



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. Břetislavu Bakalovi za odborné vedení práce a cenné rady, které mi pomohly tuto práci zkompletovat. Rád bych také poděkoval technickému řediteli Ing. Břetislavu Bakalovi ml. společnosti NETIO products a.s. za cenné rady, věcné připomínky a vstřícnost při konzultacích a vypracování maturitní 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>Teoretický úvod do problematiky</b>	<b>6</b>
2.1	Zásuvka NETIO . . . . .	6
2.2	Platforma ESP . . . . .	6
2.2.1	ESP8266 . . . . .	6
2.2.2	ESP32 . . . . .	7
2.2.3	Deep sleep . . . . .	8
2.3	LDLN030 . . . . .	9
<b>3</b>	<b>Metodika postupu</b>	<b>11</b>
3.1	Tvorba měřecího vzorku . . . . .	11
3.2	Měření připojení k WiFi a odeslání HTTP requestu . . . . .	11
3.2.1	Spotřeba . . . . .	12
3.2.2	Reakční čas . . . . .	13
3.3	Ustálené stavy . . . . .	13
3.3.1	Spotřeba . . . . .	13
3.3.2	Probuzení z ustálených časů . . . . .	14
<b>4</b>	<b>Měření spotřeby a času</b>	<b>16</b>
4.1	ESP8266 . . . . .	16
4.1.1	Spotřeba a reakční doba ustálených stavů . . . . .	16
4.1.2	Připojení k WiFi a odeslání HTTP requestu . . . . .	19
4.2	ESP32 . . . . .	20
4.2.1	Spotřeba a reakční doba ustálených stavů . . . . .	20
4.2.2	Připojení k WiFi a odeslání HTTP requestu . . . . .	21
4.3	Porovnání získaných výsledků . . . . .	23
4.3.1	Ustálené stavy . . . . .	23

<b>5</b>	<b>Vytvoření funkčních vzorků</b>	<b>24</b>
5.1	Tvorba uživatelského rozhraní . . . . .	24
<b>6</b>	<b>Závěr</b>	<b>25</b>
	<b>Přílohy</b>	<b>27</b>
<b>A</b>	<b>Příloha</b>	<b>28</b>

# Kapitola 1

## Úvod

# Kapitola 2

## Teoretický úvod do problematiky

### 2.1 Zásuvka NETIO

### 2.2 Platforma ESP

ESP jsou rodina mikročipů od společnosti **Espressif Systems**.

#### 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 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**, protože tento modul je používán ve společnosti NETIO products a. s. Je založen na mikročipu ESP8266.

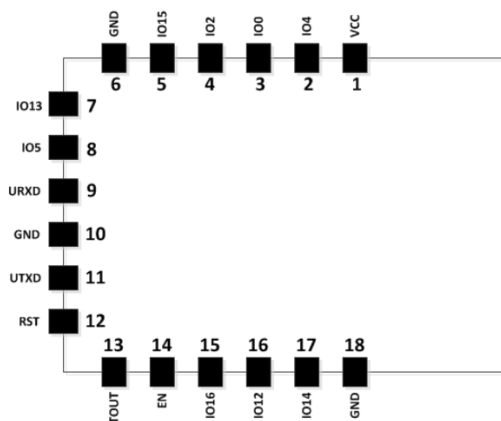
ESP8266 integruje vylepšenou verzi procesoru **L106 Diamond series 32-bit** vytvořený firmou **Tensilica** s podporou frekvencí 80 MHz a 160 MHz a RTOS<sup>1</sup>. K dispozici je 36 KB RAM a 16 Mbit Flash paměti. Integrovaný v systému je také 10 bitový analog-digitální převodník. Modul podporuje standard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

---

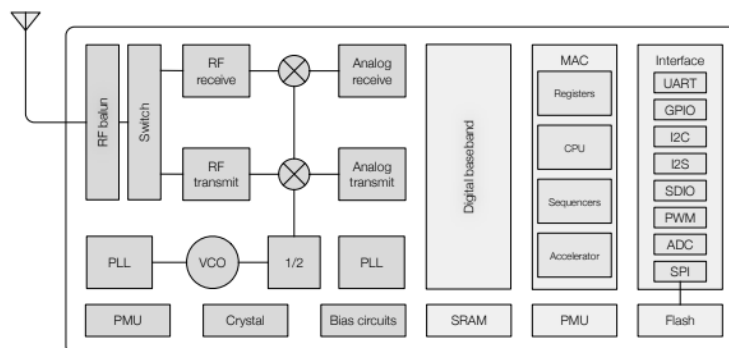
<sup>1</sup>Operační systém v reálném čase

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

WiFi 2,4 GHz umožňuje WPA/WPA2<sup>3</sup>, rychlost až 72,2 Mbps a ke komunikaci využívá anténu PCB. Zařízení má 16 GPIO pinů<sup>4</sup> (viz. obr. 2.1) z toho jsou některé piny využívány na specifické činnosti. Např. pokud při bootu ESP8266 přivedeme GROUND na GPIO 0, zařízení se přepne do UART<sup>5</sup> módu a je možné nahrát do flash paměti program. Další informace jsou na obrázku 2.2.



Obrázek 2.1: ESP8266 pinout: Literatura [1]



Obrázek 2.2: ESP8266 blokový diagram funkcí: Literatura [2]

## 2.2.2 ESP32

ESP32 je mladší model z řady ESP, který byl vydán v roce 2016. Tento mikročip má integrované WiFi i Bluetooth. Série ESP32 obsahuje mikroprocesor Tensilica Xtensa LX6

<sup>3</sup>Chráněný přístup k WiFi

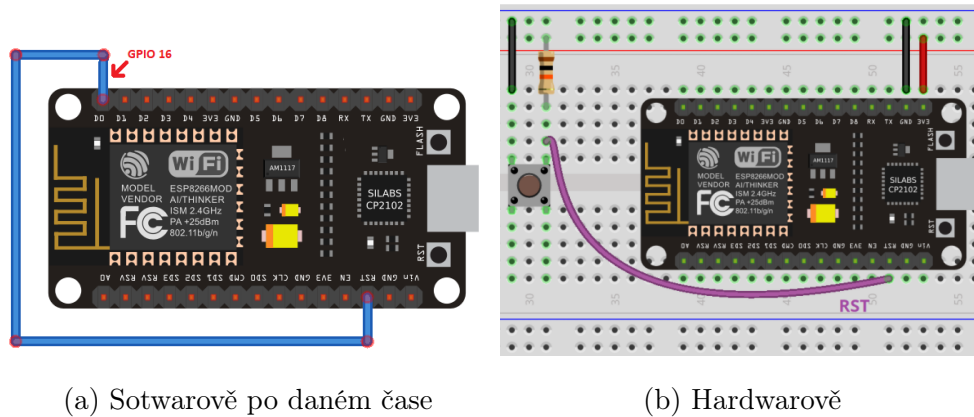
<sup>4</sup>Vstupně výstupní pin

<sup>5</sup>Univerzální asynchronní přijmač, vysílač





**16** tzv. wake-up pin na **RESET** (viz. obr. 2.4a), ten po uplynutí nastaveného času se na krátkou dobu přepne z log. 1 na log. 0 a vyresetuje ESP8266, a nebo můžeme na **RESET** přivést krátce tlačítkem logickou nulu a obnovíme ESP hardwarově (viz. obr. 2.4b).



Obrázek 2.4: ESP8266 probuzení z deep sleep po daném čase: Literatura [4]

Jelikož ESP32 je novější model, deep sleep se liší v používání. ESP32 neprobouzíme přes RESET pin protože se ani na mikročipu ani nenachází. Pokud chceme ESP probudit po určitém čase, nastavíme to pouze v programu a nemusíme žádné piny propojovat. V situaci hardwarového probuzení je nutné definovat pin probuzení a při jaké logické hodnotě se má probudit. Tento pin musí být tzv. RTC\_GPIO<sup>7</sup>.

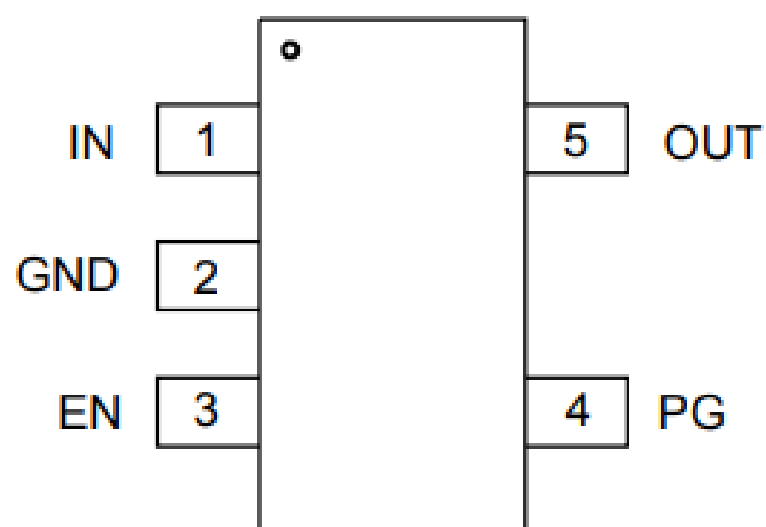
## 2.3 LDLN030

LDLN030 je napěťový regulátor, který pracuje se vstupním napětím od 1,6 V do 5,5 V. Výstupní napětí dokáže stabilizovat na 3,3 V a pokud napětí klesne pod 3 V, odpojí zařízení napojené na výstupu. Tato součástka má časté použití v mobilních telefonech, HDD a SSD discích a ostatních zařízeních napájené baterií.

Regulátor má celkem 5 pinů, a to vstup, výstup, GROUND, enable pin a Power Good. Enable pin rozhoduje zda součástka funguje či nikoliv, pokud je přivedena na zem, výstup je 0 V. Power Good je pin, který může informovat například základní desku během POST<sup>8</sup>, že napájení je v pořádku a může pokračovat v činnosti.

<sup>7</sup>Pin, který je v činnosti i během spánku ESP

<sup>8</sup>Power On Self Test



Obrázek 2.5: LDLN030 Pinout: Literatura [5]

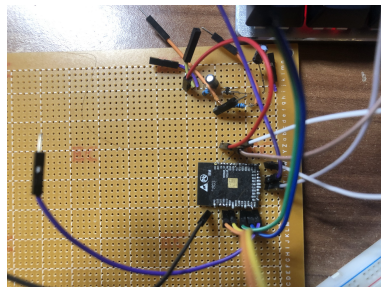
# Kapitola 3

## Metodika postupu

### 3.1 Tvorba měřecího vzorku

Pro měření jednotlivých scénářů jsem si vytvořil vzorek, který byl určen pouze na měření. Cílem bylo vytvořit vzorek jednoduše modifikovatelný, proto jsem zvolil pájivé pole na základní část.

Na pájivém poli jsem napájel testovanou ESP platformu společně s přívodem, který byl 5 V. ESP má operační napětí 3,3 V, proto bylo nutné přidat stabilizátor LDLN030, který napětí stabilizoval na operačním napětí. Stabilizátor byl zapojen dle doporučeného diagramu v datasheetu.



Obrázek 3.1: Vzorek pro měření na pájivém poli

Dále k potřebným GPIO pinům jsem napájel vodiče pro pozdější manipulaci. Pro další specifická zapojení, jako např. tlačítko, jsem použil nepájivé pole, které jsem vždy připojil specificky pro dané měření.

### 3.2 Měření připojení k WiFi a odeslání HTTP requestu

Cílem měření je zjistění rychlostí a spotřeby připojení různými způsoby k přístupovému bodu a následné porovnání případů. Do měření je také započítané odesílání HTTP

requestu zásuvce. Jsou vytvořeny celkem 4 případy akcí:

- **statická IP adresa** - Zařízení dostane IP adresu, masku a bránu staticky staticky nakonfigurovanou. Na access pointu bude vyplé DHCP a komunikace nebude zabezpečena.
- **dynamická IP adresa** - Bude využit DHCP protokol, kde zařízení požádá DHCP server o IP adresu, kterou mu access point přidělí společně s bránou, maskou a s časem, kdy tato adresa platí. Při měření nebude přístupový bod zabezpečen.
- **Zabezpečené připojení** - Připojení na access point bude šifrované pomocí WPA2 - PSK. DHCP server bude zapnut. Toto je simulace klasického uživatelského používání.
- **Odeslání HTTP requestu** - Zařízení je připojené k WiFi. Odešle zprávu zásuvce a pokud dostane zpětnou vazbu od zásuvky, ukončí se činnost.

Ve všech scénářích bude měřeno zařízením **ANALOG DISCOVERY 2** od Digilent. Frekvence procesoru je 160 MHz a AP je vzdálen 3,5 m od zásuvky a funkčního vzorku ovladače.

### 3.2.1 Spotřeba

Měření bude úbytek napětí na bočníku<sup>1</sup> o velikosti určené u měření. V programu k měřicímu zařízení je možné stejně jako na osciloskopu vytvořit graf naměřených hodnot v dané vzorkovací frekvenci. Hodnoty napětí bude převedeno na hodnoty el. proudu pomocí Ohmova zákona  $I = \frac{U}{R}$ . Amplitudy el. proudu budou sečteny a vyděleny počtem vzorků [DOPLNIT VZOREC]. Tato operace nám poskytne zjištění průměrného el. proudu ve scénáři. Spotřebu již vypočítáme dle:

$$E = U \times \bar{I} \times \frac{t}{3600}$$

Čas  $t$  v sekundách je možné zjistit v následující kapitole.

---

<sup>1</sup>nízkoohmový rezistor

### 3.2.2 Reakční čas

Z naměřeného případu bude vyjmuta klidová část a rozdíl mezi posledním a prvním vzorkem je poté reakční čas. V případě HTTP requestu nebude počítána část konfigurace WiFi. Společně s měřením byl na **serial monitor** odesílány zprávy, kde se program nachází. Poté se vyjmula část, která dle serial monitoru nepatří do scénáře.

## 3.3 Ustálené stavy

Pro ustálené stavy byly navrženy jednoduché programy pro co nejlepší porovnání. Ustálený stav je stav, kdy zařízení čeká na uživatelský podnět jako např. stisknutí tlačítka. Ideální ustálený stav musí splňovat:

1. nejkratší možnou reakční dobu
2. nejnižší možnou spotřebu pro co nejdelší výdrž baterií

Reakční doba musí být v rámci uživatelské přívětivosti tzn. probuzení z tohoto stavu nesmí trvat více jak 1,5 s. Zároveň se zařízení nesmí vypnout zdůvodu vybité baterie za krátkou dobu. Porovnání proběhne mezi třemi scénáři, které mají jiné vlastnosti. Vybrány byly následující:

- **Kontinuální režim** - V tomto stanu není žádná funkce ESP vypnuta.
- **Deep sleep** - ESP vypne vše až na RTC (kapitola viz. 2.2.3 na straně 8).
- **Enable button** - ESP je vypnuto pomocí přivedené GROUND na pin ENABLE.

### 3.3.1 Spotřeba

Cílem měření je nalézt stav s nejmenší spotřebou energie, jelikož finální prvek bude napájen z baterií je důležité co nejdelší výdrž. Programy, které byly napsány pro měření:

- **Kontinuální režim** - Zařízení má nastavený určitý GPIO na tlačítko a monitoruje zda bylo zmáčknuto. ESP také má otevřenou sériovou komunikaci.

- **Deep sleep** - Program zajišťuje probuzení ESP každých 5 s a odesílání zprávy na serial monitoring.
- **Enable button** - Pro měření této situace není třeba specifický program

Pro měření bude použit **ANALOG DISCOVERY** s jeho funkcí osciloskopu. Na napájení bude přidán bočník o velikosti  $0,7 \Omega$ ,  $10,5 \Omega$  nebo  $4,7 \Omega$ <sup>2</sup>. Pomocí Ohmova zákona  $I = \frac{U}{R}$  je možné zjistit el. proud. Příkon, který bude porovnán vypočítáme dle vztahu:

$$P = U \times I$$

## OBRÁZKY ZAPOJENÍ

### 3.3.2 Probuzení z ustálených časů

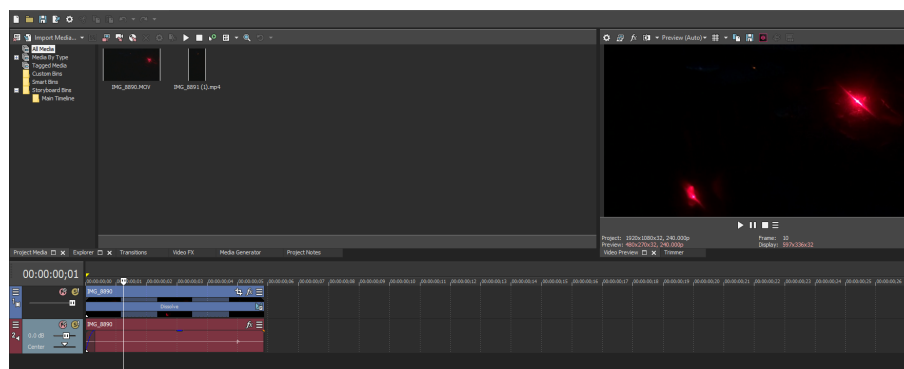
Ovladač, který je již připojen k WiFi, po stisknutí tlačítka odešle HTTP request zásuvce, která zareaguje přepnutím stavu a odesláním zprávy zpět ovladači zda se povedlo či nikoliv.

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. 3.2). 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:

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

---

<sup>2</sup>Je nutné zvolit takový odpor, aby byl úbytek napětí měřitelný.



Obrázek 3.2: Ukázka postupu pro měření reakčních časů



# Kapitola 4

## Měření spotřeby a času

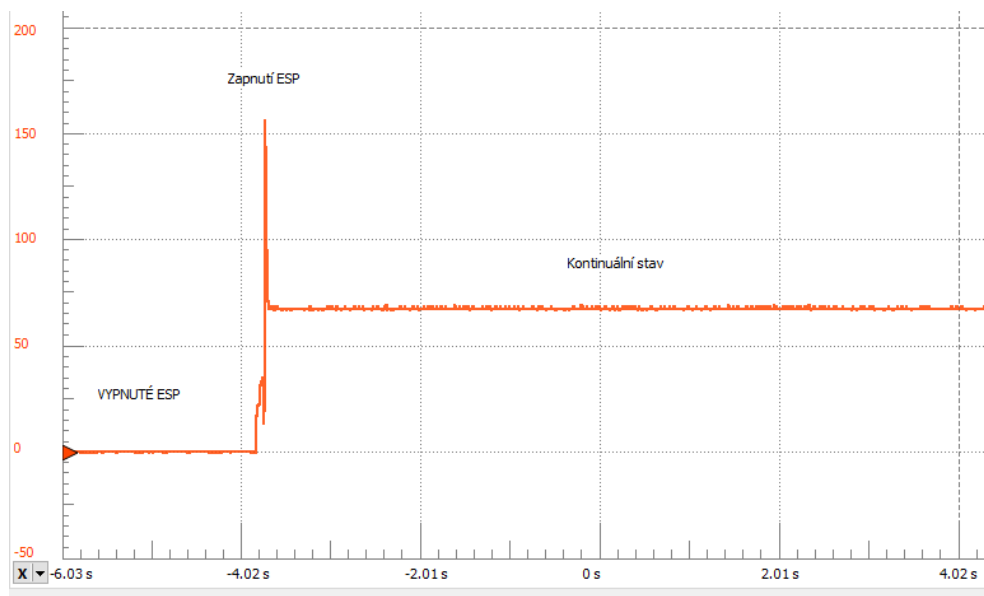
Všechny výpočty jsou k nalezení v adresáři "Měření" v souborech excel společně s použitými programy a naměřenými hodnotami v grafech.

### 4.1 ESP8266

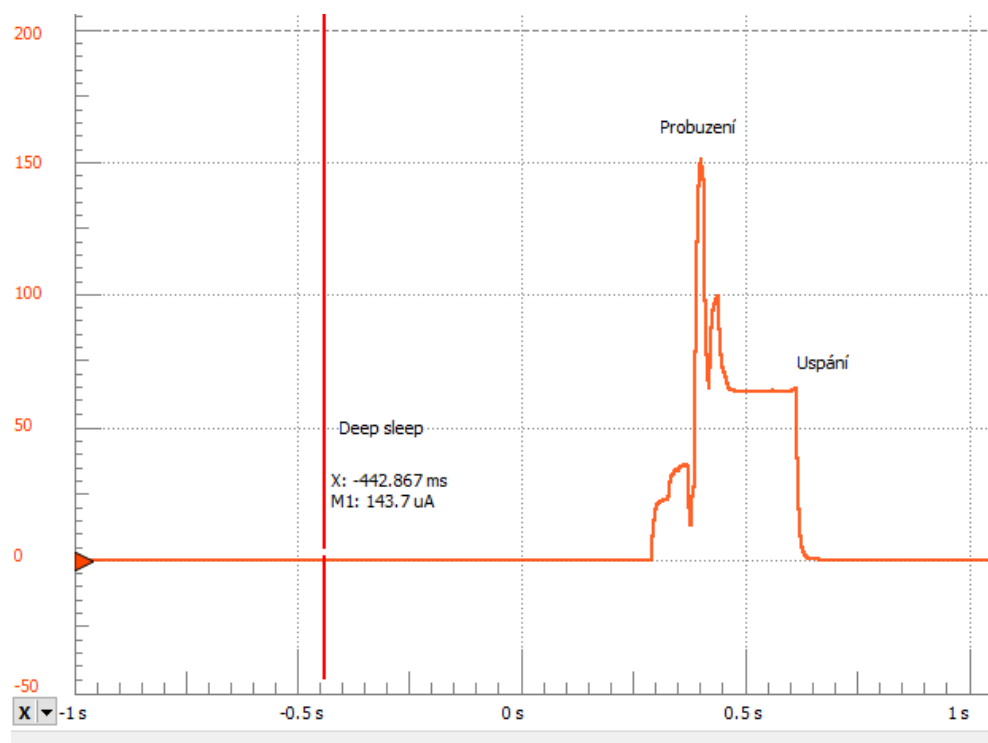
#### 4.1.1 Spotřeba a reakční doba ustálených stavů

##### Spotřeba

Spotřeba jednotlivých scénářů je měřena dle metodiky 3.3.1 na straně 13. Bočník pro měření úbytku napětí byl o velikosti  $4,7 \Omega$ .



Obrázek 4.1: ESP8266 graf kontinuálního ustáleného stavu



Obrázek 4.2: ESP8266 graf deep sleep ustáleného stavu

Situace **enable button** nebyla možná změřit zařízením **ANALOG DISCOVERY 2** ani na bočníku o velikosti  $0,7 \Omega$  kvůli velice nízkému úbytku napětí, který se v datasheetu WT8266 pohybuje v jednotkách  $\mu A$ . V tomto případě pro výpočty je použita hodnota, kterou určil výrobce v datasheetu.

Tabulka 4.1: ESP8266 porovnání spotřeby ustálených stavů

Ustálený stav	U(V)	I(A)	P(W)
Kontinuální	3,3	$67,66 \times 10^{-3}$	$223,3 \times 10^{-3}$
Deep sleep	3,3	$138,20 \times 10^{-6}$	$460,0 \times 10^{-6}$
Enable button	3,3	$3,00 \times 10^{-6}$	$9,9 \times 10^{-6}$

Dle tabulky 4.1 je možné vidět jednotlivé situace. Největším příkonem disponuje kontinuální ustálený stav z důvodu běžícího programu na kontrolu interakce. Nejmenší spotřeba je naopak vypnuté ESP přes enable pin.

Tabulka 4.2: ESP8266 porovnání reakčního času ustálených stavů

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

### Reakční čas

Reakční čas byl měřen dle metodiky 3.3.2 na straně 14. Nejrychlejší ustálený stav je kontinuální, kvůli připravenosti na interakci viz. tabulka 4.3. Nejdéle trvá vypnuté ESP, protože je nutné ESP zapnout, nastavit konfiguraci a připojit na WiFi, zatím co zapnuté ESP je již připraveno.

### Porovnání

Pro porovnání spotřeb budeme předpokládat, že zařízení je napájeno  $3 \times$  NiZn akumulátorem typu AA, které společně mají kapacitu **3750 mAh**, což je možné převést na energii vztahem  $E = U_{aku} \times \text{kapacita}$ . a dále vztahem<sup>1</sup>  $t_p = \frac{E_{bat}}{P}$  zjistit provozní čas (viz. tabulka 4.3). Zařízení není schopno dosáhnout této výdrže protože pokud napětí klesne pod 3 V, stabilizátor napájení odpojí.

Tabulka 4.3: ESP8266 porovnání ustálených stavů

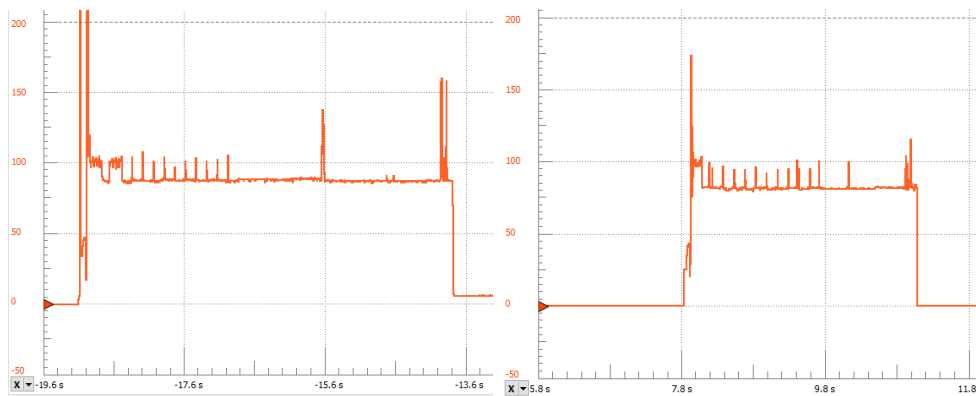
Ustálený stav	P[W]	$t_p[h]$	t[ms]
Kontinuální	$223,3 \times 10^{-3}$	75,6	196
Deep sleep	$460,0 \times 10^{-6}$	36 993,1	983
Enable button	$9,9 \times 10^{-6}$	1 704 545,0	3 208

Kontinuální stav, zatímco má výdrž v rámci několika dní, reakční čas trvá necelých 200 ms. Enable button situace naprosto opačná. Reakční čas je pro uživatele dlouhý, ale výdrž v tomto stavu bez interakce by se pohybovala v letech. Dobrý kompromis je situace deep sleep, kdy zařízení reagovalo vždy do jedné sekundy a teoretická výdrž je v jednotkách let.

<sup>1</sup> $t_p$  - provoz v ustáleném režimu,  $E_{bat}$  - energie  $3 \times$  AA akumulátorů

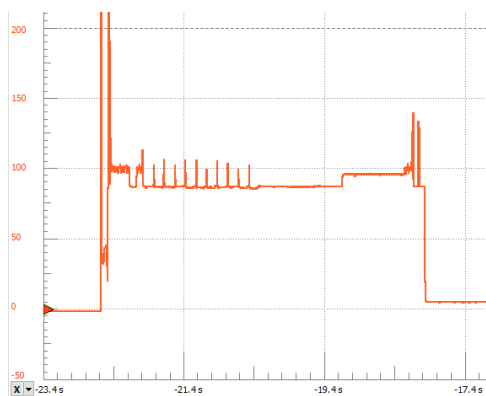
### 4.1.2 Připojení k WiFi a odeslání HTTP requestu

Měření proběhlo dle metodiky 3.2 na straně 11 na bočníku  $0,7\ \Omega$ .



(a) Dynamická IP

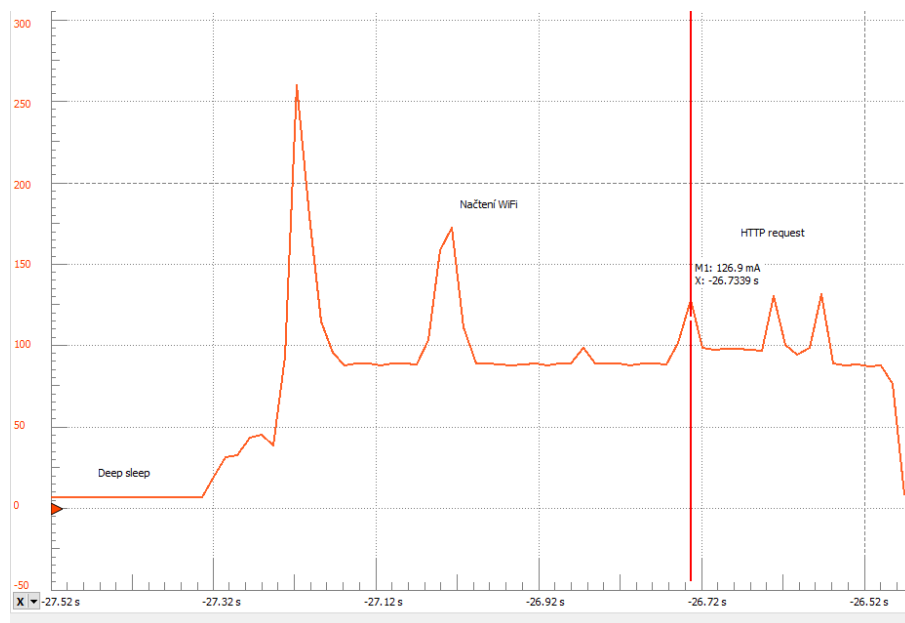
(b) Statická IP



(c) Zabezpečené, dynamická IP

Obrázek 4.3: ESP8266 grafy WiFi situací

Z tabulky 4.7 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 energie. Samotná **kommunikace HTTP** má nejnižší spotřebu ze všech definovaných operací a jeden cyklus operace spotřebuje zanedbatelné množství energie.



Obrázek 4.4: ESP8266 graf HTTP requestu

Tabulka 4.4: ESP8266 Spotřeba operací

Operace	U(V)	I[mA]	t[ms]	P[mW]	E[μWh]
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	76,7	483	253	<b>34</b>

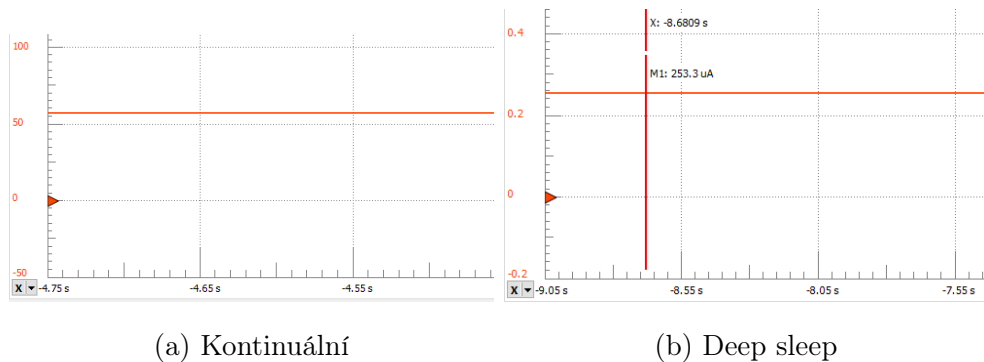
## 4.2 ESP32

### 4.2.1 Spotřeba a reakční doba ustálených stavů

#### Spotřeba

Spotřeba jednotlivých scénářů je měřena dle metodiky 3.3.1 na straně 13. Bočník pro měření úbytku napětí byl o velikosti  $10,5 \Omega$ .

Ustálený stav enable pin nebyl naměřen zařízením Analog Discovery 2 z důvodu velmi nízkému úbytku napětí na bočníku. K porovnání bude použita hodnota z datasheetu.



Obrázek 4.5: ESP32 grafy ustálených stavů

Tabulka 4.5: ESP32 porovnání spotřeby ustálených stavů

Ustálený stav	U[V]	I[A]	P[W]
Kontinuální	3,3	$57,39 \times 10^{-3}$	$189,4 \times 10^{-3}$
Deep sleep	3,3	$253,39 \times 10^{-6}$	$838,0 \times 10^{-6}$
Enable button	3,3	$1,00 \times 10^{-6}$	$3,3 \times 10^{-6}$

## Reakční čas

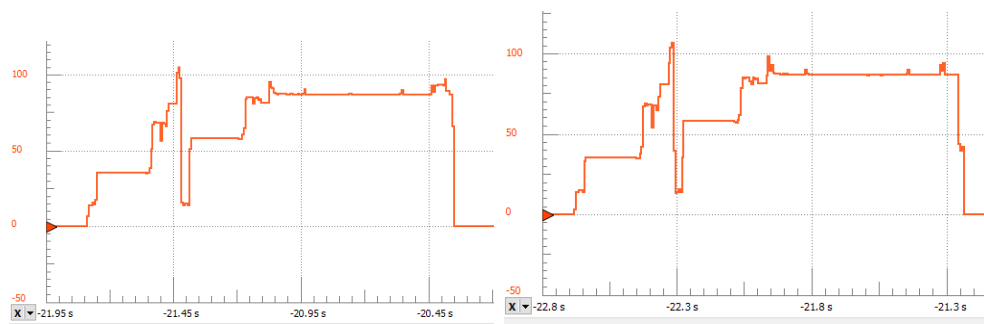
Získání reakčních časů jednotlivých možností zapojení ESP32 proběhlo způsobem dle 3.3.1 na straně 13. Tabulka 4.6 ukazuje, že nejdéle trval ustálený režim v podobě vypnutého ESP32 přes enable pin a nejkratší dobu trval kontinuální ustálený stav. Kompromis mezi těmito dvěma stavy byl deep sleep ustálený stav, který se pohyboval okolo sekundy.

Tabulka 4.6: ESP32 porovnání reakčního času jednotlivých situací

Ustálený stav	t[ms]
Kontinuální	<b>188</b>
Enable button	<b>2 333</b>
Deep sleep	<b>1 071</b>

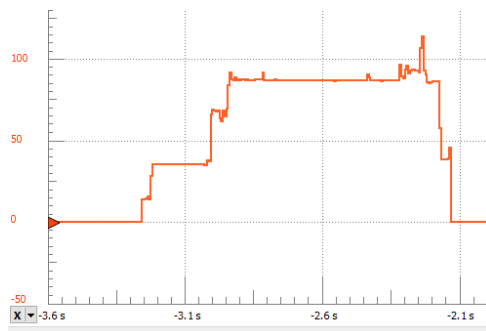
## 4.2.2 Připojení k WiFi a odeslání HTTP requestu

Měření proběhlo podle metodiky 3.2 na straně 11.



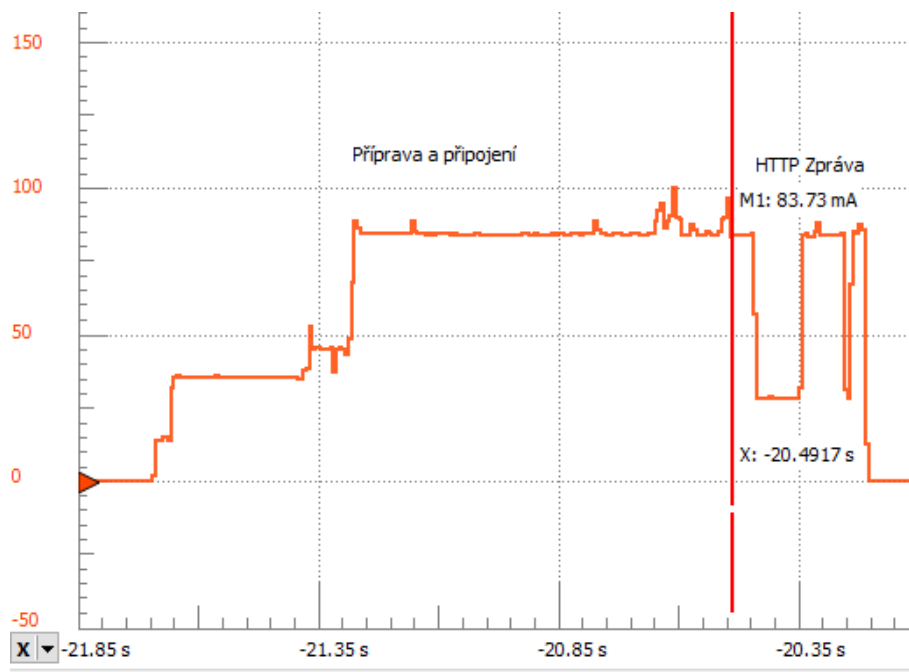
(a) Dynamická IP

(b) Statická IP



(c) Zabezpečené, dynamická IP

Obrázek 4.6: ESP32 grafy WiFi situací



Obrázek 4.7: ESP832 graf HTTP requestu

Tabulka 4.7: ESP8266 Spotřeba operací

Operace	U[V]	I[mA]	t[ms]	P[mW]	E[μWh]
Dynamické připojení	3,3	72,0	1 406	238	<b>92</b>
Statické připojení	3,3	69,2	1 400	228	<b>88</b>
Zabezpečené připojení	3,3	74,7	1 081	246	<b>74</b>
HTTP komunikace	3,3	62,6	281	207	<b>16</b>

## 4.3 Porovnání získaných výsledků

### 4.3.1 Ustálené stavy

Z tabulky 4.8 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 cirká 1 s. Zde vede ESP8266, které je zhruba o 100 ms rychlejší. Je to zdůvodu komplikovanější struktury ESP32.

Tabulka 4.8: Porovnání reakčních časů ustálených stavů platforem

Ustálený stav	ESP8266	ESP32
	[ms]	[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

### 5.1 Tvorba uživatelského rozhraní

# Kapitola 6

## Závěr

# Seznam tabulek

4.1	ESP8266 porovnání spotřeby ustálených stavů . . . . .	17
4.2	ESP8266 porovnání reakčního času ustálených stavů . . . . .	18
4.3	ESP8266 porovnání ustálených stavů . . . . .	18
4.4	ESP8266 Spotřeba operací . . . . .	20
4.5	ESP32 porovnání spotřeby ustálených stavů . . . . .	21
4.6	ESP32 porovnání reakčního času jednotlivých situací . . . . .	21
4.7	ESP8266 Spotřeba operací . . . . .	23
4.8	Porovnání reakčních časů ústálených stavů platforem . . . . .	23

# Seznam obrázků

2.1	ESP8266 pinout: Literatura [1] . . . . .	7
2.2	ESP8266 blokový diagram funkcí: Literatura [2] . . . . .	7
2.3	ESP32 pinout a blokový diagram . . . . .	8
2.4	ESP8266 probuzení z deep sleep po daném čase: Literatura [4] . . . . .	9
2.5	LDLN030 Pinout: Literatura [5] . . . . .	10
3.1	Vzorek pro měření na pajivém poli . . . . .	11
3.2	Ukázka postupu pro měření reakčních časů . . . . .	15
4.1	ESP8266 graf kontinuálního ustáleného stavu . . . . .	16
4.2	ESP8266 graf deep sleep ustáleného stavu . . . . .	17
4.3	ESP8266 grafy WiFi situací . . . . .	19
4.4	ESP8266 graf HTTP requestu . . . . .	20
4.5	ESP32 grafy ustálených stavů . . . . .	21
4.6	ESP32 grafy WiFi situací . . . . .	22
4.7	ESP832 graf HTTP requestu . . . . .	22

Příloha A

Příloha

# Literatura

- [1] WT8266. *Wireless-Tag Technology Co., Limited* [Online]. Irvine (C): Wireless-Tag, 2021 [cit.2021-02-22]. Dostupné z: <https://www.tme.com/Document/12266bdd8a30eeb152305461b089a151/WT8266-S1.pdf>
- [2] ESP8266. *Espressif Systems* [Online]. Shangai, Čína: Espressif Systems, 2020 [cit.2021-02-22]. Dostupné z: [https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf)
- [3] ESP32. *Espressif Systems* [Online]. Shangai, Čína: Espressif Systems, 2020 [cit.2021-02-22]. Dostupné z: [https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf)
- [4] ESP8266 DEEP SLEEP. *Random Nerd Tutorials* [Online]. Porto, Portugalsko: RandomNerdTutorials.com, 2019 [cit.2021-02-22]. Dostupné z: <https://randomnerdtutorials.com/esp8266-deep-sleep-with-arduino-ide/>
- [5] DHCP PROTOKOL. In: *Wikipedia: the free encyclopedia* [Online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit.2021-02-22]. Dostupné z: [https://en.wikipedia.org/wiki/Dynamic\\_Host\\_Configuration\\_Protocol](https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol)