Wifi Shield V2.0

Wi-Fi интерфейс для Arduino на базе модуля RN171 https://www.chipdip.ru/product/wifi-shield-v2.0-seeed-studio



Wifi Shield V2.0 – представляет собой плату расширения форм-фактора Arduino, которая обеспечивает беспроводное соединение по стандарту 802.11 b/g. Плата построена на базе модуля RN171 поддерживающего TCP, UDP, FTP и HTTP протоколы, которые отвечают потребностям большинства беспроводных и IoT («интернета вещей») сетевых проектов. Эта плата расширения оснащена модулем TCP/IP RN171, позволяющим вашему Arduino/ Seeeduino подключаться к беспроводным сетям 802.11b/g.

По умолчанию протокол связи с Arduino - UART/Serial, и вы можете выбрать перемычками, какие цифровые контакты (D0-D7) использовать для RX и TX. Плата также имеет два встроенных разъема Grove для I2C и Serial, для подключения устройств с Grove - интерфейсом.

Встроенная антенна позволяет плате покрывать более широкий диапазон и передавать более сильные сигналы. Модуль RN171 поддерживает протоколы связи TCP, UDP, FTP и HTTP для удовлетворения потребностей большинства беспроводных проектов, такие как, интеллектуальные домашние сети, управление роботами, персональные метеорологические станции.

Характеристики:

- RN-171: Беспроводной LAN модуль WIFLY GSX 802.11 b/g;
- 2 x Разъема Grove;
- Рабочее напряжение: 3.3 5.5 B;
- Потребляемый ток: 25 400 мА;
- Мощность передачи: 0-10 дБм;
- Частота: 2402 2480 МГц;
- Интерфейс: UART
- Скорость сети: 1-11 Мб для 802.11b, 6-54 Мб для 802.11g;
- Аутентификация Wi-Fi: WEP-128, WPA-PSK (TKIP), WPA2-PSK (AES);

- Встроенные протоколы: DHCP клиент, DNS клиент, ARP, ICMP пинг, FTP, TELNET, HTTP, UDP, TCP.

Совместимость

Обратите внимание, что «Not recomended» означает, что у модуля может быть возможность работать с платформой, однако требуется дополнительная доработка, например, подключение дополнительных проводов или переписывание кода. Если вы заинтересованы в том, чтобы внести доработки, свяжитесь с <u>techsupport@seeed.cc</u> для получения инструкций.

	Arduino Uno Seeeduino v4.2	Arduino Mega Seeeduino Mega	Arduino Leonardo Seeeduino Lite	Arduino Due 3.3v	Linkit One
Wifi Shield V2	Yes	Not recommended	Yes	Not recommended	No need

Обзор оборудования



1.Соединения последовательного периферийного интерфейса (SPI) (MOSI, SCK, MISO): эти контакты не подключены ни к одному из контактов Arduino, они независимы, а выход / вход логического уровня - 3,3 В. Они могут использоваться для связи с Arduino через SPI, но необходим логический преобразователь 3,3В между этими контактами и Arduino. Скорость передачи данных в режиме SPI может достигать 2 Мбит / сек.

RES_Wifi: Shield Wifi имеет встроенную кнопку «Rest» для модуля RN-171, вы также можете сбросить RN-171 через программное обеспечение, отправив команду сброса. Кроме того, если вы хотите подключить этот вывод к цифровому 6-контактному разъему Arduino, просто припаяйте блок, обозначенный «P5» на экране.

2. RN171: беспроводной модуль с низким энергопотреблением с встроенным стеком TCP/IP.

3. Антенна: разъем І.РЕХ.

4. Зона доступа RN171: модуль RN171 имеет свои собственные аналоговые входы и выходы GPIO, к которым шилд обеспечивает доступ через эту зону. Выводы GPIO (IO3, IO7, IO8 и IO9) являются толерантными на 3,3В, в то время как аналоговые входные контакты (S_0 и S_1) могут считывать 0-400 мВ (не превышать 1,2 В). RN171 может быть сконфигурирован для использования этих контактов с помощью программного обеспечения или может быть подключен к другим контактам для использования других функций RN171, таких как режим adhoc. Напряжение VCC зависит от мощности питания Shield WiFi.

5. UART / Serial область выбора: перемычки, позволяющие выбирать, какие выходы RX и TX вы хотите использовать для связи с Arduino.

6. Разъемы Grove: Аналоговый I2C Grove (при использовании Arduino UNO или Seeeduino) для контактов A4 и A5 и Digital Serial Grove для D8 и D9. Напряжение VCC зависит от питания платы.

Совместимые контакты

WiFi Shield использует любые два цифровых вывода по вашему выбору между D0 и D7 для связи с модулем WiFi RN171, однако имейте в виду, что D0 и D1 используются Arduino для программирования и последовательной связи, и использование их может помешать этим двум функции.

В примерах кодов используются D2 и D3 как RX и TX для экрана. В этом случае перемычки должны быть подключены, как показано ниже:



D2, выбранный для WIFI_TX, D3, выбранный для WIFI_RX

Модуль RN171 WiFi

RN-171 представляет собой автономный полный модуль беспроводной сети TCP / IP. Благодаря небольшому форм-фактору и чрезвычайно низкому энергопотреблению, RN-171 идеально подходит для мобильных беспроводных приложений. Он включает в себя радиостанцию 2,4 ГГц, 32-разрядный SPARC-процессор, стек TCP / IP, часы реального времени, крипто-ускоритель, блок управления и аналоговый сенсор-интерфейс.

В простейшей конфигурации для создания беспроводного соединения Wi-Fi требуется только четыре соединения (PWR, TX, RX и GND). Кроме того, аналоговые входы сенсора RN171 могут использоваться в качестве аналоговых входных контактов, их номинал 0-400 мВ (не превышает 1,2 В постоянного тока).

Питание: Рабочее напряжение модуля RN-171 обычно составляет 3,3В постоянного тока, поэтому на Shield WiFi предусмотрен преобразователь напряжения и логического уровня. Регулятор LD1117 на плате преобразует напряжение в 3,3В постоянного тока, который обеспечивает питанием модуль RN171. На плате присутствует схема автоматического определения напряжения питания, таким образом RN-171 может питаться от обоих контактов 3V3 и 5V.



Схема сопряжения модуля RN171 с Arduino

GPIO_6: вывод GPIO6 модуля WiFi RN171 по умолчанию подключен только к светодиоду, обозначенному D5 на шилде WiFi. Этот светодиод используется для отображения состояния соединения точки доступа (AP). Если, однако, вы хотите подключить GPIO6 к цифровому выводу 5 Arduino, просто припаяйте блок, обозначенный «P6» на WiFi Shield.

Метка	Описание	Статус	Подключение
			оборудования
D5	Зеленый светодиод. Указывает статус ассоциации.	OFF: означает, что модуль не связан с сетью. Solid ON: указывает, что он связан, и доступ в Интернет в порядке.	Подключен к GPI06 модуля RN171
D1	Красный светодиод. Указывает состояние соединения ТСР / IP.	Solid ON: подключен через TCP. Быстрое переключение(2 раза в секунду): IP-адрес или модуль не находятся в командном режиме. Медленное переключение (один раз / секунду): IP-адрес в порядке.	Подключен к GPI04 модуля RN171
RST	Красный светодиод. Статус сброса модуля WiFi.	Solid ON: нажата кнопка сброса (WIFLRST).	Подключено к сбросу модуля RN171.

PVVR	Зеленый светодиод.	Solid ON: питание модуля / шилда.	Подключается к выходу
	Указывает		3.3V регулятора
	состояние		напряжения LD1117.
	включения модуля		
	WiFi.		

Библиотека WiFi

Настройка

- Загрузите библиотеку в виде zip-файла со страницы <u>https://github.com/Seeed-Studio/WiFi_Shield</u>.
- Распакуйте загруженный файл в папку ... / arduino / libraries /.
- Переименуйте распакованную папку «WifiShield»
- Запустите Arduino IDE (или перезапустите, если он открыт).

Функции

join()

Description:

Используется для присоединения к точке доступа WiFi

Syntax:

join (const char * ssid, const char * phrase, int auth)

Parameters:

ssid: имя точки доступа, к которой вы хотите подключить шилд

phrase: пароль / фраза точки доступа, к которой вы хотите подключить шилд

auth: тип аутентификации точки доступа, к которой вы хотите подключить шилд

Может быть одной из следующих констант:

WIFLY_AUTH_OPEN

WIFLY_AUTH_WEP

WIFLY_AUTH_WPA1

WIFLY_AUTH_WPA1_2

WIFLY_AUTH_WPA2_PSK

WIFLY_AUTH_ADHOC

Returns:

boolean: true, если соединение с точкой доступа было успешным, в противном

случае-false.

Пример:

```
#include <SoftwareSerial.h>
#include "WiFly.h"
SoftwareSerial uart(2, 3); // create a serial connection to the WiFi shield
TX and RX pins.
WiFly wifly(&uart); // create a WiFly library object using the serial
connection to the WiFi shield we created above.
void setup()
{
    uart.begin(9600); // start the serial connection to the shield
    Serial.begin(9600); // start the Arduino serial monitor window connection
    wifly.reset(); // reset the shield
    while(wifly.join("mySSID", "mySSIDpassword", WIFLY_AUTH_WPA2_PSK) == false)
    {
        Serial.println("Failed to connect to accesspoint. Will try again.");
     }
     Serial.println("Connected to access point!");
}
```

Совет

Примеры основаны на Arduino UNO, и мы берем D2 / D3 в качестве контактов SoftwareSerial. Если вы используете Arduino Mega, D2 больше не доступен. Более подробную информацию смотрите в разделе Arduino Software Serial (https://www.arduino.cc/en/Tutorial/SoftwareSerialExample)



Что касается кода, вам также нужно внести некоторые изменения:

SoftwareSerial uart(10, 3); // create a serial connection to the WiFi shield TX and RX pins.

receive()

Description:

Может использоваться для чтения данных с экрана, альтернатива для функции Arduino **read**()

Syntax:

receive(uint8_t *buf, int len, int timeout)

Parameters:

buf: буферный массив, в котором хранятся байты, считанные с экрана. len: длина / размер буферной матрицы timeout: значение тайм-аута, чтобы знать, когда перестать пытаться считывать.

Returns:

int: количество байтов, считанных с экрана.

Пример:

```
char c;
while (wifly.receive((uint8_t *)&c, 1, 300) > 0) {
        Serial.print((char)c);
}
```

See File->Examples->WiFi_Shield->wifly_test sketch for a complete example.

SendCommand ()

Description:

Некоторые наши функции (такие как, **join** (), **reboot** (), **save** ()) действуют как обертки для текстовых команд, перечисленных в руководстве пользователя модуля RN171. Функция **sendCommand** () позволяет вам придумать свою собственную функцию, если наша не соответствует вашим потребностям.

Syntax:

sendCommand(const char *cmd, const char *ack, int timeout)

Parameters:

cmd: любая команда из руководства пользователя RN-171. ack: ожидаемая строка возврата из команды

timeout: время, разрешенное до того, как будет рассмотрен вывод о плохом запросе / ответе

Returns:

boolean: true, если Shield WiFi ответил строкой ack, иначе false.

Пример:



See File->Examples->WiFi_Shield->wifly_test sketch for a complete example.

Примеры и применение Wi-Fi Shield

Пример 1: Отправка команд на Wi-Fi Shield и получение ответа через окно последовательного монитора Arduino

Модуль RN-171 для WiFi Shield настраивается путем отправки команд, найденных в его таблице данных

(https://raw.githubusercontent.com/SeeedDocument/Wifi_Shield_V2.0/master/res/WiFly-RN-

<u>UM.pdf</u>). Вы можете написать скетч, чтобы отправлять команды автоматически, это отличный пример, который мы рекомендуем вам пройти, потому что он научит вас точно, как работает WiFi Shield и RN-171.

Шаг 1: Конфигурация перемычек Wi-Fi Shield

Поместите перемычки WiFi Shield так, чтобы для WIFI_TX был выбран цифровой контакт 2 (D2), а для WIFI_RX выбран цифровой контакт 3 (D3), как показано на фотографии ниже. Это выводы, которые мы будем использовать для отправки и получения информации от RN-171.



Выводы D2 для TX и D3 для RX

Шаг 2: Программное обеспечение / Код

В приведенном ниже скетче мы создали объект UART, позволяющий нам отправлять и получать данные с RN-171 / WiFi Shield. Затем мы используем этот объект совместно с библиотекой WiFly для передачи данных на экран. Серийный объект Arduino используется для печати данных, которые мы получаем от экрана, и для получения команд, которые мы хотим отправить на экран через объект WiFly / UART.



Шаг 3: Ввод командного режима

Модуль WiFly RN-171 в WiFi Shield может работать в двух режимах: данные и команда. В режиме данных экран может принимать и инициировать соединения. В командном режиме мы можем настроить модуль, используя команды, перечисленные в его таблице данных.

Чтобы войти в командный режим, выполните следующие действия:

- Откройте последовательный монитор Arduino.
- Установите последовательный монитор на "No line ending", скорость передачи до 9600.
- Введите «\$\$\$» в последовательный монитор Arduino и нажмите enter.
- Модуль будет отвечать буквами «СМD», указывая, что он вступил в командный режим.

Давайте продолжим и проверим некоторые команды, сделайте следующее:

- 1. В окне монитора Arduino Serial выберите «Carriage return» и скорость передачи в 9600.
- 2. Теперь введите каждую из приведенных ниже команд в последовательный монитор Arduino и нажмите enter.
- 3. Модуль выдает ответ, как описано в таблице, для каждой команды.

Команда	Описание
scan	Эта команда выполняет активное сканирование точек доступа на всех 13 каналах. Когда вы используете эту команду, модуль возвращает МАС-адрес, уровень сигнала, имя SSID и режим безопасности точек доступа, которые он находит.
get ip	Эта команда отображает настройки IP-адреса и номера порта

Полный список команд конфигурации смотрите в справочном руководстве RN-171, начиная со страницы 11 (<u>https://raw.githubusercontent.com/SeeedDocument/Wifi_Shield_V2.0/master/res/WiFly-RN-</u> UM.pdf).

Пример 2: подключение к точке доступа / интернет-маршрутизатору

В этом примере мы покажем вам, как подключить шилд WiFi к точке доступа (ваш интернет-маршрутизатор) с помощью и без ввода необходимых команд:

Выполните следующие действия:

1. Загрузите код в примере 1 на плату Arduino

2. Войдите в командный режим:

- Установите последовательный монитор на "No line ending", скорость передачи до 9600.
- Введите «\$\$\$» в последовательный монитор Arduino и нажмите enter.
- 3. Установите последовательный монитор на «Carriage return».

4. Сканирование доступных точек доступа:

Введите scan и нажмите enter. Последовательное окно монитора Arduino выведет список значений, разделенных запятой, для каждой точки доступа, которую нашел Shield WiFi. Слева направо третье значение - режим безопасности, последним значением является SSID. В этом примере показан режим безопасности 4 с именем SSID MySSID: 01,01, - 88,04,1104,1c,00,45:56:78⁹93:1f,MySSID

5. Из списка найденных точек доступа найдите тот, который соответствует вашему интернет-маршрутизатору, и обратите внимание на режим безопасности и SSID, поскольку нам понадобятся эти два значения для подключения к нему.

6. Установите защитный режим в шилде:

٠

Тип set wlan auth m. Замените m номером режима безопасности (в этом примере, который будет 4) точки доступа, к которой вы хотите подключиться.

• Режимы безопасности, поддерживаемые экраном WiFi, приведены на рисунке 1 ниже.

7. Установите фразу точки доступа

Введите set wlan phrase myPhrase. Замените myPhrase паролем / ключом безопасности вашей точки доступа.

Примечание

Если тип безопасности вашей точки доступа является ключом использования WEP вместо Phrase в приведенной выше команде.

Фраза точки доступа (интернет-маршрутизатор) - это пароль, который вы используете для подключения к нему с вашего ПК. В Windows вы можете найти его, как показано на изображении ниже:

R	Viidentified network No network access Wireless Network Connection	ATT409 Wireless Network Properties
	Open Network and Sharing Center	
	10:27 PM 1/10/2015	🛞 🔤 🔹 🕨 📶 10:27 PM 1/10/2015

Как найти ключ безопасности / пароль сети

8. Присоединиться к точке доступа

- Теперь, когда мы установили тип безопасности и фразу точки доступа, мы можем подключиться к ней.
- Введите соединение с MySSID. Замените MySSID своим широковещательным именем вашей точки доступа.
- Слово «Associated!» будет отображаться в окне последовательного монитора Arduino в случае успеха.

Описание команд, введенных в приведенные выше шаги, приведено в таблице ниже. Более подробное описание каждой команды можно найти в руководстве пользователя RN171.

