

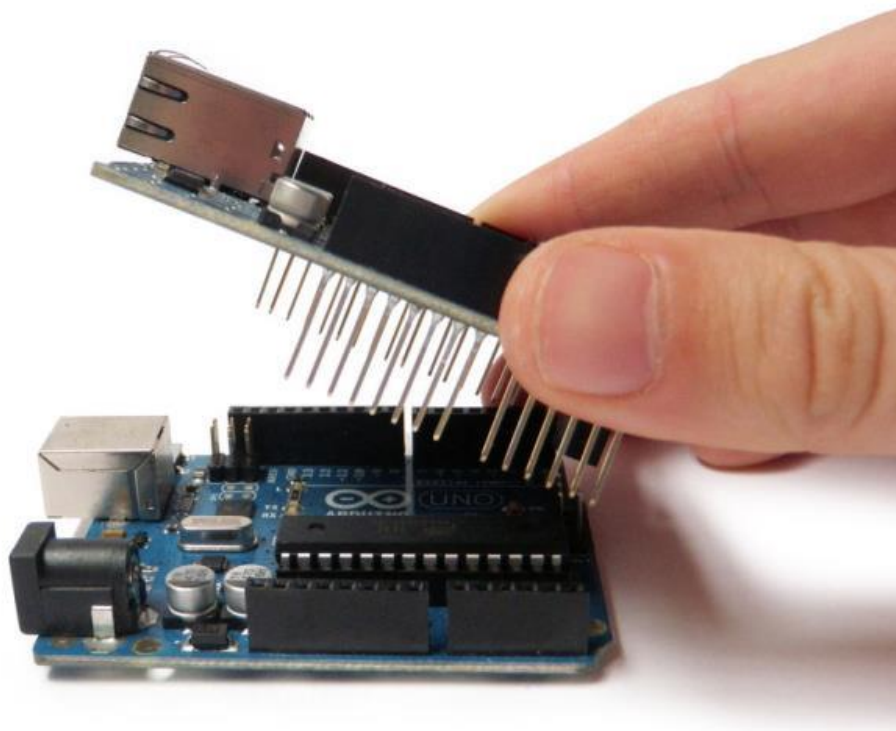
Ethernet Shield и Arduino – основы

Ethernet Shield дает вам возможность подключать Arduino к интернету. Этот шилд расширяет возможности Arduino и позволяет отсылать и принимать данные из любой точки мира, где есть интернет. Например, можно удаленно управлять вашим роботом с веб-сайта или реализовать какой-то сигнал, когда приходит новое сообщение в ваш твиттер. Ethernet Shield открывает перед вами бесконечное количество новых возможностей.

Установка Ethernet шилда

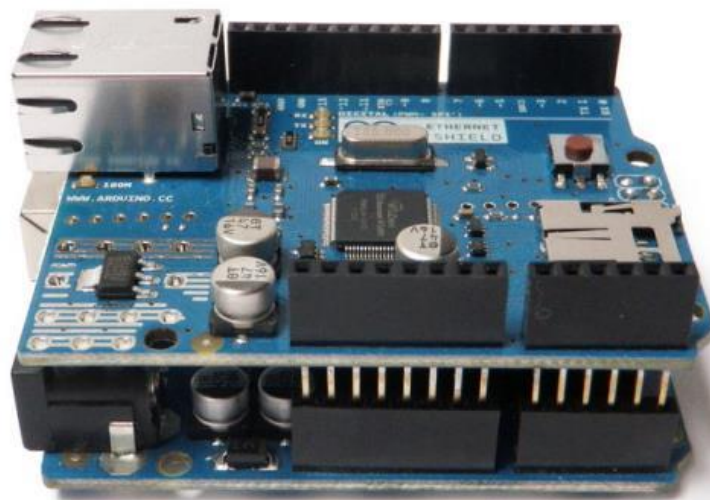
Устанавливается все предельно просто. Как и любой другой шилд, Ethernet Shield просто садится коннекторами на Arduino сверху.

