



Střední průmyslová škola a Vyšší odborná škola, Písek, Karla Čapka 402, Písek

18-20-M/01 Informační technologie

## Maturitní práce

# Dálkové ovládání zásuvek NETIO

Téma číslo 12

autor:

**Milan Jiříček, B4.I**

vedoucí maturitní práce:

**Ing. Břetislav Bakala**

Písek 2020/2021

## **Anotace**

Maturitní práce se zaměřuje na porovnání platforem ESP8266 a ESP32. Cílem je vytvořit ovladač pro ovládání zásuvek značky NETIO s webovou aplikací pro konfiguraci a zjistit, která platforma je vhodná pro realizaci funkčního vzorku z hlediska spotřeby energie a reakční doby.

## **Annotation**

The graduation thesis focuses on the comparison of the ESP8266 and ESP32 platforms. The goal is to create a driver for controlling NETIO sockets with a web application for configuration and to find out which platform is suitable for the implementation of a functional sample in terms of energy consumption and response time.

## Poděkování

Chtěl bych poděkovat panu učiteli Ing. Břetislavovi Bakalovi za odborné vedení práce a cenné rady, které mi pomohly tuto práci zkompletovat. Rád bych také poděkoval technickému řediteli Ing. Břetislavovi Bakalovi ml. společnosti NETIO products a.s. za cenné rady, věcné připomínky a vstřícnost při konzultacích a vypracování bakalářské práce. V neposlední řadě chci poděkovat Mgr. Haně Maříkové a Mgr. Vladimíře Špirhanzlové za pomoc při gramatické a stylistické kontrole.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>5</b>
<b>2</b>	<b>Základní informace</b>	<b>6</b>
2.1	Zásuvka NETIO . . . . .	6
2.2	Platforma ESP . . . . .	6
2.2.1	ESP8266 . . . . .	6
2.2.2	ESP32 . . . . .	8
2.3	Komunikace mezi ESP a NETIO zásuvkou . . . . .	8
<b>3</b>	<b>Tvorba webové stránky</b>	<b>9</b>
<b>4</b>	<b>Měření spotřeby a času</b>	<b>10</b>
4.1	ESP8266 . . . . .	10
4.1.1	Spotřeba ustálených stavů . . . . .	10
4.1.2	Reakční čas jednotlivých situací . . . . .	13
4.1.3	Rychlost WiFi připojení . . . . .	14
4.1.4	Rychlost odeslání HTTP requestu . . . . .	16
4.1.5	Spotřeba jednotlivých operací . . . . .	17
4.2	ESP32 . . . . .	19
4.2.1	Spotřeba ustálených stavů . . . . .	19
4.2.2	Reakční čas jednotlivých situací . . . . .	19
4.3	Porovnání získaných výsledků . . . . .	19
4.3.1	Reakční časy stavů . . . . .	19
<b>5</b>	<b>Vytvoření funkčních vzorků</b>	<b>21</b>
<b>6</b>	<b>Závěr</b>	<b>22</b>
	<b>Přílohy</b>	<b>24</b>



# Kapitola 1

## Úvod

# Kapitola 2

## Základní informace

### 2.1 Zásuvka NETIO

### 2.2 Platforma ESP

ESP jsou rodina mikročipů od společnosti **Espressif Systems** z Čínské Shangaje.

#### 2.2.1 ESP8266

##### Historie

ESP8266 je levný mikročip, který umí využívat WiFi. První chip, který se dostal na světlo světa byl v modulu **ESP-01**. Tento modul dokázal připojit se na WiFi síť a provádět jednoduché TCP/IP spojení. Získal si velkou oblibu u skupinek hackerů díky nízké ceně. Jsou vhodné pro IoT jako například automatizace, zabezpečení, chytré domy atd.

##### Specifikace

Pro tuto maturitní práci bude použit modul **WT8266-S1**, který je vytvořen společností **Wireless-Tag**. Je založen na mikročipu ESP8266.

