

ПРИМЕНЕНИЕ WI-FI-МОДУЛЯ HF-LPB100 В СИСТЕМАХ КОНТРОЛЯ ДОСТУПА



В век развития беспроводных технологий задача уменьшения количества мобильных устройств (а термин «мобильный» теперь все чаще означает «носимый»), реализующих такие функции, как мониторинг и управление исполнительными механизмами, весьма актуальна. И хотя концепцию носимых технологий еще нельзя назвать окончательно сформировавшейся, развивается она очень быстро. В статье предлагается один из вариантов решения проблемы избыточности индустриальных беспроводных контрольно-управляющих устройств.

Контроль датчиков автоматике в системах «умный дом», управление доступом на территорию, мониторинг состояния механизмов — ввиду устоявшейся практики применения проприетарных протоколов пользователи для каждой системы вынуждены использовать собственное/оригинальное устройство доступа (назовем его «брелок»), что, со временем и ростом подключаемых по технологии IoT (Internet of Things, «Интернет вещей») устройств, становится проблемой. На частичное решение данной задачи направлен проект ООО «Системы Индиго» по разработке устройства управления механизмами доступа на производственную территорию, в основу которого положен модуль HF-LPB100 компании High-Flying [1].

Проект, для реализации которого «Системы Индиго» выбрала модуль HF-LPB100 (рис. 1), имеет целью создать единое устройство удаленного управления автоматическими воротами, шлагбаумами, ролл-ставнями и прочими механизмами через сеть Wi-Fi.



Рис. 1. Внешний вид Wi-Fi-модуля HF-LPB100-1

Почему Wi-Fi?

Применение беспроводного канала на базе Wi-Fi позволяет сменить индивидуальные радиобрелки на приложение в смартфоне пользователя и тем самым не только сократить количество носимых устройств, но и сэкономить на их приобретении для всех пользователей (членов семьи или сотрудников организации).

Принцип работы устройства прост: пользователь со смартфона осуществляет подключение к точке доступа, после

чего может зарегистрировать и сконфигурировать устройство для работы с различными типами управляющих механизмов. При этом безопасность обеспечивается как средствами Wi-Fi (авторизация с применением механизмов WPA2-PSK), так и дополнительными индивидуальными ключами (Token). Использование беспроводного канала на основе технологии Wi-Fi позволяет не только существенно сократить расходы на развертывание и эксплуатацию системы, но и реализовать дополнительный функционал. В рамках проекта модуль HF-LPB100 (рис. 2) используется не просто как оборудование для формирования канала связи Wi-Fi, а как полноценный микроконтроллер с выполнением ряда интеллектуальных функций.

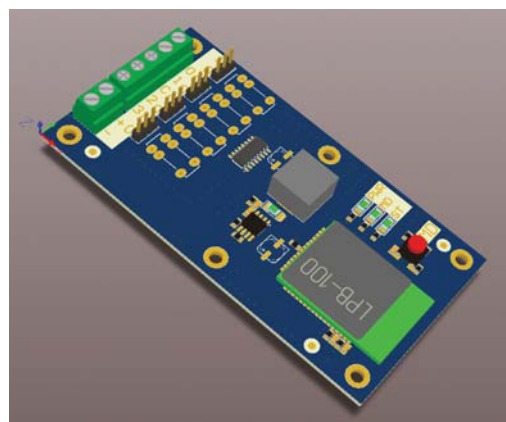


Рис. 2. Модуль управления механизмами

Почему HF-LPB100?

Модуль оснащен процессором на базе ядра Cortex-M3, позволяющим обрабатывать не только данные, необходимые для реализации стандартных функций Wi-Fi, но и пользовательское программное приложение. Набор SDK, представленный производителем, имеет достаточную документацию для реализации различных интеллектуальных функций, которые заданы требованиями проекта, например хранение и обработка пользовательских Token для контроля доступа к устройству.

