

## Литература

1 Хавербеке, М. Выразительный JavaScript. Современное веб-программирование / М. Хавербеке. – Санкт-Петербург : Питер, 2021. – 480 с.

УДК 004.62

*Р. Ю. Громыко*

### ОБ ОДНОМ СПОСОБЕ СБОРА, ХРАНЕНИЯ, ПЕРЕДАЧИ И СИСТЕМАТИЗАЦИИ БОЛЬШИХ ДАННЫХ

*В статье рассматриваются вопросы разработки мобильного приложения, которое позволяет обнаруживать и подключаться к платам ESP-32 через Bluetooth с целью получения, регулярного сбора, хранения и передачи данных. Описывается одна из возможных реализаций приложения, позволяющего получить данные с датчиков, а также обмениваться данными (отправка и приём) с выбранной платой.*

**Введение.** Большие данные, к которым относятся показатели различных датчиков, используются при решении прикладных задач в различных предметных областях. Поэтому задача их получения, хранения, передачи и систематизации актуальна и востребована в настоящее время. Благодаря своевременному контролю и обработке больших данных можно предсказывать погодные явления, контролировать работы автономных систем, отслеживать состояния людей и животных. И это только малая часть всех примеров. Многие из вышеперечисленного можно реализовать с помощью микроконтроллеров Arduino, например, на плате ESP-32. В чём их плюс? Они достаточно гибкие и легко могут подстроиться под нужды пользователя благодаря большому количеству модулей и датчиков [1]. Так же платы достаточно дешевы. Их цена является низкой, что позволяет делать большие системы, состоящие из множества датчиков. Ко всему этому микроконтроллеры достаточно легко программировать. Поэтому существуют детские клубы, посвящённые обучению программированию микроконтроллеров.

В статье описывается разработка и реализация скрипта для обработки данных, поступающих с датчиков, и записи их в файл на SD карту, подключённую через модуль к плате.

**Программная реализация получения больших данных.** Для реализации скрипта было использовано приложение Arduino IDE. Для постоянной работы платы был создан цикл, в котором предусмотрена возможность ответной реакции на команды с целью записи данных. Использовался встроенный метод loop() Arduino, который является циклом. Для предупреждения чрезмерного количества проверок внешних запросов в цикле присутствует задержка на 10 миллисекунд, выполненная с помощью команды delay(10). Предполагается, что в результате срабатывания каждые 1000 циклов в течении 10 секунд будут считываться данные с погодных датчиков.

**Организация работы с датчиками.** Сама по себе плата ESP32 оснащена встроенным датчиком Холла, который обнаруживает изменения магнитного поля в его окружении. В данном проекте предусмотрена фиксация данных с этого датчика. Фиксация происходит в том же цикле обмена данными программы. Поступление информации от датчика организовано с помощью команды hallRead().

Помимо встроенного датчика Холла к микропроцессору ESP-32 были подключены 3 датчика и модуль для вставки SD-карты. Работа с ними осуществлялась с помощью специальных библиотек.

Для того, чтобы плата могла обмениваться данными с датчиками и модулем был создан протокол последовательной связи. Это было сделано с помощью библиотеки “Wire.h”. Протокол был инициализирован с помощью метода `Wire.begin()`. После этого была начата работа с датчиками.

Первый датчик – это датчик освещенности BH1750. Он позволяет получить данные об освещении места, в котором находится датчик. Данные измеряются в ЛСК (люксах). Для работы с этим датчиком в скрипт была добавлена библиотека “BH1750.h”, а в скрипте для работы с датчиком был создан экземпляр класса `BH1750 lightMeter`. Для обращения к датчику была создана переменная `lightMeter`. Для запуска датчика использовался метод `begin()` в методе скрипта `setup()`. Далее в методе скрипта `loop()` можно получить значения с датчика методом `readLightLevel()`, который возвращает `float` значение.

Следующий датчик – это датчик атмосферного давления и температуры BMP280. Он обеспечивает измерение атмосферного давления и температуры окружающей среды. Для работы с этим датчиком в скрипт была добавлена библиотека “Adafruit\_BMP280.h”, а в скрипте был создан экземпляр класса `Adafruit_BMP280 bmp`. После этого появилась возможность обращаться к датчику через переменную `bmp`. Для того, чтобы датчик запустился, использовался метод `begin(0x76)` в методе скрипта `setup()`. Таким образом была обеспечена возможность в методе скрипта `loop()` получать данные с этого датчика. Для получения данных о температуре использовался метод `readTemperature()`, который возвращает `float` значение (значение температуры в градусах Цельсия). Для получения данных о давлении использовался метод `readPressure()`, который возвращает `float` значение (значение давления в гПа).