Обратите внимание, что некоторые шилды могут быть несовместимы с более ранними версиями [Arduino](#). Так что рекомендуется использовать именно платы Arduino версии Rev 3.



Технические характеристики Ethernet шилда

Ethernet Shield основан на чипе W51000, который имеет внутренний буфер на 16К. Скорость подключения достигает 10/100Мб. Это не самое быстрое соединение, но этого вполне достаточно, поверьте.



Шилд работает с использованием [библиотеки](#) Arduino Ethernet library, которая по умолчанию интегрирована в оболочку Arduino IDE.

На Ethernet шилде есть слот для установки micro SD карты, с помощью которой можно хранить большие массивы информации и загружать веб-сайты непосредственно с Arduino. Не забудьте, что в этом случае надо использовать дополнительную библиотеку. Более детально про: [использование SD карты](#).

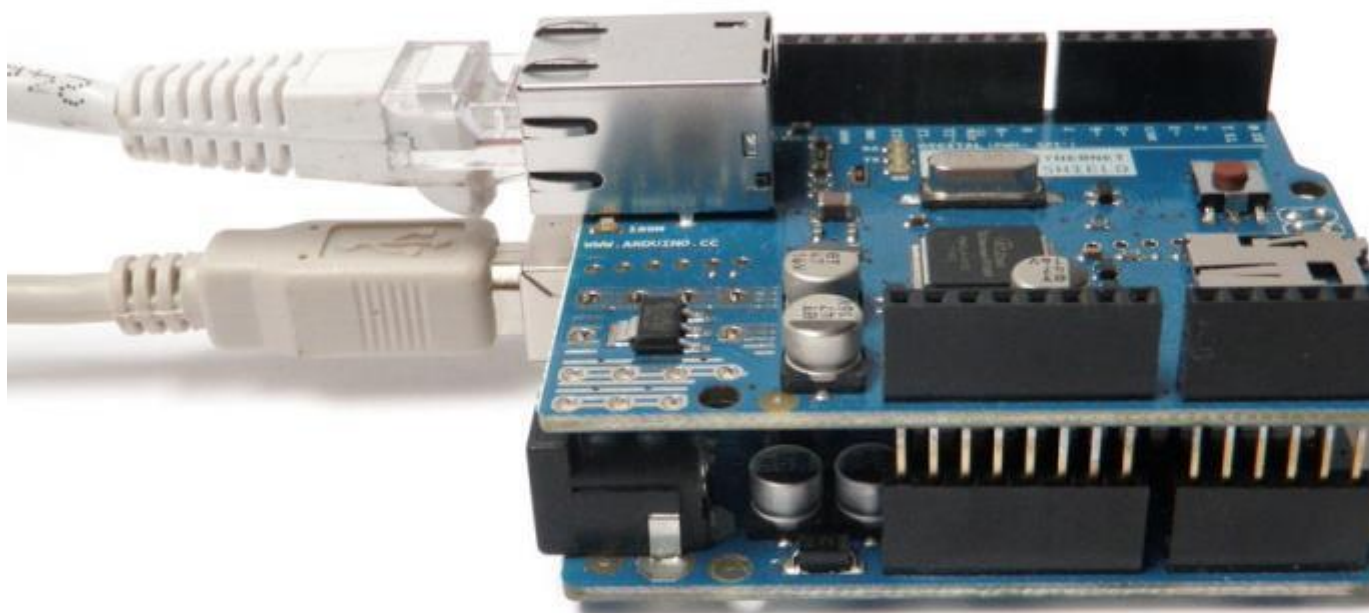
Кроме того, можно запитывать Arduino с помощью Ethernet соединения. Для этого надо использовать Power over Ethernet (PoE) модуль, место для установки есть на Ethernet шилде.

Сводный список технических характеристик Ethernet шилда:

- Для работы необходима плата Arduino
- Рабочее питание – 5 В (питается от платы Arduino)
- Ethernet контроллер: W5100 с буфером 16Кб
- Скорость подключения: 10/100Мб
- Подключается к Arduino через SPI порт

Первые шаги

Подключите Arduino к персональному компьютеру с помощью USB кабеля, а Ethernet Shield к вашему роутеру (или непосредственно вашему интернет-кабелю).



После этого откройте Arduino IDE. В версиях Arduino IDE после 1.0 есть встроенная поддержка DHCP и нет необходимости в ручной настройке IP адреса.

Для того, чтобы определить какой IP адрес присвоен вашей плате, откройте скетч DhcpAddressPrinter. Найти его можно в меню:

File --> Examples --> Ethernet --> DhcpAddressPrinter

После открытия скорее всего придется сменить мак-адрес. На более новых официальных версиях Ethernet шилда, адрес указан на стикере, прикрепленном к шилду. Если стикера нет, можно просто сгенерировать новый уникальный мак-адрес. Если вы используете одновременно несколько шилдов, мак-адреса для каждого из них должны быть уникальными.

После настройки мак-адреса, можно загружать скетч на плату Arduino и открывать серийный монитор. В результате должен отображаться используемый IP адрес.

Сервер



Можно использовать Arduino Ethernet shield в качестве веб-сервера для загрузки HTML страницы или функции. Можно отслеживать (парсить) запросы, которые отсылаются клиентом через веб-браузер. В следующих двух примерах показано как использовать наш "пирог" для обслуживания HTML страниц и парсинга URL строк.

Важно помнить: надо внести IP адрес вашей платы Arduino, чтобы приведенные ниже примеры корректно обрабатывали.

Приведенный ниже код изменяет содержимое веб страницы при нажатии кнопки:

```
/*
Web Server с использованием Ethernet шилда -
пример

Простой пример веб-сервера, который изменяет
страницу после нажатия кнопки.

Схема подключения
* Ethernet shield подключен к пинам 10, 11, 12, 13
* кнопка подключается между пинами D2 and 5V
* резистор на 10 КОм подключается между пином D2 и
землей
*/
#include
#include

// ниже необходимо ввести MAC адрес и IP адрес
вашего контроллера.

// IP адрес будет зависеть от вашей локальной
сети:

byte mac[] = { 0x00, 0xAA, 0xBB, 0xCC, 0xDA, 0x02
};

IPAddress ip(191,11,1,1); //<<< ВЕДИТЕ ВАШ IP
АДРЕС В ЭТОЙ СТРОКЕ!!!

// Инициализация библиотеки Ethernet server
// с указанием IP адреса и порта, который вы
хотите использовать

// (порт 80 является портом по умолчанию для
HTTP) :

EthernetServer server(80);

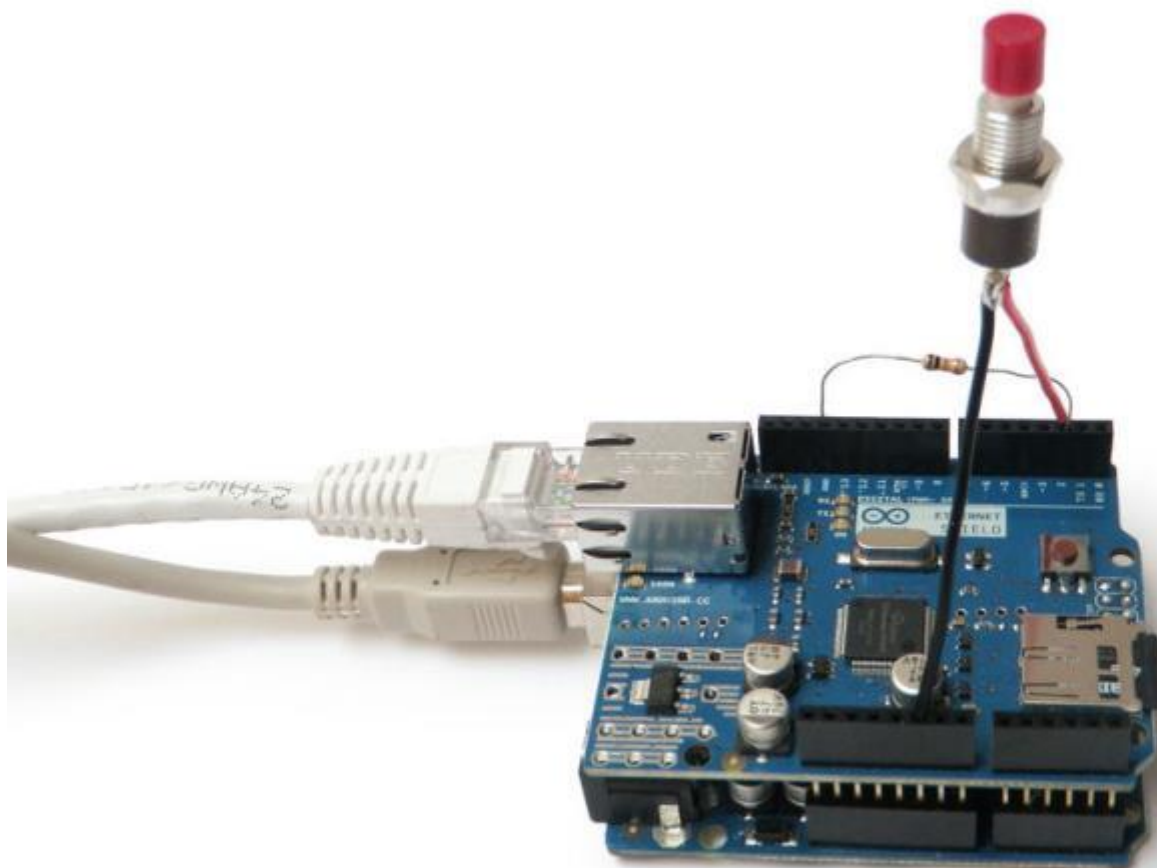
int buttonPress = 1;

void setup()
```

```
{
pinMode(2, INPUT);
// запускаем Ethernet подключение и сервер:
Ethernet.begin(mac, ip);
server.begin();
}
void loop()
{
buttonPress = digitalRead(2);
// отслеживаем входящего клиента
EthernetClient client = server.available();
if (client) {
// запрос http заканчивается пустой строкой
boolean currentLineIsBlank = true;
while (client.connected()) {
if (client.available()) {
char c = client.read();
// если вы добрались до конца строки (получили
символ для новой строки)
// и строка пустая, http запрос заканчивается.
// Соответственно, вы отсылаете ответ:
if (c == '\n' && currentLineIsBlank) {
// отсылаем стандартный http ответ
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println();
// обслуживает другую версию веб-сайта в
зависимости от состояния кнопки (нажата или нет),
// которая подключена к пину 2
if (buttonPress == 1) {
```

```
client.println("LIGHT!");
}
else if (buttonPress == 0) {
client.println("DARK!");
}
break;
}
if (c == '\n') {
// начинаем с новой строки
currentLineIsBlank = true;
}
else if (c != '\r') {
// если получен символ на данной строке
currentLineIsBlank = false;
}
}
}
// даем браузеру время для получения данных
delay(1);
// закрываем соединение:
client.stop();
}
}
```

Для того, чтобы скетч работал, подключите кнопку между пинами D2 и 5V. Резистор на 10 КОм между землей и пином D2.. После этого укажите IP адрес вашей Arduino в браузере. Должна загрузится страница с черным фоном. Нажмите и удерживайте кнопку, после этого перезагрузите страницу браузера. После этого страница должна загрузится с белым фоном.



Код, который приведен ниже, включает светодиод в зависимости от URL адреса, который отсылается на Arduino:

```
/*
```

```
Web Server - пример
```

```
Дает возможность включать и выключать светодиод  
при вводе разных URL адресов в браузере
```

```
Для того, чтобы включить:
```

```
http://ВАШ_IP_АДРЕС/$1
```

```
Для того, чтобы выключить:
```

```
http://ВАШ_IP_АДРЕС/$2
```

```
Схема подключения:
```

* Ethernet shield подключается к пинам 10, 11, 12, 13

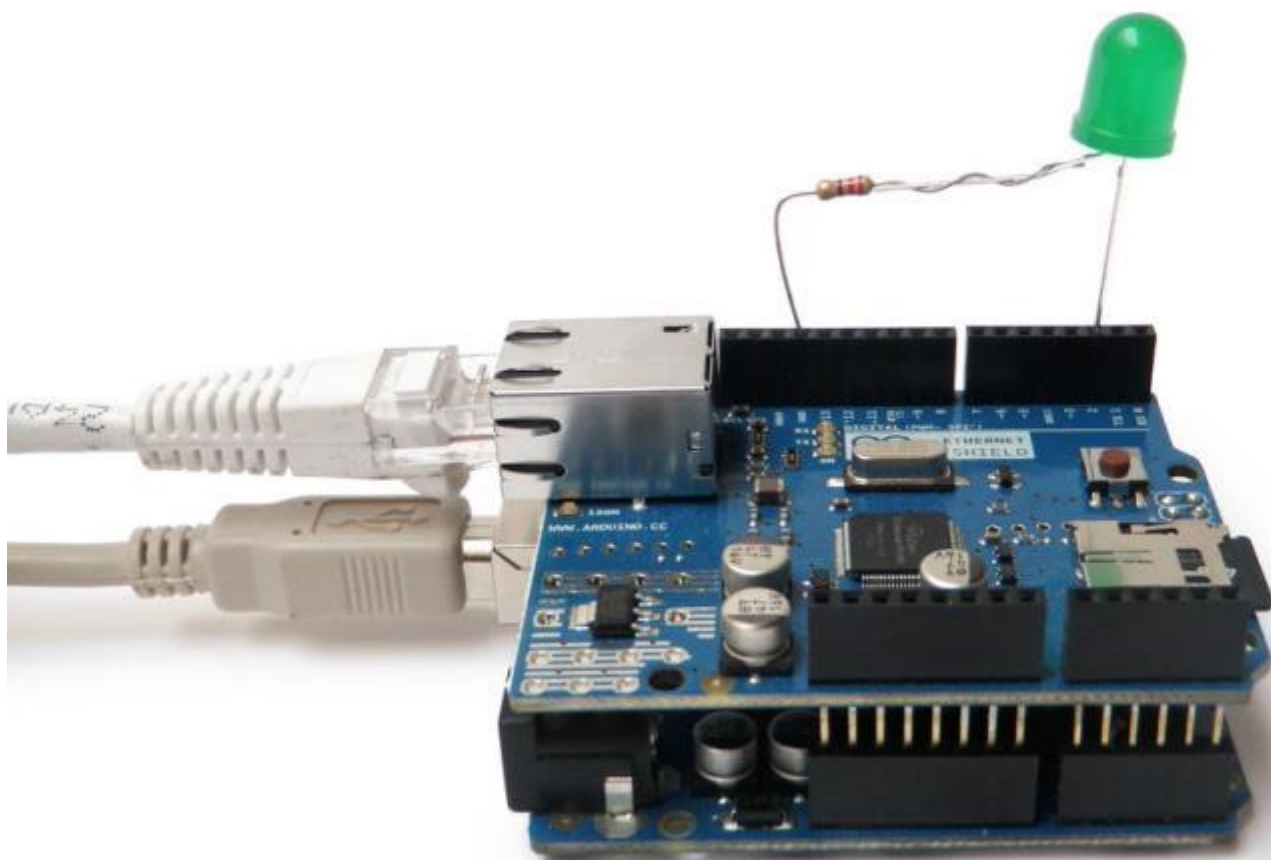
* Подключите светодиод к пину D2, а вторую ногу к земле через резистор 220 Ом

```
*/  
  
#include  
  
#include  
  
boolean incoming = 0;  
  
// ниже необходимо ввести MAC адрес и IP адрес  
вашего контроллера.  
  
// IP адрес будет зависеть от вашей локальной  
сети:  
  
byte mac[] = { 0x00, 0xAA, 0xBB, 0xCC, 0xDA, 0x02  
};  
  
IPAddress ip(191,11,1,1); //<<< ВВЕДИТЕ ВАШ IP  
АДРЕС В ЭТОЙ СТРОКЕ!!!  
  
// Инициализация библиотеки Ethernet server  
library  
  
// с использованием IP адреса и порта, который вы  
указали  
  
// (по умолчанию HTTP порт устанавливается на 80):  
EthernetServer server(80);  
  
void setup()  
{  
  pinMode(2, OUTPUT);  
  // запуск Ethernet подключения и сервера:  
  Ethernet.begin(mac, ip);  
  server.begin();  
  Serial.begin(9600);  
}  
  
void loop()  
{
```

```
// получаем данные от клиента:
EthernetClient client = server.available();
if (client) {
// HTTP запрос оканчивается пустой строкой
boolean currentLineIsBlank = true;
while (client.connected()) {
if (client.available()) {
char c = client.read();
// если вы дошли до конца строки и следующая
строка пустая,
// http запрос заканчивается и можно вывести ответ
//считывает строку URL от $ до первого пробела
if(incoming && c == ' '){
incoming = 0;
}
if(c == '$'){
incoming = 1;
}
// проверка строки URL. В ней присутствует $1 или
$2
if(incoming == 1){
Serial.println(c);
if(c == '1'){
Serial.println("ON");
digitalWrite(2, HIGH);
}
if(c == '2'){
Serial.println("OFF");
digitalWrite(2, LOW);
}
}
```

```
}  
if (c == '\n') {  
    // начинаем новую строку  
    currentLineIsBlank = true;  
}  
else if (c != '\r') {  
    // получаем символ на текущей строке  
    currentLineIsBlank = false;  
}  
}  
}  
// даем веб-браузеру время для получения данных  
delay(1);  
// закрываем соединение:  
client.stop();  
}
```

Для корректной работы приведенного примера, подключите позитивную ногу светодиода к пину D2, а негативную к земле через резистор 220 Ом.



Для того, чтобы включить светодиод, введите в строку браузера:
[http://\[ВАШ_IP_АДРЕС\]/\\$1](http://[ВАШ_IP_АДРЕС]/$1)

Для того, чтобы выключить светодиод, введите в строку браузера:
[http://\[ВАШ_IP_АДРЕС\]/\\$2](http://[ВАШ_IP_АДРЕС]/$2)

Обратите внимание: Надо заменить строку [ВАШ_IP_АДРЕС] вашим IP адресом.

Клиент

Можно использовать Ethernet Shield для отработки на стороне клиента. Другими словами, Arduino с шилдом

используются для считывания информации с веб-сайта по принципу веб-браузера.

На веб-сайтах большое количество текста. Причем текст может быть как видимым так и невидимым. Так что программирование на стороне клиента превращается в интересную задачу. Считывание информации с веб-сайта сводится к парсингу большого количества строк.

На данный момент в сети валяется куча примеров для считывания сообщений и Твиттера. В нашем случае стандартный пример будет несколько модифицирован. В результате мы сможем включить светодиод при считывании определенного сообщения.

Для работы вам надо подключить позитивную ногу светодиода к пину D2, а негативную - к земле через резистор на 220 Ом.

Не забудьте ввести ваш индивидуальный IP адрес в код ниже или скетч не отработает.

Скетч приведен ниже:

```
/*
```

```
Twitter клиент
```

```
Эта программа дает возможность подключиться к  
Twitter с использованием Ethernet шилда. Мы парсим  
XML
```

```
И ищем тут_ваш_твит
```

```
Можно использовать Ethernet шилд от Arduino, от  
компании Adafruit, да и большинство китайских тоже  
подойдут, так как на них установлен модуль Wiznet  
Ethernet.
```

```
В примере используются процедуры DHCP в библиотеке  
Ethernet library, которая входит в состав Arduino  
IDE начиная с версии 1.0 beta 1
```

```
В примере используется библиотека String library,  
которая входит в состав Arduino IDE с версии 0019.
```

```
Схема подключения:
```

```
* Ethernet шилд подключается через пины 10, 11,  
12, 13
```

```

*/
#include
#include
// ниже необходимо ввести MAC адрес и IP адрес
вашего контроллера.
// IP адрес будет зависеть от вашей локальной
сети:
byte mac[] = {
0x00, 0xAA, 0xBB, 0xCC, 0xDE, 0x01 };
IPAddress ip(191,11,1,1); //<<< ВВЕДИТЕ ВАШ IP
АДРЕС В ЭТОЙ СТРОКЕ!!!
// Инициализация библиотеки
EthernetClient client;
const int requestInterval = 60000; // задержка
между запросами
char serverName[] = "api.twitter.com"; // URL
твиттера
boolean requested; // сделали вы запрос после
соединения
long lastAttemptTime = 0; // последний раз, когда
вы подключались к серверу в миллисекундах
String currentLine = ""; // строка для хранения
текста с сервера
String tweet = ""; // строка для хранения твита
boolean readingTweet = false; // если вы читаете
ТВИТ В ДАННЫЙ МОМЕНТ
void setup() {
pinMode(2, OUTPUT);
// резервируем место для строк:
currentLine.reserve(256);
tweet.reserve(150);
// инициализация серийного протокола:

```

```
Serial.begin(9600);
// попытка DHCP подключения:
if (!Ethernet.begin(mac)) {
  // если DHCP подключение не удалось, начинаем с
адреса:
  Ethernet.begin(mac, ip);
}
// подключаемся к Twitter:
connectToServer();
}
void loop()
{
  if (client.connected()) {
    if (client.available()) {
      // read incoming bytes:
      char inChar = client.read();
      // добавляем входящий байт в конце строки:
      currentLine += inChar;
      // если мы переходим на новую строку, очищаем:
      if (inChar == '\n') {
        currentLine = "";
      }
      // если текущая строка заканчивается на , за ней
// последует твит:
      if (currentLine.endsWith(",")) {
        // твит начинается. Очищаем строку твита:
        readingTweet = true;
        tweet = "";
      }
    }
  }
}
```



```
// если вы на данный момент считываете байты из
твита,
// добавляем их в строку tweet:
if (readingTweet) {
  if (inChar != '<') {
    tweet += inChar;
  }
  else {
    // если вы пропарсили до символа "<",
    // вы достигли конца твита:
    readingTweet = false;
    Serial.println(tweet);
    if(tweet == ">Hello Cruel World"){
      digitalWrite(2, HIGH);
      Serial.println("LED ON!");
    }
    if(tweet != ">Hello Cruel World"){
      digitalWrite(2, LOW);
      Serial.println("LED OFF!");
    }
    // close the connection to the server:
    client.stop();
  }
}
}
}
}

else if (millis() - lastAttemptTime >
requestInterval) {
  // если вы не подключены и с момента последнего
подключения прошло
```

```

// две минуты, пытаемся подключиться вновь:
connectToServer();
}
}

void connectToServer() {
// пытаемся подключиться и ждем миллисекунду:
Serial.println("connecting to server...");
if (client.connect(serverName, 80)) {
Serial.println("making HTTP request...");
// делаем HTTP GET запрос к twitter:
client.println("GET
/1/statuses/user_timeline.xml?screen_name=RandyMcTe
ster&count=1 HTTP/1.1");
client.println("HOST: api.twitter.com");
client.println();
}
// отмечаем время попытки этого подключения:
lastAttemptTime = millis();
}

```

Вероятно, вам захочется считать какие-данные, которые не имеют отношения к последнему посту в Твиттере.

Для того, чтобы считать другие посты из Твиттера, внесите следующие изменения в программу:

```

client.println("GET /1/statuses/user_timeline.xml?screen_name=[NEW
TWITTER NAME HERE]&count=1 HTTP/1.1");

```

Поздравляем! Вы ознакомились с базовыми принципами работы Ethernet шилда и его возможностями. Упомянутых в статье функций более чем достаточно для покорения интернет пространства в компании с Arduino.