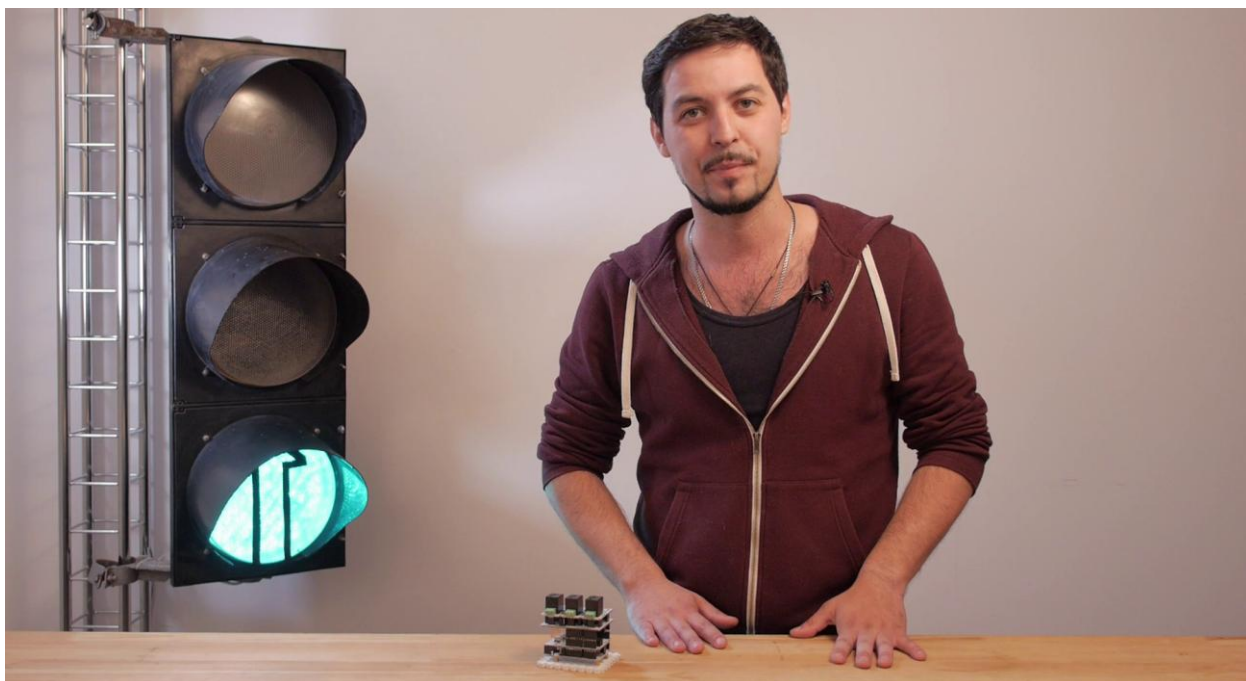


Виджет из светофора: мониторим Travis CI на Iskra JS



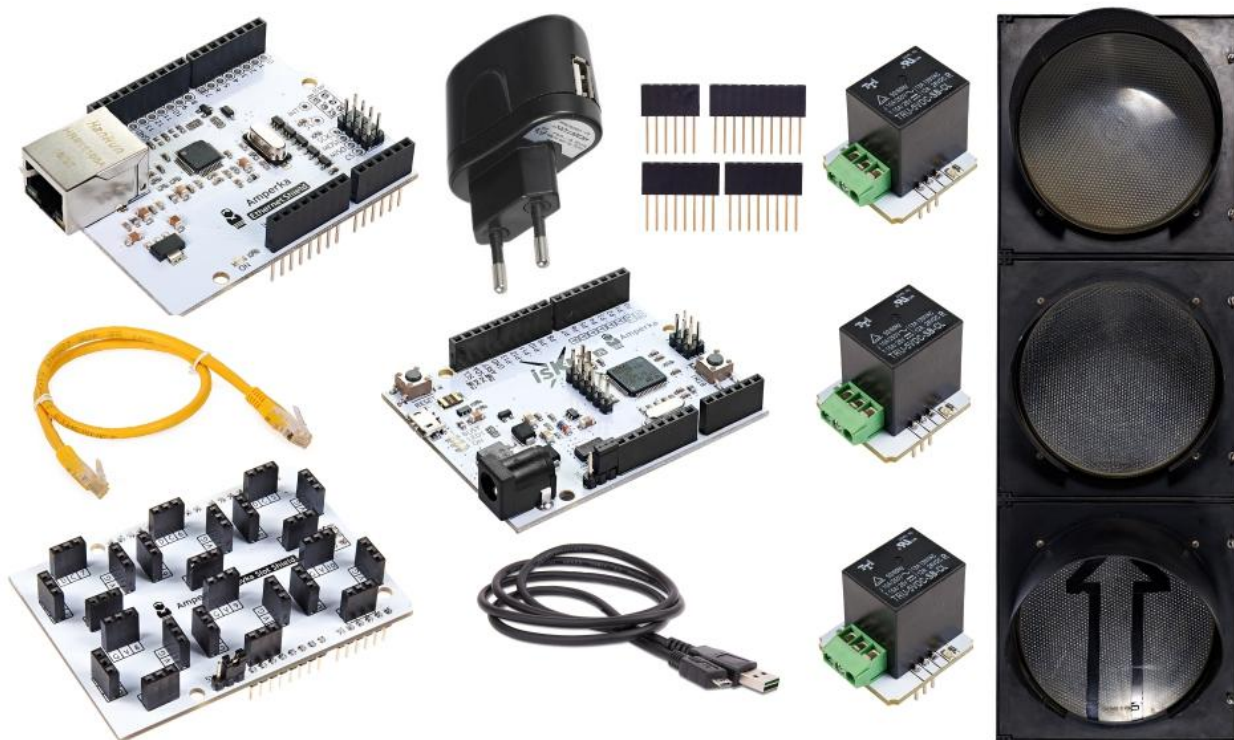
- Платформы: Iskra JS
- Языки программирования: JavaScript
- Тэги: Continuous Integration, непрерывная разработка, Travis CI, автоматическое тестирование, виджет

Что это?

Однажды нам в руки попал списанный светофор. Приспособить его для отображения пробок? Сделать индикатор занятости переговорной комнаты? Прицепить к гаражному парктронику? Нет, это всё слишком просто и банально.

Мы решили помочь нашим программистам продемонстрировать усердную и кропотливую работу всей Амперке. Разработку программного обеспечения мы ведём на GitHub-е, а обеспечить непрерывную интеграцию нам помогает сервис Travis CI. Именно к нему мы и подключили светофор, превратив его в виджет, который сообщает как прошла последняя сборка и стоит ли заводить ветку hotfix в github.

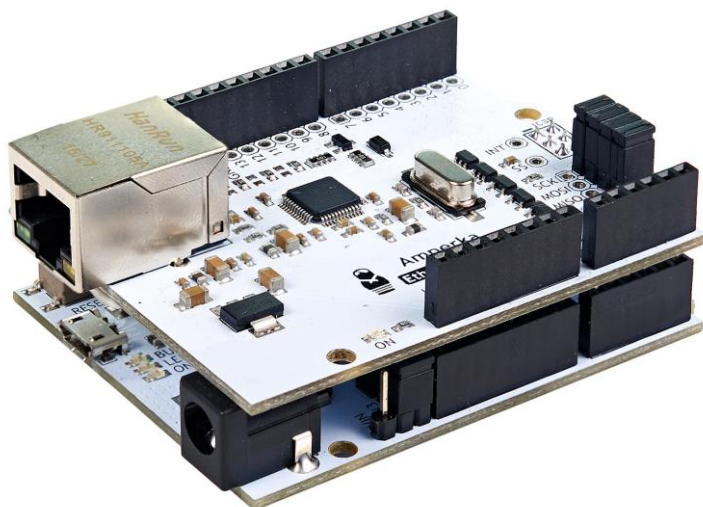
Что нам понадобится?