Небольшой размер позволяет применять модуль в проектах, требующих компактного размещения, что позволяет снизить затраты на производство. Модуль HF-LPB100 является интегрированным решением, не требует сложной обвязки, поддерживает напряжение питания 3,3 В, что дает возможность снизить требования к печатной плате и уменьшить размер всего устройства. Контроллер HF-LPB100 обеспечивает высокое быстродействие, достаточное для выполнения пользовательских приложений, обеспечивает работу прошивки в многопоточном режиме. Дополнительной особенностью данного устройства является низкое

энергопотребление, что позволяет использовать его в системах с автономным электропитанием. Широкий температурный диапазон предусматривает возможность использования модуля в различных решениях.

Технические характеристики модуля HF-LPB100 представлены в таблице 1 [2].

Таблица 1. Основные параметры HF-LPB100

Стандарт	802.11 b/g/n
Частотный диапазон, ГГц	2,412–2,484
Выходная мощность	до +16 dBi
Скорость передачи данных, Мбит/с	до 54
Интерфейсы	UART, SPI, PWM, GPIO
Напряжение питания, В	2,8–3,6
Ток потребления, мА (макс)	200
Шифрование	WEP/WPA-PSK/WPA2-PSK/AES
Режимы	STA/AP/STA+AP
Протоколы	IPv4, TCP/UDP/FTP/HTTP
Управление	AT-команды, SDK
Размеры, мм	23,1×32,8×2,7
Диапазон рабочих температур, °С	–40...+85

Более подробное описание параметров модуля HF-LPB100 приведено в [3].

Модуль поддерживает беспроводные стандарты IEEE802.11b/g/n, технологию WPS в режиме клиента. Устройство может работать как точка доступа (AP), клиент (STA) или в смешанном режиме (STA+AP), при этом поддерживается до двух соединений. Для подключения и управления модуль имеет два сокетa (SocketA и SocketB), которые могут работать в режимах клиент/сервер, поддерживают протоколы UDP и TCP. Сокеты поддерживают до 5 одновременных подключений. Необходимо отметить, что при попытке подключения большего количества клиентов модуль «сбрасывает» пытающегося подключиться клиента. При попытке подключения к сокетам большего количества соединений последнему соединению направляется отказ.

Помимо изначально задуманного предназначения (единое устройство удаленного управления автоматическими воро-

тами, шлагбаумами, ролл-ставнями и прочими механизмами через сеть Wi-Fi), данные модули могут быть востребованы в следующих областях:

- навигация;
- авторегистраторы;
- АСКУЭ и автоматизация;
- нефтегазовое оборудование;
- системы диагностики;
- СКУД;
- блоки охранно-пожарной сигнализации.

1.1. Варианты исполнения модуля

HF-LPB100 выпускается в нескольких версиях:

- HF-LPB100-0 с IPEX-разъемом для подключения внешней антенны;
- HF-LPB100-1 со встроенной керамической антенной;
- HF-LPB100-SMA с разъемом SMA.

Все версии полностью совместимы, что позволяет выпустить несколько модификаций конечного устройства с различными требованиями к уровню радиосигналов.

1.2. Интерфейсы и назначение портов ввода-вывода

В рамках проекта задействованы следующие интерфейсы модуля:

- 6 каналов PWM/GPIO — управление реле, светодиодная индикация, кнопка;
- 1 интерфейс UART — прошивка и отладка модуля.

В последовательном интерфейсе присутствуют линии контроля передачи данных RTS/CTS с пропускной способностью до 460800 кбод/с. По умолчанию производителем установлена скорость 115200 кбод/с, изменение которой можно осуществить AT-командами.

GPIO позволяют управлять реле или светодиодами, а также обрабатывать нажатие кнопок и обладают следующими характеристиками:

- выходное напряжение низкого уровня — не более 0,2 В;
- выходное напряжение высокого уровня — не менее 2,8 В;
- максимальный выходной ток — 6 мА;
- входное напряжение низкого уровня — не более 0,8 В;
- минимальное входное напряжение высокого уровня — не менее 2,2 В.

