

Прошивка (Программный код устройства)

Cliff

6 June, 2022

Оглавление

1	Работа программного кода прошивки	1
2	Версия прошивки	2
2.1	Короткая форма версии	2
2.2	Варианты пакетов	3
2.3	Флаги прошивки	3
2.4	SD-карта или внутренняя Flash-память	3
2.5	Поддержка конфигов логики	4
3	Где брать прошивку	4

1 Работа программного кода прошивки

Работа устройства построена по принципу циклической безостановочной работы, разбитой на несколько модулей, каждый из которых обрабатывается циклично, создавая таким образом несколько виртуальных потоков обработки.

Существующие на данный момент модули:

1. Системный

абота с часами, хранение ошибок и контроль их устаревания, запись в файл логов, прошивка ПО, загрузка конфига.

2. Работа с файлами на SD-карте

Инициализация SD-карты, отслеживание изменений в файлах. Здесь в целом нет фоновых задач.

3. Обработчик датчиков

Сбор информации по датчикам (а по некоторым из них информация собирается очень медленно), сохранение значений в памяти.

4. Обработчик реле

Проверка текущего состояния, работа с правилами, расписанием и флагом принудительного включения.

5. Обработчик кнопок

Проверка текущего состояния, генерация события различных вариантов нажатия кнопок.

6. WiFi-соединение

Контроль наличия WiFi-соединения, переподсоединение по необходимости.

7. Соединение с агрегирующим сервером

Создание соединения с контроллером, отправка данных, приём команд на исполнение, передача длинных данных (конфиги).

8. Отрисовка информации на дисплее

Один из базовых принципов построения каждого из этих модулей – не выполняться более 100мс за цикл. Если выполнение какой-то процедуры требует более длительного времени, то выполнение в данном цикле прерывается до следующего, давая работать остальным модулям. Таким образом, по порядку выполняются все модули, и потом всё с начала. Примеры:

1. Хранение ошибок.

После занесения ошибки в память ей присваивается отметка времени занесения. И каждый цикл обработки удаляются устаревшие.

2. Прошивка устройства.

Очевидно, что за 100мс невозможно полностью сохранить файл прошивки в nvram устройства, поэтому она пишется туда небольшими блоками и после каждого такого блока проверяется время выполнения в текущем цикле. Если время достигло 100мс, то данные по текущей записи сохраняются (файловый дескриптор, сколько уже записалось, сколько осталось), а выполнение в текущем цикле передаётся следующему модулю. Сохранённые данные могут быть использованы в другом модуле в этом же цикле, например, при отрисовке на дисплее, что сейчас выполняется процедура прошивки и выполнено 10%.

Также есть случаи, когда ждать не надо совсем. Например, если сервер ничего не шлёт, но и соединение не завершает. Модуль видит, что принимаемых данных нет, и на основании этого прерывает свою работу в данном цикле. В следующем он ещё раз проверит, есть ли данные на входе, и сразу же передаст управление дальше, если данных нет.

Предполагается, что общая задержка одного цикла не должна превышать 200мс за цикл. Потому как сильная загруженность одновременно может быть не более, чем у двух модулей (например, перекачка файла и обработка датчиков). А при слабой загруженности каждый модуль даёт задержку не более 20мс.

Значение суммарной задержки называется латентностью и отображается на дисплее на главном экране. Чем меньше латентность, тем быстрее устройство среагирует на необходимость что-то изменить. Существует минимальное значение латентности. Если суммарная задержка за один цикл по всем модулям не превысила 50 мс, то делается принудительная задержка, чтобы добить суммарную задержку до 60 мс.

2 Версия прошивки

Пример названия версия: `espfw.ver0.94.release.fully.dev.esp8266`

Через точку перечислены:

1. `[espfw]` Основное название проекта, других вариантов пока нет.
2. `[ver0.94]` Номер версии. Чем больше, тем позднее.
3. `[release]` Подверсия.

Варианты:

- `alphaN` – выходит до релиза в тестовых целях (N указывает на последовательный номер такой подверсии),
- `release` – официальный релиз,
- `fixN` – багфиксы после выхода релиза.

4. `[fully]` Пакет – включает все возможные компоненты.
5. `[dev]` Для разрабатываемой версии тут указан `dev`, для боевой (продакшн) – ничего. Разница только в том, на какой контроллер будет соединяться устройство – тестовый или продакшн.
6. `[debug]` Это продакшн-версия прошивки, однако в ней включён режим расширенного логирования, как на тестовой версии. Суффикс `debug` будет указан вместо суффикса `dev`.
7. `[esp8266]` Модель чипа, под который собрана прошивка. Варианта пока два: `esp8266` и `esp32`.

