

Связка Arduino + Ethernet shield позволяет создавать устройства, реализующие концепция Internet of things ("Интернет вещей"), согласно которой доступ к данным датчиков или управление исполнительными устройствами может быть обеспечено из любой точки мира. Но ни плата Arduino, ни Ethernet shield не содержат ни датчиков, ни исполнительных устройств. Их надо подключать. [Easy Module Shield v 1](#) - это универсальная плата для построения устройств для Интернета вещей (как измерение, так и управление), потому что на ней уже присутствуют датчики и устройства.



На плате установлены следующие датчики и устройства:

- цифровой датчик температуры и влажности DHT11 (подключен к D4);
- аналоговый датчик температуры LM35 (A2);
- аналоговый датчик освещенности (A1);
- приемник ИК сигналов с пульта (D6);
- динамик для генерации простейших звуковых сигналов (D5);
- две кнопки (D2, D3);
- потенциометр (A0);
- светодиод синий D13;
- светодиод красный D12;
- RGB светодиод (D9, D10, D11).

a place to push your data.

Why are you building this?

We want to bring a dose of reality to the Internet of Things hype. data.sparkfun.com is a free, robust service for use with all of your projects. The underlying engine is open source so if you don't want to use our servers you can install [phant](#) on the server of your choice.

Wait, this is totally free? What's the catch?

Yep. There are limits, but we wanted to give our users a good, free place to store data and give data scientists more fun things to analyze. Our hope is that you buy a SparkFun widget to connect your next beehive.

How do I use it?

We thought data storage should be as easy as string concatenation. First, create a stream and you'll receive a public url and a private key. Then, using the hardware of your choice (Ethernet shield, RasPi, BBB, Electric Imp, WiFi) simply create a string with the data you want to post. Here's a simple example:

```
https://data.sparkfun.com/input/Jxyjr7DmxwTD5dG1D1Kv?  
private_key=gzgnB4Vazklg7GN1g1qA&brewTemp=33.4
```

Go ahead and try it! Paste the string above into a browser window and hit return. You should see a confirmation message. Then view the results on the [public stream](#). Congrats! You just added some data to the internet.

What types of projects would benefit from this service?

Almost anything. Your weather station in the backyard. A classroom of kids working on science experiments. A community concerned with pollution and crowdsourcing data collection. The weight of your pug's food dish. You get the idea. If your project can send an http request, you can push data out to our servers.

Create a free data stream immediately
at data.sparkfun.com

CREATE

Explore all of the public data
streams on data.sparkfun.com

EXPLORE

Learn how to create data streams and
post data to them.

DOCS

Configure and deploy your own copy
of the phant server.

DEPLOY

Help improve the service by reporting
any issues you encounter.

BUGS

Для отправки данных в сервис необходимо создать поток. Нажимаем на кнопку **CREATE** и на открывшейся странице заполняем данные для нашего потока – Название, описание, названия передаваемых в поток параметров (Fields), теги, местоположение.

Вводим в качестве параметров Fields:

- **dht11_humidity** – для показаний влажности воздуха датчика DHT11;
- **lm35_temp** – для показаний температуры датчика LM35.

Create a Data Stream

If you need more info about creating a stream, check out the [stream creation documentation](#).

Title*

Arduino-Kit_example_01

Description*

Передача данных с устройства
- Arduino Uno
- Ethernet shield W5100
- Easy Module shield v 1

Параметры



Show in Public Stream List?*

Visible Hidden

Fields*

dht11_humidity x **lm35_temp** x humidity, temp

Stream Alias

arduinokit_easy_module_shield_v1 ✓

This will be used as an alias for your stream when sharing.

e.g. http://data.sparkfun.com/arduinokit_easy_module_shield_v1

Tags

arduinokit x **easy_module_shield_v1** x nist, weather

Location

Пятигорск, Ставропольский край, Россия

Save



Для сохранения нажимаем кнопку **Save**. Открывается страница с данными для нашего потока. Для передачи данных с нашего устройства в сервис нам понадобятся следующие данные:

- **Public Key**;
- **Private Key**.

Формат отправки данных

[http://data.sparkfun.com/input/\[publicKey\]?private_key=\[privateKey\]&dht11_humidity=\[value\]&lm35_temp=\[value\]](http://data.sparkfun.com/input/[publicKey]?private_key=[privateKey]&dht11_humidity=[value]&lm35_temp=[value])

Страница для просмотра отправленных данных

<https://data.sparkfun.com/streams/LQm5nXj7E1fjONaJLoGw>

New Stream: Arduiuno-Kit_example_01

Передача данных с устройства - Arduino Uno - Ethernet shield W5100 - Easy Module shield v 1

Fields: `dht11_humidity` `lm35_temp`

Tags: `arduinokit` `easy_module_shield_v1`

Public URL

`http://data.sparkfun.com/streams/LQm5nXj7E1fj0NaJLoGw`

Public Key

`LQm5nXj7E1fj0NaJLoGw`

Private Key

`A19pd0zJeksbxBNVvn19`

Keep this key secret, and in a safe place. *You will not be able to retrieve it.*

Delete Key

`AEXbVNa7kotbLn6Z8vdY`

This key can only be used once. Keep this key secret, and in a safe place. *You will not be able to retrieve it.*

Download Your Keys

Download your keys as a JSON file.

Keep this file secret and in a safe place.

Logging using query string parameters

Format:

`http://data.sparkfun.com/input/[publicKey]?private_key=[privateKey]&dht11_humidity=[value]&lm35_temp=[value]`

Example:

`http://data.sparkfun.com/input/LQm5nXj7E1fj0NaJLoGw?private_key=A19pd0zJeksbxBNVvn19&dht11_humidity=13.80&lm35_temp=19.19`

Теперь собираем наше устройство (Arduino + Ethernet shield + Easy Module Shield), подключаем к сети Интернет и загружаем в него скетч (вам необходимо изменить параметры сетевого подключения на свои).

```
// датчик влажности воздуха
#define PIN_DHT 4
// датчик температуры LM35
#define PIN_LM35 A2
// dht
#include "DHT.h"
```

```

// тип датчика dht
#define DHTTYPE DHT11
// создать экземпляр DHT
DHT dht(PIN_DHT, DHTTYPE);

// Ethernet
#include <SPI.h>
#include <Ethernet.h>
// 1-dhcp, 0 - manual
#define DHCP 0
// mac-адрес Ethernet shield
byte arduino_mac[] = { 0xFF, 0xFE, 0xFB, 0xFA, 0xF9, 0xF8 };
EthernetClient client;
// если адрес статический
// (установить #define DHCP 1)
int mip[4]={192,168,0,121};
int mmask[4]={255,255,255,0};
int mgateway[4]={192,168,0,28};
int mdns[4]={192,168,1,1};
// для data.sparkFun
char server[] = "data.sparkfun.com";
const String publicKey = "LQm5nXj7E1fjONaJLoGw";
const String privateKey = "A19pd0zJeksbxBNVvn19";
const byte NUM_FIELDS = 2;
const String fieldNames[NUM_FIELDS] = {"dht11_humidity",
"lm35_temp"};
int fieldData[NUM_FIELDS];

void setup() {
  Serial.begin(9600);
  // установка сетевого соединения
  ethernet_begin();
  print_ip();
  // инициализация dht11
  dht.begin();
}

void loop() {
  fieldData[0]=get_dht11();
  fieldData[1]=get_lm35()-20;
  Serial.print(fieldNames[0]);Serial.print("=");
  Serial.println(fieldData[0]);
  Serial.print(fieldNames[1]);Serial.print("=");
  Serial.println(fieldData[1]);
  sendData();
  delay(30000);
}

// запустить ethernet
void ethernet_begin()
{
  if(DHCP==1)
  {

```

```

    if(Ethernet.begin(arduino_mac)==0)
        Serial.println("dhcp - error");
    else
        Serial.println("dhcp - ok");
    }
else
{
    IPAddress arduino_ip(mip[0],mip[1],mip[2],mip[3]);
    IPAddress dns_ip(mdns[0],mdns[1],mdns[2],mdns[3]);
    IPAddress
gateway_ip(mgateway[0],mgateway[1],mgateway[2],mgateway[3]);
    IPAddress subnet_mask(mmask[0],mmask[1],mmask[2],mmask[3]);
    Ethernet.begin(arduino_mac, arduino_ip, dns_ip, gateway_ip,
subnet_mask);
}
}
// вывести ip-адрес
void print_ip()
{
    Serial.print("My IP address: ");
    for (byte thisByte = 0; thisByte < 4; thisByte++) {
        // print the value of each byte of the IP address:
        Serial.print(Ethernet.localIP()[thisByte], DEC);
        Serial.print(".");
    }
    Serial.println();
    Serial.print("My SUBNET: ");
    for (byte thisByte = 0; thisByte < 4; thisByte++) {
        Serial.print(Ethernet.subnetMask()[thisByte], DEC);
        Serial.print(".");
    }
    Serial.println();
    Serial.print("My GATEWAYIP: ");
    for (byte thisByte = 0; thisByte < 4; thisByte++) {
        Serial.print(Ethernet.gatewayIP()[thisByte], DEC);
        Serial.print(".");
    }
    Serial.println();
    Serial.print("My DNS: ");
    for (byte thisByte = 0; thisByte < 4; thisByte++) {
        Serial.print(Ethernet.dnsServerIP()[thisByte], DEC);
        Serial.print(".");
    }
    Serial.println();
}
// получить показания с датчика DHT
int get_dht11()
{
    float fhumidity;

    fhumidity=dht.readHumidity();
    if (isnan(fhumidity))
    {

```

```

        return 1000;
    }
    return int(fhumidity);
}
// получить показания LM35 - датчик температуры
int get_lm35()
{
    double val=analogRead(PIN_LM35);
    double voltage=val*5.00/1024;
    double temp=voltage*100-273;

    return int(temp);
}
// отправка данных
void sendData()
{
    // Подключиться к хосту
    if (client.connect(server, 80))
    {
        // Host: data.sparkfun.comn
        // Connection: closed
        //
        client.print("GET /input/");
        client.print(publicKey);
        client.print("?private_key=");
        client.print(privateKey);
        for (int i=0; i<NUM_FIELDS; i++)
        {
            client.print("&");
            client.print(fieldNamees[i]);
            client.print("=");
            client.print(fieldData[i]);
        }
        client.println(" HTTP/1.1");
        client.print("Host: ");
        client.println(server);
        client.println("Connection: close");
        client.println();
        client.println();
        client.stop();
        client.flush();
    }
    else
    {
        Serial.println(F("Connection failed"));
        client.stop();
        client.flush();
    }
}

```

Вывод данных в последовательный порт


```
My IP address: 192.168.0.121.
My SUBNET: 255.255.255.0.
My GATEWAYIP: 192.168.0.28.
My DNS: 192.168.1.1.
dht11_humidity=37
lm35_temp=19
dht11_humidity=38
lm35_temp=19
dht11_humidity=38
lm35_temp=19
dht11_humidity=38
lm35_temp=19
dht11_humidity=38
lm35_temp=19
dht11_humidity=38
lm35_temp=19
dht11_humidity=38
lm35_temp=19
dht11_humidity=38
lm35_temp=19
dht11_humidity=38
lm35_temp=19
dht11_humidity=37
lm35_temp=19
dht11_humidity=37
lm35_temp=19
dht11_humidity=37
lm35_temp=19
dht11_humidity=37
lm35_temp=19
```

И на странице <https://data.sparkfun.com/streams/LQm5nXj7E1fjONaJLoGw> видим отображение поступающих данных.

dht11_humidity	lm35_temp	timestamp
38	19	2016-11-03T11:53:18.810Z
38	19	2016-11-03T11:52:48.070Z
38	19	2016-11-03T11:52:17.327Z
38	19	2016-11-03T11:51:46.584Z
38	19	2016-11-03T11:51:15.785Z
37	19	2016-11-03T11:50:45.044Z
37	20	2016-11-03T11:42:09.969Z
37	20	2016-11-03T11:41:52.402Z
38	22	2016-11-03T11:37:38.190Z
38	22	2016-11-03T11:33:52.388Z
38	35	2016-11-03T11:28:06.496Z
38	35	2016-11-03T11:26:49.367Z
37	32	2016-11-03T11:24:50.494Z

Скачать данный скетч (ссылка) и библиотеку DHT11 (ссылка).