1. [Iskra JS](#)
2. [Ethernet Shield](#)
3. [Тройка Slot Shield](#)
4. [Контактные колодки Arduino](#)
5. 3 × [Мини-реле \(Тройка-модуль\)](#)
6. [Патч-корд витой пары](#)
7. [Кабель USB \(A — Micro USB\)](#)
8. [Импульсный блок питания с USB-разъёмом \(5 В, 1000 мА\)](#)
9. Светофор

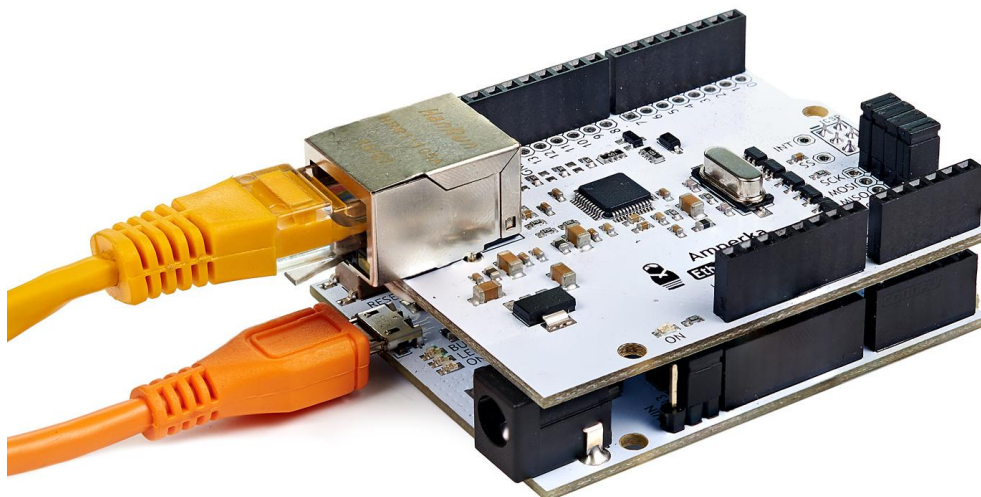
Как собрать?

1. Установите Ethernet Shield на Iskra JS.



Эти платы уже позволят взаимодействовать с API [travis-ci](#). Travis-ci — это хостинг непрерывной интеграции для open source сообщества. Он используется для проведения сборки и тестирования исходного кода проекта.

2. Подключите Iskra JS кабелем micro-USB к компьютеру. Подключите Ethernet Shield к своей локальной сети через патч-корд витой пары.



3. Настроим связь с Интернетом — напишем простой GET-запрос по URL. Загрузите код в Iskra JS. Устройство один раз сделает запрос к DHCP-серверу для получения IP-адреса, затем будет каждые 5 секунд отправлять запрос на URL-адрес example.com. В ответ придёт html-код запрашиваемой страницы.

[get-request.js](#)

```
// Настраиваем соединение с Ethernet Shield по протоколу SPI.
SPI2.setup({baud: 3200000, mosi: B15, miso: B14, sck: B13});
var eth = require('WIZnet').connect(SPI2, P10);
// Подключаем модуль http.
var http = require('http');

// Получаем IP-адрес от DHCP-сервера
eth.setIP();

// Задаём опции подключения к запрашиваемому серверу
var options = {
  host : 'google.ru',
  port : '80',
  path : '/',
  protocol : 'http:',
  headers : {
    'Accept': 'text/html'
  }
};

var request = function () {
  var response = '';
  http.get(options, function(res) {
    // Большие ответы от сервера могут приходить по частям.
    // В переменной response собираем весь ответ целиком.
    res.on('data', function(data) {
      response += data;
    });
    // После закрытия соединения обрабатываем весь пришедший ответ
    res.on('close', function() {
      if (response === undefined) {
        print('Error. Response is undefined.');
```

```

    });
  });
};
// Производим запрос
request();

```

4. Подключимся к сервису travis-ci. Будем использовать функцию [branches](#) API, при вызове которой сервис travis-ci отдаёт информацию об указанной ветке в репозитории. Для начала произведём запрос через адресную строку браузера. Введите URL <https://api.travis-ci.org/repos/niggor/buildMonitor/branches/master>. В ответ travis-ci вернёт строку в формате JSON. С помощью [онлайн редактора](#) JSON можно посмотреть эту строку в понятном виде.

```

5. {
6.   "branch": {
7.     "id": 151571537,
8.     "repository_id": 91736666,
9.     "commit_id": 43049835,
10.    "number": "8",
11.    "config": {
12.      "language": "node_js",
13.      "node_js": [
14.        "5"
15.      ],
16.      ".result": "configured",
17.      "group": "stable",
18.      "dist": "precise"
19.    },
20.    "state": "passed",
21.    "started_at": "2016-08-11T17:10:50Z",
22.    "finished_at": "2016-08-11T17:11:32Z",
23.    "duration": 42,
24.    "job_ids": [
25.      151571538
26.    ],
27.    "pull_request": false
28.  }
}

```

29. В этом ответе нам интересны два поля: `number` и `state`. Поле `number` сообщает номер последней сборки. По нему виджет будет определять наличие новой сборки. Если новой сборки нет, сигнализировать о старых не стоит. Поле `state` отвечает за статус сборки и может принимать 4 значения: `created`, `started`, `passed`, `failed`.
30. Сделаем то же самое, но с помощью Iskra JS.

Travis-ci предоставляет API только по защищённому соединению `https`. Если у вас появляются ошибки TLS соединения, [обновите прошивку](#) Iskra JS.