2.1 Короткая форма версии

На дисплее отображается более сокращённое название, например: `f0.98f1p`

1. `[f]` Первая буква от названия пакета `fully`.

2. [0.98] Номер версии без префикса ver.
3. [f1] Подверсия: f1 = fix1, a2 = alpha2, для release ничего не отображается.
4. [p] Это продакшн-версия: p = продакшн, d = тестовая, pd = продакшн debug (с расширенными логами).

2.2 Варианты пакетов

1. Fully – полный набор всех поддерживаемых функций.
2. Jaggy – экспериментальная версия прошивки, где либо испытываются нестабильные функции, либо уже давно работающие функции работают на ином, чем ранее принципе. Например, часы используются не внешние, а внутренние, требующие синхронизации при первом старте устройства, а вместо внешней SD-карты используется внутренняя Flash-память.
3. Pussy – облегчённая версия. Полностью отсутствует логика (таймеры, события, правила реле, переменные) и поддержка SD-карты. Устройство может передавать собственные события (само обрабатывать эти события не умеет), данные с датчиков и состояния реле. А также может принимать прямые управляющие команды (например, включить/выключить реле).

Планируется в будущем выход экономной прошивки, которая не будет поддерживать входящие команды, работу реле и будет даже без дисплея, а сможет только передавать данные о своём состоянии (с увеличенным интервалом обновления) и некоторые события. При этом появится возможность работать такому устройству автономно от батареек или аккумуляторов.

2.3 Флаги прошивки

Флаги определяют параметры работы прошивки, при этом не конфигурируются (защиты на стадии компиляции прошивки).

- dev – тестовая версия прошивки (соединяется на тестовый сервер).
- alpha – альфа-версия (ещё до релиза).
- esp8266 – скомпилировано под чип esp8266.
- esp32 – скомпилировано под чип esp32.
- master – устройство может принимать и логически обрабатывать данные.
- economu – устройство работает в экономном режиме (увеличены интервалы обновления информации).
- txtcfg – поддержка конфигов логики.
- display – поддержка работы дисплея.
- pinextend – поддержка работы расширителей пинов реле/кнопок.
- relay – поддержка управления реле.
- button – поддержка работы кнопок.
- onewire – поддержка one-wire устройств (термодатчик и остальные простые).
- sdcard – поддержка внешней SD-карты.
- sdint – использование внутренней Flash-памяти для сохранения конфигов.
- clockext – поддержка внешних часов на чипе DS3231.
- clockint – поддержка внутренних часов (требуется синхронизация при старте устройства).
- filelog – поддержка хранения текстовых логов на SD-карте.
- memlog – поддержка хранения логов в памяти устройства.
- loclink – поддержка связи нескольких устройств между собой.
- servo – поддержка серво-приводов.
- analog – поддержка аналоговых датчиков (съём значений с аналоговых пинов).

2.4 SD-карта или внутренняя Flash-память

SD-карта (либо внутренняя Flash-память) играет ключевую роль в работе устройства, многое сильно меняется при отсутствии её поддержки.

Если не прошивкой не поддерживается внешняя SD-карта:

1. Конфиги больше не сохраняются. Они всё ещё могут поддерживаться, но при старте устройства их надо инициализировать через сеть.
2. В том числе отсутствует сохранение whitelist у GSM-модема. Это означает, что после старта устройства до инициализации whitelist модем принимает звонки и sms с любых телефонов.

3. По умолчанию в конфиге GSM-модем выключен, иначе при отсутствии самого модема могут быть шумы на гх-входе, которые будут постоянно интерпретироваться как ошибки. Поэтому при старте устройства надо так же включить модем.
4. Удалённая прошивка устройства через сохранение файла на SD-карте невозможна. Сохранение файла прошивки возможно только на внешнюю SD-карту, без её поддержки возможно обновление прошивки только командой `/firmwareupload`.
5. Без внешней SD-карты не сохраняются состояния реле и переменных.

2.5 Поддержка конфигов логики

Если нет поддержки конфигов логики: В этом случае устройство перестаёт автономно принимать решения, т.к. отсутствует поддержка правил реле, событий, расписания.

И как следствие за ненадобностью отсутствуют флаги реле: «принудительное включение» и «расписание».

Однако, сохраняется возможность удалённого управления – непосредственное включение/выключение реле.

3 Где брать прошивку

- Последний протестированный релиз можно скачать в [Панели управления](#). Раздел Конфигурация.
- [Архив прошивок](#).