты ввода-вывода, которые позволяют объединить модуль с различными датчиками и другими специальными устройствами, нуждающимися в сетевом обмене

данными.

На рисунке 2 представлено изображение модуля *Wi-Fi ESP8266*.



Рисунок 2. Модуль *Wi-Fi ESP8266*

Автоматизированная система управления реализована в виде лабораторного макета, имеющего комплектацию, которая позволяет использовать его в рамках учебной дисциплины «Встраиваемые системы».

Схема электрическая структурная лабораторного макета состоит из 13 блоков. Она представлена на рисунке 3.

Микроконтроллер является основным блоком отладочной платы. В микроконтроллер загружается исполняемая программа через программатор.

ST-LINK/V2 – внутрисхемный программатор/отладчик для микроконтроллеров серии *STM32* производства фирмы *STMicroelectronics*. Программатор подключается к программному обеспечению, и с его помощью загружает код программы в микроконтроллер.

Имеется возможность подключить программатор к иным отладочным платам на базе микроконтроллеров на базе ядра *STM32* посредством выходного отладочного интерфейса.

Устройство индикации является пользовательским устройством отображения и может быть задействована для любого рода задач.

Кнопка сброса – кнопка очистки памяти. При нажатии осуществляется сброс параметров микроконтроллера отладочной платы к изначальным настройкам.

С помощью модуля *Wi-Fi ESP8266* осуществляется беспроводная связь отладочной платы *STM32* с устройством управления и отображения информации. В рамках автоматизированной системы управления *ESP8266* выступает в роли адаптера.

Пользовательская программа загружается в модуль через преобразователь, который преобразует компьютерный интерфейс в последовательный.

Интерфейсы для программирования автоматизированной системы управления на базе микроконтроллера *STM32*. На рисунке 3 представлено расположение выводов отладочной платы относительно разъемов, а также интерфейсы взаимодействия.

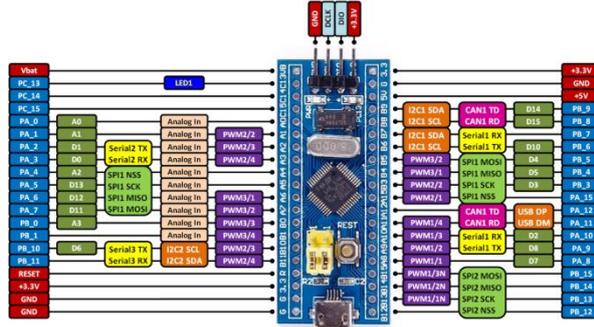


Рисунок 3. Расположение выводов и интерфейсы в отладочной плате на базе STM32

Метод программирования последовательных конфигурационных постоянных запоминающих устройств, который осуществляется через интерфейс *JTAG*, что упрощает процесс отладки. Программа микроконтроллера, разработанная в программном обеспечении на персональном компьютере, поступает через интерфейс *mini-USB*, после чего программатор подключается к микроконтроллеру на плате, либо через *SWD*-интерфейс к иному устройству. Для программирования и отладки по интерфейсу *SWD* используются следующие выводы микроконтроллера:

- *SWDIO* – *Serial Wire Data Input/Output*;
- *SWCLK* – *Serial Wire Clock*;

Линии интерфейса имеют внутреннюю подтяжку, поэтому нет необходимости устанавливать внешние резисторы.

Имеется возможность подключить программатор к иным отладочным платам на базе микроконтроллеров на базе ядра *STM32* посредством выходного отладочного интерфейса.

Светодиодная индикация состоит из пользовательского светодиода и индикатора питания. Они подключены напрямую к информационным выводам микроконтроллера.

Кнопка сброса представляет собой кнопку очистки памяти. При необходимости ее можно перепрограммировать на иное использование.

Для взаимодействия светодиодной индикации и кнопки сброса с контроллером используется интерфейс ввода/вывода общего назначения (*general-purpose input/output, GPIO*), который осуществляет прямое подключение компонентов к контроллеру.

Вторым блоком в функциональной схеме лабораторного макета является модуль *Wi-Fi*, который реализован на микроконтроллере *ESP8266*, тактовая частота которого составляет 80 МГц, возможно программное переключение в режим 160 МГц. Система может легко работать как операционная система реального времени (*RTOS*). Микроконтроллер может взаимодействовать с остальными частями чипа через интерфейсы: программируемый *RAM/ROM* интерфейс (*iBus*), соединенный с контроллером памяти и имеющий доступ к внешней *flash*-памяти; интерфейс памяти данных (*dBus*), соединенный с контроллером памяти.