Линии PWM позволяют реализовать управление посредством ШИМ, например мигание светодиодом или управление сервоприводом, и обладают следующими характеристиками:

- выходное напряжение низкого уровня — не более 0,2 В;
- выходное напряжение высокого уровня — не менее 2,8 В;
- максимальный выходной ток — 6 мА;
- диапазон частот — 500 Гц...60 кГц;
- скважность — 0...100%.



Важным вопросом при проектировании изделия является выбор модуля Wi-Fi, который будет не только отвечать заявленным характеристикам, но и способен сам решать простые задачи, без применения внешнего микроконтроллера. И на примере изделия компании «Системы Индиго» мы убедились, что модуль HF-LPB100-1 способен решать поставленные задачи. Модуль поддерживает необходимые режимы работы как точка доступа или клиент, обеспечивает стабильный канал связи, обладает промышленным температурным диапазоном. Наличие SDK позволяют существенно миниатюзировать изделие, т.е. сделать его более компактным, сохранив устойчивость связи и высокую производительность.

Иван Гончаров,
инженер по внедрению холдинга PT Electronics,
ivan.goncharov@ptelectronics.ru



ДАННЫЕ МОДУЛИ НАЙДУТ ШИРОКОЕ ПРИМЕНЕНИЕ В СЛЕДУЮЩИХ СФЕРАХ:

- умный дом;
- навигация;
- регистраторы;
- АСКУЭ и автоматизация;
- нефтегазовое оборудование;
 - системы диагностики;
 - СКУД;
- блоки охранно-пожарной сигнализации.

2. Беспроводной интерфейс и функционал канала связи

Модуль представляет широкие сетевые возможности, поддерживая беспроводные стандарты IEEE802.11b/g/n, технологию WPS в режиме клиента. Устройство может работать как точка доступа (AP), клиент (STA) или в смешанном режиме (STA+AP), при этом поддерживается до 2 соединений. Для подключения и управления модуль имеет 2 сокетa (SocketA и SocketB), которые могут работать в режимах клиент или сервер, поддерживают протоколы UDP и TCP. Сокеты поддерживают до 5 одновременных подключений.

Стоит отметить, что при попытке подключения большего количества клиентов модуль сбрасывает пытающегося подключиться клиента. При попытке подключения к сокетам большего количества соединений последнему соединению направляется отказ.

3. SDK модуля. Основные возможности

Компания Hi-Flying представляет SDK для работы с модулем. На сайте в разделе SDK Support представлен Hi-Flying Software Framework. На момент написания статьи актуальная версия v1.40. Внутри архива имеются необходимые утилиты для прошивки и обновления модуля, примеры проектов, документация. SDK рассчитан на разработку в среде mVision4 от Keil. В комплекте присутствует инструкция по настройке среды разработки и интеграции в нее Hi-Flying Software Framework. В SDK включены примеры программ, поясняющие процесс разработки с учетом основных функциональных частей модуля HF-LPB100 — работой с сокетами, Wi-Fi, GPIO, таймерами, UART, флэш-памятью и т. д. Модуль позволяет осуществлять работу с AT — командами непосредственно из программного кода, в том числе имеется возможность анализа и обработки ответов. Это позволяет, например, во время исполнения программы на готовом устройстве изменить режим работы Wi-Fi, что дает простор при создании сложных или многофункциональных устройств.

4. Структура программы, пример исходного кода

Точка входа в приложение — файл app_main.c.

В качестве примера в SDK файл содержит:

- функцию инициализации всех линий ввода-вывода;
- функции инициализации сетевых сервисов.

Функция обратного вызова для получения событий с сетевого сокета (SOCKET A, по умолчанию — TCP-сервер):

```
static int USER_FUNC socketa_recv_callback(uint32_t event, char *data, uint32_t len, uint32_t buf_len)
{
    if(event==HFNET_SOCKETA_DATA_READY)
    {
        HF_Debug(DEBUG_LEVEL_LOW, «socketa recv %d bytes %d\n», len, buf_len);
    }
    else if(event==HFNET_SOCKETA_CONNECTED)
        u_printf(«socket a connected!\n»);
    else if(event==HFNET_SOCKETA_DISCONNECTED)
        u_printf(«socket a disconnected!\n»);
    return len;
}
```

Подобным образом выглядят и другие функции обратного вызова, например для таймеров.

Ниже представлена функция для тестирования работы четырех управляющих реле. Задача — проинициализировать четыре линии GPIO, по таймеру раз в секунду по очереди посылать управляющие команды на реле.

На листинге ниже происходит инициализация четырех линий GPIO на 11, 12, 18, 20.

Задается смещение для пользовательских линий GPIO для инициализации карты линий ввода-вывода:

```
#define HFGPIO_F_GATEWAY_GPIO1 (HFGPIO_F_USER_DEFINE+0)
#define HFGPIO_F_GATEWAY_GPIO2 (HFGPIO_F_USER_DEFINE+1)
#define HFGPIO_F_GATEWAY_GPIO3 (HFGPIO_F_USER_DEFINE+2)
#define HFGPIO_F_GATEWAY_GPIO4 (HFGPIO_F_USER_DEFINE+3)
```

Инициализация используемых пользователем пинов GPIO:

```
const int hf_gpio_fid_to_pid_map_table[HF_MAX_FUNC_CODE]=
```

```
{
    HF_M_PIN(2), //HFGPIO_F_JTAG_TCK
    HF_M_NOPIN, //HFGPIO_F_JTAG_TDO
    HF_M_NOPIN, //HFGPIO_F_JTAG_TDI
    HF_M_PIN(5), //HFGPIO_F_JTAG_TMS
    HF_M_NOPIN, //HFGPIO_F_USBDP
    HF_M_NOPIN, //HFGPIO_F_USBDM
    HF_M_PIN(39), //HFGPIO_F_UART0_TX
    HF_M_PIN(40), //HFGPIO_F_UART0_RTS
    HF_M_PIN(41), //HFGPIO_F_UART0_RX
    HF_M_PIN(42), //HFGPIO_F_UART0_CTS
```

```
    HF_M_PIN(27), //HFGPIO_F_SPI_MISO
    HF_M_PIN(28), //HFGPIO_F_SPI_CLK
    HF_M_PIN(29), //HFGPIO_F_SPI_CS
    HF_M_PIN(30), //HFGPIO_F_SPI_MOSI
```



```

HFM_NOPIN, //HFGPIO_F_UART1_TX,
HFM_NOPIN, //HFGPIO_F_UART1_RTS,
HFM_NOPIN, //HFGPIO_F_UART1_RX,
HFM_NOPIN, //HFGPIO_F_UART1_CTS,
HF_M_PIN(43), //HFGPIO_F_NLINK
HF_M_PIN(44), //HFGPIO_F_NREADY
HF_M_PIN(45), //HFGPIO_F_NRELOAD
HF_M_PIN(7), //HFGPIO_F_SLEEP_RQ
HFM_NOPIN, //HFGPIO_F_SLEEP_ON

HF_M_PIN(15), //HFGPIO_F_WPS
HFM_NOPIN, //HFGPIO_F_RESERVE1
HFM_NOPIN, //HFGPIO_F_RESERVE2
HFM_NOPIN, //HFGPIO_F_RESERVE3
HFM_NOPIN, //HFGPIO_F_RESERVE4
HFM_NOPIN, //HFGPIO_F_RESERVE5

HF_M_PIN(11), //HFGPIO_F_USER_DEFINE HFGPIO_F_
GATEWAY_GPIO1
HF_M_PIN(12), //HFGPIO_F_USER_DEFINE HFGPIO_F_
GATEWAY_GPIO2
HF_M_PIN(18), //HFGPIO_F_USER_DEFINE HFGPIO_F_
GATEWAY_GPIO3
HF_M_PIN(20), //HFGPIO_F_USER_DEFINE HFGPIO_F_
GATEWAY_GPIO4
};

```

В главной функции вызывается системная функция `hfgpio_fmap_check()`, которая применяет карту конфигурации портов ввода-вывода (если в ней нет ошибок).

Написанная пользовательская функция `gpio_test()` при каждом вызове по очереди выбирает реле.

```

int gpioTestMachine = 0;
static void USER_FUNC gpio_test(void) {
//

```

```

switch(gpioTestMachine)
{
case 0: // init
{
gpioTestMachine = 1;
break;
}
case 1:
{
hfgpio_fset_out_high(HFGPIO_F_
GATEWAY_GPIO1);
hfgpio_fset_out_low(HFGPIO_F_
GATEWAY_GPIO4);
gpioTestMachine = 2;
break;
}
case 2:
{
hfgpio_fset_out_high(HFGPIO_F_
GATEWAY_GPIO2);
hfgpio_fset_out_low(HFGPIO_F_
GATEWAY_GPIO1);
gpioTestMachine = 3;
break;
}
case 3:
{
hfgpio_fset_out_high(HFGPIO_F_
GATEWAY_GPIO3);
hfgpio_fset_out_low(HFGPIO_F_
GATEWAY_GPIO2);
gpioTestMachine = 4;
break;
}
case 4:
{
hfgpio_fset_out_high(HFGPIO_F_
GATEWAY_GPIO4);
hfgpio_fset_out_low(HFGPIO_F_
GATEWAY_GPIO3);
gpioTestMachine = 1;
}
default:
break;
}
}

```

Функция `hfgpio_fset_out_high(HFGPIO_F_GATEWAY_GPIO4)` устанавливает высокий уровень напряжения на выходе `GPIO4`, соответственно, функция `hfgpio_fset_out_low(HFGPIO_F_GATEWAY_GPIO3)` устанавливает низкий уровень напряжения на выходе `GPIO3`, таким образом осуществляется изменение состояния соответствующих выходов GPIO. Далее рассмотрим настройку тестового таймера, срабатывающего каждую секунду.

Выполним инициализацию:

```

hftimer_handle_t gpioTestTimer = NULL;
#define GPIO_TEST_TIMER_ID (1)

```

```

int USER_FUNC app_main (void)
{
.....
if((gpioTestTimer = hftimer_create(«gpioTestTimer», 1000,
true, GPIO_TEST_TIMER_ID, gpio_test_timer_callback, 0))!=
NULL)

```

```

{
    u_printf(«[ERR] GPIO test timer fail \n»);
}
else
{
    u_printf(«[OK] GPIO test timer make \n»);
    hftimer_start(gpioTestTimer);
}
}
.....
}

```

Функция из SDK HF-LPB100 `hftimer_create()` создаст таймер, в аргументы функции передается имя таймера (`gpioTestTimer`), время срабатывания в миллисекундах (`1000`), тип таймера — срабатывающий один раз (`false`) или регулярно (`true`), ID таймера (`GPIO_TEST_TIMER_ID`), а также функцию обратного вызова (`gpio_test_timer_callback`). Функция `hftimer_start()` запускает созданный таймер, в нашем случае — после успешного создания таймера.

Функция обратного вызова, которая исполняется в случае срабатывания таймера:

```

int gpioTimerCountCallback = 0;
void gpio_test_timer_callback(hftimer_handle_t htimer)
{
    int flagZero = -1;

    if(hftimer_get_timer_id(htimer) == GPIO_TEST_TIMER_ID)
    {
        flagZero = 1;
        if(gpioTimerCountCallback > 0) gpio_test();

        gpioTimerCountCallback++;

        if(flagZero == 1) gpioTimerCountCallback = 0;
    }
}

```

В процессе работы с таймерами выявлена следующая особенность — при инициализации циклического таймера происходит его первое немедленное срабатывание. С целью этого избежать была введена переменная `gpioTimerCountCallback = 0`, приращение которой происходит каждое срабатывание таймера, выполнение программы возможно только после первого приращения.

Работу периферийных интерфейсов необходимо инициализировать следующим образом (присутствует отладочный вывод в UART):

```

if(hfnet_start_uart(HFTHREAD_PRIORITIES_LOW,(hfnet_callback_t)NULL)!=HF_SUCCESS)
{
    HF_Debug(DEBUG_WARN,«start uart fail!\n»);
}
if(hfnet_start_socketa(HFTHREAD_PRIORITIES_LOW,(hfnet_callback_t)socketa_recv_callback)!=HF_SUCCESS)
{
    HF_Debug(DEBUG_WARN,«start socketa fail!\n»);
}

```

Принципы работы с функциями, значения переменных и примеры использования детально описаны в документа-

ции на модуль в пакете High-Flying Software Framework, API Reference Manual.

5. Прошивка и отладка ПО модуля

Модуль LPB100 поддерживает обновление прошивки несколькими способами:

- Через отладочный комплект ULINK от KEIL. Модуль позволяет производить отладку приложения в процессе разработки через интерфейс JTAG.
- Через UART. Необходимо запустить модуль с зажатым кнопкой RELOAD и одновременно зажатым пробелом в активном окне терминала. BIN-файл необходимо отправить с помощью протокола XMODEM. Производитель Hi-Flying рекомендует использовать для этих целей программу SecureCRT. Способ удобен для прошивки единичных модулей.
- Используя HF Production Upgrade Tool. С помощью этой утилиты можно обновлять программное обеспечение через Wi-Fi. Возможна автоматическая прошивка до 8 устройств одновременно. Способ предназначен для загрузки программ в готовую продукцию. Используется BIN-файл с пометкой UPGRADE. Таким образом, обеспечивается максимальное быстродействие при обновлении прошивок на большом количестве готовой продукции. Загрузка программы через Wi-Fi — не самый лучший способ в процессе разработки, т.к. при допущении серьезной ошибки в коде и невозможности загрузки устройства повторное обновление на рабочую прошивку возможно только через UART или ULINK.
- Возможна загрузка прошивки через WEB-интерфейс, что может быть удобно при удаленном обновлении устройства.

6. Заключение

Применение в проекте модуля HF-LPB100 позволило отказаться от использования отдельного микроконтроллера. Разработанное с использованием Hi-Flying Software Framework программное обеспечение модуля позволяет осуществлять аутентификацию и идентификацию пользователей, управлять Token-ами пользователей, осуществлять управление двумя группами исполнительных механизмов с выдачей управляющих сигналов. Управляющие сигналы на исполнительные механизмы имеют программируемые профили и могут быть изменены с авторизованного мобильного устройства пользователя. Таким образом, при использовании модуля HF-LPB100 удалось построить универсальное решение, которое в дальнейшем не потребует изменения непосредственно программной части прошивки модуля для добавления новых типов ворот, шлагбаумов, ролл-ставен и прочих механизмов. Данные о новых поддерживаемых устройствах будут обновляться в пользовательском приложении смартфона, которое легко обновляется через Интернет, и при конфигурации модуля записываются в память в виде изменяемого профиля.



Литература

1. www.hi-flying.com
2. <http://ptelectronics.ru/product/wi-fi-modul-high-flying-lbp100/>
3. Применение HF-LPB100 для удаленного управления зарядно-разрядным устройством // Вестник электроники. 2015. № 1 (51).