Третий датчик – это датчик влажности и температуры AHT20. Он обеспечивает фиксацию информации о влажности и температуре окружающей среды. Для работы с этим датчиком в скрипт была добавлена библиотека “Adafruit\_AHTX0.h” и создан в скрипте экземпляр класса `Adafruit_AHTX0 aht`. Для обращения к датчику использовалась переменная `aht`. Для того, чтобы датчик запустился, использовался метод `begin()` в методе скрипта `setup()`. Все эти действия обеспечили получение данных с этого датчика в методе скрипта `loop()`. Следует отметить, что механизм получения информации с этого датчика отличается от остальных. Данные получают с типом `sensors_event_t` (в проекте – это 2 переменные: `humidity`, `temp_aht`), а их получение обеспечивает метод `getEvent(&humidity, &temp)`, где параметры передаются по ссылке в функцию и в ней записывается значение. Для получения данных о температуре использовался `temp_aht.temperature`, который возвращает `float` значение (значение температуры в градусах Цельсия). Получение данных о влажности было реализовано с помощью `humidity.relative_humidity`, что возвращает `float` значение (значение влажности в процентах).

**Сохранение и структурирование больших данных.** В ходе работы программы в каждый тысячный цикл метода `loop()` все данные с датчиков записываются в строку (рисунок 1), а после и в файл. Первое значение – это данные об изменении магнитного поля в окружении датчик Холла. С использованием набора датчиков платы собирались данные о температуре с помощью двух датчиков (третье и пятое значение). При этом было отмечено, что данные, полученные с них, немного отличаются. Это может обусловлено одной из следующих этих причин: точностью и калибровкой датчиков, реакцией датчиков на окружающую среду, различием в расположении, длительностью нагрева и температурными градиентами.

22.00, 37.50, 26.29, 99190.61, 25.51, 41.66

Рисунок 1 – Строка с полученными данными от датчиков

Данные с датчиков записывались на SD-карту. Для этого были подключены 2 библиотеки: “FS.h” для работы с файлами и “SD.h” для работы с SD-картой. Для того, чтобы

SD-модуль запустился, был использован метод `SD.begin()` в методе скрипта `setup()`. Далее с файлами стало возможно работать через класс `File`. При этом им присваивалось значение `SD.open("/data.txt", FILE_READ)`, где первый параметр – это имя файла, а второй тип работы с файлом (в скрипте используются также типы `FILE_APPEND` и `FILE_WRITE`) (рисунок 2).



Рисунок 2 – Содержимое текстового файла с данными, полученными с датчиков, присоединенных к ESP32

В результате работы программного обеспечения на SD-карте были сформированы 2 файла, в один из которых с заданной частотой записываются данные с датчиков. При получении сигнала на чтение и отправку данных с платы данные одновременно отправляются на другое устройство и записываются на второй файл. После того, как все строки первого файла обрабатываются, первый файл очищается. Таким образом в первом файле хранятся неотправленные данные, а во втором – данные, которые уже были отправлены. Данные во втором файле могут быть очищены с помощью специального сигнала. Такой механизм нужен для того, чтобы иметь резервные данные.

**Автоматизация обмена данными с платой ESP-32 по BLUETOOTH.** Микроконтроллеры ESP32 обладают встроенными Wi-Fi и Bluetooth. Поэтому было разработано мобильное приложение, которое позволяет обнаруживать и подключаться к платам ESP-32 через Bluetooth, а также обмениваться данными с выбранной платой. Был создан канал обмена данными посредством Bluetooth между микроконтроллером ESP-32-WROOM-DA и смартфоном на платформе Android. Поскольку устройства не похожи друг на друга, то для каждого из них были написаны отдельные скрипты. Канал обеспечивает получение и сохранение данных с датчиков. На рисунке 3 представлены данные, полученные с датчика Холла, которые можно просматривать и контролировать с использованием мобильного приложения.

28/11/2023 10:34:04:	59
28/11/2023 10:34:08:	48
28/11/2023 10:34:12:	63
28/11/2023 10:34:16:	36
28/11/2023 10:34:20:	67
28/11/2023 10:34:24:	49

Рисунок 3 – Просмотр информации с датчика Холла

**Заключение.** Разработанное приложение позволяет контролировать платы ESP-32 в зоне действия Bluetooth.

## Литература

1 Ревич, Ю. В. ESP32-C3. Беспроводное приключение. Полное руководство по IoT / Ю. В. Ревич (перевод с англ.). – Москва : ДМК-Пресс, 2023. – 442 с.