Модуль *Wi-Fi ESP8266* имеет также периферийные интерфейсы, такие как *SPI, UART, I2C, I2S*.

На рисунке 4 представлено расположение выводов *ESP8266* относительно разъемов, а также интерфейсы взаимодействия.

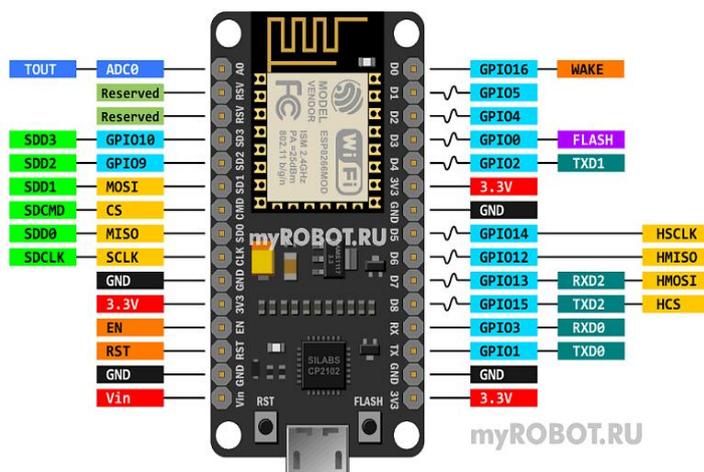


Рисунок 4. Расположение выводов и интерфейсы в ESP8266

Интерфейс *I2C* – последовательная асимметричная шина. Он используется для подключения датчиков и периферийных устройств. *ESP8266* не имеет аппаратных выводов *I2C*, но интерфейс можно реализовать программно. Поддерживаются как *I2C Master*, так и *I2C Slave*. В качестве контактов *I2C* используются выводы *GPIO5: SCL* и *GPIO4: SDA*.

Программа, разработанная в программном обеспечении на персональном компьютере, поступает через интерфейс *mini-USB*, после чего преобразуется в переходнике *USB-UART*, который осуществляет преобразование компьютерного интерфейса *USB* в последовательный интерфейс *UART*.

UART (Universal Asynchronous Receiver-Transmitter) – асинхронный последовательный интерфейс. У контроллера *ESP8266* два модуля *UART*. Один из которых устанавливает связь с отладочной платой на базе микроконтроллера *STM32F103C8T6* по шине *UART*, а другой – для загрузки кода по последовательной связи.

Светодиодная индикация включает в себя два светодиода. Один из них используется для индикации рабочего состояния *Wi-Fi*, другой – для индикации состояния связи между устройством и сервером.

Кнопка сброса предназначена для сброса контроллера *ESP8266*, кнопка *Flash* – для загрузки новой программы в модуль.

Для поддержания различных функций светодиодов и кнопок используется интерфейс ввода-вывода общего назначения (*GPIO*). Каждый *GPIO* являются двунаправленным и неинвертированным, имеет внутренние подтяжки на напряжение питания или на землю, и может быть переведен в состояние с высоким импедансом. Когда *GPIO* сконфигурирован как вход, входные значения могут быть прочитаны из регистров. Вход также может быть использован как источник прерываний по уровню или перепаду напряжения.

Запоминающее устройство представлено в виде *Flash*-памяти, объем которой составляет 32 Мбит. *Flash*-память используется для хранения пользовательских программ. Программа переносится в оперативную память с помощью интерфейса *SPI*.

SPI (Serial Peripheral Interface) – последовательный периферийный интерфейс. *ESP8266* имеет два *SPI (SPI и HSPI)* в ведущем и подчиненном режимах.

Приемо-передающий модуль включает в себя *Wi-Fi*-антенну. Она представляет собой радиотехническое устройство для приема и передачи сигналов беспроводного интернета. Приемопередатчик работает в диапазоне 2,4 ГГц. В нем реализован полный 802.11 *b/g/n* протокол.

Данные с *Wi-Fi*-антенны передаются по беспроводному интерфейсу *Wi-Fi* в устройство управления и отображения состояния системы, и наоборот.

Разработка программной части автоматизированной системы управления. Для построения встраиваемых систем используются аппаратные и программные средства. В процессе создания этапы проектирования и разработки неотделимы друг от друга. Решение конечной задачи достигается путем нахождения компромисса в выборе проектной платформы и среды разработки.

Проектные платформы, используемые для создания встраиваемых систем, определяют процесс проектирования и разработки. Широко используемыми в производстве платформами можно назвать промышленные персональные компьютеры, микроконтроллеры, сигнальные процессоры, программно-аппаратные комплексы и контроллеры с программируемой логикой, программируемые логические интегральные схемы и сверхбольшие интегральные схемы программируемой логики.