ESP8266 integruje vylepšenou verzi procesoru **L106 Diamond series 32-bit** vytvořený firmou **Tensilica** s podporou frekvencí 80 MHz a 160 MHz a RTOS<sup>1</sup>. K dispozici je 36 KB RAM a 16 Mbit Flash paměti. Integrovaný v systému je také 10 bitový analog-digitální převodník. Modul podporuje standard **IEEE802.11 b/g/n**<sup>2</sup> a sadu protokolů TCP/IP. WiFi 2,4 GHz umožňuje WPA/WPA2<sup>3</sup> a ke komunikaci využívá anténu PCB. Zařízení

---

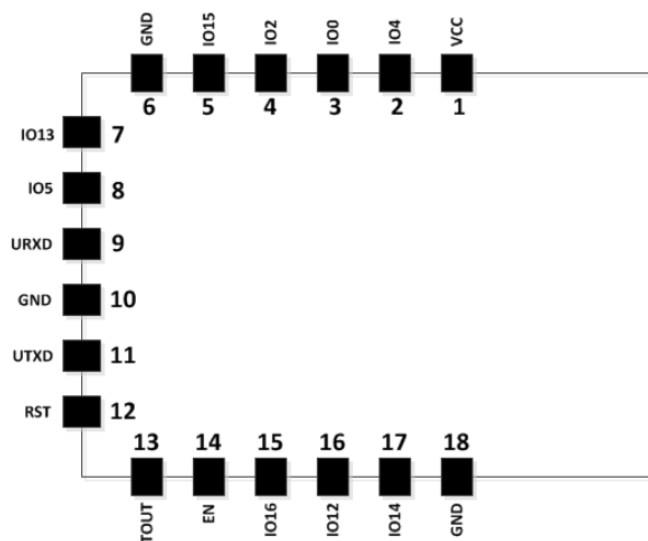
<sup>1</sup>Operační systém v reálném čase

<sup>2</sup>Standard pro lokální bezdrátové sítě

<sup>3</sup>Chráněný přístup k WiFi

má 16 GPIO pinů<sup>4</sup> (viz. obr. 2.1). Také obsahuje:

- **UART** - univerzální asynchronní přijímač-vysílač pro sériový přenos
- **I<sup>2</sup>C** - seriová sběrnice o dvou vodičích pro nízkorychlostní zařízení
- **I<sup>2</sup>S** - seriová sběrnice pro digitální audio
- **SPI** - seriové periferní rozhraní pro komunikaci mezi mikroprocesorem a integrovanými obvody
- **rozhraní HSPI** - externí SPI pro připojení displeje či externí Flash
- **PWM rozhraní** - 4 kanálová pulzně šířková modulace pro přenos analogového signálu pomocí dvouhodnotného signálu



Obrázek 2.1: ESP8266 pinout

## Deep sleep

Platforma ESP lze také uvést do úsporného režimu. Jednoduše to znamená vypnutí nejdůležitějších částí pro ustálený stav. ESP8266 celkově nabízí 3 druhy úsporných režimů.

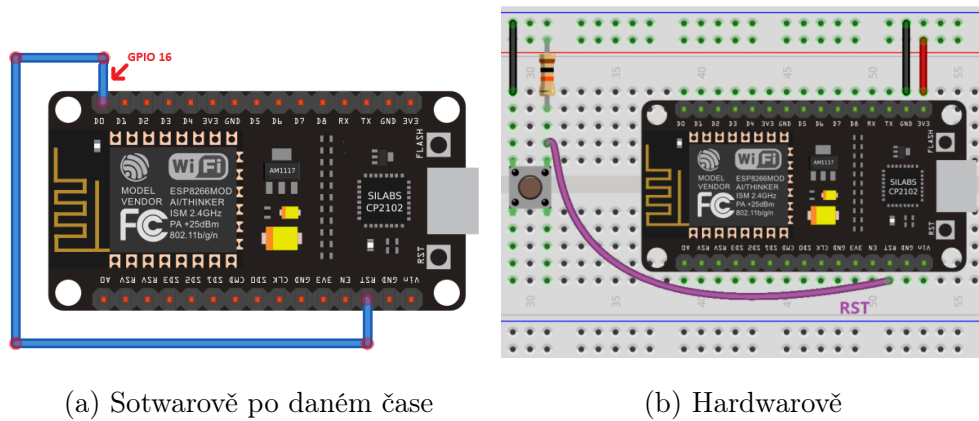
---

<sup>4</sup>Vstupně výstupní pin



Modem, light a deep. Pro moji maturitní práci použiji deep sleep, který je nejúspěšnějších. Kromě **RTC**<sup>5</sup> se vše vypne, včetně CPU či WiFi. V tomto režimu je také možné uchovat data v tzv. RTC paměti, která jsou dostupná i po obnovení. Průměrná spotřeba je 20  $\mu\text{A}$  a je vhodný pro jakýkoliv projekt na akumulátor či baterii.

Probuzení ESP probíhá, buď po nastaveném čase, kde je nutné připojit pin **GPIO 16** tzv. wake-up pin na **RESET** (viz. obr. 2.2a), nebo můžeme na **RESET** přivést krátce tlačítkem logickou nulu a obnovíme ESP hardwarově (viz. obr. 2.2b).



Obrázek 2.2: ESP8266 probuzení z deep sleep po daném čase

### 2.2.2 ESP32

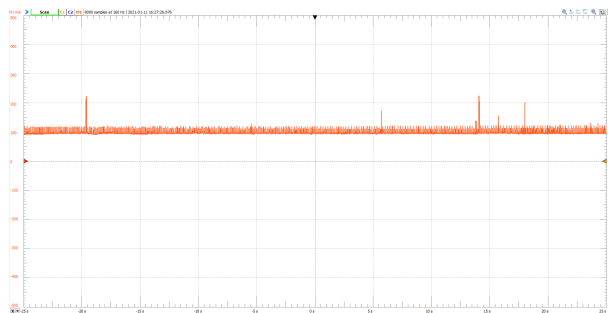
## 2.3 Komunikace mezi ESP a NETIO zásuvkou

<sup>5</sup>Hodiny reálného času

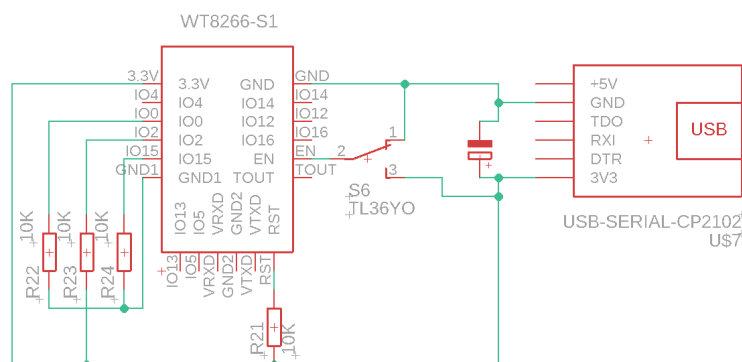
# Kapitola 3

## Tvorba webové stránky





Obrázek 4.2: ESP8266 měření klidového stavu kontinuálního režimu



Obrázek 4.3: ESP8266 schéma zapojení vypnutého ESP přes ENABLE pin

- ESP je neustále zapnuté, probíhá loop funkce pro kontrolu zmáčknutí
- Je připojeno k WiFi, je zaplý access point ESP, běží webserver

Při klidovém stavu byl naměřen elektrický proud průměrně 96,81 mA viz. obr. 4.2. Měření probíhalo 50 s. Pro jednotné porovnání je třeba vypočítat příkon:

$$P = 96,81 \times 10^{-3} \text{ A} \times 3,3 \text{ V}$$

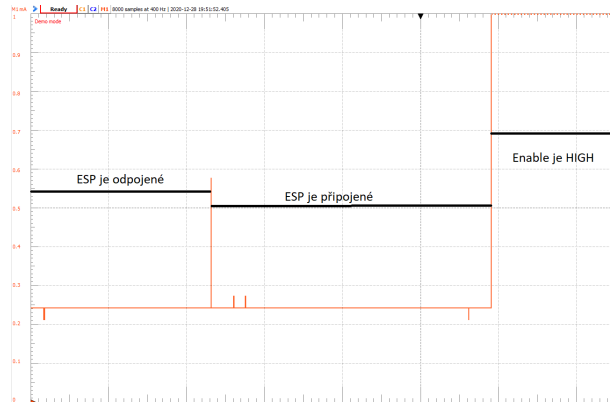
Dle rovnice se příkon rovná  $319,5 \times 10^{-3} \text{ W}$

### ESP vypnuté přes ENABLE pin

Schéma zapojení viz. obr. 4.3

Měření proběhlo za podmínek:

- Měřicí rezistor má odpor 10  $\Omega$



Obrázek 4.4: Měření klidového režimu enable případu

- pin enable byl připojen manuálně

Po připojení ESP8266 proud nevzrostl a drží se stále na 240  $\mu\text{A}$ , což neodpovídá teoretickým hodnotám, které by se měly pohybovat okolo 3  $\mu\text{A}$  viz. obr. 4.4.

Pro výpočet bude jako průměrný odebraný proud použita hodnota uvedená v datasheetu což je 3  $\mu\text{A}$ . Víme, že napětí je 3,3 V takže jsme schopni spočítat elektrický příkon:

$$P = 3 \times 10^{-6} \text{A} \times 3,3 \text{V}$$

Výsledek je  $9,9 \times 10^{-6} \text{ W}$ .

### Deep sleep režim

Schéma zapojení viz. obr. 4.5

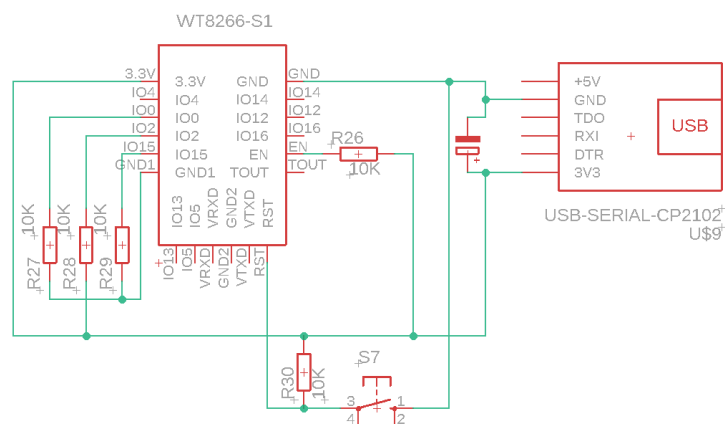
Kvůli citlivosti Analog Discovery 2 nejsme schopni změřit spotřebu deep sleep režimu, je nutné změřit microampérmetrem. Pro výpočet spotřebované energie dosadíme za průměrný elektrický proud hodnotu z datasheetu, která odpovídá 20  $\mu\text{A}$ . Spočítáme elektrický příkon:

$$P = 20 \times 10^{-6} \text{A} \times 3,3 \text{V}$$

Ten v této situaci odpovídá hodnotě  $66 \times 10^{-6} \text{ W}$ .

### Shrnutí výsledků měření spotřeby

Dle měření (viz. tabulka 4.1) nejmenší spotřeba je pokud je ESP8266 vypnuté protože nepracuje žádná část ovladače, díky tomu ovladač je schopen vydržet na bateriích dlouhou



Obrázek 4.5: ESP8266 schéma uvedené v deep sleep stavu

dobu. Na druhé straně stojí kontinuální stav, kde ESP nevypne žádnou část a neustále běží. Kompromis je deep sleep, kdy ESP určité části aby ušetřil energii.

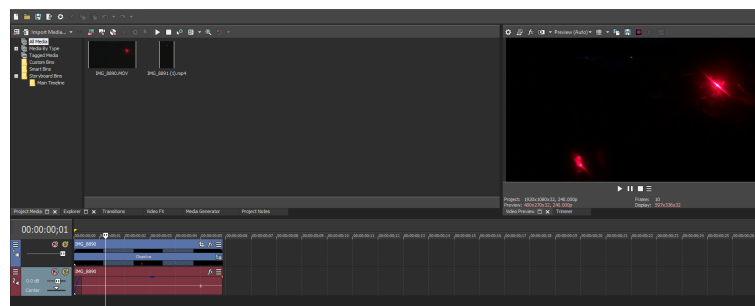
Tabulka 4.1: ESP8266 porovnání ustálených stavů

Ustálený stav	$I(A)$	$P(W)$
Kontinuální	$96,81 \times 10^{-3}$	$319,5 \times 10^{-3}$
Enable	$3,00 \times 10^{-6}$	$9,9 \times 10^{-6}$
Deep sleep	$20,00 \times 10^{-6}$	$66,0 \times 10^{-6}$

#### 4.1.2 Reakční čas jednotlivých situací

Reakční doba byla změřena pomocí kamery. K tlačítku jsem připojil LED, místnost jsem izoloval od světla a zmáčknutí tlačítka a reakci zásuvky jsem natočil ve zpomaleném režimu s **240 snímky za sekundu**. Pro upravení videa jsem použil trial verzi programu **Sony Vegas** (viz. obr. 4.6). Našel jsem rozsvícení LED tlačítka a rozsvícení LED zásuvky ve videu a ustříhl jsem tento úsek od zbytku videa. Program poskytuje zobrazení počtu snímku daného úseku. Reakční čas je následovně možný zjistit pomocí:

$$t = \frac{\text{Počet snímků úseku}}{\text{Počet snímků za sekundu}}$$



Obrázek 4.6: Ukázka postupu pro měření reakčních časů

Tabulka 4.2: ESP8266 porovnání reakčního času jednotlivých situací

Ustálený stav	t(ms)
Kontinuální	<b>196</b>
Enable	<b>3 208</b>
Deep sleep	<b>983</b>

### Porovnání reakčních časů

Nejrychlejší reakce byla pokud ESP8266 bylo neustále zapnuto. Nejpomalejší naopak bylo pokud ESP8266 bylo nutné zapnout, je to z důvodu načtení sketchu do operační paměti, načtení konfigurace WiFi a následnému připojení viz. tabulka 4.2.

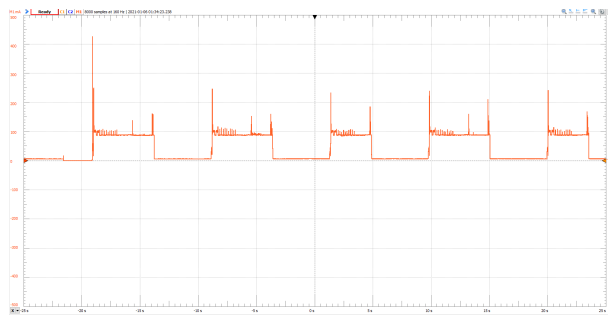
### 4.1.3 Rychlost WiFi připojení

Cílem měření je zjistění rychlostí připojení různými způsoby k přístupovému body, spotřeby a následné porovnání případů. Všechna měření byla provedena za podmínek:

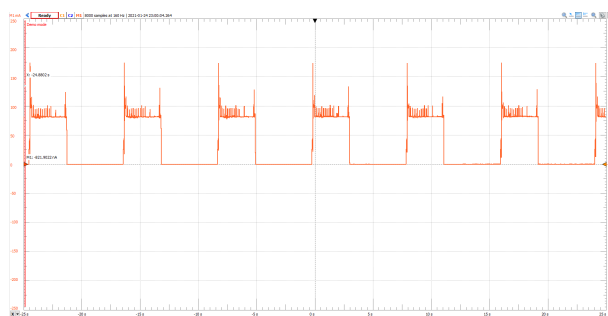
- Zařízení bylo napájeno z USB
- Měřeno bylo pomocí úbytku napětí na rezistoru o velikosti  $0,7 \Omega$
- Přístupový bod se nachází 3,5 m od zařízení

### Dynamické přidělení IP adresy

Měření proběhlo za použití DHCP protokolu, kdy zařízení požádá DHCP server o IP adresu, kterou mu Access point přidělí společně s bránou, maskou a s časem, kdy tato



Obrázek 4.7: Měření dynamického připojení k AP



Obrázek 4.8: Měření statického připojení k AP

adresa platí. Při měření nebyl přístupový bod zabezpečen.

Měření bylo provedeno 5x. Průměrný čas se pohybuje okolo 4 652 ms. Jak je možno vidět na grafu (viz. obr. 4.7), tak dvě WiFi připojení trvaly o 2 sekundy kratší dobu. ESP8266 se totiž zapíše do **DHCP client listu**, a má tak rezervovanou IP adresu, což znamená, že přiřazení proběhne rychleji.

### Statické přidělení IP adresy

Použita byla statická adresa, která byla přidělena ESP8266 před připojením na AP. Přístupový bod nebyl zabezpečen. DHCP server byl vypnut. Měření proběhlo 5x. Průměrný čas byl 3238 ms.

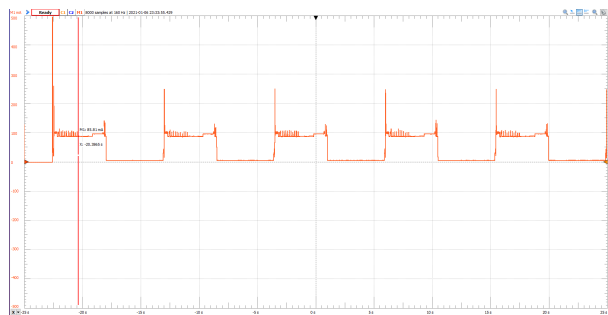
viz. obr. 4.8

### Zabezpečený AP

Připojení na access point je šifrované pomocí WPA2-PSK. IP adresa je na ESP nastavena staticky. DHCP server je zapnut. Průměrný čas byl 4709 ms.

viz. obr. 4.9





Obrázek 4.9: Měření zabezpečeného připojení k AP

Tabulka 4.3: ESP8266 Porovnání reakční doby připojení k WiFi

Pořadí	Dynamické (ms)	Statické (ms)	Zapezpečené (ms)
1.	5 300	3 143	4 606
2.	5 300	3 143	4 606
3.	3 619	3 143	4 606
4.	5 300	3 143	4 606
5.	3 627	3 143	4 606
<b>Průměr</b>	<b>4 629</b>	<b>3 143</b>	<b>4 606</b>

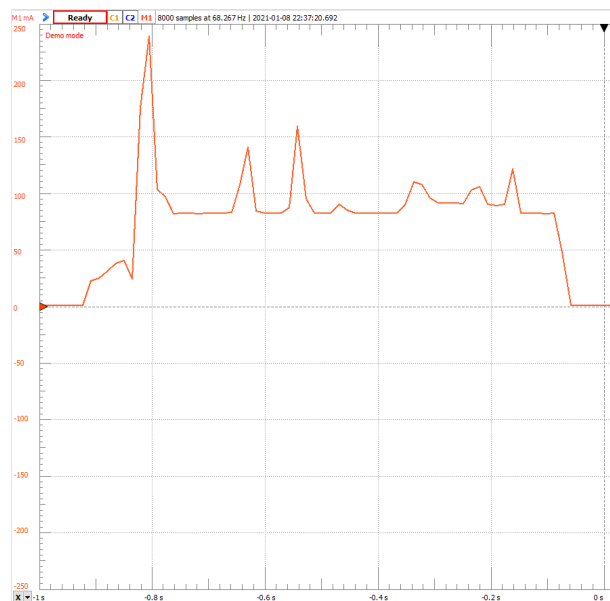
### Shrnutí výsledků rychlostí WiFi nastavení

Z výsledků měření (viz. tabulka 4.3) je nejrychlejší připojení pomocí statické IP adresy, nicméně je velice náročné nastavit IP adresu, masku a bránu pro běžného uživatele. Připojení s DHCP je pomalejší průměrně o 1 s než případ se statickou IP adresou. DHCP vyniká jednoduchostí použití pro běžného uživatele. K zabezpečené WiFi trvá stejně dlouho jako s DHCP.

#### 4.1.4 Rychlost odeslání HTTP requestu

Cílem měření je zjistit čas odesílání HTTP requestu a následné odpovězení zásuvky NETIO. Pokus byl proveden za podmínek:

- Napájeno z USB
- Měřeno pomocí úbytku napětí na rezistoru o velikosti  $0,7 \Omega$



Obrázek 4.10: ESP8266 měření odesílání HTTP requestu včetně reakce zásuvky

- ESP8266 zkontroluje připojení k WiFi a pokud není navázáno, pokusí se ho navázat
- Načtení uložené konfigurace WiFi z flash paměti trvá 300 ms
- ESP ukončí reakci, pokud dostane zpětnou vazbu od zásuvky

viz. obr. 4.10 Jelikož ESP přestane reagovat až po odpovězení zásuvky, dokážeme zjistit celkový čas včetně zapnutí, zkontrolování WiFi připojení, sestavení a odeslání HTTP requestu, reakce zásuvky a zpracování HTTP zprávy viz. tabulka ??.

#### 4.1.5 Spotřeba jednotlivých operací

Spotřebu budeme měřit pro jednotlivé situace WiFi a odesílání HTTP requestu. Pro tyto situace využijeme data z měření v kapitolách 4.1.3 a 4.1.4. Měření WiFi připojení probíhalo za vzorkovací frekvence 160 Hz a HTTP komunikace za 68,275 Hz. Z veličin, které známe, lze vypočítat **spotřebovanou energii**:

$$E = P \times t$$

$$E = U \times I \times t$$

Tabulka 4.4: Čas odeslání HTTP requestu včetně reakce zásuvky

Pořadí	HTTP request (ms)
1.	732
2.	645
3.	732
4.	732
5.	747
<b>Průměr</b>	<b>718</b>

Tabulka 4.5: ESP8266 Spotřeba operací

Operace	U(V)	I(mA)	t(ms)	P(mW)	E( $\mu Wh$ )
Dynamické připojení	3,3	89,5	5 300	295	<b>435</b>
Statické připojení	3,3	83,0	3 143	274	<b>239</b>
Zabezpečené připojení	3,3	90,9	4 606	300	<b>383</b>
HTTP komunikace	3,3	84,8	732	280	<b>57</b>

### Shrnutí výsledků spotřeby operací

Z tabulky 4.5 je možné vidět, že **příkon P** je téměř totožný v každé operaci. Největší rozdíl, který ovlivňuje spotřebu, je čas, za který se úkon vykoná. Jelikož **připojení pomocí statické IP adresy** trvalo nejkratší dobu, má také nejmenší spotřebu.

Samotná **komunikace HTTP** má nejnižší spotřebu ze všech definovaných operací a jeden cyklus operace spotřebuje zanedbatelné množství energie.

## 4.2 ESP32

### 4.2.1 Spotřeba ustálených stavů

Kontinuální

Enable

Deep sleep

### 4.2.2 Reakční čas jednotlivých situací

Získání reakčních časů jednotlivých možností zapojení ESP32 proběhlo stejným způsobem jako u ESP8266 (viz. obr. 4.6 na straně 14) tzn. Náhrál jsem reakci na kameru v 240 snímcích za sekundu, vystříhl jsem úsek, kde zásuvka reaguje, a převedl jsem snímky na čas.

Tabulka 4.6: ESP32 porovnání reakčního času jednotlivých situací

Ustálený stav	t(ms)
Kontinuální	<b>188</b>
Enable	<b>2 333</b>
Deep sleep	<b>1 071</b>

## 4.3 Porovnání získaných výsledků

### 4.3.1 Reakční časy stavů

Z tabulky 4.7 je možné vidět jak si platformy vedly proti sobě.

Pokud ovladač neustále běží s připojenou WiFi čas reakce je velmi rychlý a v obou případech uživatel nepostřehne prodlevu mezi aktivací ovladače a sepnutí relé v zásuvce. Časy mezi platformami se liší minimálně a rozdíl se pohybuje v rámci milisekund, přesto je ESP32 lehce rychlejší.

Zapnutí vypnutého ESP, následné načtení a připojení k WiFi je nejdelší ze všech scénářů,

který trvá několik sekund. Pro uživatele je velice pomalé a v reálném produktu dle mého názoru zcela nepoužitelné. I přesto je ESP32 je mnohem rychlejší a to zhruba o 1 s. Deep sleep je alternativa mezi kontinuálním a vypnutým stavem. ESP totiž vypne většinu svých částí. Díky tomu je schopno relativně rychle zareagovat na budící signál. Reakce obou ovladačů je cirka 1 s. Zde vede ESP8266, které je zhruba o 100 ms pomalejší.

Tabulka 4.7: Porovnání reakčních časů ustálených stavů platforem

Ustálený stav	ESP8266 (ms)	ESP32 (ms)
Kontinuální	<b>196</b>	<b>188</b>
Enable	<b>3 208</b>	<b>2 333</b>
Deep sleep	<b>983</b>	<b>1 071</b>

# Kapitola 5

## Vytvoření funkčních vzorků

# Kapitola 6

## Závěr

# Seznam tabulek

4.1	ESP8266 porovnání ustálených stavů . . . . .	13
4.2	ESP8266 porovnání reakčního času jednotlivých situací . . . . .	14
4.3	ESP8266 Porovnání reakční doby připojení k WiFi . . . . .	16
4.4	Čas odeslání HTTP requestu včetně reakce zásuvky . . . . .	18
4.5	ESP8266 Spotřeba operací . . . . .	18
4.6	ESP32 porovnání reakčního času jednotlivých situací . . . . .	19
4.7	Porovnání reakčních časů ustálených stavů platformou . . . . .	20



# Seznam obrázků

2.1	ESP8266 pinout . . . . .	7
2.2	ESP8266 probuzení z deep sleep po daném čase . . . . .	8
4.1	ESP8266 schéma zapojení kontinuálního ustáleného stavu . . . . .	10
4.2	ESP8266 měření klidového stavu kontinuálního režimu . . . . .	11
4.3	ESP8266 schéma zapojení vypnutého ESP přes ENABLE pin . . . . .	11
4.4	Měření klidového režimu enable případu . . . . .	12
4.5	ESP8266 schéma uvedené v deep sleep stavu . . . . .	13
4.6	Ukázka postupu pro měření reakčních časů . . . . .	14
4.7	Měření dynamického připojení k AP . . . . .	15
4.8	Měření statického připojení k AP . . . . .	15
4.9	Měření zabezpečeného připojení k AP . . . . .	16
4.10	ESP8266 měření odesílání HTTP requestu včetně reakce zásuvky . . . . .	17

Příloha A

Příloha

# Literatura

- [1] ESP8266. *Wireless-Tag Technology Co., Limited* [Online]. Irvine (CA): Wireless-Tag, 2021 [cit.2021-02-04]. Dostupné z: <https://www.tme.com/Document/12266bdd8a30eeb152305461b089a151/WT8266-S1.pdf>
- [2] DHCP PROTOKOL. In: *Wikipedia: the free encyclopedia* [Online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit.2021-02-04]. Dostupné z: [https://en.wikipedia.org/wiki/Dynamic\\_Host\\_Configuration\\_Protocol](https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol)