



Středoškolská technika 2015

Setkání a prezentace prací středoškolských studentů na ČVUT

WiPower

IoT zásuvka

Matouš Hýbl

Klvaňovo gymnázium Kyjov

[Úvod](#)
[ESP8266](#)
[NodeMCU](#)
[Mechanická konstrukce](#)
[Elektronika](#)
[Firmware](#)
 [init.lua](#)
 [wipower.lua](#)
[Android aplikace](#)
[Závěr](#)
[Zdroje](#)

Úvod

Internet of Things, česky internet věcí je jedním ze nejdiskutovanějších technologických zaměření současnosti. Jedná se o možnost propojení zařízení mezi sebou v rámci internetu, které přinese integraci mnoha senzorů a výstupních zařízení do běžného života. Díky internetu věcí bude například možné, aby uživatel mimo domov pomocí svého mobilního telefonu zjistil, jaké je teplota vzduchu u něj doma a na základě těchto informací si případně přitopil nebo naopak zapnul klimatizaci. Internet věcí by měl přinést také revoluci v zabezpečovacích systémech - například detektor požáru bude schopen odeslat informace o požáru okamžitě svému majiteli a hasičským jednotkám, což zajistí mnohem větší efektivitu. O důležitosti tohoto odvětví elektroniky a informatiky svědčí také fakt, že většina velkých technologických společností vynakládá nemalé zdroje na výzkum v této oblasti.

WiPower sice není žádným takovým převratným zařízením, jedná se ale o mezikrok, který bylo potřeba udělat, než dojde k vytvoření něčeho mnohem užitečnějšího v amatérských podmínkách. WiPower je přes WiFi ovládaná zásuvka, kterou lze ovládat jak přes webové rozhraní tak pomocí aplikace v mobilním telefonu. Zařízení je vystavěno na platformě NodeMCU a SoC ESP8266.

ESP8266

ESP8266 je SoC (System on Chip), které obsahuje programovatelný WiFi modul, který lze používat jak v AP módu, tak v módu připojení k již existující WiFi. Tento modul se objevil na konci roku 2014 a znamenal revoluci v IoT pro amatéry, vzhledem ke své ceně, velikosti paměti, a celkové vybavenosti - například obsahuje rozhraní I²C, UART, 1-Wire a další. Pro tento modul bylo v komunitě vytvořeno několik kompilátorů pro spoustu programovacích jazyků např. pro Lua nebo Python. Modul je k dostání zejména na e-Bay a AliExpressu, kde je k dostání v konfiguraci s různými podpůrnými moduly - například s USB portem, kterým lze modul jednodušeji programovat, protože v základu jej lze programovat pouze přes UART rozhraní.



NodeMCU

NodeMCU je jedním z rozšiřujících modulů pro ESP8266 vhodný zejména pro prototypovou výrobu, zejména pro jeho snadnou montáž do nepájivého kontaktního pole. Hlavní výhodou tohoto modulu je převodník UART <-> USB, díky kterému je programování SoC velmi snadné. Modul je dodáván s firmwarem NodeMCU, který je naprogramovaný tak, že obsahuje interpreter jazyka Lua. NodeMCU má také integrovaný převodník z 5V na 3V, díky kterému jej lze napájet přímo z USB. Modul obsahuje také jedno programovatelné tlačítko.



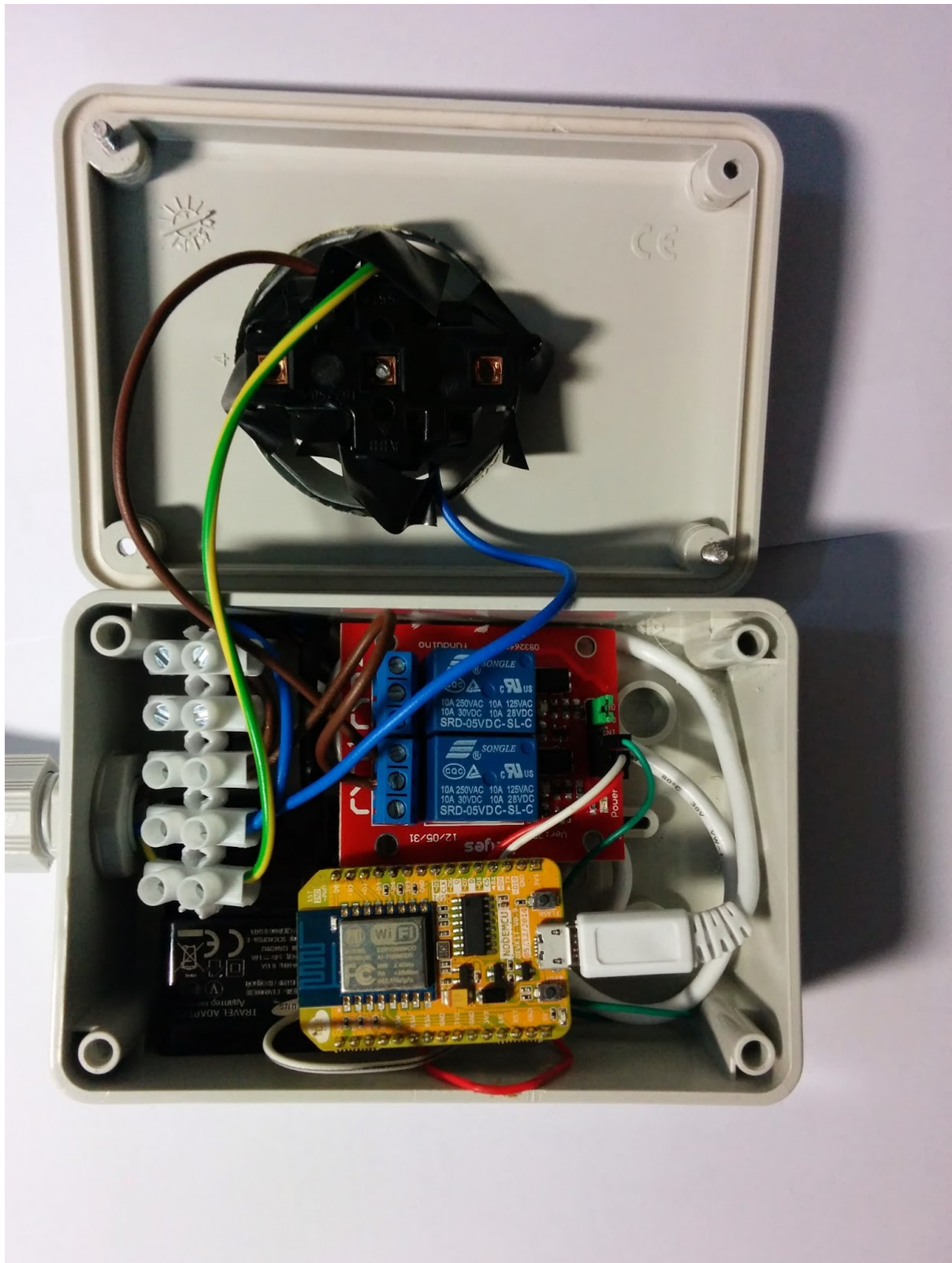
Mechanická konstrukce

Celé zařízení je vestavěno do elektroinstalční krabičky o rozměrech přibližně 130x85x57 mm, na které je namontována standardní zásuvka. Do krabičky byla také provrtána díra, kterou byla vedena průchodka, kterou vede kabel, sloužící k připojení zařízení do elektrické sítě. Dále bylo vyvrtáno několik otvorů pro kontrolní LED a otvor pro konektor externího přepínače.



Elektronika

Z elektronického hlediska se zařízení skládá z řídicí jednotky NodeMCU, dvojitého relé modulu, nabíjecího adaptéru pro mobilní telefon, jakožto napájení pro řídicí systém.



Firmware

Firmware řídicí jednotky NodeMCU byl naprogramován v programovací jazyce Lua a skládá se ze dvou základních částí - **init.lua** souboru, který je řídicí jednotkou spuštěn při startu a stará se o korektní spuštění dalších skriptů, konkrétně skriptu **wipower.lua**, který má na starost obsluhu událostí změny stavu relé.

init.lua

Init.lua je skript, který po startu řídicí jednotky připojuje jednotku k síti WiFi, která je předem definovaná v tomto souboru, zároveň přiřazuje jednotce statickou IP adresu.

```
print(wifi.sta.getip())
wifi.setmode(wifi.STATION)
wifi.sta.config("SSID", "PASSWD")
print(wifi.sta.getip())
cfg = {ip="192.168.1.60",
       netmask="255.255.255.0",
       gateway="192.168.1.1"}
wifi.sta.setip(cfg)
wifi.sta.connect()
tmr.alarm(1, 1000, 1, function()
  if wifi.sta.getip()== nil then
    print("IP unavaiable, Waiting...")
  else
    tmr.stop(1)
    print("Config done, IP is "..wifi.sta.getip())
    dofile("led_over_ethernet.lua")
  end
end)
```

wipower.lua

Tento skript je implementací obsluhy HTTP GET requestů na dvou portech - na portu 80, který slouží pro standardní webové rozhraní, a na portu 666 který slouží jako API entrypoint pro aplikace. Instrukce pro činnost, kterou má řídicí jednotka provést je předávána jako parametr GET requestu. Při přístupu na port 666 odpoví řídicí jednotka pomocí HTTP response OK. Řídicí jednotka obsluhuje tři typy paramtetrů - **/?LED=0**, který slouží pro vypnutí relé, **/?LED=1**, který slouží k sepnutí relé a parametr **/status**, který ve svém response vrací true, nebo false podle toho, zda-li je relé sepnuto, nebo ne.


```

ledPin = 1
gpio.mode(ledPin, gpio.OUTPUT)
gpio.write(ledPin, gpio.LOW)
srv=net.createServer(net.TCP)
srv:listen(80,function(conn)
  conn:on("receive",function(conn,payload)
    print(payload)
    head = string.sub(payload, 0, string.find(payload, "\n"))
    print(head)
    if string.find(head, "?LED=1") then
      gpio.write(ledPin,gpio.HIGH)
      print("on")
    else
      if string.find(head, "?LED=0") then
        gpio.write(ledPin, gpio.LOW)
        print("off")
      end
    end
    conn:send("<h1> Hello, NodeMcu.</h1>")
    conn:send("<a href=?LED=1?>Turn LED on</a><br/>")
    conn:send("<a href=?LED=0?>Turn LED off</a><br/>")

    conn:on("sent", function(conn) conn:close() end)
  end)
end)
srv:listen(666, function(conn)
  conn:on("receive",function(conn,payload)
    print(payload)
    head = string.sub(payload, 0, string.find(payload, "\n"))
    print(head)
    if string.find(head, "/?LED=1") then
      gpio.write(ledPin,gpio.HIGH)
      print("on")
      conn:send("HTTP/1.0 200 OK\r\n\r\nset")
    else
      if string.find(head, "/?LED=0") then
        gpio.write(ledPin, gpio.LOW)
        print("off")
        conn:send("HTTP/1.0 200 OK\r\n\r\nset")
      else
        if string.find(head, "/status") then
          conn:send("HTTP/1.0 200 OK\r\n\r\n" .. ((gpio.read(ledPin) == 1) and "true"
or "false"))
        end
      end
    end
    conn:on("sent", function(conn) conn:close() end)
  end)
end)

```

Android aplikace

Android aplikace WiPower je jednoaktivitová aplikace obsahující pouze jedno tlačítko na přepnutí stavu relé. Aplikace je vystavěna na knihovně Retrofit, která byla navržena pro jednoduchou implementaci REST API. Deklarace API vypadá takto:

```
package com.matoushybl.wipower.api;

import retrofit.Callback;
import retrofit.http.GET;
import retrofit.http.Query;

/**
 * @author <a href="mailto:hyblmatous@gmail.com">Matous Hybl</a>
 */
public interface WiPowerService {

    @GET("/")
    void toggle(@Query("LED") int state, Callback<String> callback);

    @GET("/status")
    void status(Callback<String> callback);
}
```

Pro přepnutí stavu WiPower slouží tento kód:

```
application.getPowerService().toggle(toggleButton.isChecked() ? 1 : 0, new
Callback<String>() {
    @Override
    public void success(String s, Response response) {
        // FIXME this should change the text of the button
    }

    @Override
    public void failure(RetrofitError error) {
        Toast.makeText(MainActivity.this, R.string.failMessage,
Toast.LENGTH_SHORT).show();
    }
});
```

Aplikace si po každém startu, případně obnovení (onResume) zjistí, v jakém stavu je momentálně WiPower a podle toho nastaví počáteční stav přepínacího tlačítka.



WiPower



Závěr

Cílem stavby tohoto zařízení bylo otestovat možnosti IoT s SoC ESP8266 a NodeMCU, zařízení není ani zdaleka hotové a je očekáván větší vývoj v nejbližší době. Osobně mohu ESP8266 doporučit všem, kteří chtějí propojit svá zařízení s internetem bezdrátově, protože jeho konfigurace je extrémně snadná a použití ještě snazší. Myslím také, že se v budoucnosti objeví čím dál více podporovaných programovacích jazyků, knihoven a API, které umožní s tímto SoC vyvíjet čím dál pokročilejší zařízení, a doufám, že bude možné jej propojit s čím dál více zařízeními. Zároveň bych chtěl apelovat na společnosti zabývající se výrobou a výzkumem IoT, aby se spojili a vydali jednotnou specifikaci komunikačního protokolu, díky kterému by IoT zařízení bylo možné bezproblémově integrovat se službami, které již existují, čímž mám na mysli například Google Now.

Zdroje

- NODEMCU. 2015. *Nodemcu-firmware* [online]. [cit. 2015-05-09]. Dostupné z: <https://github.com/nodemcu/nodemcu-firmware>
- SQUARE. 2013. *Retrofit* [online]. [cit. 2015-05-09]. Dostupné z: <http://square.github.io/retrofit/>