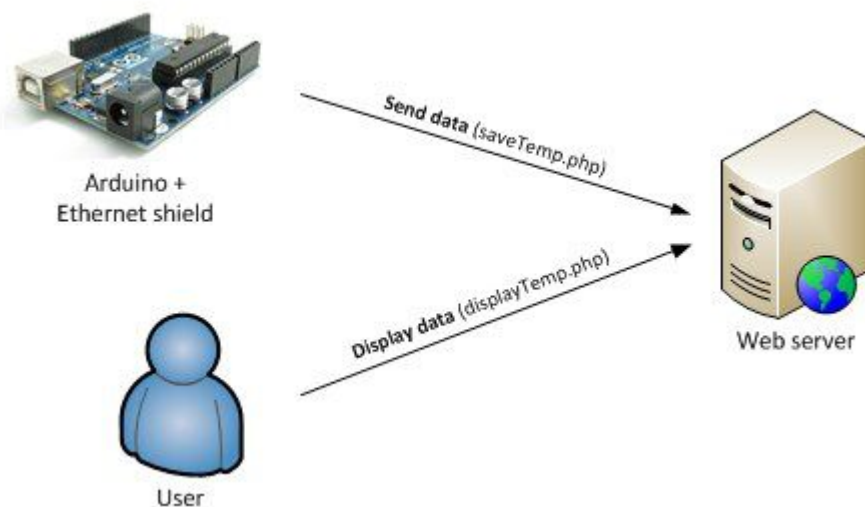


在本教程中，我将介绍如何开发一个小网站从 Arduino 的（使用 Enc28j60 盾）获取数据，存储并显示出图标曲线。

逻辑架构

您的网站将会由两页：

- saveTemp.php，从 Arduino 获取数据，并将其存储在文本文件；
- displayTemp.php，读取文本文件，并显示数据。



在这个例子中，Arduino 使用 SD18B20 传感器获取房间的温度，我已经在博客的[这篇文章](#)介绍了该传感器。

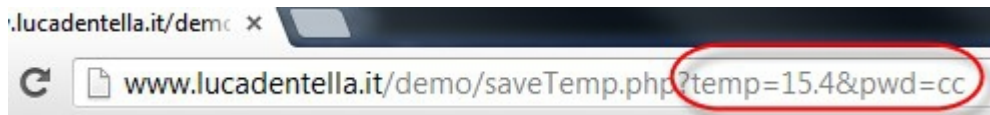
对于网页编程，我选择了在网络上广泛使用 PHP。

注意：为了保持简单的例子，数据被存储在一个 CSV（逗号分隔值）文件，在实际应用中，最好使用一个数据库，例如 MySQL。

发送数据

我使用的 HTTP 协议的 GET 方法将数据发送到 Web 服务器。这种方法请求的数据会附在 URL 之后（就是把数据放置在 HTTP 协议头中），以?分割 URL 和传输数据，参数之间以&相连，如：

login.php?name=hyddd&password=idontknow&verify=%E4%BD%A0%E5%A5%BD。如果数据是英文字母/数字，原样发送，如果是空格，转换为+，如果是中文/其他字符，则直接把字符串用 BASE64 加密，得出如：%E4%BD%A0%E5%A5%BD，其中%XX 中的 XX 为该符号以 16 进制表示的 ASCII。



为了防止其他用户发送数据，除了温度参数，你也必须发送到该网页的密码，使用 `pwd` 参数。

我的代码读取温度传感器，将其转换成字符串类型，并把它“print”到一个 `Stash` 对象：

```
1. sensors.requestTemperatures();
2. float float_temp = sensors.getTempCByIndex(0);
3.
4. char string_temp[7];
5. dtostrf(float_temp, 4, 2, string_temp);
6.
7. Stash stash;
8. byte sd = stash.create();
9. stash.print(string_temp);
10. stash.save();
```

