

Базовый маршрутизатор. Руководство программиста

Назначение и условия применения программы

Программа реализует логику работы компонента цифровой системы передачи данных. Программа прошивки базового маршрутизатора рассчитана на работу на микроконтроллере esp8266. Для компиляции программы помимо стандартных библиотек платформы Arduino требуются библиотеки RadioHead, Regex и ArduinoJson.

Характеристики программы

Глобально программа состоит из функций setup и loop. Первая выполняется единожды сразу после включения прибора, а тело функции loop повторяется в бесконечном цикле.

Обращение к программе

Программа запускается сразу после включения питания микроконтроллера.

Входные и выходные данные

Программа получает JSON-данные от конечных приборов, кодирует их в двоичном формате и отправляет ретранслятору-шлюзу.

Сообщения

Для отладки и наблюдения за процессом работы базового маршрутизатора в монитор последовательного порта выводятся текстовые сообщения, говорящие о совершении той или иной операции. Например, при получении развёрнутым на базовом маршрутизаторе веб-сервером JSON-строки от конечного прибора, она выводится в монитор порта, а при анализе JSON-значения и переводе его в бинарный формат в монитор последовательного порта выводится сообщение вида "<JSON-ключ> : <JSON-значение> is a <Тип JSON-значения>[размер значения]". Типом JSON-значения может быть целое число (integer), вещественное число (custom float), строка (string), IP-адрес (IP), MAC-адрес (MAC), дата (date) и время (time). Размер значения относится к целым и вещественным числам и может быть равен 8, 16 или 32 бита (8 bit, 16 bit, 32 bit). Если число не умещается в 32 бита, выводится строка вида "error, sending as string", говорящая о том, что число будет передано как текстовая строка.

Перед отправлением каждого пакета выводится строка "Bytes to be sent : ", после которой выводится общее число байт в отсылаемом ретранслятору-шлюзу пакете. В случае, если от ретранслятора-шлюза не приходит подтверждения получения пакета, в монитор порта выводится сообщение "sendtoWait failed". Если же от ретранслятора-шлюза пришло подтверждение, и затем от него пришёл ответный пакет, выводится сообщение "Got reply from LoRa gateway". Если пришло подтверждение, но не пришёл ответный пакет, выводится строка "No reply, is LoRa gateway running?".

Для сокращения объёма передаваемых данных каждому уникальному JSON-ключу ставится в соответствие номер, и это соответствие синхронизировано между базовым маршрутизатором и ретранслятором-шлюзом. На базовом маршрутизаторе оно хранится в виде хеш-таблицы (C++-контейнера unordered_map). Если в процессе разбора входного JSON-текста встречается ключ, отсутствующий в хеш-таблице, то в монитор последовательного порта выводится сообщение "Requesting registration", что означает посылку ретранслятору-шлюзу запроса о регистрации JSON-ключей. Если на этот запрос приходит ответ, то есть пакет, содержащий

номера каждого присланного ретранслятору-шлюзу JSON-ключа, то эти номера заносятся в хеш-таблицу базового маршрутизатора, а в монитор порта выводится сообщение "Registration complete", что свидетельствует о завершении процедуры регистрации JSON-ключей. После этого выводится содержимое хеш-таблицы: множество пар "JSON-ключ: номер".

От ретранслятора-шлюза в ответном пакете может прийти запрос на очистку хеш-таблицы, хранящей соответствие между JSON-ключами и номерами – об этом будет свидетельствовать сообщение "Got registry clear request". После сброса хеш-таблицы выведется сообщение "Registry cleared".

При включении или перезагрузке в монитор последовательного порта выводится сообщение "Access point IP address: ", за которым сразу следует IP-адрес программной точки доступа, развёрнутой на базовом маршрутизаторе. Значение этого адреса жёстко задано как 188.35.161.31. Затем следует сообщение "server started", свидетельствующее о начале работы развёрнутого на базовом маршрутизаторе веб-сервера.

Полученный веб-сервером JSON-документ проходит процедуру десериализации, то есть формирования из текстовой строки некой структуры данных. При возникновении ошибки десериализации в монитор порта будет выведено сообщение "deserializeJson() failed with code ", за которым следует описание ошибки:

- EmptyInput – означает, что входные данные пусты или содержат только пробелы;
- pcompleteInput – входные данные сформированы верно, но заканчиваются раньше, чем нужно;
- InvalidInput – входные данные сформированы неверно с точки зрения формата JSON;
- NoMemory – объект типа JsonDocument имеет недостаточный размер;
- TooDeer – входные данные сформированы верно, но содержат JSON-записи со слишком большим уровнем вложенности.