Одним из важнейших критериев при создании встраиваемых систем является выбор среды программирования.

STM32CubeMX представляет собой графический инструмент, который позволяет легко конфигурировать микроконтроллеры и микропроцессоры производства *STM32*, а также генерировать соответствующий *C*-код инициализации для ядра *ARM Cortex-M* или *ARM Cortex-A*. Его плюсы: простота, совместимость, возможность создание дополнительных пакетов расширений.

Для программирования был выбран язык Си (с англ. *C*), потому что он близко сопоставляется типичным машинным инструкциям, благодаря чему он нашёл применение в проектах, для которых был свойственен язык ассемблера, в том числе как в операционных системах, так и в различном прикладном программном обеспечении для множества устройств – от суперкомпьютеров до встраиваемых систем.

Его компиляторы разрабатываются сравнительно легко, поэтому данный язык доступен на самых различных платформах.

Программирование и разработку на языке *C* поддерживают и устройства компании *STMicroelectronics*.

Для написания программного кода для микроконтроллера *ESP8266* используется программное обеспечение *Visual Studio Code*. Благодаря большому количеству многоплатформенных и кросс-архитектурных инструментов *Visual Studio Code* активно используется для написания программного обеспечения для встраиваемых систем. Одним из этих инструментов является *PlatformIO*.

PlatformIO включает в себя утилиту командной строки, через которую можно запускать компиляцию и загрузку программ на несколько семейств микроконтроллеров (*Atmel AVR, Atmel SAM, ST STM32, TI MSP430* и другие). При этом поддерживаются

разные наборы библиотек (*frameworks*): *Arduino*, *Energia*, *mbed*, а также нативный код для *Atmel AVR*, *espressif*, *MSP430*.

Использована библиотека *HAL* (*Hardware Abstraction Layer*, с англ. слой аппаратных абстракций) встроенное программное обеспечение уровня абстракции *STM32Cube HAL*. *HAL API* доступны для всех периферийных устройств, низкоуровневые *API* (*application programming interface*, с англ. интерфейс прикладного программирования), предлагающие быстрый и легкий уровень, ориентированный на экспертов, который ближе к конфигурируемому оборудованию, чем *HAL*, универсальные *API*, которые предоставляют общие и универсальные функции для всех серий *STM32*.

Уровень *HAL* предоставляет простой универсальный набор *LL* (*low-layer*, с англ. низкоуровневые) *API* для взаимодействия с верхним уровнем программирования (приложением, иными библиотеками и стеками).

Использованные библиотеки и *API* упрощают реализацию пользовательских приложений, реализуют обнаружение сбоев во время выполнения путем проверки входных значений всех функций.

При программировании автоматизированной системы управления используется библиотека, которая отображает прием и передачу данных по *MQTT* протоколу.

MQTT – это протокол обмена данными между издателями и подписчиками, который был разработан специально для технологии интернета вещей (*IoT*). В основе протокола *MQTT* лежит идея пересылки небольших сообщений, например показаний датчиков, между устройствами.

```
void UART_MQTT_init(void);
void UART_MQTT_Check_net(void);
void UART_MQTT_Send_data_bool(uint8_t VARIABLE_ID, bool data, uint8_t *tx_buffer);
void UART_MQTT_Send_data_int8_t(uint8_t VARIABLE_ID, int8_t data, uint8_t *tx_buffer);
void UART_MQTT_Send_data_uint8_t(uint8_t VARIABLE_ID, int8_t data, uint8_t *tx_buffer);
void UART_MQTT_Send_data_int16_t(uint8_t VARIABLE_ID, int16_t data, uint8_t *tx_buffer);
void UART_MQTT_Send_data_uint16_t(uint8_t VARIABLE_ID, uint16_t data, uint8_t *tx_buffer);
void UART_MQTT_Send_data_int32_t(uint8_t VARIABLE_ID, int32_t data, uint8_t *tx_buffer);
void UART_MQTT_Send_data_uint32_t(uint8_t VARIABLE_ID, uint32_t data, uint8_t *tx_buffer);
void UART_MQTT_Send_data_float(uint8_t VARIABLE_ID, float data, uint8_t *tx_buffer);
bool UART_MQTT_Receive_data_bool(uint8_t *rx_buffer);
int8_t UART_MQTT_Receive_data_int8_t(uint8_t *rx_buffer);
uint8_t UART_MQTT_Receive_data_uint8_t(uint8_t *rx_buffer);
int16_t UART_MQTT_Receive_data_int16_t(uint8_t *rx_buffer);
uint16_t UART_MQTT_Receive_data_uint16_t(uint8_t *rx_buffer);
int32_t UART_MQTT_Receive_data_int32_t(uint8_t *rx_buffer);
uint32_t UART_MQTT_Receive_data_uint32_t(uint8_t *rx_buffer);
float UART_MQTT_Receive_data_float(uint8_t *rx_buffer);
bool UART_MQTT_Checksumm_validation(uint8_t *rx_buffer);
void UART_MQTT_Receive_Callback(void);
```

Передача и прием данных по *MQTT* протоколу на языке *C* представлены на рисунке 5.

Рисунок 5. Передача и прием данных по *MQTT* протоколу

Далее, написанный файл библиотеки *.h* (в рассматриваемом случае это *uart_mqtt_lib.h*), добавляется директивой *#include* в библиотеку *.c*, в которой при помощи рассмотренных ранее указаний, пишутся функции передачи данных по *MQTT* протоколу. Пример передачи переменной *bool* представлен на рисунке 6.

```
/*Отправка bool в массив*/  
/// \param VARIABLE_ID - номер переменной (должен совпадать на  
передающем и принимающем устройстве)  
/// \param data - переменная, которую следует отправить  
/// \param *tx_buffer - буфер, который используется для отправки  
void UART_MQTT_Send_data_bool(uint8_t VARIABLE_ID, bool data, uint8_t  
*tx_buffer) {  
    tx_buffer[0] = 0xFF;  
    tx_buffer[1] = VARIABLE_ID;  
    tx_buffer[2] = 0x01;  
    tx_buffer[3] = 0x0;  
    tx_buffer[4] = 0x0;  
    tx_buffer[5] = 0x0;  
    tx_buffer[6] = data;  
    uint16_t CRC_tx_buffer = (~(tx_buffer[1] + tx_buffer[2] + tx_buffer[3] +  
tx_buffer[4] + tx_buffer[5] + tx_buffer[6])) + 1;  
    tx_buffer[7] = CRC_tx_buffer >> 8;  
    tx_buffer[8] = CRC_tx_buffer;  
    if (huart_used.gState != HAL_UART_STATE_BUSY_TX) {  
        HAL_UART_Transmit_IT(&huart_used, tx_buffer, 9);  
    }  
}
```

Рисунок 6. Передача переменной *bool*

Заключение. Автоматизированная система управления в виде лабораторного макета на базе микроконтроллера *STM32* для дисциплины «Встраиваемые системы», отвечает требованиям учебной программы.

Совместное использование *STM32* и *ESP8266* снижает затраты на реализацию лабораторного макета за счет интеграции нескольких решений для подключения в один модуль. Тщательное разделение программного обеспечения позволяет решать проблемы с подключением и программными ограничениями в режиме реального времени в одном модуле.

С помощью разработанного лабораторного макета автоматизированной системы управления на базе микроконтроллера *STM32* можно создать прототип умного дома, который решает конкретные задачи пользователей.

Список литературы

- [1] Предко М. Руководство по микроконтроллерам: В 2-х т. – Пер. с англ. – М.: Постмаркет, 2001. – Т. 1 – 415с., – Т. 2 – 487 с.
- [2] Керниган Брайан У. Язык программирования *C* / Ритчи Деннис М., Керниган Брайан У. – М.: Вильямс, 2017. – 288с.
- [3] *STM32F103x6* [Электронный ресурс]: *Datasheet / STMicroelectronic* – Режим доступа: *stm32f103c8t6.pdf*.

Авторский вклад

Дворникова Татьяна Николаевна – постановка задачи исследования, руководство разработкой автоматизированной системы управления на базе микроконтроллера *STM32*.

Ипатов Герман Романович – написание программного кода, формирование структуры статьи.

AUTOMATED CONTROL SYSTEM BASED ON STM32 MICROCONTROLLER

G.R. Ipatov

*Student of the Information Radio
Technologies Department of
BSUIR*

T.N. Dvornikova

*Senior lecturer of the Information
Radio Technologies Department of
BSUIR, Master of Technical
Sciences*

Abstract. Advanced technologies of embedded systems, which allow automating household tasks, are rapidly developing with each day. An automated control system based on the microcontroller STM32 is implemented as a laboratory prototype for the «Embedded Systems» discipline, as well as an algorithm of system functioning is developed. Developing up-to-date and in-demand laboratory equipment allows students to get acquainted with the latest technologies of embedded and cyber-physical systems, and is an important aspect of ensuring high-quality training of highly qualified specialists.

Keywords: MQTT architectures, technical requirements, electrical schematic structural, electrical schematic functional, electrical schematic principal, microcontroller, embedded systems.