[get-travis-ci.js](#)

```
// Настраиваем соединение с Ethernet Shield по протоколу SPI.
```

```
SPI2.setup({baud: 3200000, mosi: B15, miso: B14, sck: B13});
var eth = require('WIZnet').connect(SPI2, P10);
// Подключаем модуль http.
var http = require('http');

// Получаем IP-адрес от DHCP-сервера
eth.setIP();

// Задаём репозиторий в github и ветку, которую хотим отслеживать
var repo = 'niggor/buildMonitor';
var branch = 'master';

// Задаём опции подключения к запрашиваемому серверу
// Поля key, ca и cert необходимы для установления зашифрованного
соединения https
var options = {
  key :
    atob('MI IJKQIBAAKCAgEApB40JT9Y9n05aDagDqXvohgEySccefWtxwQUQQNpVuv0+KCiRX
FqLPkzIswK5ZLA43qDKTdnfvpScC1GeR3BxAda/uGy+BbpUozKI1CTQ/+Z+B1QWbMweRARm
rctNSAhYcelCsYfBU7DprWXP63ZoNPCxtNl0y4QEY6PfuPnG/dStu4iyaqqseF1MVWML2xp
sHWrKhxvu3sASrOwR3KazKd4gkPqfTPLEDz4p8YlzLuovtOlXU1WxX9nOUYd1xv5U7rIw02
OBIPztyICHxH2KX5aIlMnkgR7fMgNXS4jwUyWApYpe/+gPvD09+QmcLqpNzkTiyAc+wycE/
aLH/2QEZXEBLEaP7l6vwzI8CwZ4tzsBKWB4whiJ7JJjeSYfB6U6keAcMGCixMNuSG9giWIZ
VlrJgIztSJ2MnjMEWE1Y8CUOc9P41OFM0dw8t2UfcqgQLbY4X0wQ0SHpOGmnP9uNE/mcSG
pavk8sM9pgGA8GS+BicJPwJVex2iZsLei/n08fZDqfI9xuMvPrJOqTEuY8zQaJvZvfrBYNL
EAKZKzfuLKOJ1zjSybj6TZseJsI4sObrNrMs8n52rDuw+uWq8J76rnN/zqdCBysE1U2lQtj
Ly79EHA+/lUyUyS5AyxvEPdVwA+di0Qzb/jUs0ysGbs3PqJwMTC5FaUejDF7n7FbBsCAwEA
AQKCAgAe/9pWKMgKDw834xGKksdXy8ejozKbch8TgXxXc6Y74rfJ6BfE5QlmJCTIwlv3a93
iF2b/G4dKvVFYRftqrQ97dG/USx39gPuZs+Lqh2W+4G+vvK9br5aL7HTkoQEg+QqTwaaqV+
UphMIu0ZBEvKKhVg6LBYQtKOUXUo3NUWSn73M9eA2TXlejiKPxpG/WigAf8lr8+EA/ZnJgn
VqmXgp7fwyRtFu/GJ/17R0JFZhfJtCtLqgOnpWkQ/NgeyEJz6vhp+diFJUdwhA/yjmyWrcvk
UE/OIvB5fzZTgRQNHXmfjDE3E5LsLY5DVSIOl0Ub/2kg7BwWAMBvhlPH+mZilCbMIOChD5r
9LjHIPjr4jKwPpim3plgtNLca52x+HmVXwkFoU15s3ybvAn0q4OZpjOxZjUrsmhgh/ibzQd
vpCNRdESd08QsBJJf4L35pAo58LYceCQI4vqzFqzOnAIS9we0MUstadsGccUs6n8wVdn9zY
fcF3fGFPJ4PmNF4a+n9a/vh/X006T2DUBfirtAOzwb2xeUqR+T90UpBbAM4D4320NynvhX075
8fCREfT0Zc/P+yR6+xHzJ4jppqszoIPvfKxvZmSSS/L5oNcaEDzweM0+U5iv4GbJHQ1ldzCN
zPJoOhflvsqdVhCSiMgScGnuU3V9lykMesH28STZc8Mooigt7QQKCAQEA2ZuSfGLB4kNsh9
R26uwLstg6lC2bJbM5oAX5SvTMAgXeRkw192d1T86F5bCyvWzS8AjGxSH5t//cs8snsXHDv
IqCh28P5t4b9/+2M/DEDzxWkfgswVVU9JFELgfgioxuq+fc1X+ZLEYxibzNqNDI+Y019us2
S6R7b3oPzWXhdY4sbLdThT9vSi/CwxMCCqVLjnZHWic7Z/ZIcXoc97I4KyYK1sLusLa850+
FELoKk+SN7rXOSIqiqFqVWIXSt8hHeHg37Lg15YBFH8f16Bte9G3V0qDyIKVqlcs/dwzJOn
pTxOZgdSy5ChDPS9010yIjHguL6kXQRvLaTfxB4788YQKCAQEAwRKM5qsCvUGf+gTV0OGjZ
mGun9G4EEsIbXyh8HoIdI5JsQY0d+sr2jGxP3Bh3sHQGTkWKI/u/yx7KuglfQo6gGHgx0pY
LchZb5u+uJTACgXES/U/G2hbB10znMDqxo1PGuQ6N11pmFv5/OLqinuh6Nmsjfa59Ev5Scp
gsMhCj+Ifz3X+zirGRhuvQGjyWwAmpNEU0C5xinKVx3SimdkxjOb1RiJ6mwcPp6XqV1A49W
q6f6ircuecE7vpL9PCiFPclH7WrPbu6wDGLQeZd+JHeQ0TGRErmqx0D0b3fxS1Myaa52039S
QAtoxcQ7WT/PFS1rzv9UIW05E8Ukhd3Z3fz+wKCAQEAxvDGpDzqgbvYXfms2isAqvmqFs62
92159kVf2togghsIuUX1VIDeLB8IJKw9TgzGzU+IwTLFhSVC1ZDWchAqvZKulTjBFfTbJwg
9gBDOFYtRLdPwvROPnqOsS2ljUFUA3OJ17y3fUC3AHkdg0cRG6Hbj7iCtm8v9GTZDeil5TK
q1DJMxR5lHulAdw4/+HtNtVwnlQ5N4aRwssKPCzkq9tRzrbdn6s83stXQQd/FNuFWMOL83n
vQ3rqhZvvthYKmWRZnKrvzeMTnqeBqsQw266sX7ZLvFgHkaD2YoWGN1GKQ7L1jauGUxqfUi
Px42CyCzcEVoHb7Dog9XMTk7PkZYIqKCAQEAjZmOTyqCwszbiGF9c0nZLiBcEwibhw+QqcP
rfNPzWZBlIWh0mLKYodkqFs3UaawZdDZDaxK1BN7ZollKDU+wIjImyeLDkyhKN6mxEey+f1
tFYTQHYZEameAQbdmrFl3S83UJIJAX/s8OJCNEQRzZw7PJcWdN1tbViQhTmhpdfusny17Nt
uT2L7rkKEgP3x/YN67EJf659QEeyn7HNjBtv9ovbvWgmAxwtdLgb9c8Pvm8uB27SHWHKY36
csJK/PMMZrgzt/0OsroJnkLhkmQZaTVmmU7XfQP0ZWJtOEuW7pnh225yYdtIo8EqAAG/k76
9GSzVFvZI+dzpyE9/mhnzewKCAQB6ZeaPI+k7CsNfyt9j4HX9JKi5ZyCSBUdBJ+kJ7xOd1F
a0Y/OcnQh6vLeC5oqtJLB4mLnRvutyK6NP3sI6dLNvsqjVNex15pSI0/ZlNXniiFcc119
pnA4znialGLgh+IO72OQrWMrEhTSuSBZnCr71+r0ACIVv5t1A57PlXr+bqv/JIKMwiocFeoW
Dh9NfuhvA5ecv/Q1OS0rsvK1tSFW+dKzMQgE1P8yB9DmyVZ4F160Yf5dERcRzYLd+hFs4eQ
+LpP11sMhF8TIN1optRT4+NdY+ZdJbhBYeriFOPVUcfn4bdkmGezmy9UFg+IArnHB8bXZow
pFwfcieNZuwWNE'),
  cert :
    atob('MIIFijCCA3KgAwIBAgIJANLNN7E5scNGMA0GCSqGSIb3DQEBBQUAMIGBMQswCQYDV
```


QQGEwJVUzELMAkGA1UECAwCTUExDzANBgNVBACMBkVjc3RvbjEETMBEGA1UECgwKRXhhbXBsZSBDZzEQMA4GA1UECwwHdGVjaG9wczELMAkGA1UEAwwCY2ExIDAeBgkqhkiG9w0BCQEWENN1cnRzQGV4YW1wbGUuY29tMB4XDTE2MDgxmDEOMDEyNFoXDTE5MDUwNjE0MDEyNFowgYYxCzAJBgNVBAYTAlVTMQswCQYDVQQIDAJNQTETPMA0GA1UEBwwGQm9zdG9uMRMwEQYDVQQKDApFeGFtcGx1IENvMRAwDgYDVQQLDAd0ZWNob3BzMRwDgYDVQQDDAdjBGl1bnQxMSAwHgYJKoZIhvcNAQkBFhFjZXJ0c0BleGFtcGx1LmNvbTCCAIIwDQYJKoZIhvcNAQEBBQADggIPADCCAgocGgIBAKQeDiU/WPZ9OWg2oA6l76IYBMknHH1rccEFEEEDaVbr9PigokVxaiZ5MyLMCuWSwON6gyk3Z376UnAtRnkdwcQHwv7hsvgW6VKMyiNqk0P/mfgdUFmzMHkQEzq3LTUgIWHHPqRGHWOW6a116et2aDTwsbTZdMuEBGOj31KZxv3UrbuIsmgqrHhdTFVjJdsabB1qyocb7t7AEqzsEdymsyneIJD6n0zyxA8+KfGJcy7qL7TpV1NVsV/Zz1GHdcb+VO6yMNNjgSKWbciAh8R9i1+WiJTJ5IEE3zIDV0uI8FMsaKWKXv/oD7w9PfkJnC6qTc5E4sgHPsMnBP2ix/9kBGVxASxGj+5er8MyPAsGeLc7ASlgeMIYieySY3kmHwelOpHgAphgosTDbkhvYIliM1SKyYCM7UidjJ4zBFhNWPAlDnPT8eNThTNhcPLdlH3KoEC22OF9MENEh6Thppz/bjRP5nEhQWr5PLDPaYBgPBkvGYNCT8CVXsdombC3ov5zvH2Q6nyPcbjLz6yTqkxLmPM0Gib2b36wWDSxAcMss37iyjdc40sm4+k2bHibCOLDm6zazLPJ+dqw7sPrlqvCe+q5zf86nQgcrBNVnPUly8u/RBwPv5VmlMkuQMSbxD3VcAPnYtEM2/41LNMrbm7Nz6icDEwURWlHowxe5+xWwbAgMBAAEwDQYJKoZIhvcNAQEFBQADggIBAIq0fsXM/NWlqHecu6EQM18x5JQIECDrruwrRe9Gj7jEdMka1Uz1HwLLnTd6fiyBo0R0Vcdvp96Ic20o5WzK21x5srBz+B604/Oud682ZE6CogQZUSjw4kxP2GBDjnB8wytgeb15QaLLIHCZTS33Wvme6K2cquxMNmeG9CW4ki/LKJpcBf7t3Cb5fbusosB32oyRt8BpKi3476AlhhcFRTse0P3RV1NrIyrwgxq0PY+uEtEz9BQRCT1fZohPK1EUMWYzV3Wj8eboaWohW+Tonr2zuulWhF3bxKe9vd0E3hWoe7bI5I010XldxogFFokr6MVLzXV95q0zuULu1NqCIAwgttzwe9fsXmd+k+rD9YNNW3fXu4WtqVoovwLIUYNDRQENIbyigyhCUzIXo97Xn43vKxxTtyu546QrcfgX59hFjxK0yRxwh2EjuUG10bo2UCes6k6QzBvWrH8gN14mWFYXaqXJJI4o0R5hTIRyVp/lquXYz6pdWNcuR/5Szd/1F0+Z6S4xPbxRZTVolPRJ7FF1J5GaFvkInA4pDskYDvFtfmcy0hdY/0nCJgvYUMtZfMtWYtFKL+4EefjFkl/yqZsNAv/gkP87QwqA+tn1NJ0kfm0I9n28XDlhdU92f0yHjCSscy1o3ewcB7I0smFcUDD4OCYFUINahxAUA0mwiP'),

ca :

atob('MIIFgDCCA2gCCQDQNeSIXshM8DANBgkqhkiG9w0BAQUFADCBgTELMakGA1UEBhMCVVMxZzAJBgNVBAGMAk1BMQ8wDQYDVQQHDAZCb3N0b24xEzARBGNVBAoMCKv4YW1wbGUgQ28xEDA0BgNVBAsMB3R1Y2hvcHMxCzAJBgNVBAMMAMhMSAwHgYJKoZIhvcNAQkBFhFjZXJ0c0BleGFtcGx1LmNvbTAeFw0xNjA4MTAxMzUxMjZaFw0xOTA1MDYxMzUxMjZaMIGBMBQswCQYDVQQGEwJVUzELMAkGA1UECAwCTUExDzANBgNVBACMBkVjc3RvbjEETMBEGA1UECgwKRXhhbXBsZSBDZzEQMA4GA1UECwwHdGVjaG9wczELMAkGA1UEAwwCY2ExIDAeBgkqhkiG9w0BCQEWENN1cnRzQGV4YW1wbGUuY29tMIICIJANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAs0Tdt33C5DJ1fBjBeqbbGDU+ipNNHJeTFU0WC28BUNOPSMcziWqc99LOCHXaW3M66PYJ9qRoHSOPG2dUBdCTFDeRgh32vVrtKGEgIeILJ19k4d0s2Px8fyOgQQybekDR/o2PWxYx1I+SRS+0SvWJVpQEDzjCaQC7RvMD/r98VvK3uiHzisLQVWTARpbkYWU6PFh9S+rD4Mf6fjY6IKotebHKj5CuS+nv71mUv1l4DmGwTn9kf7ZtLgKEFP51JsuuQ4rXuYWmK2HOn92dquKyXwY+rWwHbXmSW9T oxg4AtQutXd4A6AiNmgnj7Gaa69nnXe7ug4wsMDSLqYWG+1z1FGomTTMngq5eXrURAE5yIPD6KhookZSowdzJKE13S6ijbR/qx8HjaBQD0bi8yrvjGOaXaQwU0NBPBsTBEG6xSQDPgx0nv+4YSLrF9CeDNdWIBPrjrc7wv7xkG4U6n8ZNG9N9RkC+J6UJQC3N57X3xefvi8PbqlMom0ERK9phLjIzyNWqtIm018egS2caYhncmuHeulFBlfZEoDggdlN5i0I9QG3Fu9LCE1cziA18q6qbve+sweAH4vVSX/G9ImYfodKwitbDov8lgi+fqaPBK50lqqgT6ub+NonVPIeicfGBY7VZzDTDImgt8ucA1vXiOSng7ZmLfm3fHD3g3ScCAwEAATANBgkqhkiG9w0BAQUFAAOCAgEAs0SAB49Ws0DudLG+xkFUVMXju+K3o8kJCFNXwzHe6djfTr5dKDwl+Tsx2maQR8JMzfuDcUBeQK/sMx2Ps3RmQ1rA4OcRYZPwPJXs7LeREZlmeKJwm4HwOcmnNbuIQ6vqd/TEhc5tpSupxHytez7QERmncnch4PSJngJCAVgGBUr6sVTJCTBX3hhcsR6JXMVoaXS3OPPPMDwnH4oos5MV1uZqa9zsubjlJUGEKKFN7Z0FCvPJXI4XC6pHbxQGz4/LeKSeKvBPHG6sTkwJuQFyHm7Sxf0E0708tX1FKOCvQFPHAAakccoKdGvXyJhOZ23ZK2ooi43pb7Yu706wgUyZ9B1PPvRymKq2KMaowVpFC72JuUlLhkIEBONLhsV6gc+rY9HBx6WP6DATRnm5o5HiNQ0gnEj1Unb+ou0BwKVRbIin+iowmYEXuw9/nze8mpC62jufqTu9Uu5NjMr911A4E4VUIHxk7BQ9rLdDppv49bSMU4ccVzXcjChip/fo1pPpPHCC96En2sXKGxSEv3HVyJUN2n5E/vVcP/9A7nXZi1GLIMD1HHuTqUzs1H6ggtPKWHPLPx0XGiYhoKMWviP41oPJqRJC11vrXQV2Ae1+Giz1Ah6s8dr7ufZ0tI+mv2Rg4VOa+gPd2lI/bMhID2hq2DSeZ+am7HHGQrh15R4OI=') ,

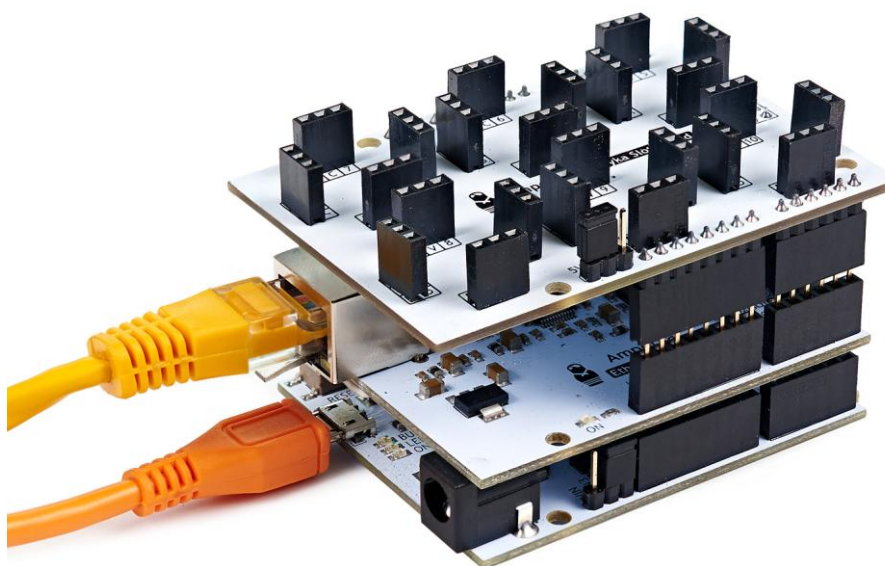
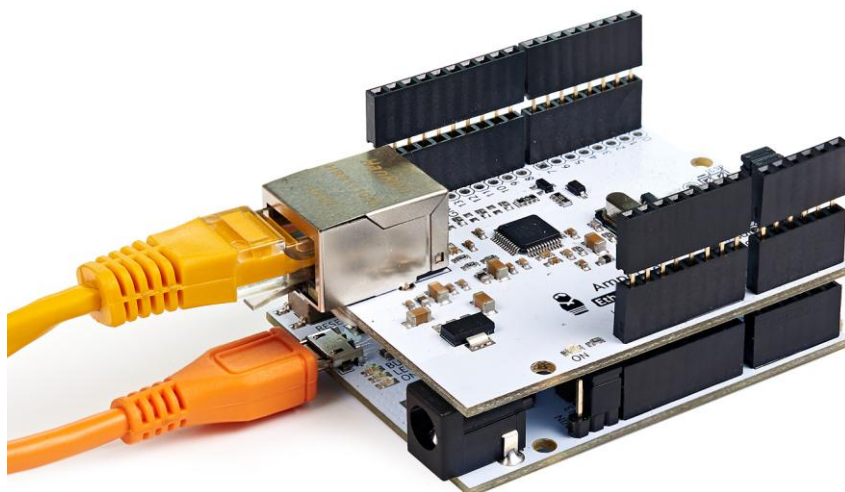
```
// задаем опции подключения к серверу
host : 'api.travis-ci.org',
port : '443',
// Собираем URL-адрес для вызова функции branch
path : '/repos/'+repo+'/branches/'+branch,
protocol : 'https:',
headers : {
  'Accept': 'application/vnd.travis-ci.2+json'
}
```

```

};
// функция запроса данных с сервера Travis CI
var request = function() {
  var response = '';
  http.get(options, function(res) {
    // Большие ответы от сервера могут приходить по частям.
    // В переменной response собираем весь ответ целиком.
    res.on('data', function(data) {
      response += data;
    });
    // После закрытия соединения обрабатываем весь пришедший ответ
    res.on('close', function() {
      // Преобразуем ответ сервера из формата JSON в объект JavaScript
      var answer = JSON.parse(response);
      // проверяем результат преобразования на корректность
      if (answer === undefined) {
        print('answer is undefined.');
```

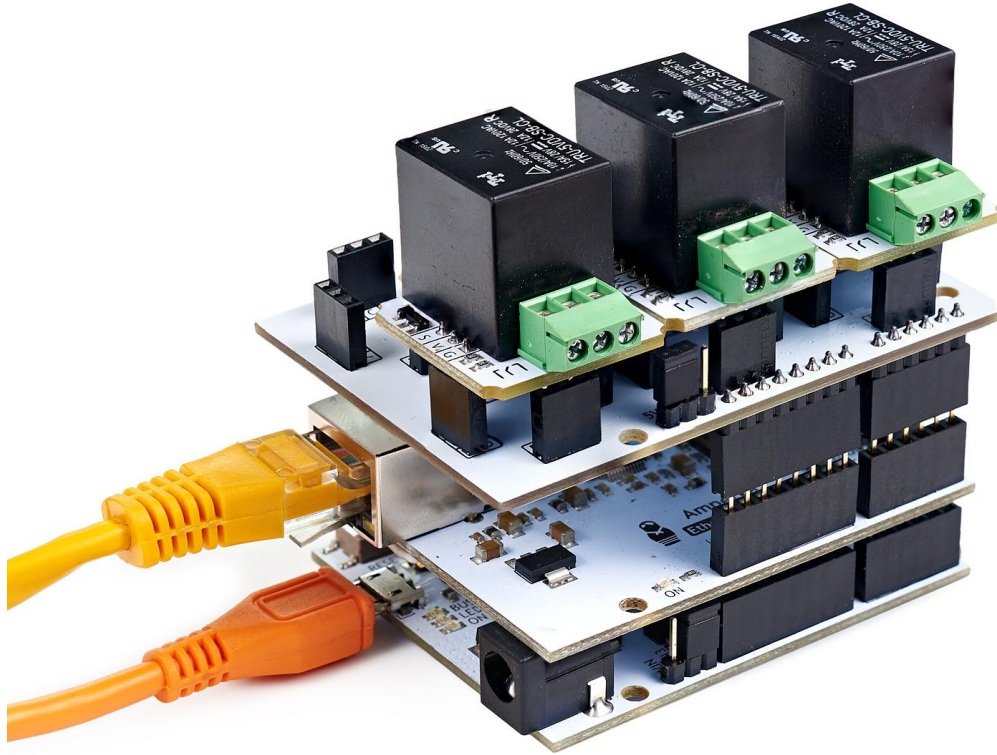
31. Установите Slot Shield сверху на Ethernet Shield.

Разъём RJ-45 на Ethernet Shield довольно высокий, для установки Slot Shield потребуются контактные колодки Arduino. Они приподнимут Slot Shield над Ethernet Shield.