номер	Команды	Описание
1	Scan	Эта команда выполняет активное сканирование точек доступа на всех 13 каналах. Когда вы используете эту команду, модуль возвращает МАС-адрес, уровень сигнала, имя SSID и режим безопасности точек доступа, которые он находит.
2	set wlan auth 4	Найдите значение, соответствующее протоколу безопасности вашей точки доступа. Затем укажите WiFly, какой протокол безопасности следует использовать, это число, показанное на рисунке 1, соответствует протоколу безопасности точки доступа. Здесь мы выбираем «4».
3	set wlan phrase seeed- mkt	Укажите Wi-Fi-Shield свою кодовую фразу.
4	join SEEED-MKT	Укажите Wi-Fi-Shield, чтобы присоединиться, «SEEED- МКТ» - это имя точки доступа, которую мы выбираем для подключения. После отправки команды модуль должен теперь подключиться и распечатать информацию о соединении. (Если соединение не выполнено, попробуйте отправить команду еще раз, пока она не будет работать)

Значение	Режим проверки подлинности
0	Открыто (по умолчанию)
1	WEP-128
2	WPA1
3	Смешанные WPA1 и WPA2-PSK
4	WPA2-PSK
5	Не используется
6	AD hoc режим (присоединяется к любой сети ad hoc)
8	WPE-64
	Рисунок 1

Подключение с использованием наших WiFi-библиотек

Теперь, когда вы знаете, как подключиться к точке доступа, набрав каждую команду, пришло время использовать библиотеки и примеры, которые мы предоставляем.

Чтобы увидеть код, необходимый для подключения к точке доступа, перейдите в «File -> Examples -> Wifi_Shield -> wifi_test». Измените код, чтобы использовать свой собственный SSID (имя точки доступа) и КЕҮ (пароль вашей точки доступа), затем загрузите эскиз в свою среду разработки Arduino.

#define SSID " SEEED-MKT "
#define KEY " seeed-mkt "

Со скетчем, загруженным на плату Arduino, откройте окно последовательного монитора. Если экран был успешно присоединен к точке доступа, будет отображаться сообщение «ОК» вместе с информацией о соединении, полученной в результате команды "get everything". Если экран не смог присоединиться к точке доступа, появится сообщение «Failed».

Настройка шилда для автоматического подключения к точке доступа

Шилд можно настроить для автоматического подключения к точке доступа при включении питания, вам нужно сделать это только один раз:

- Отправьте команду "set wlan ssid mySSID", заменив mySSID на ваш SSID
- Отправьте команду "set wlan join 1".
- Отправьте команду "save".

Теперь шилд автоматически подключится к точке доступа при включении питания.

Описание того, что делает каждая команда, можно найти в таблице данных RN-171 и в таблице ниже.

Номер	Команда	Описание
1	set wlan ssid	"" - это имя точки доступа, к которой вы хотите подключиться автоматически
2	set wlan join 1	Это говорит о том, что модуль пытается автоматически подключиться к SSID, хранящемуся в памяти.
3	save	Сохранение и хранение этих настроек в конфигурационном файле Wifi

Установка статического IP-адреса

Чтобы экран получил статический IP-адрес от точки доступа, после подключения к точке доступа, отправьте следующие команды:

Номер	Команда	Описание
1	set ip dhcp 0	Возврат DHCP.
2	set ip address	Задайте требуемый IP-адрес.

Пример 3: Взаимодействие с сетью ()

Этот пример покажет вам, как устройство, такое как ваш компьютер и / или телефон, может взаимодействовать с Shield WiFi.

Следуй этим шагам:

- 1. Настройте модуль с шагом 1-7 в примере 2 раздела «Подключение по командам ввода»
- 2. Установите прослушивание IP-порта на "80", отправив команды "set ip local 80"
- 3. Подключите свой шилд к точке доступа, как показано в шаге 8 раздела «Пример 2».
- 4. Сохраните эти настройки, отправив команду "save"
- 5. Получите IP-адрес вашего экрана с помощью команды "get ip". IP-адрес и порт будут отображаться справа от "IP=" в ответе (например, IP = 192.168.0.10: 80)
- 6. Откройте веб-браузер и введите IP-адрес своего шилда в строке URL вашего веббраузера и нажмите Enter, чтобы просмотреть его.
- 7. В окне последовательного монитора Arduino будет отображаться HTTP-ответ, подобный приведенному ниже. Это информация, которую ваш браузер отправил на шилд для запроса данных.

*OPEN*GET / HTTP/1.1
Host: 192.168.0.10
Connection: keep-alive
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/39.0.2171.95 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8

Браузер теперь ждет данных, модуль Wifi может отправлять значения датчика, обслуживать веб-страницы или любые другие данные прямо обратно в браузер! В этом случае браузер ждет веб-страницы. Если модуль Wifi отвечает на HTML-форматированную страницу, браузер отобразит его.

Пример 4. Использование WiFi-Shield в качестве веб-сервера (обслуживание вебстраниц с экрана)

Как вы видели в примере 3, интернет-браузер может подключаться к Shield WiFi. Как только соединение будет установлено (когда браузер отправит свой HTTP-запрос), Shield WiFi может затем отправить обратно HTML-код для просмотра браузером в качестве вебстраницы. В этом примере вы узнаете, что нужно, чтобы экран отвечал на веб-браузер.