然后在另外一个 `Stash` 对象中准备请求（`Http Get` 请求），并用 `tcpSend()` 方法发送到服务器：

```
1. Stash::prepare(PSTR("GET          /demo/saveTemp.php?temp=$H&pwd=$F
HTTP/1.0" "\r\n"
2.   "Host: $F" "\r\n" "\r\n"),
3.   sd, password, website);
4. session_id = ether.tcpSend();
```

由于会话标识符（`SESSION_ID`），你可以得到一个来自服务器的响应，并检查它是否包含错误消息（“KO”）：

```
1. const char* reply = ether.tcpReply(session_id);
2. [...]
3. if(strstr(reply, "KO - ") != 0) Serial.println(strstr(reply, "KO - "));
```

在 `PHP` 中，你可以通过关联数组 `$_GET` 获得这些 `HTTP GET` 发送过来的数据：

```
1. $temp = $_GET["temp"];
2. $pwd = $_GET["pwd"];
```

验证密码后，温度值以及对应的时间戳一起存储在文件中：

```
1. define("LOG_FILE", "./temperatures.csv");
```

```

2. [...]
3. $file_handler = fopen(LOG_FILE, "a+");
4. fwrite($file_handler, time() . "," . $temp . "\n");

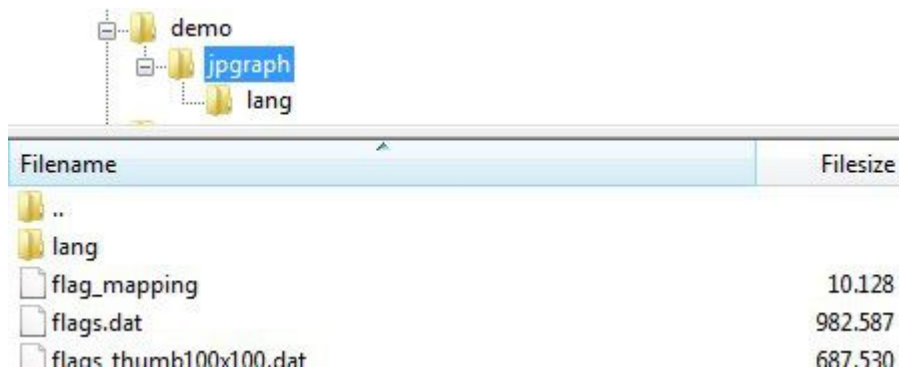
```

旁白请忽略，版主如果觉得不合适可以删除：哎要不要给我的[维基](#)拉点流量嘞。。。算了，还是让各位看官直接在这儿爽完吧。。。因为我饱受那样的痛苦。。。从此直接忽略掉这类文章。。。

数据显示

为了使这个例子更有趣，我决定把这些数据显示在图表上。

对于 PHP，有一个很好的开源的图表库可供选择：[JpGraph](#)。要使用它，你必须把库文件上传到您的网站的子文件夹中，通常为“/JpGraph”：



在 PHP 代码中，你就必须包括需要的库，这取决于你要创建的图表类型：

```

1. require_once('jpgraph/jpgraph.php');
2. require_once('jpgraph/jpgraph_line.php');
3.

```

JpGraph 库不负责获取图表所需的数据而是通过 PHP 代码，打开文件 - 按行 - 把数据保存到两个数组中：

```

1. $file_lines = file(LOG_FILE,
    FILE_IGNORE_NEW_LINES|FILE_SKIP_EMPTY_LINES);
2. foreach($file_lines as $line_num => $line_value) {
3.     $line_elements = explode(",", $line_value);
4.     $times[] = date("H:i:s", $line_elements[0]);
5.     $values[] = $line_elements[1];
6. }

```

JpGraph 可以为每组数据创建许多不同类型的图表，你可以做很多的自定义：有关使用该库的例子和教程，请参考[官方手册](#)。

运行

Arduino 的草图和 PHP 网页的源代码可以在 [GitHub](#) 上我的[存储库](#)中下载，这里有两个截图显示运行状态：