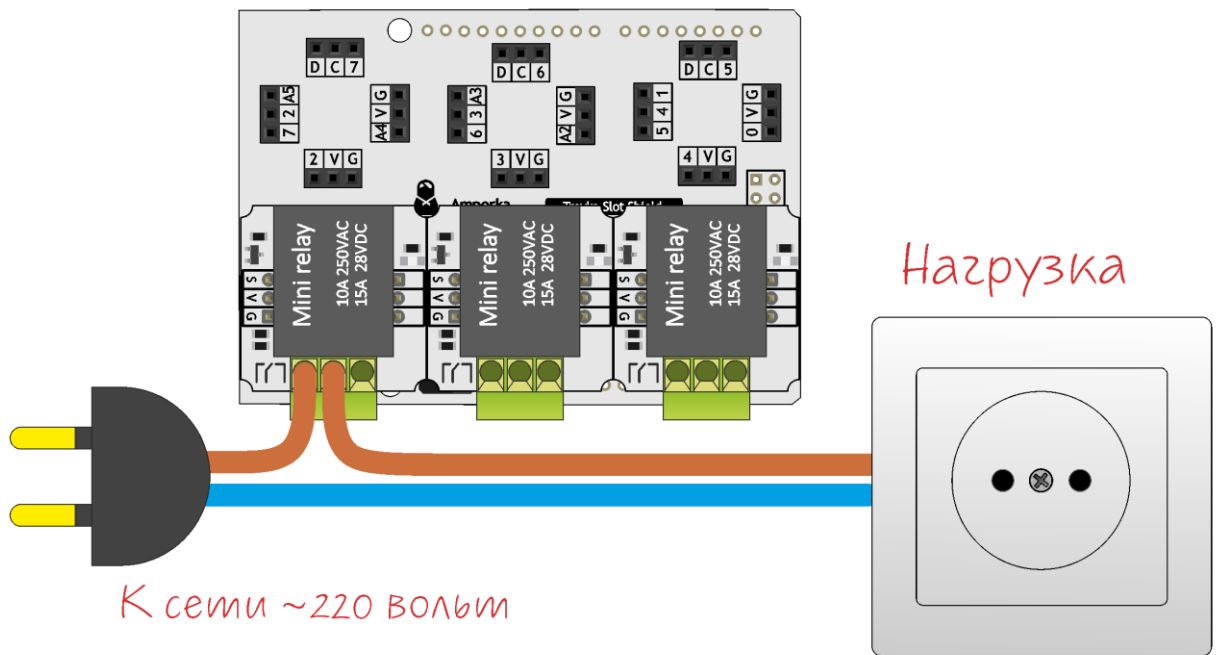


32. Установите на Slot Shield реле. Подключите их к пинам P11, P12 и P13. Реле будут отображать состояние текущего теста. Реле на P12 будет говорить, что сборка в данный момент тестируется (поле state в состоянии created или started). Реле на P11 — тесты провалены (поле state в состоянии failed), а реле на P13 —

успешно пройдены (поле state в состоянии passed).



33. Подключите к реле нагрузку. Мы использовали дорожный светофор. Лампы светофора питаются от напряжения 220 вольт. Если под рукой нет светофора, вы можете взять любой бытовой прибор. Например, подключите чайник к реле на контакте P12 — чайник включится, как только система начнёт тестирование. Тестирование — процесс не быстрый, поэтому кипятик будет очень кстати.



1. Добавим в код управление реле в зависимости от принятого ответа.

ci-trafficlight.js

```
// Настраиваем соединение с Ethernet Shield 2
SPI2.setup({baud: 3200000, mosi: B15, miso: B14, sck: B13});
var eth = require('WIZnet').connect(SPI2, P10);
// Подключаем модуль http.
var http = require('http');

// Подключаем реле
var relay = require('@amperka/relay');
var red = relay.connect(P11);
var yellow = relay.connect(P12);
var green = relay.connect(P13);

// Задаём репозиторий в github и ветку, которую хотим отслеживать
var repo = 'niggor/buildMonitor';
var branch = 'master';
var lastBuild = 0;

// Задаём опции подключения к запрашиваемому серверу
// Поля key, са и cert необходимы для установления зашифрованного
соединения https
var options = {
  key :
atob('MIIJKQIBAAKCAgEApB4OJT9Y9n05aDagDqXvohgEyScfWtxwQUQQNpVuv0+KCiRX
FqLPkzIswK5ZLA43qDKTdnfvpScClGeR3BxAda/uGy+BbpUozKI1CTQ/+Z+B1QWbMweRARm
rctNSAhYcelCsYfBU7DprWXP63ZonPCxtNl0y4QEY6PfUpnG/dStu4iyaqqseF1MVWML2xp
sHWrKhxvu3sASrOwR3KazKd4gkPqfTPLEDz4p8YlZLuovtOlXU1WxX9nOUYd1xv5U7rIw02
OBIPztyICHxH2KX5aIlMnkgR7fMgNXS4jwUyWApYpe/+gPvD09+WmcLqpNzkTiyAc+wycE/
aLH/2QEZXEBLEaP7l6vwzI8CwZ4tzsBKWB4whiJ7JJjeSYfB6U6keACmGcixMNUsg9giWIz
VIRjGIZtSJ2MnjMEWE1Y8CUoc9Px41OFM0dw8t2UfcqgQLbY4X0wQ0SHpOGmnP9uNE/mcSG
pavk8sm9pgGA8GS+BicJPwJVex2iZsLei/n08fZDqfI9xuMvPrJOqTEuY8zQaJvZvfrBYNL
EAKZKzfuLKOJ1zjSybj6TZseJsI4sObrNrMs8n52rDuw+uWq8J76rnN/zqdCBysE1U2lQtj
Ly79EHA+/lUyUyS5AyxvEPdVwA+di0Qzb/jUs0ysGbs3PqJwMTC5FaUejDF7n7FbBsCAwEA
AQKCAgAe/9pWKMgKdW834xGKksdXy8ejojzKbch8TgXxXc6Y74rfJ6BfE5Q1mJCTIw1v3a93
iF2b/G4dKvVFYRftqrQ97dG/USx39gPuZs+Lqh2W+4G+vvK9br5aL7HTkoQEg+QqTwaaqV+
UphMIu0ZBEvKKhVg6LBYQtKOUXUo3NUWSn73M9eA2TXlejiKPxpP/WigAf8lr8+ea/ZnJgn
VqmXgp7fwyRtFu/GJ/17R0JfZhfJtCLqgOnpWkQ/NgeyEJz6vhp+diFJUdwhA/yjmyWrcvk
UE/OIvB5fzZTgRQNHXmfjDE3E5LsLY5DVSIO10Ub/2kg7BwWAMBvhlPH+mZilCbMIOChD5r
9LjHIPjr4jKwPpim3plgtNlca52x+HmVXwkFoU15s3ybvAn0q4OZpjOxZjUrsmhgh/ibzQd
vpCNRdESd08QsBJJf4L35pAo58lYceCQI4vqzFqzOnAIS9we0MUstadsGccUs6n8wVdn9zY
fcF3fGPJ4PmNF4a+n9a/vh/XO06T2DUBfirtAOzwb2xeUqR+T90UpBbAM4D4320NynvhX075
8fCREFT0Zc/P+yR6+xHzJ4jppqszoIPvfKxvZmSSS/L5oNcaEDzweM0+U5iv4GbJHQ1ldzCN
zPJoOhflvsqdVhCSiMgScGNuU3V9lYkMesH28STzC8Mooigt7QQKCAQEA2ZuSfG1B4kNsh9
R26uwLstg61C2bJbM5oAX5SvTMAgxeRkw192dlT86F5bCyvwZs8AjGxSH5t//cs8snsXHDv
IgcH28P5t4b9/+2M/DEDzxWkfgswVVU9JFELgjfqiouxq+fc1X+ZLEYxibzNqNDLsI+Y019us2
S6R7b3oPzWXhdY4sbLdThT9vSi/CwxMCCqFVLjnZHWic7Z/ZIcX0c97I4KyYK1sNUSLa850+
FELoKk+SN7rXOSIqiqFqVWIXSt8hHeHg37Lg15YBFH8f16Bte9G3V0qDyIKVqlcs/dwzJOn
pTxOZgdSy5ChDPS901OyIjHguL6kXQRvlatfxB4788YQKCAQEAwrKM5qsCvUGf+gTV0OGjZ
mGun9G4EEsIbXyh8HoIdI5JsQY0d+sr2jGxP3Bh3sHQGTkWKI/u/yx7KuglfQo6gGHgx0pY
LchZb5u+uJTACgXES/U/G2hbB1OznMDqxolPGuQ6N1lpmFv5/OLqinuh6Nmsjfa59Ev5Scp
gsMhCj+Ifz3X+zirGRhuvQGjyWwAmpNEU0C5xinKVx3SimdkxjOb1RiJ6mwcPp6XqV1A49W
q6f6ircuecE7vpL9PCiFPclH7WrPbu6wDGLQeZd+JHeQ0TGRErmqx0D0b3fxS1Myaa52039S
QAtocxQ7WT/PFSlrzv9UIW05E8Ukhd3Z3fz+wKCAQEAxvDGPdZqgbvYXfms2isAqvmqFs62
92159kVf2togghsIuUX1VIDeLB8IJKw9TgzGzU+IwTLFhSVC1ZDWchAqvZKulTjBffTbJwg
9gBDofYtRLdPwvrOPnqOsS2ljUFUA3OJ17y3fUC3AHkdg0cRG6Hbj7iCtm8v9GTZDeil5TK
q1DJMxR5lHulAdw4/+HtNtVwnlQ5N4aRwssKPCzkq9tRzrbdn6s83stXQQd/FNuFWMOL83n
vQ3rqhZvvthYKmWRZnKrvzeMTnqeBqsQw266sX7ZLvFgHkaD2YoWGN1GKQ7L1jauGUxqfUi
Px42CyCzcEVoHb7Dog9XMtK7PkZYIQKCAQEAjZmOTyqCwszbiGF9c0nZLiBcEwibhw+QcP
rfNPzWZBlIWh0mLKYodkqFs3UaawZdDZDaxK1BN7ZollKDU+wIjImyeLDkyhKN6mxEey+f1
tFYTQHyzEameAQbdmrFl3S83UJIJAX/s8OJCNEQRzZw7PJcWdN1tbViQhTmhpdfusny17Nt
uT2L7rkKEgP3x/YN67EJf659QEeyn7HNjBtv9ovbvWgmAxwtdLgb9c8Pvm8uB27SHWHKY36
csJK/PMMZrgzt/0OsroJnkLhkmQZaTvmMU7XfQ0P0ZwJtOEuW7pnh225yYdtIo8EqAAG/k76
9GSzVfVzI+dzpyE9/mhnzewKCAQB6ZeaPI+k7CsNfyt9j4HX9JKi5ZycsBUDBJ+kJ7xOd1F
a0Y/OcnQh6vLeC5oqtJLB4mLnRvutyK6NP3sI6dLNvsqjVNexl5pSI0/ZlNxNiinFcchl19
```

pnA4znialGLh+IO72OQrwMrEhTSuSBZNcR71+roACIVv5t1A57PlXr+bqv/JIKMwiocFeoW
Dh9NfuhvA5ecv/Q1OS0rsvK1tSFw+dKzMQgE1P8yB9DmyVZ4F16Oyf5dERcRzYLd+hFs4eQ
+LpP11sMhF8TINloptRT4+NdY+ZdJbhBYEriFOPVUcfn4bdkmGezmy9UFg+IArnHB8bXZow
pFwfcieNZuwWNE'),

cert :

atob('MIIFijCCA3KgAwIBAgIJANLNn7E5scNGMA0GCSqGSIB3DQEBBQUAMIGBMQswCQYDV
QQGEwJVUzELMAkGA1UECAwCTUEExDzANBgNVBACMBkJvc3RvbjETMBEGA1UECgwKRXhhbXBs
ZSBDbzEQMA4GA1UECwwHdGVjaG9wczELMAkGA1UEAwwCY2ExIDAeBgkqhkiG9w0BCQEW
EWN1cnRzQGv4YW1wbGUuY29tMB4XDTE2MDgxMDEyNFoXDTE5MDUwNjE0MDEyNFowgYYxCz
AJBgNVBAYTA1VTMzQYDVQQLIDAJNQTU0GA1UEBwwGQm9zdG9uMRMwEQYDVQKDApFe
GFtcGx1IENvMRAwDgYDVQQLDAd0ZWNob3BzMRwDgYDVQDDAdjBGl1bnQxMSAwHgYJKoZI
hvcNAQKBFBhFjZXJ0c0BleGFtcGx1LmNvbTCCAIwDQYJKoZIhvcNAQEBBQADggIPADCCAg
CggIBAKQeDiU/WPZ9OWg2oA6l76IYBMknHH1rccEFEEDaVbr9PigokVxaiZ5MyLMCuWSwON
6gyk3Z376UnAtRnkdwCQHwv7hsvgW6VKMyiNqk0P/mfgdUFmzMHkQEzq3LTUgIWHHPqRgHw
VOW6a116et2adTwsbTzdMuEBGOj31KzXv3UrbuIsmqqrHhdTFVjJdsabB1qyocb7t7AEqzs
EdymsyneIJD6n0zyxA8+KfGJcy7qL7TpV1NVsV/ZzLGHdcb+VO6yMNNjgSKWbciAh8R9il+
WiJTJ5IEE3zIDV0uI8FMsAKWKXv/oD7w9PfkJnC6qTc5E4sgHPsMnBP2ix/9kBGVxASxGj+
5er8MyPAsGeLc7ASlgeMIYieySY3kmHwe1OpHgAphgosTDbkhvYIliM1SKyYCM7UidjJ4zB
FhNWPAlDnPT8eThTNHcPLdlH3KoEC22OF9MENEh6Thppz/bjRP5nEhQwR5PLDPaYBgPBkv
gYnCT8CVXsdombC3ov5zvH2Q6nyPcbjLz6yTqkxLmPM0Gib2b36wWDSxAcMss37iyjdc40
sm4+k2bHibCOLDM6zazLPJ+dqw7sPr1qvCe+q5zf86nQgcrBNVnpULYy8u/RBwPv5VM1mku
QMSbx3D3VcAPnYtEM2/41LNMrbm7Nz6icDEwUwRlHowxe5+xWwbAgMBAAEDQYJKoZIhvcNA
QEFBQADGgIBAIq0fsXM/NWlqHecu6EQM18x5JQieCdrurwrRe9Gj7JedMka1Uz1HwLLnTd6
fiyBo0R0Vcdvp96Ic20o5WzK21x5srBz+B604/Oud682ZE6CogQZUSjw4kxP2GBDjnB8wyt
gebl5QaLLIHCZTS33WvmE6K2cquxMNMzE9CW4ki/LKJpcBf7t3Cb5fbusosB32oyRt8BpK
i3476A1hhcFRtse0P3RV1NrIyrwgxq0PY+uEtEz9BQRCT1fZohPK1EUMWYzV3Wj8eboaWoh
W+Tonr2zuu1WhF3bxKe9vd0E3hWoe7bI5I010X1DxogFFokr6MVLzXV95q0zuULu1NqcIAw
gttze9fsXmd+k+rD9YNNW3fXu4WtqVoovwLIUYNDRQENIiByigyhCUzIXo97Xn43vKxxTt
yu546QrcfgX59hFjxK0yRxwh2EjuUG10bo2UCes6k6QzBvWrH8gN14mWFYXaqXJJI4o0R5h
TIRyVp/lquXYZ6pdWNcuR/5SzD/1F0+Z6SJ4xPbxRZTVolPRJ7FF1J5GaFvkinA4pDskYDv
Ftfmcy0hdY/0nCJgvYUMtZfMtWYtFKL+4EfjFk1/yqZsNav/gkP87QQwqA+tn1NJ0kfm0I9
n28XDLhdU92f0yHjCSylo3ewcB7I0smFcUDD4OCYFUINahxAUA0mwiP'),

ca :

atob('MIIFgDCCA2gCCQDQNeSIxshM8DANBgkqhkiG9w0BAQUFADCBgTELMAkGA1UEBhMCV
VMxChAJBgNVBAGMAk1BMQ8wDQYDVQHDZCb3N0b24xExARBgNVBAoMckV4YW1wbGUgQ28x
EDA0BgNVBAsMB3R1Y2hvcHMxChAJBgNVBAMMAMNhMSAwHgYJKoZIhvcNAQKBFBhFjZXJ0c0B
leGFtcGx1LmNvbTAeFw0xNjA4MTAxMzUxMjZaFw0xOTA1MDYxMzUxMjZaMIGBMQswCQYDVQ
QGEwJVUzELMAkGA1UECAwCTUEExDzANBgNVBACMBkJvc3RvbjETMBEGA1UECgwKRXhhbXBs
ZSBDbzEQMA4GA1UECwwHdGVjaG9wczELMAkGA1UEAwwCY2ExIDAeBgkqhkiG9w0BCQEW
EWN1cnRzQGv4YW1wbGUuY29tMIICiANBgkqhkiG9w0BAQEFAAOCAg8AMIICgKCAgEAs0Tdt33
C5Dj1FbJBeqbbGDU+ipNNHJeTFU0WC28BUNOPSMCziWqc99LOCHXaW3M66PYJ9qRoHsoPG2
dUBdCTFDeRgh32vVrtKGEgIeLLJ19k4d0s2Px8fyOgQYqybekDR/o2PWxYx1I+SRs+0SvwJV
PqEDzjCaQC7RVMD/r98VvK3uiHzis1QVWTARpbkYWU6PFh9S+rD4Mf6fjY6IKotebHKj5Cu
S+nv71mUv114DmGwTn9kf7ZtLgKEFP51JsuuQ4rXuYwMk2HOn92dquKyXwY+rWwHbXmSW9T
oxg4AtQutXd4A6AiNmgNj7Gaa69nnXe7ug4wsMDSLqYWG+1z1FGomTmngq5eXrURAE5yIP
D6KhookZSowdzJKE13S6ijbR/qx8HjaBQD0bi8yrvjGOaXaQwU0NBPBsTBEg6xSQDPgx0nv
+4YSLrF9CeDNdWIBPrjrc7wv7xkG4U6n8ZNG9N9RkC+J6UJQ3C9N57X3xefvi8PbqlMom0
ERK9phLjIzyNWqtImO18egS2caYhncmuHeulFB1fZEoDggdlnL5i0I9QG3Fu9LCE1czia1
8q6qbve+sweAH4vVSX/G9ImYfodKWitbDov8lgI+fqapBK50lqqgT6ub+NonVPIeicfGBy7
VZzDTDImgt8ucA1vXiOSng7ZmLfm3fHD3g3ScCAwEAATANBgkqhkiG9w0BAQUFAAOCAgEAs
0SAB49Ws0DudLG+xkFUvMXju+K3o8kJCFNXwzHe6djfTr5dKDwl+Tsx2maQR8JMzfuDcUBe
QK/sMx2Ps3RmQ1rA40cRYZPwPJXs7LeREZ1mEKJwm4HwOcmnNbuIQ6vqd/TEhc5tpSUpXH
ytez7QERmncnch4PSJngJCAVgGBUr6sVTJCTBX3hhcsR6JXMVoaXS3OPpPdwnH4ooS5MV1
uZqa9zsubjLJUGEKKFN7Z0FCvPJXI4XC6pHbxQGz4/LeKSeKvBPHG6sTkWJuQFyHm7SXf0E
0708tX1FKOCvQFPAAakccoKDgVxYJhOZ23ZK2ooi43pb7Yu706wgUyZ9B1PPvRymKq2KMa
oWvPFC72JuU1LhkIEBONLhsV6gc+rY9HBx6WP6DATRnm5o5HiNQ0gnEj1Unb+ou0BwkVRbI
in+iowmYEXuw9/nze8mpC62jufqTu9Uu5NjMr911A4E4VUIHxk7BQ9rLdDppv49bSMU4ccV
zXcjCbiP/folpPFPHC96En2sXGxSEv3HVyJN2n5E/vCp/9A7nXZi1GILMD1HHuTqUzs
1H6qgtPKWHPLPx0XGiYhoKMWviP41oPjQRJc11vrXQV2Ae1+Giz1Ah6s8dr7ufZ0ti+mv2
Rg4VOa+gPd21I/bMhID2hq2DSeZ+am7HHGQrh15R40I='),

// задаем опции подключения к серверу

host : 'api.travis-ci.org',

port : '443',

```

    path : '/repos/'+repo+'/branches/'+branch,
    protocol : 'https:',
    headers : {
      'Accept': 'application/vnd.travis-ci.2+json'
    }
  };

  // Функция включения реле в зависимости от состояния state
  // blink(5) включает реле на 5 секунд, а затем автоматически выключает
  его
  var lightUp = function(state) {
    switch(state) {
      case 'failed': red.blink(5); break;
      case 'passed': reen.blink(5); break;
      case 'created': yellow.blink(5); break;
      case 'started': yellow.blink(5); break;
      default: yellow.blink(5);
    }
  };

  // функция запроса данных с сервера Travis CI
  var request = function() {
    var response = '';

    http.get(options, function(res) {
      // Большие ответы от сервера могут приходить по частям.
      // В переменной response собираем весь ответ целиком.
      res.on('data', function(data) {
        response += data;
      });
      // После закрытия соединения обрабатываем весь пришедший ответ
      res.on('close', function(noError) {
        // Преобразуем ответ сервера из формата JSON в объект JavaScript
        var answer = JSON.parse(response);
        // проверяем результат преобразования на корректность
        if (answer !== undefined) {
          // Получаем номер и статус последнего теста
          var build = answer.branch.number;
          var state = answer.branch.state;
          // Если текущий тест в процессе выполнения,
          // ещё успеваем отобразить его статус
          if (state === 'created' || state === 'started') {
            lightUp(state);
          } else {
            // Если появились свежие выполненные тесты,
            // на которые мы ещё не реагировали.
            if (lastBuild !== build) {
              // Если lastBuild равен нулю, светофор
              // только что включили в питание, а результаты
              // тестов уже известны без виджета.
              if (lastBuild !== 0) {
                lightUp(state);
              }
              // Запоминаем номер последнего теста,
              // чтобы больше на него не реагировать.
              lastBuild = build;
            }
          }
        }
      });
      // Повторяем запрос к Travis-ci через 5 секунд
      setTimeout(request, 5000);
    });
  };
};

```

```
};  
// Получаем IP-адрес от DHCP-сервера  
eth.setIP();  
// Делаем первый запрос к сервису  
request();
```

Хак

Библиотека `wiznet` и `https`-соединение выбрасывают в консоль много отладочной информации. При запуске `Iskra JS` без подсоединения к консоли, код выполняться не будет, так как вывод в консоль замораживает выполнение программы, пока компьютер не запросит новую порцию данных. Чтобы запустить код, необходимо в первой строке программы прописать `Serial3.setConsole(true);`. Эта строка заставит `Iskra JS` выводить информацию в консоль через последовательный интерфейс на пинах `P0` и `P1`. Замкните эти пины проводком, чтобы информация в консоль могла передаваться, не заполняя буфер последовательного интерфейса.