Шаг первый: Код Arduino

Загрузите следующий код на плату Arduino, заменив «myssid» и «mypassword» значениями вашей точки доступа соответственно:

#include <SoftwareSerial.h>
#include "WiFly.h"
#define SSID "myssid"
#define KEY "mypassword"
// check your access point's security mode, mine was WPA20-PSK
// if yours is different you'll need to change the AUTH constant, see the
file WiFly.h for avalable security codes
#define AUTH WIFLY_AUTH_WPA2_PSK

int flag = 0;

// Pins' connection // Arduino WiFly // 2 <----> TX // 3 <----> RX

SoftwareSerial wiflyUart(2, 3); // create a WiFi shield serial object WiFly wifly(&wiflyUart); // pass the wifi siheld serial object to the WiFly class

```
void setup()
{
    wiflyUart.begin(9600); // start wifi shield uart port
    Serial.begin(9600); // start the arduino serial port
    Serial.println("----- WIFLY Webserver ------");
    // wait for initilization of wifly
    delay(1000);
```

wifly.reset(); // reset the shield delay(1000); //set WiFly params

wifly.sendCommand("set ip local 80\r"); // set the local comm port to 80
delay(100);



Шаг второй: получите ІР-адрес шилда

Откройте окно последовательного монитора и дождитесь появления cooбщения "Web server ready". Последовательный монитор также отобразит IP-адрес экрана WiFi:

© COM19	
	Send
WIFLY Webserver	
Join AAABBB	
OK	1
get ip	
IF=UP	
DHCP=ON	
IP=192.168.0.10:80	
NM=255.255.255.0	
GW=192.168.0.1	
HOST=0.0.0.0:2000	
PROTO=HTTP, SMTP,	
MTU=1524	
FLAGS=0x7	
TCPMODE=0x0	
BACKUP=0.0.0.0	
<4.00> Web server ready	
Autoscroll Carriage return 👻	9600 baud 👻

Вывод последовательной связи Arduino. IP-адрес экрана подсвечивается.

Шаг третий: посещение веб-страницы

Теперь посетите этот IP-адрес в своем веб-браузере. Веб-страница ниже должна отображаться, она содержит ссылку на Yahoo! и Google и кнопка, которая ничего не делает (пока):



Простая веб-страница с двумя ссылками и одной кнопкой, которая подается с шилда WiFi.

Когда веб-страница будет посещена, окно последовательного монитора также отобразит "New Browser Request!", как показано ниже:



Окно последовательной связи Arduino, показывающее, что оно обнаружило новое соединение / запрос браузера.

Примечание

В случае некоторых браузеров, таких как Google Chrome, даже ввод URL-адреса в строке отправляет запрос на веб-страницу, это связано с тем, что эти браузеры пытаются получить заголовок веб-страницы для удобства пользователя даже до того как, кто-то посетит веб-страницу.

Пример 5: Управление цифровыми выводами Arduino с веб-страницы (переключение светодиодов с веб-страницы)

В этом примере мы создадим веб-страницу с тремя кнопками для управления тремя различными цифровыми выводами в Arduino.

Шаг 1: Аппаратное обеспечение

Подключите три светодиода и резистор к цифровым контактам 11, 12 и 13, как показано на схеме ниже:



Три светодиода и 1k резисторы подключены к контактам 11, 12 и 13.

Шаг 2: Скетч Arduino

Загрузите следующий код на свою плату Arduino, но замените «mySSID» и «myPassword» на имя SSID вашего доступа и пароль:

#include <SoftwareSerial.h> #include "WiFly.h" #define SSID "mySSID" #define KEY "myPassword" // check your access point's security mode, mine was WPA20-PSK // if yours is different you'll need to change the AUTH constant, see the file WiFly.h for avalable security codes #define AUTH WIFLY AUTH WPA2 PSK int flag = 0; // Pins' connection // Arduino WiFly // 2 <---> ΤX 3 <---> RX SoftwareSerial wiflyUart(2, 3); // create a WiFi shield serial object WiFly wifly(&wiflyUart); // pass the wifi siheld serial object to the WiFly **class** char ip[16]; void setup() { pinMode(11,OUTPUT); digitalWrite(11,LOW); pinMode(12,OUTPUT); digitalWrite(12,LOW);

pinMode(13,OUTPUT); digitalWrite(13,LOW);

```
wiflyUart.begin(9600); // start wifi shield uart port
   Serial.begin(9600); // start the arduino serial port
   Serial.println("----- WIFLY Webserver -----");
   // wait for initilization of wifly
   delay(1000);
   wifly.reset(); // reset the shield
   delay(1000);
   //set WiFly params
   wifly.sendCommand("set ip local 80\r"); // set the local comm port to 80
   delay(100);
  wifly.sendCommand("set comm remote 0\r"); // do not send a default string
when a connection opens
  delay(100);
   wifly.sendCommand("set comm open *OPEN*\r"); // set the string that the
wifi shield will output when a connection is opened
  delay(100);
   Serial.println("Join " SSID );
   if (wifly.join(SSID, KEY, AUTH)) {
       Serial.println("OK");
   } else {
       Serial.println("Failed");
   }
delav(5000);
wifly.sendCommand("get ip\r");
   wiflyUart.setTimeout(500);
   if(!wiflyUart.find("IP="))
   {
       Serial.println("can not get ip");
       while(1);;
   }else
   {
       Serial.print("IP:");
   }
   char c;
   int index = 0;
   while (wifly.receive((uint8 t *)&c, 1, 300) > 0) { // print the response
from the get ip command
       if(c == ':')
       {
           ip[index] = 0;
           break;
       }
       ip[index++] = c;
       Serial.print((char)c);
       ?
   }
   Serial.println();
   while (wifly.receive((uint8 t *)&c, 1, 300) > 0);;
   Serial.println("Web server ready");
```

void loop()





Шаг 3: Окно последовательного монитора

Откройте окно последовательного монитора и дождитесь появления cooбщения "Web server ready". Последовательный монитор также отобразит IP-адрес экрана WiFi:



Вывод последовательной связи Arduino. IP-адрес экрана подсвечивается.

Шаг 4: Посетите веб-страницу

Посетите IP-адрес в веб-браузере. Должна отображаться веб-страница с тремя кнопками, как показано ниже. Нажмите на кнопки для управления светодиодами.



Веб-страница управления светодиодами с WiFi Shield.

Arduino также вернет веб-браузеру состояние «pin», браузер отобразит это в окне предупреждения.

	The page at 192.168.0.10 says:		×
	Pin 12 is ON		
1		ОК	

Диалоговое окно оповещения, отображающее состояние Pin12, Строка Pin12 включена, была отправлена из Arduino.

Окно последовательного монитора также покажет, когда браузер отправляет запрос на посещение веб-страницы или управление светодиодными выводами.

💿 СОМ19	
	Send
Join AAABBB	
OK	
get ip	
IF=UP	
DHCP=ON	
IP=192.168.0.10:80	
NM=255.255.255.0	
GW=192.168.0.1	
HOST=0.0.0.0:2000	
PROTO=HTTP, SMTP,	
MTU=1524	
FLAGS=0x7	=
TCPMODE=0x0	
BACKUP=0.0.0.0	
<4.00> Web server ready	
New Browser Request!	
LED Control	
Data sent to browser	
New Browser Request!	
LED Control	
Data sent to browser	
	-
V Autoscroll	Carriage return 🖌 9600 baud 🔶

Вывод Arduino последовательной связи, когда HTTP-запрос отправляется на экран.

Пример 6: WiFi Shield и приложение для Android

	Pin 2	
	Pin 3	-
	Pin 5	c
P Address:		÷
e.g. 192.168.0.10		
ort Number:		
e.g. 80		

Приложение Android, которое вы можете использовать для управления контактами Arduino через Wi-Fi или Ethernet-Shield

Программного обеспечения

Загрузите проект / источник Android Studio по этой ссылке:

https://raw.githubusercontent.com/SeeedDocument/Wifi_Shield_V2.0/master/res/WiFiShieldLEDControl.zip

Пример 7: Отправка данных и получение данных с внешнего сервера

Модуль RN-171 в WiFi Shield имеет возможность выступать в роли HTML-клиента (текстовый веб-браузер по существу), это означает, что мы можем использовать экран для отправки и получения данных с веб-сервера. В этом примере вы научитесь использовать экран с веб-интерфейсом прикладного программирования (API), который отображает данные о погоде в городе (т. е. температуру, влажность и т. д.).

Имя API, которое мы будем использовать, это OpenWeatherMap (<u>https://openweathermap.org/api</u>), когда вы отправляете название города и страны на этот сайт, оно возвращает строку JSON с информацией о погоде. Например, если вы хотите отобразить погоду для Лондона, Великобритании, пожалуйста, обратитесь к этой ссылке <u>http://openweathermap.org/appid</u>. Начиная с 9 октября 2015 года, веб-сайт требует, чтобы пользователи подписывались на ключ API перед посещением API. После того, как вы получите ключ API, вы сможете посетить следующий URL-адрес

http://api.openweathermap.org/data/2.5/weather?q=London,uk, который вернет строку JSON, как показано ниже, данные о погоде и другая информация.

"coord":{"lon":-0.13,"lat":51.51},

```
"sys":{"type":3,"id":60992,"message":0.0079,"country":"GB","sunrise":14213950
87,"sunset":1421425352},
    "weather":[{"id":802,"main":"Clouds","description":"scattered
clouds","icon":"03n"}],
    "base":"cmc stations",
    "main":{
        "temp":277.25,"humidity":79,"pressure":998.4,
        "temp_min":277.25,"temp_max":277.25
    },
    "wind":{
        "speed":2,"gust":5,"deg":180},
        "rain":{"3h":0},"clouds":{"all":32},
        "dt":1421372140,"id":2643743,"name":"London","cod":200
}
```

Шаг 1: URL-адрес

Давайте перейдем и возьмем строку JSON погоды для Сан-Франциско, США. URL-адрес, который должен посетить наш Shield WiFi, следующий (вы можете проверить его в своем веб-браузере):

http://api.openweathermap.org/data/2.5/weather?q=San%20Francisco,US

Шаг 2: Код Arduino

В разделе 13 руководства WiFly

(https://raw.githubusercontent.com/SeeedDocument/Wifi Shield V2.0/master/res/WiFly-RN-UM.pdf) приведены различные способы подключения к веб-серверу, но во всех случаях нам нужно указать имя сервера (или IP-адрес, если сервер не имеет имени домена), а затем данные, которые желаю отправить.

Команды, которые необходимо отправить на Shield WiFi для получения строки JSON с сервера OpenWeatherMap, следующие:

```
set ip proto 18 //enable html client
set dns name api.openweathermap.org //name of your webserver
set ip address 0 // so WiFly will use DNS
set ip remote 80 // standard webserver port
set com remote 0 // turn off the REMOTE string so it does not interfere with
the post
open // to open the connection
GET /data/2.5/weather?g=San%20Francisco,US \n\n // to send the data
```

Это код arduino, который будет отправлять команды:

#include <SoftwareSerial.h> #include "WiFly.h" #define SSID "mySSID" #define KEY "myPassword"

// check your access point's security mode, mine was WPA20-PSK
// if yours is different you'll need to change the AUTH constant, see the
file WiFly.h for avalable security codes
#define AUTH WIFLY AUTH WPA2 PSK

// Pins' connection // Arduino WiFly // 2 <---> TX // 3 <---> RX

SoftwareSerial wiflyUart(2, 3); // create a WiFi shield serial object WiFly wifly(&wiflyUart); // pass the wifi siheld serial object to the WiFly <mark>class</mark>

void setup() {

wiflyUart.begin(9600); // start wifi shield uart port Serial.begin(9600); // start the arduino serial port Serial.println("------ OpenWeatherMap API ------");

```
// wait for initilization of wifly
delay(3000);
wifly.reset(); // reset the shield
Serial.println("Join " SSID );
if (wifly.join(SSID, KEY, AUTH)) {
    Serial.println("OK");
} else {
    Serial.println("Failed");
}
```

delay(5000);

wifly.sendCommand("set ip proto 18\r"); //enable html client
delay(100);

wifly.sendCommand("set dns name api.openweathermap.org\r"); // name of the webserver we want to connect to delay(100);

wifly.sendCommand("set ip address 0\r"); // so WiFly will use DNS
delay(100);

wifly.sendCommand("set ip remote 80\r"); /// standard webserver port
delay(100);

```
wifly.sendCommand("set com remote 0\r"); // turn off the REMOTE string so
it does not interfere with the post
delay(100);
wifly.sendCommand("open\r"); // open connection
delay(100);
wiflyUart.print("GET /data/2.5/weather?q=San%20Francisco,US \n\n");
delay(1000);
}
void loop()
{
   //As soon as the data received from the Internet ,output the data
through the UART Port .
   while (wifly.available())
   {
      Serial.write(wifly.read());
   }
}
```

Шаг 3: Результат

Откройте окно последовательного монитора, вы сможете увидеть ту же строку JSON, которую вы видели в браузере.

💿 COM19	
Send	ר
OpenWeatherMap API	
Join AAABBB	1
ok	
Open	
<pre>""""""""""""""""""""""""""""""""""""</pre>	"
<	F
☑ Autoscroll Carriage return ↓ 9600 baud ↓	3

JSON показан в окне последовательного монитора Arduino.

Пример 8: Связь ТСР с терминалом

В этом примере мы покажем вам, как отправлять информацию с Shield WiFi в программу ПК-терминала. Мы создадим простую консоль Arduino с меню, которое даст вам возможность увидеть состояние цифрового контакта Arduino и переключить их.

Шаг 1. Загрузите ТСР-терминал

Загрузите и установите RealTerm

(https://sourceforge.net/projects/realterm/files/Realterm/2.0.0.70/Realterm_2.0.0.70_setup.exe/d ownload), служебный терминал, который позволит нам подключиться к Shield WiFi.

Шаг 2: Код Arduino

delay(1000);

Загрузите приведенный ниже код на свою плату Arduino, заменив «mySSID», «myPassword» и код аутентификации с информацией о вашей собственной точке доступа:

```
#include <SoftwareSerial.h>
#include "WiFly.h"
#define SSID "mySSID"
#define KEY "myPassword"
// check your access point's security mode, mine was WPA20-PSK
// if yours is different you'll need to change the AUTH constant, see the
file WiFly.h for avalable security codes
#define AUTH WIFLY AUTH WPA2 PSK
#define FLAG MAIN MENU 1
#define FLAG SUB MENU 2 2
int flag = FLAG MAIN MENU;
// Pins' connection
// Arduino WiFly
// 2 <---> TX
// 3 <---> RX
SoftwareSerial wiflyUart(2, 3); // create a WiFi shield serial object
WiFly wifly(&wiflyUart); // pass the wifi siheld serial object to the WiFly
<mark>class</mark>
void setup()
{
    // define the pins we can control
   pinMode(11,OUTPUT);
   digitalWrite(11,LOW);
    pinMode(12,OUTPUT);
   digitalWrite(12,LOW);
   pinMode(13,OUTPUT);
   digitalWrite(13,LOW);
    pinMode(7,OUTPUT);
   digitalWrite(7,LOW);
 wiflyUart.begin(9600); // start wifi shield uart port
    Serial.begin(9600); // start the arduino serial port
    Serial.println("----- TCP Communication -----");
    // wait for initilization of wifly
   delay(1000);
    wifly.reset(); // reset the shield
```

```
wifly.sendCommand("set ip local 80\r"); // set the local comm port to 80
   delav(100);
wifly.sendCommand("set comm remote 0\r"); // do not send a default string
when a connection opens
  delay(100);
   wifly.sendCommand("set comm open *\r"); // set the string or character
that the wifi shield will output when a connection is opened "*"
 delay(100);
   wifly.sendCommand("set ip protocol 2\r"); // set TCP protocol
   delay(100);
   Serial.println("Join " SSID );
   if (wifly.join(SSID, KEY, AUTH)) {
       Serial.println("OK");
   } else {
       Serial.println("Failed");
   }
delay(5000);
   wifly.sendCommand("get ip\r");
   char c;
   while (wifly.receive((uint8 t *) &c, 1, 300) > 0) { // print the response
from the get ip command
       Serial.print((char)c);
   }
  Serial.println("TCP Ready");
void loop()
{
   if(wifly.available())
    {
       delay(1000); // wait for all the characters to be sent to the WiFi
shield
       char val = wiflyUart.read(); // read the first character
       if(flag == FLAG MAIN MENU)
        {
           switch(val)
            {
               case '*': // search for the new connection string
               printMainMenu();
               break;
               case '1': // the user typed 1, display the pin states
               printPinStates();
               printMainMenu();
               break;
               case '2': // the user typed 2, display the sub menu (option
to select a particular pin)
               printSubMenu2();
               flag = FLAG SUB MENU 2; // flag to enter the sub menu
               break;
               default:
               wiflyUart.print("INVALID SUBMENU\r\n");
               break;
           }
```





Шаг 3: Получите IP-адрес и порт Shield

Откройте окно последовательного монитора Arduino, чтобы получить IP-адрес и номер порта WiFi Shield, выделенный на изображении ниже.



Вывод окна последовательного монитора Arduino из примера, подсвечивается IP-адрес и номер порта.

На изображении выше ІР-адрес и порт будут следующие:

192.168.0.10:80

Шаг 4. Настройка терминала ТСР и подключение к шилду

Откройте RealTerm и на вкладке «Display» введите «30» для «Rows» и выберите опцию «Scrollback»:

📲 RealTerm: Serial Capture Program 2.0.0.70	
· · · · · · · · · · · · · · · · · · ·	
Display Port Capture Pins Send Echo Port 12C 12C-2 In Clear	Freeze
Display As □ Half Duplex Ascii □ newLine mode C Ansi □ Invest C Hextspace] □ Invest C Hextspace] □ Invest C Hextspace] □ Big Endian C unt8 □ Data Erames C unt16 □ Bytes C wint16 □ Single C wint16 □ Single C hext Bytes 2 € C hext □ Rows C hext CSV Terminal Eonth 30 € Image: Scrollback	Status Connec RXD (2 TXD (3) CTS (8) DCD (1) DSR (6) Ring (9) BREAK Error
You can use ActiveX automation to control me! Char Count:0	CPS:
III III	

Окно RealTerm: rows = 30 и опция Scrollback.

На вкладке "Port" программы RealTerm введите IP-адрес и порт вашего экрана, например. 192.168.0.10:80, затем нажмите кнопку "Open", главное меню Arduino в жестком кодировании должно отображаться в терминале.

RealTerm: Serial Capture Program 2.0.0.70		
1. Show digital pin states(NF 2. Toggle a digital pin's state(NF(NF(NF(NF))))	- CgLF	
Hrduino Console Menu: 184 1. Show digital pin states 1845		
Z. loggie a digital pin's stateskeskeskes		
< III		
Display Port Capture Pins Send Echo Port 12C	12C-2 In Clear Freeze	?
Baud 57600	pen Spy Change Boy Control	tec
Parity Data Bits Stop Bits	Receive Xon Char: 17 TXD (3)	
C Odd C 7 bits Hardware Flow Control	Transmit Xoff Char: 19	1
C Mark C 6 bits C None C RTS/CTS C Space C 5 bits C DTR/DSR C RS485-tts	─Winsock i ■ DSR (6) ⊂ Raw Ring (9) ເ⊂ BREAK	
You can use ActiveX automation to control me!	Char Count:694 CPS:0	3
K		۲

Окно RealTerm. Порт имеет IP-адрес и номер порта WiFi. Отображается меню Arduino

На вкладке "Send" выберите один из вариантов из меню «1» или «2», введите его в текстовое поле и нажмите "Send ASCII", чтобы отправить значение.

Например, чтобы переключить контакт 13, введите «2» и нажмите "Send ASCII", затем, когда будет предложено "Enter the pin number you wish you toggle", введите «13» и нажмите "Send ASCII". Arduino должен ответить «Pin Toggled!» и вернитесь в главное меню, теперь введите «1» и нажмите "Send ASCII", чтобы увидеть текущее состояние контактов.

Par RealTerm: Serial Capture Program 2.0.0.70
Enter the pin number you wish to toggle: UNF You want to toggle pin 13? OKGMF Pin Toggled (MaffAffAfA 2. Toggle a digital pin's stateGMFfAffAffAffAffAffAffAffAffAffA Arduino Console Menu: GMF 1. Show digital pin statesGMF 2. Toggle a digital pin's stateGMFfAffAff
Pin 7 is OFF04F Pin 11 is OFF04F Pin 12 is OFF04F Pin 13 is OFF04F Off Off Arduino Console Menu: 04F 1. Show digital pin's state04F 2. Toggle a digital pin's state04F (04F
INVALID SUBMENU (st.f
Display Port Capture Pins Send Echo Port I2C I2C-2 In Clear Freeze ?
13 ▼ Send Numbers Send ASCII E0 Status 13 ▼ Send Numbers Send ASCII FXD (2) 1 ▼ Send Numbers Send ASCII TXD (3) 0 ^C LF Repeats 1 € Literal Strip Spaces DCTS (8) 0 D C Literal Strip Spaces DCSB (6)
C:\temp\capture.txt Send Eile X Stop De Pring (9)
You can use ActiveX automation to control me! Char Count:1892 CPS:0

Окно RealTerm. Состояние контакта 13 было изменено с OFF на ON, как показано в желтом тексте.

Пример 9: WiFi Shield и Relay Shield

Теперь, когда вы знаете, как отправлять и получать информацию с WiFi Shield, вы можете видеть, как легко было бы управлять любым устройством через Интернет.

Если вы хотите управлять устройствами высокой мощности, такими как настольная лампа, мотор или водяной насос через веб-страницу или приложение для телефона, мы рекомендуем использовать Relay Shield (<u>https://www.chipdip.ru/product/relay-shield-v3.0?from=suggest_product</u>)

Пример 10: Adhoc режим

Чтобы использовать экран в режиме Adhoc в качестве точки доступа, просто подключите вывод IO9 от шилда к выходу 3.3V в Arduino, как показано ниже, и сбросьте шилд, если он включен.



Подключение экрана требуется для режима adhoc. Контакт IO9 экрана подключен к 3.3V.

Чтобы получить SSID экрана, загрузите код, в примере 1, в Arduino и откройте последовательный монитор, экран укажет свой SSID, как показано в примере ниже, где WiFly-EZX-1b является SSID.

AP mode as WiFly-EZX-1b on chan 1

Теперь вы должны подключиться к WiFi Shield в качестве точки доступа, например, SSID должен быть виден в списке доступных сетей WiFi вашего компьютера.

Чтобы узнать больше о режиме adhoc, ознакомьтесь с разделом 16 «Руководство пользователя WiFly RN» «Adhoc Networking Mode» (https://raw.githubusercontent.com/SeeedDocument/Wifi_Shield_V2.0/master/res/WiFly-RN-UM.pdf)

Техническая поддержка

Любые технические вопросы можно направить на форум производителя <u>https://forum.seeedstudio.com/</u>.