

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2019

Bc. Miroslav Franek



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV MIKROELEKTRONIKY

DEPARTMENT OF MICROELECTRONICS

## VÝVOJOVÝ KIT PRO KOMUNIKACI MIKROKONTROLÉRŮ S POČÍTAČEM

DEVELOPMENT KIT FOR MICROCONTROLLERS COMMUNICATION WITH COMPUTER

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

**Bc. Miroslav Franek**

### VEDOUCÍ PRÁCE

SUPERVISOR

**Ing. Vladimír Levek**

**BRNO 2019**

# Diplomová práce

magisterský navazující studijní obor **Mikroelektronika**

Ústav mikroelektroniky

**Student:** Bc. Miroslav Franek

**ID:** 173645

**Ročník:** 2

**Akademický rok:** 2018/19

## NÁZEV TÉMATU:

### Vývojový kit pro komunikaci mikrokontrolérů s počítačem

#### POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte vývojový kit sloužící pro výuku laboratorních cvičení, který bude umožňovat komunikaci vybraného mikrokontroléru s počítačem prostřednictvím sběrnice USB a sítěmi Bluetooth, Ethernet a Wifi. Systém bude komunikovat s aplikací v počítači pomocí vybraných modulů a bude ovládán pomocí tlačítek a displeje. Prostudujte technické možnosti dotčené problematiky, popište jednotlivé komunikační platformy, vyberte vhodné součástky a moduly, proveďte návrh vývojového přípravku a celý jej realizujte včetně obslužného programu v mikrokontroléru. Dále proveďte ověření funkčnosti a otestujte použité komunikační moduly. Všechny etapy vývoje a realizace popište v textové části diplomové práce.

#### DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce

**Termín zadání:** 4.2.2019

**Termín odevzdání:** 21.5.2019

**Vedoucí práce:** Ing. Vladimír Levek

**Konzultant:**

**doc. Ing. Lukáš Fajcik, Ph.D.**  
*předseda oborové rady*

#### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Diplomová práce se zabývá návrhem vývojového kitu pro komunikaci mikrokontroléru s počítačem pomocí rozhraní USB, Bluetooth, WiFi a Ethernet. V teoretickém rozboru jsou popsána tato rozhraní a také rozhraní, která používá pro komunikaci mikrokontrolér. V další části jsou vybrány součástky pro realizaci vývojového kitu. Následuje návrh schématu zapojení a desky plošného spoje. Dále je zde popsán návrh programového vybavení. Poslední kapitola se věnuje realizaci a testování vývojového kitu. Příloha obsahuje návrh laboratorní úlohy pro seznámení studentů s vývojovým kitem.

## **KLÍČOVÁ SLOVA**

Vývojový kit, Mikrokontrolér, Ethernet, WiFi, USB, Bluetooth, UART, I<sup>2</sup>C, SPI, OLED displej, ESP32, W5500

## **ABSTRACT**

Master's thesis deals with the design of development kit for communication between a microcontroller and a computer using USB, Bluetooth, WiFi and Ethernet interfaces. In the theoretical introduction is the analysis of these interfaces and the interfaces used for communication by the microcontroller. In the next part, components for the development kit are selected. Then a wiring diagram and the printed circuit board is designed. Next, the firmware design is described. The last chapter deals with the implementation and testing of the development kit. The appendix contains a proposal for a laboratory task to introduce the development kit to students.

## **KEYWORDS**

Development Kit, Microcontroller, Ethernet, WiFi, USB, Bluetooth, UART, I<sup>2</sup>C, SPI, OLED Display, ESP32, W5500

FRANEK, M. *Vývojový kit pro komunikaci mikrokontrolérů s počítačem*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2019. 81 s.  
Vedoucí diplomové práce Ing. Vladimír Levek.

## **Prohlášení autora o původnosti díla**

*„Prohlašuji, že svou diplomovou práci na téma Vývojový kit pro komunikaci mikrokontrolérů s počítačem jsem vypracoval samostatně pod vedením vedoucí/ho diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.*

*Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.*

V Brně dne: **20. května 2019**

.....  
Miroslav Franek, podpis

## **PODĚKOVÁNÍ**

Děkuji vedoucímu diplomové práce Ing. Vladimíru Levkovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce. Dále děkuji své rodině a blízkým za podporu.

V Brně dne .....

.....

(podpis autora)

Experimentální část této diplomové práce byla realizována na výzkumné infrastruktuře  
vybudované v rámci projektu CZ.1.05/2.1.00/03.0072

**Centrum senzorických, informačních a komunikačních systémů (SIX)**  
operačního programu Výzkum a vývoj pro inovace.

# OBSAH

<b>1</b>	<b>Úvod</b>	<b>12</b>
1.1	Co je to vývojový kit?.....	12
1.2	Funkce navrženého vývojového kitu .....	13
<b>2</b>	<b>Popis komunikačních rozhraní</b>	<b>16</b>
2.1	Komunikace mikrokontroléru s počítačem.....	16
2.1.1	Rozhraní Ethernet .....	16
2.1.2	Síť WiFi .....	17
2.1.3	Protokoly TCP a UDP.....	19
2.1.4	Rozhraní USB .....	20
2.1.5	Síť Bluetooth.....	21
2.2	Komunikace mikrokontroléru s periferiemi .....	22
2.2.1	Rozhraní UART .....	23
2.2.2	Sběrnice I <sup>2</sup> C .....	24
2.2.3	Sběrnice SPI.....	26
<b>3</b>	<b>Návrh zařízení</b>	<b>27</b>
3.1	Výběr součástek .....	27
3.1.1	Mikrokontrolér.....	27
3.1.2	Převodník USB na UART.....	28
3.1.3	Modul pro připojení k Bluetooth a WiFi .....	29
3.1.4	Modul pro připojení k Ethernetu .....	32
3.1.5	Displej OLED .....	33
3.1.6	Napájení .....	35
3.2	Návrh zapojení .....	35

3.2.1	Ošetření tlačítek .....	36
3.2.2	Zapojení LED .....	37
3.2.3	Pullup rezistory .....	37
3.2.4	Blokovací kondenzátory .....	38
3.2.5	Měření proudu.....	38
3.2.6	Testovací piny .....	39
3.2.7	Připojení diferenciálního páru USB.....	39
3.2.8	OLED displej .....	39
3.3	Návrh desky plošného spoje .....	39
<b>4</b>	<b>Návrh programového vybavení</b> .....	<b>41</b>
4.1	Funkce Firmware .....	41
4.1.1	Hlavní smyčka .....	41
4.1.2	Nastavení rozhraní WiFi a Ethernet.....	42
4.1.3	Nastavení rozhraní Bluetooth .....	45
4.1.4	Nastavení rozhraní UART a USB.....	45
4.1.5	Popis funkce terminálu .....	45
4.2	Grafické prostředí .....	46
4.2.1	Seznam.....	47
4.2.2	Terminál.....	47
4.2.3	Informační okno.....	48
4.2.4	Okna pro zadávání hodnot .....	49
4.3	Komunikace s rozhraními .....	50
4.3.1	Výměna dat mezi rozhraním a terminálem.....	50
4.3.2	Komunikace mikrokontroléru s modulem ESP32 .....	51
4.3.3	Komunikace mikrokontroléru s USB převodníkem .....	53

4.3.4	Komunikace mikrokontroléru s modulem W5500 .....	54
<b>5</b>	<b>Realizace a testování</b>	<b>55</b>
5.1	Měření osciloskopem .....	56
5.1.1	Komunikace UART .....	56
5.1.2	Komunikace I <sup>2</sup> C.....	57
5.1.3	Komunikace SPI .....	59
5.2	Měření spotřeby .....	60
5.2.1	Spotřeba OLED displeje .....	60
5.2.2	Spotřeba modulů ESP32 a W5500.....	61
5.3	Měření datové propustnosti .....	62
<b>6</b>	<b>Závěr</b>	<b>64</b>
	<b>Literatura</b>	<b>65</b>
	<b>Seznam obrázků</b>	<b>67</b>
	<b>Seznam tabulek</b>	<b>69</b>
	<b>Seznam veličin a zkratk</b>	<b>70</b>
	<b>Seznam příloh</b>	<b>73</b>

# 1 ÚVOD

Diplomová práce se zabývá návrhem vývojového kitu pro komunikaci mikrokontroléru s počítačem. Vývojový kit bude sloužit pro výuku programování v laboratořích. Cílem práce bylo prostudovat jednotlivá rozhraní, vybrat vhodný mikrokontrolér a také součástky, pomocí kterých bude mikrokontrolér s počítačem přes daná rozhraní komunikovat. Práce se skládá z teoretického rozboru, kde jsou popsána jednotlivá rozhraní, která mikrokontrolér využívá ke komunikaci s počítačem, nebo se součástkami na DPS (Desce plošného spoje). Následuje výběr součástek, jejich popis a výčet hlavních parametrů. Dále se práce zabývá návrhem elektrického zapojení a rozmístěním součástek na DPS. Práce pokračuje programováním FW (*Firmware*) pro mikrokontrolér, kde jsou na vývojových diagramech popsány jeho hlavní funkce. Poslední kapitola obsahuje měření a testování vývojového kitu.

## 1.1 Co je to vývojový kit?

Vývojový kit je přípravek, osazený různými elektronickými součástkami. Slouží k demonstraci a ověření funkce jednotlivých částí. Uživatel si na vývojovém kitu může vyzkoušet například naprogramovat komunikaci mikrokontroléru s nějakou součástkou či modulem.

Vývojář si může vyzkoušet práci se součástkami, které potřebuje použít na svém budoucím projektu dříve, než k tomuto projektu vznikne DPS, na které budou součástky napevno připojeny. Může tak předejít chybám v návrhu, které nejsou na první pohled zřejmé, a přišlo by se na ně až po výrobě prvního prototypu. Šetří se tak výrobní náklady a také čas, který by byl potřeba na výrobu dalšího prototypu.

V současné době je k dispozici velké množství vývojových kitů. Obvykle se jedná o zařízení s jedním řídicím mikrokontrolérem. Deska bývá dále osazena několika LED (*Light emitting diode*), tlačítky a případně také několika periferními zařízeními, jako jsou akcelerometr, kompas, paměť, či displej. Záleží na určení vývojového kitu. Neobsazené vývody mikrokontroléru bývají obvykle vyvedeny na pinové konektory. K těm je možné



Vývojový kit bude pro interakci s uživatelem využívat grafický displej, LED a tlačítka. Firmware bude obsahovat grafické prostředí, s jehož pomocí bude možné nakonfigurovat jednotlivá rozhraní pro komunikaci s počítačem. Také zde bude naprogramován terminál, který bude sloužit pro zobrazování příchozích zpráv, nebo počtu přijatých bajtů. LED budou sloužit pro indikaci příchozích dat v terminálu. Každá LED bude indikovat jedno rozhraní.

Vývojový kit bude pracovat ve třech módech.

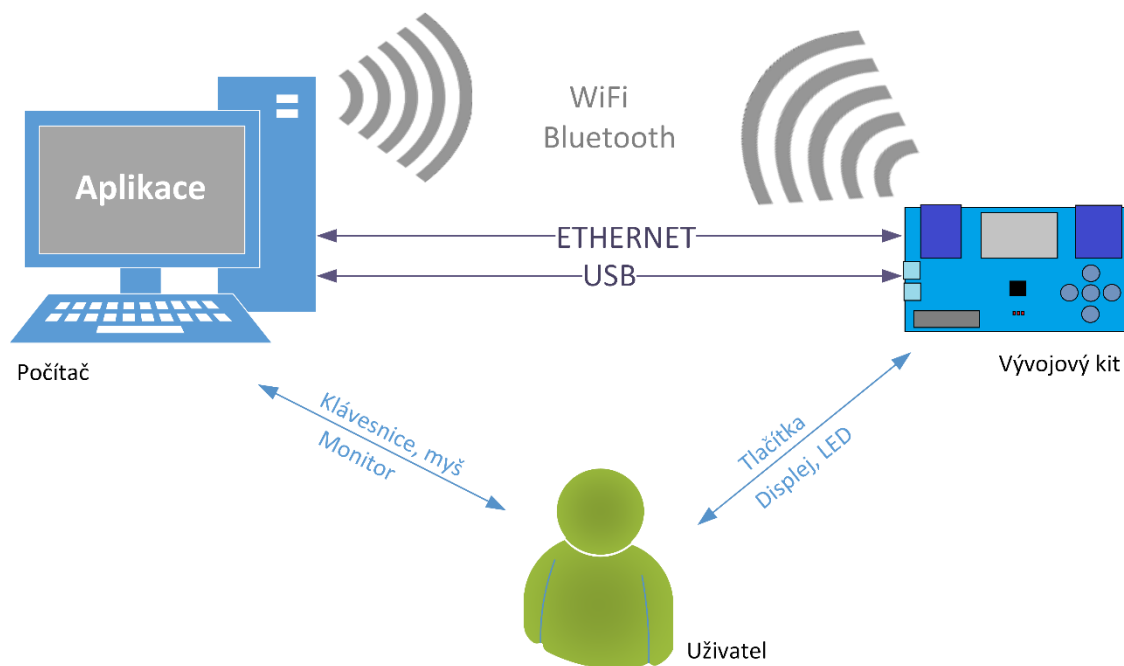
1. ENDPOINT-Koncový bod
2. MOST
3. ECHO-Ozvěna

V módu ENDPOINT bude vývojový kit pracovat pouze s jedním rozhraním. Přijatá data se zobrazí na displeji v terminálu. Při stisku některého z tlačítek se přes zvolené rozhraní odešle zpráva o stisku.

MOST bude komunikovat se dvěma rozhraními. Přijatá data z rozhraní č.1 budou přeposlána na rozhraní č.2 a naopak. V terminálu bude zobrazen počet přeposlaných bajtů jedním a druhým směrem.

V módu ECHO bude vývojový kit komunikovat pouze přes jedno rozhraní. Přijatá data bude odesílat zpět po stejném rozhraní a na displeji bude v terminálu zobrazovat pouze počet přeposlaných bajtů.

Obrázek 1-2 zobrazuje schématické znázornění komunikace mezi počítačem, vývojovým kitem a uživatelem. Na počítači bude spuštěna aplikace, pomocí níž bude možné odesílat a přijímat data přes zvolená rozhraní. Nastavení rozhraní na straně počítače bude také probíhat v této aplikaci. Pro tyto účely poslouží aplikace Hercules.



Obrázek 1-2: Komunikace mezi počítačem, vývojovým kitem a uživatelem

Pokud bude vývojový kit nastaven do módů ENDPOINT nebo ECHO, bude k připojení potřeba spustit aplikaci pouze v jednom okně. V případě módu MOST bude potřeba spustit aplikaci ve dvou oknech a v každém nastavit jiné rozhraní. Odeslaná data z okna č.1 se poté zobrazí v okně č.2 a naopak.

## 2 POPIS KOMUNIKAČNÍCH ROZHRAŇÍ

Mikrokontrolér používá pro komunikaci s počítačem rozhraní Ethernet, WiFi, USB (*Universal serial bus*) a Bluetooth. Ethernet a WiFi jsou příklady rozhraní síťové vrstvy referenčního modelu TCP/IP (*TCP-Transmission control protocol, IP-Internet protocol*) a jsou používána pro připojení a komunikaci v rámci sítě internet. Dále mikrokontrolér pro komunikaci s periferiemi používá rozhraní UART (*Universal asynchronous receiver and transmitter*), I<sup>2</sup>C (*Inter-integrated circuit*) a SPI (*Serial peripheral interface*).

### 2.1 Komunikace mikrokontroléru s počítačem

Počítače se dnes k internetu nejčastěji připojují pomocí rozhraní Ethernet a WiFi. Periferie, jako je například myš, klávesnice nebo tiskárna zase využívají k připojení sběrnici USB. Bezdrátové připojení periferií pak bývá často zprostředkováno pomocí sítě Bluetooth. Vývojový kit bude obsahovat všechny zmíněné rozhraní, aby se s nimi mohli studenti v rámci výuky seznámit a aspoň částečně pochopit jejich funkci.

#### 2.1.1 Rozhraní Ethernet

IEEE 802.3, který je dnes často označován jako Ethernet, definuje strukturu přenášených dat a je nezávislý na přenosovém médiu. Rozhraní bylo původně navrženo pro připojení počítačů do sítě pomocí sdíleného koaxiálního kabelu. Následně byla pro přenos používána telefonní kroucená dvojlinka. Dnes se používá pro drátový přenos čtyř párů kroucených dvojlinek, nebo optických vláken. Původně docházelo při komunikaci ke kolizím, protože mohlo více zařízení vysílat do stejné linky. Dnes je rozhraní Ethernet realizováno jako dvoubodové, plně duplexní, takže ke kolizím nedochází. Spojení je rozvětvováno pomocí rozbočovačů (*Hub*-tyto stále způsobují kolize, takže je od nich odstupováno), síťových prepínačů (*Switch*) a směrovačů (*Router*).

K určení odesílatele a příjemce slouží MAC (*Media access control*) adresa. Adresa je přidělována výrobcem zařízení, má 6 bajtů a je jedinečná. První 3 bajty slouží k identifikaci výrobce, zbylé 3 bajty slouží k identifikaci zařízení. Při odesílání zprávy vysílač nejprve odešle tzv. preamble, série 1 a 0 indikující začátek vysílání. Dále

následuje hlavička. Hlavička obsahuje MAC adresu příjemce a odesílatele a délku odesílaných dat. Následují odesílaná data. Vysílání je zakončeno pomocí patičky, která obsahuje kontrolní součet. MAC slouží k identifikaci konkrétního síťového zařízení v počítačové síti. [2]



Obrázek 2-1: Ethernetová přípojka RJ45 [3]

Nejčastěji se dnes spojení přes rozhraní Ethernet vytváří pomocí kabelů s kroucenými dvojlinkami, které mají na koncích konektory RJ45 (viz Obrázek 2-1). Přenosová rychlost těchto kabelů je 100 Mb/s nebo 1 Gb/s. Vyšších přenosových rychlostí dosahují optická vlákna.

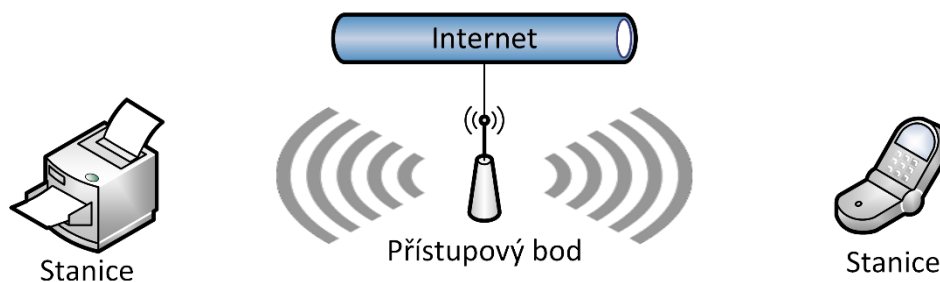
### 2.1.2 Síť WiFi

WiFi využívá bezlicenční pásma 2,4 a 5 GHz. V pásmu 2,4 GHz se může vysílat v USA na 11 kanálech, v Evropě na 13 kanálech, v Japonsku na 14 kanálech. První kanál se nachází na frekvenci 2412 MHz. Každý další kanál se nachází na frekvenci o 5 MHz vyšší, pouze 14. kanál je na frekvenci o 12MHz vyšší. Šířka každého kanálu je 22 MHz. Aby se kanály navzájem rušily, nesmí se protínat v celé šířce pásma. Bez rušení mohou ve stejném prostoru fungovat pouze 3 různé sítě např. na kanálech 1, 6 a 11, resp. 4 sítě v Japonsku na kanálech 1, 6, 11 a 14. Pokud se ve stejném místě nachází více sítí, dochází k jejich vzájemnému rušení a klesá jejich přenosová rychlost. Srovnání jednotlivých standardů WiFi je v tabulce Tabulka 2-1. [4]

Tabulka 2-1: Přenosová frekvence a maximální rychlost jednotlivých standardů [4]

Standard	Frekvence [GHz]	Maximální rychlost [Mb/s]
802.11	2,4	2
802.11b	2,4	11
802.11g	2,4	54
802.11a, 802.11h	5	54

O vzájemnou kompatibilitu WiFi zařízení se stará organizace WiFi aliance, která po splnění všech požadavků uděluje zařízením certifikaci WiFi. V síti WiFi se nejčastěji nacházejí dva typy zařízení: přístupové body a stanice. Obrázek 2-2 zobrazuje sdílení internetu pomocí sítě WiFi. [4]



Obrázek 2-2: Sdílení internetu pomocí WiFi

Přístupový bod spravuje wifi síť. Řídí komunikaci s ostatními zařízeními v síti. Vysílá do okolí SSID (*Service set identifier*) síť, aby ostatní zařízení o její přítomnosti věděly a mohly se k ní připojit. Dále uděluje povolení připojení k síti. Pokud obsahuje DHCP (*Dynamic host configuration protocol*) server, přiděluje připojeným zařízením IP adresy, jinak si IP adresu vyžádá zařízení od nadřazeného DHCP serveru, který bývá k přístupovému bodu obvykle připojen pomocí rozhraní ethernet. Nastavení sítě WiFi je součástí nastavení přístupového bodu. Obvykle se nastavuje frekvence vysílání, kanál, vysílací výkon, SSID, metoda šifrování a přihlašovací údaje. Přístupový bod může být například router nebo switch s možností WiFi. Stanice je zařízení v síti WiFi, které se připojuje k již existující síti pomocí přihlašovacích údajů. Stanice je například mobilní telefon, notebook, tiskárna, nebo chytrá televize.

### 2.1.3 Protokoly TCP a UDP

Aplikace využívají k síťové komunikaci různé protokoly. Například protokol IP se stará o to, aby byla data doručena správnému koncovému zařízení. Protokoly TCP a UDP (*User datagram protocol*) zajišťují doručení dat ke konkrétní aplikaci, běžící na koncovém zařízení. To je zajištěno pomocí hlavičky, která obsahuje zdrojový a cílový port programu. Číslo portu určuje jakýsi kanál, po kterém si aplikace posílají data.

Protokol IP je příkladem druhé vrstvy referenčního modelu TCP/IP. Protokoly TCP a UDP jsou zase příkladem třetí vrstvy. Všechny tři protokoly jsou nezávislé na použitém rozhraní, které tvoří první vrstvu modelu. Proto je možné je použít jak pro komunikaci přes rozhraní Ethernet, tak WiFi, apod.

UDP protokol se využívá u přenosů, kde nezáleží nutně na tom, zda jsou všechny pakety přeneseny. Pokud se některý paket ztratí, příjemce obvykle nevyžaduje jeho opětovné odeslání. Využití je například u streamování videí, internetových hovorů, online her apod.

Komunikace pomocí UDP funguje tak, že si všichni účastníci otevřou port, na kterém naslouchají příchozím zprávám. Jedná se o lokální port. Při odesílání zprávy jinému účastníkovi je potřeba zadat jeho IP adresu a port, na kterém účastník naslouchá. V případě odesílatele se jedná o vzdálený port. Ten musí být shodný s lokálním portem příjemce. Při UDP komunikaci jsou si všichni účastníci rovni, rozdělení rolí se případně řeší až na úrovni aplikace. Odeslané zprávy nejsou potvrzovány, takže odesílatel nemá jistotu, že byla jeho zpráva doručena.

Pokud je pro úspěšný přenos potřeba každý paket, využije se protokol TCP. Zde je navázáno spojení a každý odeslaný paket je příjemcem potvrzen. Pakety jsou číslovány, aby bylo možné zrekonstruovat původní data. Při ztrátě paketu je odeslán požadavek o jeho opětovné odeslání. Zařízení mají v TCP komunikaci dvě role: server a klient.

Server naslouchá příchozím připojením na lokálním portu. Při navázání spojení proběhne mezi klientem a serverem tzv. Three way handshake, neboli třicestné ověření. Nejprve klient žádá o připojení. Server odpoví, že se klient může připojit. Nakonec klient odesílá zprávu, že se připojil, a spojení je navázáno. Veškerá komunikace je dále

ověřována a v případě ztráty dat odešle odesílatel data znovu. Při nastavení TCP serveru se zadává pouze lokální port, na kterém má server naslouchat příchozím zprávám. U TCP klienta je potřeba zadat IP adresu serveru a vzdálený port, na který mají být zprávy odeslány.

## 2.1.4 Rozhraní USB

V devadesátých letech 20. století se různá zařízení připojovala pomocí různých rozhraní. Pokud počítač dané rozhraní neobsahoval, musela se k počítači koupit rozšiřující karta. Navíc ovladače různých zařízení stejného typu nebyly mezi sebou kompatibilní. Zařízení také nebylo možné připojit k počítači za jeho běhu.

Proto byla vyvinuta sběrnice USB. Ta dovoluje připojit až 127 zařízení k jednomu hostujícímu zařízení. Zařízení se mohou na sběrnici připojit i za běhu počítače. Další výhodou je podpora standardizovaných ovladačů, díky kterým je možné nainstalovat většinu zařízení bez potřeby vlastních ovladačů. Tabulka 2-2 znázorňuje, jak se postupně vyvíjel standard USB.

Tabulka 2-2: Vývoj specifikace USB [5]

Verze	Rok	Označení	Maximální rychlost
1.0	1996	Low Speed	1.5 Mbit/s
1.1	1998	Full Speed	12 Mbit/s
2.0	2000	High Speed	480 Mbit/s
3.0 (USB 3.1 Gen 1)	2008	SuperSpeed	5 Gbit/s
3.1 (USB 3.1 Gen 2)	2013	SuperSpeed+	10 Gbit/s

Sběrnice USB využívá k připojení standardizované konektory tří typů: A, B a C. Konektory typu A a B jsou ve třech variantách: Standard, Mini a Micro. Konektory A a B existují již od první specifikace USB a je možné je připojit do zdířky pouze s jednou orientací. Konektor typu C přišel se specifikací 3.0 a je možné jeho orientaci otočit o 180 stupňů. USB 2.0 je zpětně kompatibilní na všechny předchozí standardy a USB 3.0 je zase zpětně kompatibilní s USB 2.0. Konektor typu Standard A má stejný tvar jak pro verzi 3.0 tak pro starší verze, takže je možné do něj připojit zařízení s jinou verzí

specifikace. Ostatní konektory mají vždy jiný tvar pro verzi 3.0 a jiný pro starší verze. Konektor typu C má pouze variantu Standard a verzi specifikace 3.0. Zařízení starší verze specifikace je k němu možné připojit pomocí redukce.

V síti USB se nacházejí tři druhy zařízení: Host, Hub (Rozbočovač) a Device (Zařízení). Host je vždy jeden a řídí veškerou komunikaci na síti. Také přiděluje ostatním zařízením čas, po který mohou ony komunikovat s ním. Žádné zařízení nemůže komunikovat s jiným než s hostitelským zařízením. Host může komunikovat přímo pouze s jedním zařízením. Pokud potřebuje host komunikovat s více zařízeními, je potřeba k němu připojit rozbočovač, který toto umožňuje. Některá zařízení spojují dohromady funkci rozbočovače a koncového zařízení, například klávesnice s rozšiřujícími USB porty.

Až do verze 2.0 probíhala komunikace této sběrnice po dvou vodičích, které tvořily diferenciální pár. Verze 1.0 posílala data pomocí změn napětových úrovní na datových vodičích, od verze 1.1 funguje odesílání dat pomocí proudové smyčky. Komunikace u jednoho diferenciálního páru může v jednu chvíli fungovat pouze v jednom směru. Proto host musí komunikaci řídit a přiděluje ostatním zařízením časové okno, ve kterém mu mohou vysílat svá data. Od verze 3.0 byla sběrnice rozšířena o další dva diferenciální páry, které dovolují současnou oboustrannou komunikaci.

### **2.1.5 Síť Bluetooth**

Bluetooth je označení bezdrátového připojení na krátké vzdálenosti. Stejně jako WiFi funguje v bezlicenčním pásmu na frekvencích v rozsahu 2,402 – 2,480 GHz. Síť vznikla v roce 1994 za účelem bezdrátové komunikace mezi dvěma zařízeními. Původní verze dosahovala pouze nízkých přenosových rychlostí, zařízení byla energeticky náročná a připojení nebylo dostatečně zabezpečené. Postupně docházelo k navyšování rychlosti a spolehlivosti a také ke snižování energetické náročnosti. Dnes slouží Bluetooth nejčastěji k připojení mobilního telefonu k bezdrátovým sluchátkům, reproduktorům nebo handsfree. Dále je možné pomocí Bluetooth odesílat data z jednoho telefonu do druhého telefonu, nebo notebooku. Bluetooth v dnešní době obsahují také automobily, chytré hodinky a podobně.

Zařízení jsou rozdělena do několika tříd podle vysílacího výkonu. Jejich rozdělení je možné vidět v tabulce Tabulka 2-3. Čím vyšší je vysílací výkon, tím vyšší je maximální vzdálenost pro úspěšnou komunikaci. Některé státy povolují používání pouze některých výkonových tříd. Výrobci se musí těmito omezeními řídit. Certifikaci o kompatibilitě Bluetooth zařízení s ostatními zařízeními vydává společnost Bluetooth.

Tabulka 2-3: Vysílací výkon podle třídy zařízení [6]

Třída	Vysílací výkon	
	[mW]	[dBm]
1	100	+20
1,5	10	+10
2	2,5	+4
3	1	0

Existují dva druhy Bluetooth, základní BR/EDR (*BR-Basic rate, EDR-Enhanced data rate*) a nízko energetický BLE (*Bluetooth low energy*). BR/EDR pracuje na 79 kanálech, mezi kterými je rozestup 1 MHz. Je určen pro trvalý a rychlejší přenos větších balíků dat. Rychlost přenosu se pohybuje od 1 do 3 Mbit/s. Běžně zde funguje spojení Bod – Bod, ale je také možné spojit více zařízení do jedné sítě, kde jedno zařízení řídí komunikaci a ostatní poslouchají a případně odpovídají na dotazy.

BLE pracuje na 40 kanálech, mezi kterými je rozestup 2 MHz. Je určeno pro zařízení, která posílají data pouze občas. Přenosová rychlost se pohybuje od 125 Kb/s do 2 Mbit/s. Výhodou BLE je až 100x nižší spotřeba energie oproti BR/EDR. Dále je možné propojit zařízení také do sítě typu Mesh, ve které může kterékoli zařízení začít vysílat, aniž by mu to jiné zařízení muselo povolit.

## 2.2 Komunikace mikrokontroléru s periferiemi

Mikrokontrolér využívá různá rozhraní pro komunikaci s jeho periferními zařízeními. Tato rozhraní obvykle slouží pro komunikaci na krátké vzdálenosti a mají nízkou přenosovou rychlost.

Rozhraní se dělí podle synchronizace vysílače a přijímače na synchronní a asynchronní. Synchronní rozhraní obsahuje kromě datových vodičů také vodič, který přenáší hodinový signál pro synchronizaci všech zařízení, propojených tímto rozhraním. Obvykle je u synchronních rozhraní jedno řídicí zařízení (master), které udává rychlost komunikace a řídí komunikaci. Ostatní zařízení jsou podřízená (slave). Slave zařízení odesílají data po rozhraní pouze tehdy, pokud je o to master požádá.

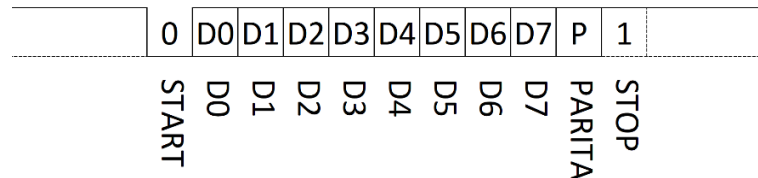
Dále je možné dělit rozhraní podle způsobu přenosu dat na potenciálové a diferenční. U potenciálových rozhraní je spolu s datovými vodiči veden také zemnicí vodič. Logická hodnota přenášených dat je dána napětím mezi datovým a zemnicím vodičem. U diferenčních rozhraní jsou datové vodiče vždy v páru a logická hodnota přenášených dat je určena rozdílem napětí na těchto vodičích. Diferenční rozhraní se používají hlavně na větší vzdálenosti kvůli vyšší odolnosti proti rušení. Pokud jsou data odesílána po jednotlivých bitech, jedná se o sériovou komunikaci. Odesílá-li se několik bitů najednou, jde o paralelní komunikaci.

### **2.2.1 Rozhraní UART**

Jedná se o asynchronní sériové rozhraní. Je určeno ke spojení dvou zařízení. Má oddělenou vysílací a přijímací linku. Vysílací linka TX jednoho zařízení je přivedena na přijímací linku RX druhého zařízení a naopak. Rozhraní dovoluje současné vysílání obou zařízení. Pokud se nevysílá, je na lince hodnota log. 1. K synchronizaci vysílacího a přijímacího zařízení slouží START a STOP bity. Při zahájení vysílání se nejprve odešle START bit (log.0). Za ním následuje domluvený počet datových bitů. Obvykle se odesílá 8 bitů (1 bajt). První datový bit je LSB (*Least significant bit*) a poslední je MSB (*Most significant bit*). LSB má v odesílaných datech nejmenší váhu, MSB největší. Následuje paritní bit. Vysílání je zakončeno STOP bitem (log. 1). Někdy se odesílá také 1,5 nebo 2 stop bity, podle nastavení vysílače.

Paritní bit slouží k detekci chyby při čtení dat. Parita může být sudá, lichá, nebo žádná. Paritní bit pro sudou paritu se spočítá jako xor jednotlivých datových bitů. Negací tohoto se získá paritní bit pro lichou paritu. U sudé parity je sudý počet datových bitů v log. 1, u liché parity je lichý. Kontrola paritního bitu dokáže detekovat vždy pouze lichý počet chyb. Vysílač i přijímač musí mít nastavený stejný typ paritního bitu, aby vysílač

ve funkčním stavu posílal stejný paritní bit, jaký očekává přijímač. Pokud se s daty neposílá žádný paritní bit, obvykle kontrolu správnosti zajišťuje nadřazený protokol (např. Modbus). V některých případech slouží paritní bit, místo kontroly správnosti, k rozlišení dvou typů dat (např. 0 – data, 1 – příkaz), nebo k určení začátku přenosu většího balíku dat. Popsaný rámeček rozhraní UART zobrazuje Obrázek 2-3.



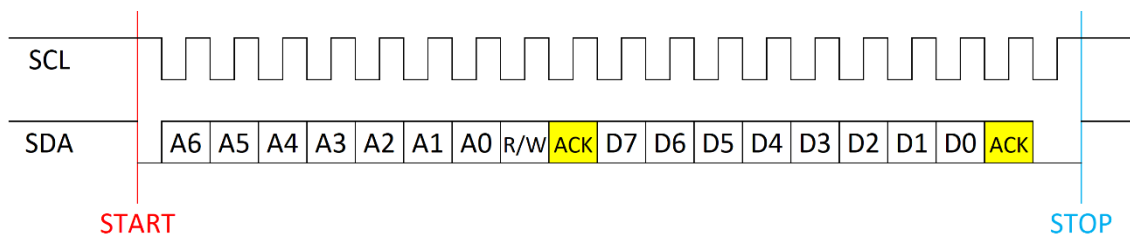
Obrázek 2-3: Rámeček rozhraní UART

Modulační rychlost rozhraní se udává v jednotce Baud. Jedná se o maximální počet změn úrovně na komunikační lince za sekundu. Skutečná přenosová rychlost je nižší a je závislá na nastavení rozhraní (počet datových bitů, parita, počet stop bitů).

## 2.2.2 Sběrnice I<sup>2</sup>C

Pro obousměrnou komunikaci jednoho řídicího zařízení s více podřízenými zařízeními na krátkou vzdálenost byla vynalezena sběrnice I<sup>2</sup>C. Pomocí I<sup>2</sup>C je možné připojit různé druhy integrovaných obvodů, například paměti, RTC (*Real-time clock*), IO (*Input output*) extendery, řadiče displeje, digitální potenciometry apod. [7]

Komunikace (viz Obrázek 2-4) probíhá po dvou vodičích, SDA (*Synchronous data*) a SCL (*Synchronous clock*). Vodič SDA slouží k odesílání dat. Vodič SCL přenáší hodinový signál z řídicího zařízení k podřízeným a slouží k jejich synchronizaci. Komunikace je vždy zahájena START podmínkou, kdy SDA mění stav z log. 1 do log. 0, zatímco SCL je v log. 1. Kromně START existuje ještě STOP podmínka (změna SDA z log. 0 do log. 1). V ostatních případech se může stav SDA měnit pouze, pokud je stav SCL v log. 0. Po odeslání START se odešle několik bajtů dat (první bit je MSB), každý bajt je zakončen ACK bitem, kdy slave zařízení může odeslat potvrzení o přijetí. Nakonec je odeslána STOP podmínka. Obvykle se v prvním bajtu nachází 7bitová adresa slave zařízení a je zde specifikováno, zda se jedná o čtení nebo zápis.



Obrázek 2-4: Rámec I<sup>2</sup>C, odeslání adresy a dat

Na vodič SDA jsou zařízení připojena pomocí vstupně výstupních pinů typu OC (*Open collector*) nebo OD (*Open drain*). Při čtení jsou tyto piny vždy ve stavu vysoké impedance. To znamená, že se chovají jako by k lince vůbec připojeny nebyly. Při odesílání dat jsou buď ve stavu vysoké impedance a na SDA bude log. 1, nebo zkratují SDA na zem a bude zde log. 0. Jelikož výstupy typu OC a OD nedokáží na výstup přivést kladné napětí, musí být napětí na vodič SDA i SCL přivedeno pomocí pullup rezistorů, přes které jsou tyto vodiče přivedeny na kladné napájecí napětí.

Tabulka 2-4: Maximální přenosové rychlosti sběrnice [7]

Mód sběrnice	Maximální přenosová rychlost	Druh komunikace
Standard	100 kbit/s	Obousměrná
Fast	400 kbit/s	Obousměrná
Fast Plus	1 Mbit/s	Obousměrná
High speed	3.4 Mbit/s	Obousměrná
Ultra Fast	5 Mbit/s	Jednosměrná

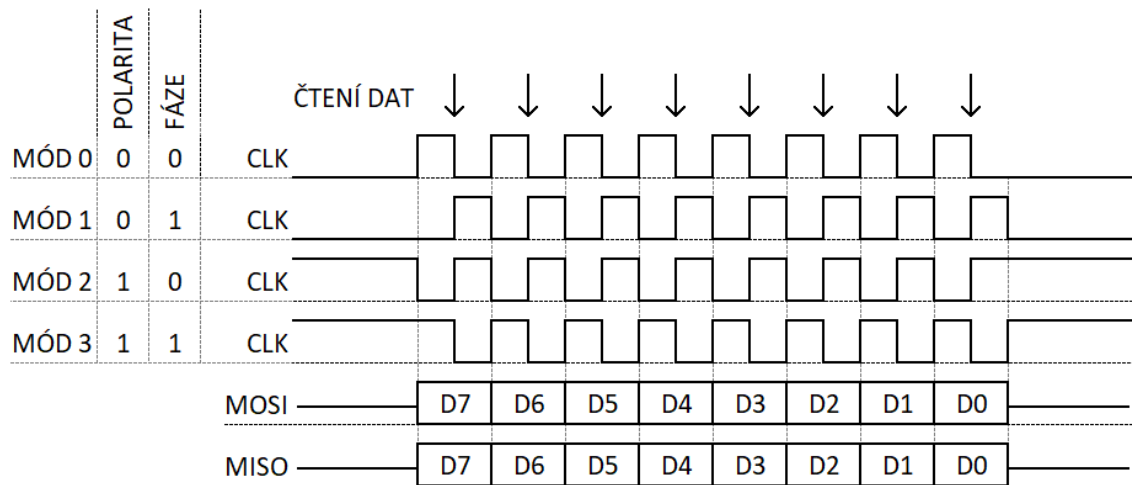
Velikost pullup rezistorů se volí podle požadované přenosové rychlosti sběrnice a maximálního dosahu. Čím nižší je hodnota rezistoru, tím rychlejší je změna ze stavu log. 0 do log. 1, a tím vyšší může být přenosová rychlost a maximální dosah. Pullup rezistory však nelze použít libovolně malé. Jejich minimální velikost se odvíjí od maximálního proudu, který dokáže každé připojené zařízení sepnout. Čím nižší je zvolená přenosová rychlost, tím vyšší je dosah. Rozdělení maximálních rychlostí podle módu sběrnice shrnuje Tabulka 2-4.

### 2.2.3 Sběrnice SPI

Je to sériové synchronní rozhraní, ve kterém je jedno řídicí a několik podřízených zařízení. Všechna zařízení jsou obvykle připojena na stejnou sběrnici. Master určuje pomocí povolovacího signálu, se kterým zařízením chce komunikovat. Každé zařízení má svůj povolovací signál. Frekvenci hodinového signálu určuje master, pohybuje se od 100 kHz do 20 MHz.

Sběrnice má oddělenou datovou linku pro data, které posílá master a data, které posílá slave. Je tak možné zároveň vysílat data oběma směry.

SPI dokáže pracovat ve čtyřech módech, které se liší podle toho, jakým způsobem je řízen hodinový signál. Polarita znamená logickou hodnotu, ve které je CLK (*Clock*) linka po dobu, kdy se nevysílá. Fáze určuje, zda se polarita CLK linky poprvé mění na začátku nebo až v půlce periody CLK signálu. Příklady jednotlivých módů jsou na obrázku Obrázek 2-5.

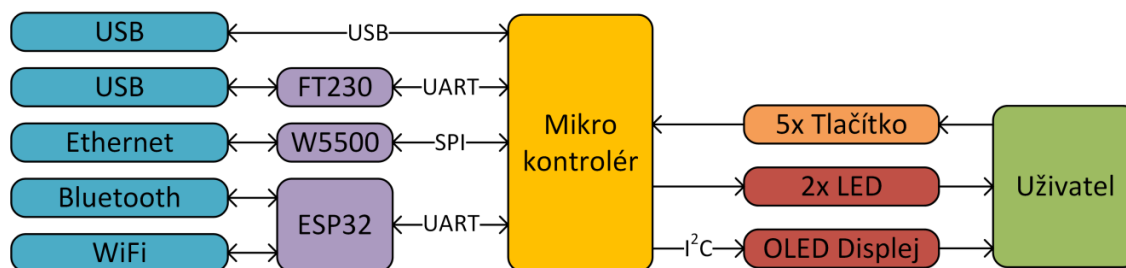


Obrázek 2-5: Ukázka řízení CLK signálu v různých módech. [8].

Některé zařízení dokáží komunikovat v módu „Daisy chain“. V tomto případě se vodič MOSI (*Master out, slave in*) připojuje z master zařízení pouze do prvního slave zařízení. Vodič MISO (*Master in, slave out*) slave zařízení je přiveden na vodič MOSI následujícího zařízení. MISO posledního zařízení je připojeno na MISO master zařízení.

## 3 NÁVRH ZAŘÍZENÍ

Jednou z možností realizace tohoto vývojového kitu bylo použití mikrokontroléru, který v sobě integruje většinu rozhraní pro připojení k počítači, zbytek by byl realizován pomocí modulů. Mikrokontrolér by však byl drahý a složitý. Navíc by poznatky z následné diplomové práce byly omezeny pouze na daný mikrokontrolér.



Obrázek 3-1: Blokové schéma realizace kitu

Proto byl nakonec použit mikrokontrolér, který integruje pouze rozhraní USB, zbytek rozhraní je zprostředkován pomocí modulů, se kterými mikrokontrolér komunikuje přes rozhraní UART a SPI. Obrázek 3-1 znázorňuje blokové schéma zapojení mikrokontroléru s moduly a ostatními součástkami.

### 3.1 Výběr součástek

Existuje velké množství modulů, které je možné použít pro komunikaci mikrokontroléru s různými rozhraními. Důležité bylo vybrat takové moduly, které jsou moderní a jsou často používány v různých projektech. K těmto modulům je na internetu dostupných mnoho příkladů použití. Také je u nich menší pravděpodobnost, že budou v krátké době nahrazeny novější verzí.

#### 3.1.1 Mikrokontrolér

Pro vývojový kit byl vybrán mikrokontrolér ARM s 32bitovým jádrem Cortex-M4. Mikrokontroléry ARM jsou dnes velmi moderní a používají se v celém spektru elektroniky od drobných a jednoduchých řešení až po složité systémy. Mikrokontroléry ARM jsou nejčastěji používány také v mobilních telefonech a jsou také přítomny

ve výkonnějších čípech FPGA (*Field Programmable Gate Array*). Nejdůležitější parametry použitého obvodu jsou popsány níže, viz Tabulka 3-1. Mikrokontrolér je připojen na externí krystal 8MHz, tato frekvence je vnitřní násobičkou zvýšena na 72MHz.

Tabulka 3-1: Parametry mikrokontroléru STM32F302K6U8 [9]

Parametr	Hodnota
Napájecí napětí	2 – 3,6 V
Pracovní teplota	-40 až +85 °C
Pouzdro	UFQFPN32
Taktovací frekvence	Až 72 MHz
Paměť Flash/RAM	64 kB / 16 kB
Rozhraní	CAN, I2C, SPI, USART, USB, I2S

Mikrokontrolér je možné naprogramovat pomocí externího programátoru ST-Link. Ten se připojuje pomocí 20 pinového konektoru na DPS. Programátor ST-Link dovoluje také ladění programu za jeho běhu.

### 3.1.2 Převodník USB na UART

Rozhraní USB je na vývojovém kitu realizováno hned dvěma způsoby. Jednak přímo pomocí mikrokontroléru, ale také nepřímo pomocí převodníku z USB na UART. Výhoda přímého připojení USB je úspora převodníku, a hlavně mnohem větší rozsah funkčnosti. Mikrokontrolér se tak může chovat jako sériový port, ale také jako myš, klávesnice, nebo paměťové zařízení. Nevýhodou je náročnější implementace ve zdrojovém kódu a větší náročnost na paměť programu i výkon mikrokontroléru. Naopak řešení pomocí externího převodníku USB na UART je zase jednoduché na implementaci a nenáročné pro mikrokontrolér.

Tabulka 3-2: Parametry převodníku FT230XQ [10]

Parametr	Hodnota
Napájecí napětí	3,3 nebo 5 V
Proud (aktivní / spánek)	8 mA / 125 uA
Pracovní teplota	-40 až +85 °C
Velikost bufferu (RX / TX)	512 B / 512 B
Baudrate	300 baud až 3 Mbaud

Pro vývojový kit byl vybrán obvod FT230XQ, který je velmi jednoduchý na implementaci a dodává se v malém pouzdru QFN16. Disponuje také čtyřmi rozšiřujícími piny, které je možné využít například pro připojení LED k indikaci vysílání dat po lince UART. K tomuto účelu jsou piny využity i na vývojovém kitu. Základní technické parametry převodníku shrnuje Tabulka 3-2.

### 3.1.3 Modul pro připojení k Bluetooth a WiFi

Spojení počítače s vývojovým kitem pomocí Bluetooth a WiFi je realizováno pomocí SOC (*System on chip*) ESP32. Mikrokontrolér s ním komunikuje pomocí AT (*attention*) příkazů přes rozhraní UART. Výhodou ESP32 je to, že integruje dohromady dvě rozhraní a poskytuje jejich různé nastavení. Dále je celý systém velmi malý a má nízkou spotřebu.

ESP32 v sobě integruje výkonný dvoujádrový mikrokontrolér, rozhraní BLE, Bluetooth a WiFi. Kromě toho obsahuje další rozhraní jako jsou UART, SPI, I2C. Vzhled modulu ESP32 zobrazuje Obrázek 3-2 a jeho technické parametry popisuje Tabulka 3-3.

Firmware mikrokontroléru ESP32 je možné přeprogramovat pomocí rozhraní UART0. Ladění programu je pak možné přes JTAG (*Joint Test Action Group*). Od výroby je v něm nahrán firmware, který naslouchá na rozhraní UART2 a řídí se AT příkazy, které dostává od nadřazeného systému. Pomocí těchto příkazů je možné provádět nastavení BLE a WiFi, odesílat a přijímat data z těchto rozhraní.



Obrázek 3-2: SOC ESP32-WROOM-32 [11]

WiFi oddíl modulu může pracovat v režimu stanice, kdy se modul připojí k existující WiFi síti. Dále může pracovat jako přístupový bod, vysílá WiFi signál a můžou se k němu připojit ostatní stanice. Také dokáže spojit oba tyto režimy a fungovat zároveň jako stanice i přístupový bod (*AP-Access point*).

Tabulka 3-3: Parametry modulu ESP32 [12]

Parametr	Hodnota
Napájecí napětí	2,7 až 3,6 V
Proud	Až 500 mA (při vysílání WiFi i Bluetooth) Průměr 80 mA
Pracovní teplota	-40 až +85 °C
Frekvence procesoru	80 až 240 MHz
Paměť (RAM, ROM, FLASH)	520 KB, 448 KB, 4 MB
Rozhraní modulu	SD karta, UART, SPI, SDIO, I2C, LED PWM, I2S, IR, GPIO, ADC, DAC
Protokoly WiFi	TCP, UDP
Specifikace Bluetooth	4.2 BR / EDR, BLE

Bluetooth oddíl lze nastavit jako server nebo klient. Server vytváří připojení, ke kterému se může připojit jeden nebo více klientů. Vysílací výkon WiFi a BLE je možné konfigurovat.

## AT příkazy

Některé zařízení nebo moduly je možné konfigurovat pomocí textových AT (*Attention*) příkazů. Pomocí AT příkazů jsou konfigurovány například GSM (*Groupe spécial mobile*), Bluetooth a Wifi moduly. Dříve sloužily ke konfiguraci telefonních zařízení. AT příkazy se posílají obvykle přes rozhraní RS232 nebo UART.

Příkaz vždy začíná zkratkou AT, následuje znak „+“ a za ním název příkazu, případně parametry. Každý příkaz musí být zakončen pomocí znaků pro nový řádek (CR-LF, hexadecimálně 0x0a 0x0d). Existují 4 druhy AT příkazů, které shrnuje Tabulka 3-4.

Tabulka 3-4: Přehled typů AT příkazů

Typ příkazu	Formát	Popis
Test	AT+<x>=?	Dotazuje se na parametry příkazu <x> a jejich rozsah
Dotaz	AT+<x>?	Dotazuje se na aktuální hodnoty parametrů příkazu <x>
Nastavení	AT+<x>= <...>	Nastavuje hodnoty parametrů příkazu <x> a spouští příkaz
Provedení	AT+<x>	Spustí příkaz <x> bez parametrů

Některé obecné AT příkazy má většina zařízení společné. Jedná se o příkazy pro nastavení rozhraní, dotaz na pozornost, a reset. Příklady některých AT příkazů a popis, co dané příkazy dělají, zobrazuje Tabulka 3-5.

Tabulka 3-5: Příklady AT příkazů

Příkaz	Popis
AT	Dotaz na pozornost. Slouží ke kontrole, že je zařízení aktivní a komunikuje
AT+RST	Restartuje zařízení
AT+GMR	Dotaz na verzi firmwaru s AT příkazy
AT+UART_CUR	Vyčítání a změna nastavení komunikačního rozhraní. Po restartu zařízení se nastaví na hodnoty UART_DEF
AT+UART_DEF	Vyčítání a změna výchozího nastavení komunikačního rozhraní

Modul ESP32 má k dispozici ke stažení firmware s AT příkazy, který si koncový uživatel může podle potřeby upravit a přidat k sadě příkazů vlastní příkazy. Výchozí firmware má k dispozici kromě obecných AT příkazů také příkazy pro nastavení WiFi a BLE. Příkazy pro nastavení Bluetooth bohužel výchozí firmware nepodporuje. Na internetu se však nachází upravený firmware s příkazy pro Bluetooth.

Firmware ESP32 je možné přeprogramovat i na vývojovém kitu pomocí připojení převodníku UART na konektor JP4. Při resetu kitu je potřeba držet stisknuté tlačítko S7, modul tak přejde do režimu, kdy očekává nahrání nového firmwaru.

### 3.1.4 Modul pro připojení k Ethernetu

Pro komunikaci přes rozhraní Ethernet je použit modul s čipem W5500. Byl vybrán hlavně díky svým malým rozměrům a snadnému připojení do celého vývojového kitu. Vzhled modulu zobrazuje Obrázek 3-3, jeho technické parametry pak Tabulka 3-6.



Obrázek 3-3: Modul s obvodem W5500 [13]

Obvod podporuje funkci Auto Negotiation, což znamená, že dokáže automaticky omezit svou přenosovou rychlost, pokud komunikuje s pomalejším zařízením. Bohužel nedisponuje funkcí Auto MDIX (*Medium dependent interface crossover*), která ošetřuje překřížení RX a TX linek. Pokud počítač, ke kterému se vývojový kit připojuje má tuto funkci, může se pro propojení použít jak přímý, tak křížený kabel. Pokud touto funkcí nedisponuje ani počítač, je nutné použít křížený kabel.

Tabulka 3-6: Parametry obvodu W5500 [14]

Parametr	Hodnota
Napájecí napětí	2,97 - 3,63 V
Nejvyšší spotřeba (100M vysílání, 3,3V, 25°C)	132 mA
Počet Socketů	8
Paměť RX/TX bufferů (celková / jednoho)	32 KB / 2 KB
Protokoly	TCP, UDP, ICMP, IPv4, ARP, IGMP
Přenosová rychlost	10 M / 100 M, Half/Full duplex

Mikrokontrolér komunikuje s obvodem W5500 pomocí rozhraní SPI v módu 0. Maximální frekvence SPI, kterou obvod zvládne, je 80MHz. To je větší frekvence, než jakou dokáže vyvinout ARM. Obvod disponuje jedním výstupním pinem přerušení. Když toto přerušení zachytí mikrokontrolér, z registru obvodu W5500 si přesně vyčte, o jaké přerušení se jedná a provede požadovanou akci.

### 3.1.5 Displej OLED

Zatímco LCD má celou plochu podsvícenou a jednotlivé pixely slouží jako stínítko, OLED (*Organic light emitting diode*) displej podsvícení nemá, protože každý pixel je zdrojem světla. Spotřeba OLED displeje je tedy úměrná počtu rozsvícených pixelů, zatímco spotřeba LCD je stále stejná. Tím, že OLED displej nemá podsvícení, dosahuje také mnohem lepšího kontrastu než LCD.

Jako zobrazovací jednotka byl vybrán grafický OLED displej OLED-128O064D-WPP3N00000. Displej má rozlišení 128x64 pixel, úhlopříčku 1.3 palce a je černobílý. Displej se rozsvěcuje po jednotlivých řádcích, takže jednotlivé pixely svítí se střídou 1/64.



Obrázek 3-4: OLED displej [15]

Obrázek 3-4 ukazuje, jak displej vypadá, jeho základní technické parametry vypisuje Tabulka 3-7. Plochý spoj displeje je k vývojovému kitu připájen.

Tabulka 3-7: Parametry OLED displeje [16]

Parametr	Hodnota
Napájecí napětí	2,8 - 3,3 V
Rozlišení	128 x 64 px
Rozhraní	I <sup>2</sup> C, SPI, 6800, 8080
Řadič	SSD1306

Displej je k DPS připojen pomocí 30 vodičového ohebného pájecího konektoru. První a poslední vodič konektoru nemají žádnou elektrickou funkci, slouží k tomu, aby se snížilo mechanické namáhání ostatních vodičů. Dvojice vodičů 2, 3 a 4, 5 slouží k připojení externích 2,2uF kondenzátorů k nábojové pumpě displeje.

Tabulka 3-8: Nastavení rozhraní OLED displeje [16]

Popis vodiče	Číslo vodiče	Rozhraní				
		I <sup>2</sup> C	3-wire SPI	4-wire SPI	68xx	80xx
BS0	10	0	1	0	0	0
BS1	11	1	0	0	0	1
BS2	12	0	0	0	1	1

Vodiče 10 až 12 plochého spoje displeje slouží k nastavení rozhraní pro komunikaci mezi mikrokontrolérem a displejem, viz Tabulka 3-8. V tomto případě komunikuje mikrokontrolér s displejem pomocí rozhraní I<sup>2</sup>C, takže na vodič BS1 bude přivedena hodnota log. 1, na vodiče BS0 a BS2 pak log. 0. Adresa displeje je závislá na vstupu SA0. Na tomto vstupu bude log. 0, takže adresa bude 0x3C.

### 3.1.6 Napájení

Všechny použité komponenty jsou napájeny napětím 3,3 V. Aby bylo možné vývojový kit spolehlivě napájet, bylo potřeba jej osadit stabilizátorem, který je schopný dodat dostatečný proud. K tomuto účelu byl použit stabilizátor LD1086DT33TR, který má na svém výstupu napětí 3,3 V a dokáže dodat proud až 1,5 A. [17]

$$P_{STAB} = \frac{U_{IN} - U_{OUT}}{I_{OUT}} = \frac{5 - 3,3}{1} = 1,7 W \quad (3-1)$$

Stabilizátor je napájen z USB portu napětím 5 V, vzniká na něm úbytek 1,7V. Za předpokladu maximálního proudového odběru 1 A by maximální ztrátový výkon stabilizátoru mohl činit 1,7W (viz vzorec 3-1). Aby nedocházelo k přehřívání stabilizátoru, bude DPS přizpůsobena tak, aby bylo teplo ze stabilizátoru lépe odváděno do okolí. Podle specifikace by měl USB port zvládnout dodat maximálně 0,5 A. Z toho důvodu je DPS opatřena piny pro přivedení externího napájení.

## 3.2 Návrh zapojení

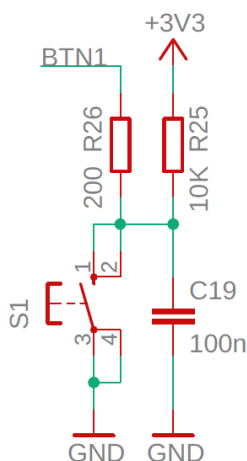
Tato kapitola popisuje, jaká obvodová řešení jsou použita ve schématu zapojení a vysvětluje jejich funkci. Nachází se zde také výpočty hodnot jednotlivých součástek. Schéma zapojení a DPS byly navrženy v programu Eagle 9.2.2 free. Tato neplacená verze podporuje návrh jednovrstvých a dvouvrstvých plošných spojů o maximálních rozměrech 100 x 80 mm. Schéma není omezeno počtem součástek, ale je zde omezený počet listů schémat pro jeden projekt. Velikost jednoho listu není nijak omezena, ale schéma může být nepřehledné při velkém počtu součástek na jednom listu.

### 3.2.1 Ošetření tlačítek

Proces spínání a rozpínání tlačítka není nikdy dokonalý. Při jeho stisku nebo uvolnění dochází k několikanásobnému sepnutí a rozepnutí, dokud se poloha mechanických částí tlačítka neustálí. Ustálení trvá jednotky až desítky milisekund v závislosti na mechanickém uspořádání tlačítka a také na jeho kvalitě. Vývojový kit je osazen pěti běžnými tlačítky, která vytvářejí zákmity. Pokud by se tyto zákmity nijak neošetřily, mikrokontroler by mohl jeden stisk tlačítka vyhodnotit jako několik stisků.

Jeden způsob ošetření zákmitů je hardwarově pomocí RC (*Resistor capacitor*) článku, jehož zapojení zobrazuje Obrázek 3-5. Při stisku tlačítka dochází k vybití kondenzátoru a mikrokontrolér detekuje sepnutí. Při následných zákmitech dochází ke krátkodobému uvolnění tlačítka a kondenzátor se začne nabíjet. Vždy když se kontakty tlačítka opět sepnou, kondenzátor se opět vybije, takže se nestihne nabít na napětí, při kterém mikrokontrolér detekuje rozepnutí. Až při uvolnění tlačítka se kondenzátor nabije a mikrokontrolér poté detekuje rozepnutí. V dnešní době se toto řešení z ekonomických důvodů téměř nepoužívá. RC článek není osazen ani na vývojovém kitu, ale pro účely testování je dobré mít možnost jej osadit.

Druhé, dnes již používanější řešení ošetření zákmitu tlačítka je softwarové. Mikrokontrolér periodicky vyčítá stav tlačítka a změnu stavu považuje za platnou až po vyčtení N vzorků stejného stavu. Řešení je levné a téměř nezatěžuje výpočetní výkon procesoru.



Obrázek 3-5: Zapojení tlačítka

Tlačítko je k pinu mikrokontroléru připojeno přes rezistor 220 Ω. Rezistor slouží jako ochrana proti nadměrnému zatížení pinu. Pokud by byl pin omylem nastaven jako výstup a měl na výstupu napětí 3,3 V, došlo by při stisku tlačítka ke zkratování pinu na zem. Výstupní obvod pinu by se nadměrným proudem mohl poškodit. Z tabulky č. 21 katalogového listu mikrokontroléru [9] lze vyčíst, že maximální proud odebíraný z jednoho pinu je 25 mA. Minimální hodnota rezistoru, potřebná na ochranu před nadměrným proudem se vypočítá z Ohmova zákona (vzorec 3-2).

$$R_{min} = \frac{U_{CC}}{I_{max}} = \frac{3,3}{25 * 10^{-3}} = 132 \Omega \quad (3-2)$$

Nakonec byly pro ochranu pinů použity rezistory 220 Ω, které splňují podmínku  $R_{min}$  a navíc již jsou rezistory stejné hodnoty použity i na jiných místech ve schématu. Ochranné rezistory jsou kromě tlačítek použity také u rozhraní UART. Zde slouží jako ochrana pinů mikrokontroléru i pinů ostatních zařízení.

### 3.2.2 Zapojení LED

Zapojení svítivých diod vychází z doporučeného zapojení pro obvod FT230. Ten využívá zapojení s inverzní logikou. To znamená, že pokud je na výstupu pinu s LED hodnota log.0, LED svítí, při log.1 dioda nesvítí. Stejná logika je použita i u mikrokontroléru a to jak pro LED, tak pro tlačítka. K diodám jsou v sérii připojeny rezistory 220 Ω. Předpokládaný úbytek napětí na LED podle katalogového listu je 1,7V. [18] Podle vzorce 3-3 by v sepnutém stavu měl diodou protékat proud 7,3 mA.

$$I_{LED} = \frac{U_{CC} - U_{LED}}{R} = \frac{3,3 - 1,7}{220} = 7,3 \text{ mA} \quad (3-3)$$

Pozdějším měřením bylo zjištěno, že na LED vzniká úbytek napětí 2,08V a teče zde proud 5,5mA.

### 3.2.3 Pullup rezistory

Pro nastavení výchozího stavu některých vodičů do log. 1 slouží pullup rezistory. Hodnoty těchto rezistorů jsou obvykle 10 KΩ. Pullup rezistory jsou přítomny u všech tlačítek. Dále se nacházejí na vodičích sběrnic SPI a I<sup>2</sup>C. Právě pro funkci sběrnice I<sup>2</sup>C jsou pullup rezistory velmi důležité, protože zařízení, komunikující po této sběrnici, jsou obvykle typu OC nebo OD, takže dokáží nastavit pouze stav log. 0, stav log. 1 musí být

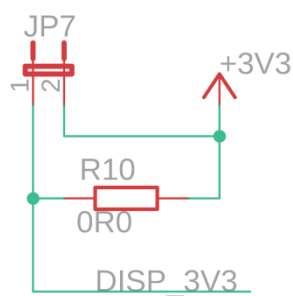
nastaven právě externím pullup rezistorem. Čím vyšší je rychlost komunikace po I<sup>2</sup>C, tím nižší hodnoty rezistorů se doporučují používat.

### 3.2.4 Blokovací kondenzátory

Většina integrovaných obvodů odebírá v krátkých špičkách mnohem větší proud, než je proud souvislý. Tyto špičky jsou způsobeny nejčastěji přepínáním tranzistorů v integrovaném obvodu. Frekvence špiček je obvykle dána taktovací frekvencí integrovaného obvodu. Pokud by se tyto špičky nijak neošetřily, došlo by k šíření šumu od integrovaného obvodu po napájecích vodičích. To by mohlo ovlivňovat funkci ostatních komponent. Navíc při chvilkovém zvýšení odběru proudu může dojít k takovému poklesu napětí na napájecím pinu, že se může integrovaný obvod resetovat, nebo může přestat správně fungovat. Z toho důvodu jsou co nejbliže k napájecím pinům integrovaných obvodů umísťovány blokovací kondenzátory. Ty mívají obvykle hodnotu v rozsahu 100nF až 10uF. Jsou určeny k pokrytí krátkodobých špiček, zabráňují šíření rušení po napěťových vodičích a také omezují vyzařování elektromagnetického rušení do okolí.

### 3.2.5 Měření proudu

Aby bylo možné změřit, jaký proud protéká celým obvodem i jednotlivými součástmi, jsou napájecí vodiče rozděleny pomocí pinových konektorů, viz Obrázek 3-6.



Obrázek 3-6: Pinový konektor pro měření proudu

Tyto konektory jsou osazeny zkratovací propojkou, nebo jsou vodiče spojeny pomocí rezistoru 0R0. Pokud je potřeba měřit proud, který vodičem protéká, místo zkratovací propojky se připojí ampérmetr. Rezistor 0R0 nesmí být při měření osazen.

### **3.2.6 Testovací piny**

Na sběrnice UART, SPI a I<sup>2</sup>C jsou připojeny pinové konektory, takže je možné připojit si snadno kteroukoli linku sběrnice na osciloskop a sledovat průběhy napětí při odesílání dat. Sběrnice USB pinové konektory nemají, aby nedošlo k jejich zarušení. Každý pinový konektor pro testování sběrnic má první vývod přiveden na zem, ostatní vývody jsou linky sběrnic. Je tomu tak, aby bylo možné snadno připojit zemní vodič osciloskopu.

### **3.2.7 Připojení diferenciálního páru USB**

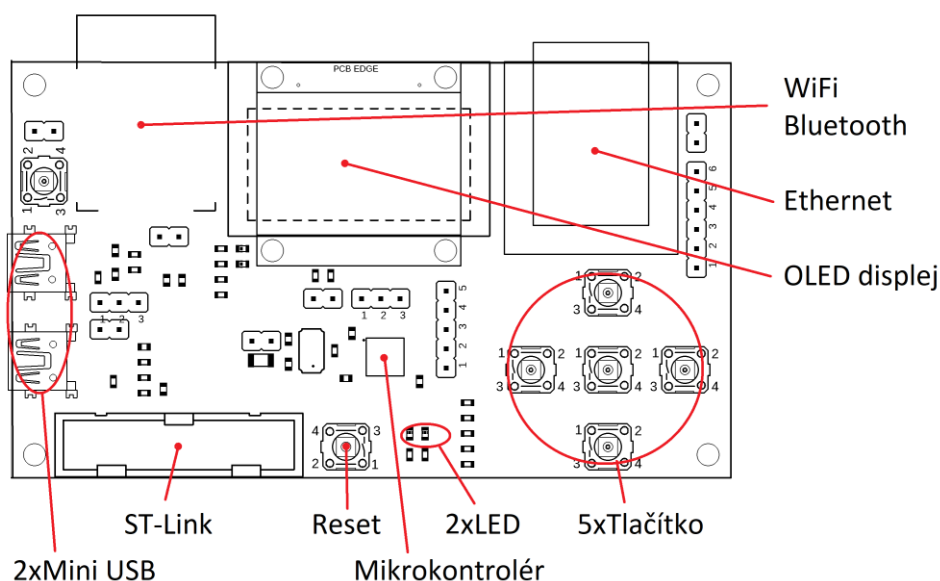
Při vedení diferenciálního páru sběrnice USB je potřeba vést oba vodiče vedle sebe a pokud možno daleko od vodičů, které mohou být zdrojem rušení. Dále je potřeba zajistit, aby byly oba vodiče přibližně stejně dlouhé. Maximální tolerovaná doba nesouběhu těchto vodičů je 100 ps. Pokud by se elektrony ve vodičích šířily rychlostí světla, mohly by se délky vodičů lišit až o 3 cm. Reálně je rychlost elektronů ve vodiči asi poloviční, takže je potřeba dodržet rozdíl maximálně 15 mm. [19]

### **3.2.8 OLED displej**

Vývojový kit podporuje osazení jak modulu s OLED displejem, tak i samotný OLED displej. Modul s displejem obsahuje všechny součástky pro jeho správnou funkci, pokud bude na DPS osazen samotný displej, jsou jeho podpůrné součástky osazeny přímo na DPS zespod.

## **3.3 Návrh desky plošného spoje**

Deska plošného spoje byla navržena jako dvouvrstvá. Návrh začal rozmístěním nejdůležitějších součástek a umístěním montážních děr do každého rohu. Díry budou sloužit k přišroubování distančních sloupků, aby plošný spoj neležel na součástkách, které budou připájeny na spodní straně. Rozměry DPS jsou 95 x 55 mm. Rozmístění součástek ukazuje Obrázek 3-7. Mini USB konektory, ST-Link konektor a modul W5500 musí být na okraji desky, aby bylo možné do nich připojit kabel. Stejně tak OLED displej, protože z něj vede ohebný spoj, který bude přehnut přes okraj a bude připájen na spodní straně DPS. Na kraji musí být také modul ESP32 tak, aby anténa byla až za okrajem DPS.



Obrázek 3-7: Rozložení součástek na DPS

V pravém dolním rohu se nacházejí tlačítka pro ovládání vývojového kitu. Jejich rozmístění je do kříže, aby měly tlačítka 1 až 5 funkce nahoru, doprava, dolů, doleva a potvrzení. Mikrokontrolér je umístěn do středu vývojového kitu, aby měl co nejkratší cestu ke všem periferiím. Otočen je tak, aby bylo zajištěno minimální křížení vodičů. Stabilizátor má pod sebou umístěny prokovy na druhou stranu DPS, kde se nachází měděná plocha. Dochází tak k jeho lepšímu chlazení.

Cesty mají minimální tloušťku 10 milů, minimální mezera mezi vodiči je 8 milů. Prokovy mají velikost 0,35 mm. Napájecí vodiče, kterými poteče větší proud jsou širší, aby na nich byl menší úbytek napětí. Pokud tyto vodiče přecházejí z jedné strany desky na druhou, je k tomu použito více prokovů.

## 4 NÁVRH PROGRAMOVÉHO VYBAVENÍ

Ke konfiguraci a demonstraci funkce všech modulů byl naprogramován FW, jehož zdrojový kód je možné rozdělit do tří kategorií. První kategorii tvoří kód, který se stará o ovládání jednotlivých rozhraní a součástí mikrokontroléru. Zde patří například obsluha komunikace UART, I<sup>2</sup>C, SPI. Druhá kategorie se stará o komunikaci s moduly, které jsou k mikrokontroleru připojeny, prostřednictvím funkcí z první kategorie. Třetí kategorií je GUI (*Graphic user interface*), které se zajišťuje výměnu dat mezi kitem a uživatelem. Zpracovává vstupy z tlačítek od uživatele a zobrazuje výstupy na displeji a pomocí LED. Pomocí GUI může uživatel nakonfigurovat jednotlivá rozhraní a přes tyto rozhraní odesílat či přijímat data. Firmware byl naprogramován v prostředí Atollic TrueSTUDIO for STM32, ve verzi 9.3.0. Pro vygenerování části, obsluhující rozhraní USB, byl použit program MX Cube, dodávaný výrobcem ST Microelectronic.

### 4.1 Funkce Firmware

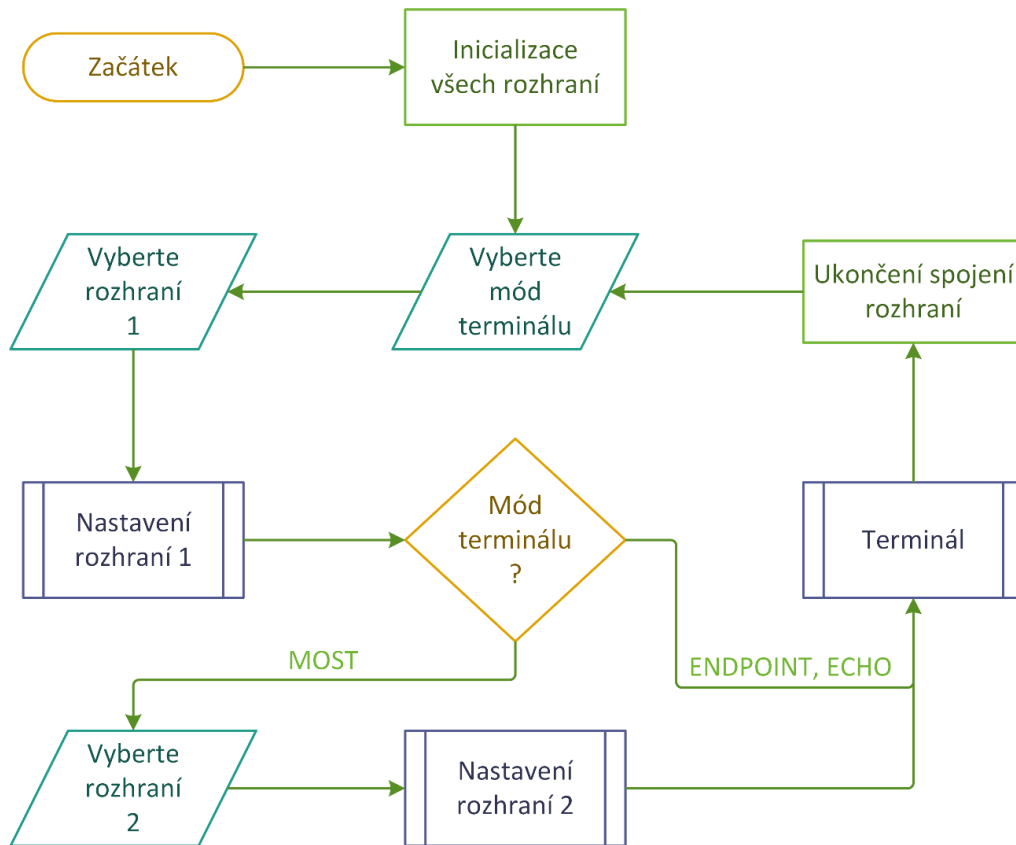
Hlavní funkce FW jsou znázorněny pomocí vývojových diagramů a doplněny popisem. Popsána je zde hlavní smyčka programu, nastavení jednotlivých rozhraní a terminálu.

#### 4.1.1 Hlavní smyčka

Obrázek 4-1 zobrazuje vývojový diagram hlavní smyčky FW, kterým je naprogramován vývojový kit. Po zapnutí vývojového kitu se nejprve provede inicializace všech rozhraní. Pokud inicializace skončí úspěšně, zobrazí se na displeji seznam módů, do kterých lze nastavit terminál vývojového kitu. Po zvolení módu se zobrazí seznam rozhraní, přes která je možné komunikovat.

Mezi tyto rozhraní patří UART, USB, WiFi, Ethernet a Bluetooth. Výběrem ze seznamu se zvolí rozhraní č. 1 terminálu. Po volbě rozhraní se spustí jeho nastavení. Toto nastavení je pro každé rozhraní jiné a vyžaduje od uživatele zadání různých vstupních hodnot. Inicializace jednotlivých rozhraní jsou popsány v dalších podkapitolách. Jakmile se úspěšně dokončí nastavení rozhraní č. 1, další postup závisí na zvoleném módu terminálu. Pokud byl zvolen mód ENDPOINT nebo ECHO, spustí

se terminál, protože v tomto módu je potřeba nastavit pouze jedno rozhraní. V případě, že byl zvolen mód MOST, spustí se volba rozhraní č. 2, kde lze zvolit druhé rozhraní pro terminál. Seznam se skládá ze stejných rozhraní, jako při volbě rozhraní č. 1, pouze neobsahuje rozhraní, které bylo zvoleno jako rozhraní č. 1.

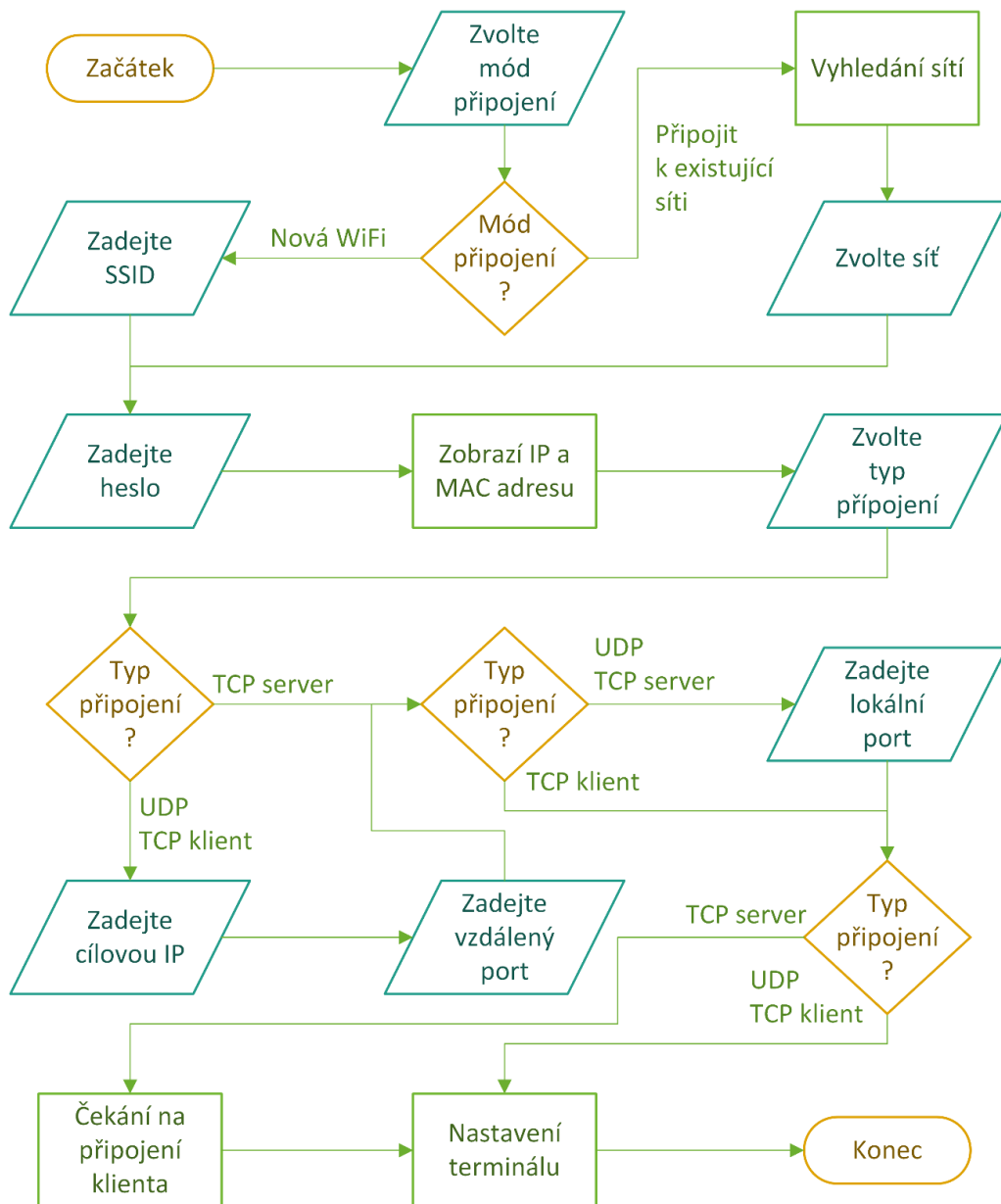


Obrázek 4-1: Vývojový diagram, hlavní smyčka FW

Po úspěšném nastavení rozhraní č. 2 se spustí terminál. Vzhled i funkce terminálu se liší podle zvoleného módu. Po ukončení terminálu se ukončí spojení všech otevřených rozhraní a vývojový kit se opět dostane na začátek smyčky k volbě módu terminálu.

#### 4.1.2 Nastavení rozhraní WiFi a Ethernet

Nastavení rozhraní WiFi se zavolá tehdy, pokud si jej uživatel zvolí pro komunikaci v terminálu. Obrázek 4-2 zobrazuje, jakým způsobem nastavení WiFi probíhá a jaké vstupní hodnoty jsou potřeba zadat od uživatele.

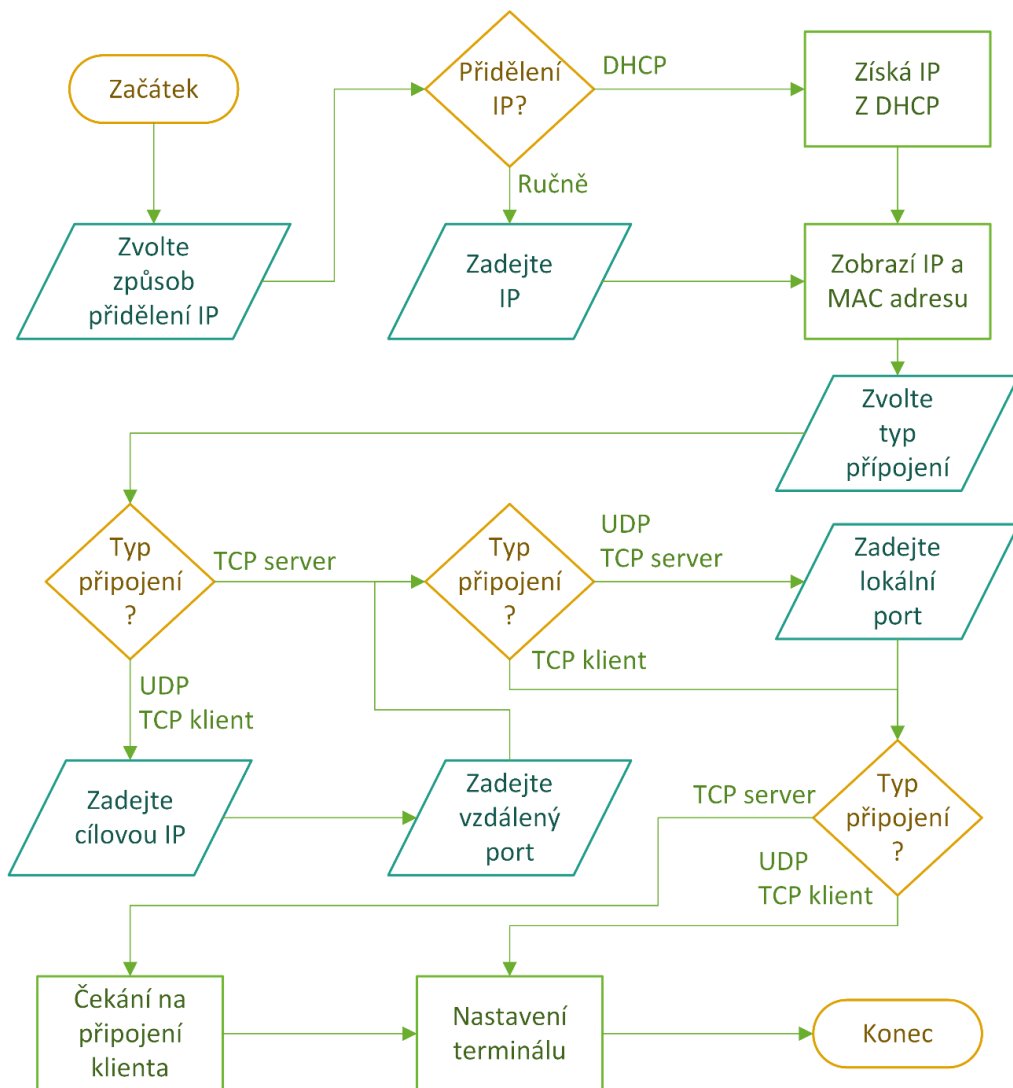


Obrázek 4-2: Vývojový diagram nastavení rozhraní Wifi

Nejprve musí být uživatelem zadáno, zda se má kit připojit k existující WiFi, či zda má být založena nová WiFi. Tím se zároveň určí, jestli se má WiFi modul chovat jako stanice nebo přístupový bod. Při volbě existující sítě vývojový kit vyhledá dostupné sítě, na které je schopen se připojit. Uživateli se zobrazí jejich seznam. Volbou některé ze sítí se pokračuje v nastavení k zadání hesla sítě. Pokud bylo zvoleno založení nové sítě, je uživatel vyzván k zadání jejího názvu. Po potvrzení se zobrazí výzva k zadání hesla sítě.

Po úspěšném zadání hesla se na displeji zobrazí IP adresa, která byla vývojovému kitu přidělena, a také MAC adresa WiFi modulu. Dále je potřeba zadat typ připojení. Je možné vybrat mezi UDP, TCP klientem a TCP serverem.

Pokud bylo zvoleno připojení UDP nebo TCP klient, je uživatel vyzván k zadání cílové IP adresy a vzdáleného portu. Tyto údaje potřebuje pro navázání spojení s jiným zařízením, kterému patří daná IP adresa a které naslouchá na daném portu. V případě připojení UDP a TCP serveru je potřeba ještě zadat číslo lokálního portu, na kterém bude vývojový kit naslouchat příchozím datům. Pokud bylo zvoleno připojení TCP server, kit tento server na zadaném portu vytvoří a čeká na připojení klienta. Po úspěšném nastavení rozhraní a vytvoření připojení pokračuje nastavení terminálu.



Obrázek 4-3: Vývojový diagram nastavení rozhraní Ethernet

Vývojový diagram nastavení rozhraní Ethernet zobrazuje Obrázek 4-3. Při nastavení Ethernetu je potřeba nejprve zvolit, zda se má IP adresa získat ze serveru DHCP, nebo se zadá ručně pomocí formuláře pro zadávání IP adresy. Dále nastavení pokračuje podobně jako u nastavení WiFi rozhraní. Vybere se typ připojení, případně se zadá cílová adresa, cílový port, nebo lokální port.

### **4.1.3 Nastavení rozhraní Bluetooth**

Rozhraní Bluetooth v tomto případě pracuje vždy v režimu slave. K úspěšnému nastavení je potřeba zadat pouze název zařízení. Bluetooth rozhraní se zapne a je dostupné pod zadaným názvem. Po připojení master zařízení je nastavení ukončeno a pokračuje se buď ke spuštění terminálu nebo k výběru dalšího rozhraní.

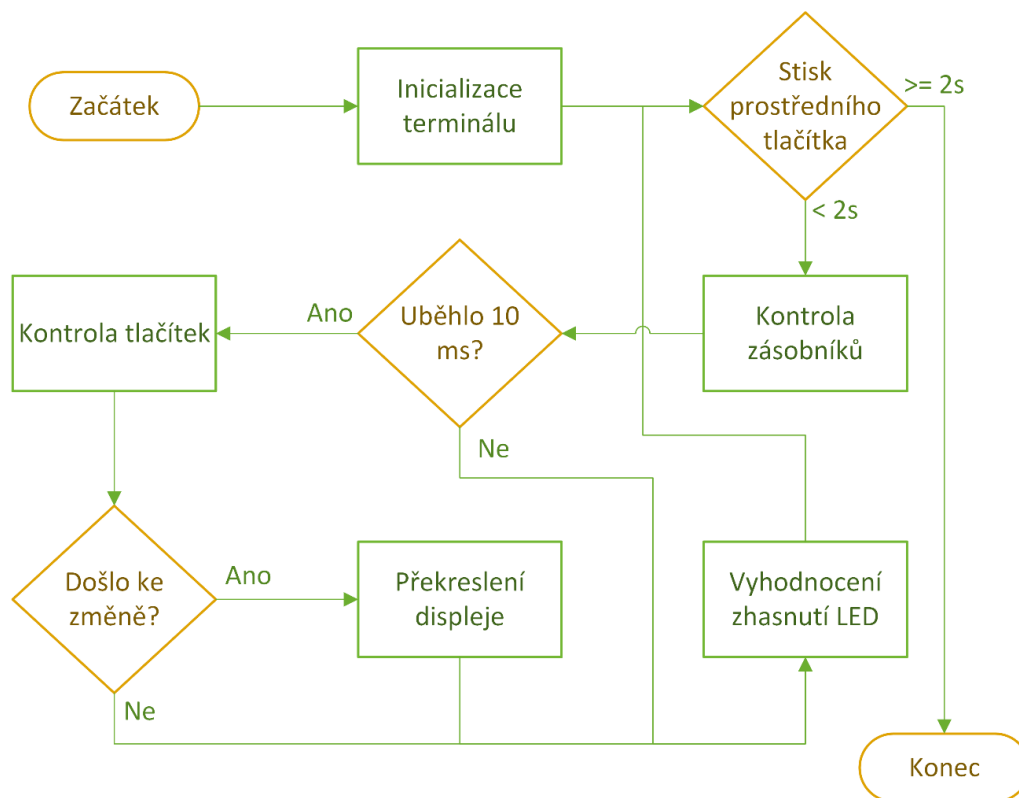
### **4.1.4 Nastavení rozhraní UART a USB**

U těchto dvou rozhraní není potřeba zadávat žádné doplňující údaje pro nastavení. Rozhraní UART je nastaveno na 115200 baud, 8 datových bitů, bez parity. Rozhraní USB (resp. jeho virtuální port) se nastavuje až v aplikaci počítače.

### **4.1.5 Popis funkce terminálu**

Při vstupu do funkce terminálu jsou již všechna potřebná rozhraní nastavena a jsou připravena ke komunikaci. Obrázek 4-4 zobrazuje jeho vývojový diagram. Nejprve proběhne inicializace. Zde se nastaví všechny proměnné do výchozího stavu a smažou se dosavadní příchozí zprávy (například pokud na rozhraní č. 1 dorazila zpráva během nastavení rozhraní č. 2).

Následuje hlavní smyčka terminálu, kterou je možné opustit stiskem prostředního tlačítka po dobu 2 sekund. V hlavní smyčce se nejprve kontroluje, zda nedorazila zpráva na některé z nastavených rozhraní. Pokud ano, zpráva se zpracuje a rozsvítí se LED, indikující příslušné rozhraní. Zároveň se nastaví příznak pro překreslení obrazovky.



Obrázek 4-4: Vývojový diagram terminálu

Dále se rozhoduje o tom, zda od poslední kontroly tlačítek uběhlo aspoň 10 milisekund. Pokud ano, zkontrolují se tlačítka. Zde se také kontroluje, jak dlouho je stisknuto prostřední tlačítko. Pokud je prostřední tlačítko stisknuto po dobu 2 vteřin, je terminál ukončen a program se vrací opět na úvodní obrazovku. Podrobněji je funkce terminálu popsána v části Grafické prostředí.

## 4.2 Grafické prostředí

K interakci s uživatelem je využito grafického displeje, LED diod a tlačítek. Po úspěšné inicializaci všech rozhraní se na displeji zobrazí menu. Pomocí něj může uživatel zvolit mód terminálu a nastavit rozhraní, přes která se bude v terminálu komunikovat. Menu se skládá z různých typů obrazovek. Jednotlivé obrazovky jsou popsány v následujících podkapitolách.

## 4.2.1 Seznam

Tato obrazovka se skládá ze dvou částí, nadpisu a položek seznamu. Nadpis je od seznamu oddělen horizontální čarou. Vybraná položka seznamu je zvýrazněna tak, že má invertovanou barvu textu a pozadí.

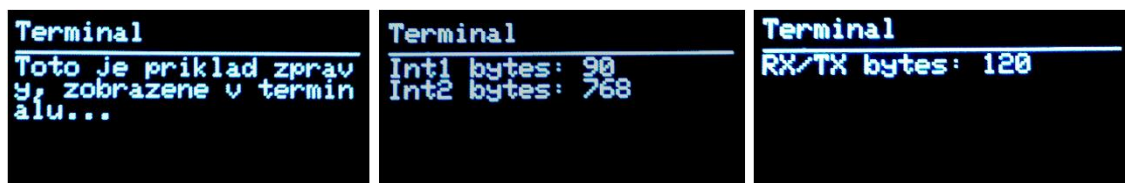


Obrázek 4-5: Příklad obrazovky se seznamem

Pomocí tlačítek nahoru a dolů je možné měnit vybranou položku. Volba se potvrzuje stiskem prostředního tlačítka. Stiskem tohoto tlačítka dochází k ukončení této obrazovky. Následuje zpracování zadaných dat a zobrazení jiné obrazovky. Stiskem levého tlačítka se obrazovka ukončí s tím, že nebyla zvolena žádná z položek seznamu. Obvykle následuje návrat na předchozí obrazovku. Obrázek 4-5 je příklad úvodní obrazovky, kde se nachází seznam módů terminálu. V grafickém rozhraní vývojového kitu se nachází hned několik seznamů.

## 4.2.2 Terminál

Terminál slouží pro demonstraci komunikace vývojového kitu přes různá rozhraní. Obrazovka tohoto typu je v menu pouze jedna. Terminál se zobrazí po úspěšném nastavení rozhraní a jeho podoba se liší podle módu, který si uživatel zvolí na úvodní obrazovce (viz. Obrázek 4-5).



Obrázek 4-6: Různé módy terminálu

Pokud je zobrazena obrazovka Terminál, LED slouží k signalizaci příchozích dat přes jednotlivá rozhraní. LED1 (vlevo) signalizuje příchozí data přes rozhraní č.1, LED2 rozhraní č.2. Terminál je možné ukončit stiskem prostředního tlačítka po dobu 2 vteřin.

Před opuštěním terminálu se na nakonfigurovaná rozhraní odešle zpráva: „[BYE BYE] \n“ a menu přejde na úvodní obrazovku. Obrázek 4-6 zobrazuje zleva terminál v módu ENDPOINT, MOST a ECHO.

## **ENDPOINT**

V tomto módu komunikuje vývojový kit pouze přes jedno rozhraní. Přijatá data jsou zobrazena jako text v terminálu a řadí se za sebe. Pokud se text již na displej nevejde, smaže se první řádek a ostatní řádky se o jeden řádek posunou nahoru. Při stisku některého z tlačítek je přes nastavené rozhraní odeslána zpráva o stisku daného tlačítka. Například při stisku prostředního tlačítka se odešle následující zpráva: „[MIDDLE] \n“.

## **MOST**

Při této konfiguraci funguje kit jako most mezi dvěma rozhraními. Přijatá data z rozhraní č. 1 přeposílá na rozhraní č. 2 a naopak. Terminál počítá, kolik bajtů přeposlal kterým směrem a tyto hodnoty zobrazuje na displeji. „Int1 bytes: 90“ znamená, že na rozhraní č. 1 bylo přijato 90 bajtů dat, a ta byla přeposlána na rozhraní č. 2.

## **ECHO**

Mód ECHO je něco mezi módy ENDPOINT a MOST. Stejně jako u módu ENDPOINT je využito pouze jedno rozhraní. Přijatá data se však nezobrazí na displeji, pouze jsou po vzoru módu MOST přeposlána po stejném rozhraní zpět. Stejně jako u módu MOST vývojový kit počítá přeposlané bajty a hodnotu zobrazuje na displeji.

### **4.2.3 Informační okno**

K zobrazení informace pro uživatele slouží informační okno. Obsahuje nadpis a zprávu. V pravém dolním rohu je tlačítko OK. Po stisku prostředního tlačítka se okno ukončí a zobrazí se obrazovka, která má následovat. V tomto firmwaru slouží informační okno například k zobrazení IP adresy a MAC adresy po připojení k wifi, aby bylo možné se ke kitu snadno připojit pomocí UDP nebo TCP klienta a nemusela se složitě zjišťovat jeho IP adresa. Obrázek 4-7 zobrazuje, jak informační okno vypadá.

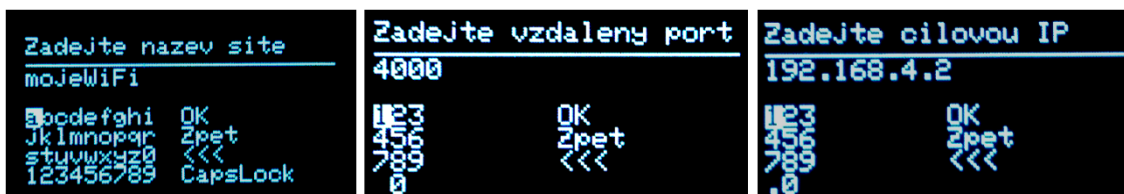


Obrázek 4-7: Informační okno

Grafické rozhraní obsahuje také informační okna, která nemají tlačítko OK. Tyto okna slouží k informování uživatele o stavu, např. „Vyhledavam site WiFi...“, aby si uživatel nemyslel, že se program zacyklil. Tato okna nevyžadují potvrzení od uživatele a zmizí, jakmile program dokončí časově náročnou činnost a může zobrazit jinou obrazovku.

#### 4.2.4 Okna pro zadávání hodnot

Uživatelské rozhraní se skládá ze tří druhů oken pro zadávání hodnot. Pomocí nich je možné zadat text, číslo, nebo IP adresu. Obrázek 4-8 zobrazuje zleva: okno pro zadání textu, čísla a IP adresy.



Obrázek 4-8: Různé obrazovky pro zadávání hodnot

Okna pro zadávání hodnot se skládají z nadpisu, který je, stejně jako je tomu u ostatních oken, oddělen od zbytku obrazovky pomocí horizontální čáry. Na řádku pod nadpisem se nachází aktuálně zadaná hodnota. V levém dolním rohu je umístěna klávesnice se znaky, ze kterých se zadaná hodnota může skládat. V případě zadávání textu jsou zde písmena a čísla. Při zadávání čísla je paleta omezena pouze na ně a pro zadávání IP adresy je k číslům přidána ještě tečka. Vpravo od klávesnice se nachází volby, které spouští různé akce (viz. Tabulka 4-1).

Tabulka 4-1: Volby a akce okna pro zadání hodnot

Volba	Akce
OK	Potvrdí zadání hodnoty, opouští okno
Zpět	Nastaví hodnotu na prázdný řetězec nebo 0, opouští okno
<<<	Smaže poslední zadaný znak
CapsLock	Změní malá písmena za velká a naopak

Všechna okna pro zadávání hodnot mají stejné ovládání. Kurzor ukazuje pomocí invertování barev textu a pozadí na zvolený znak nebo volbu. Pohybuje se s ním tlačítka nahoru, dolů, doleva a doprava. Prostřední tlačítka slouží pro potvrzení volby.

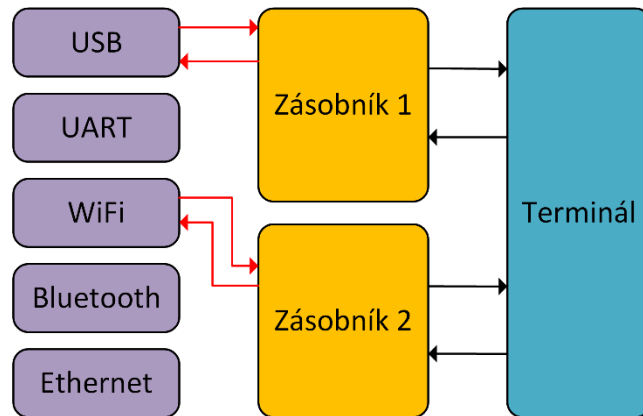
### 4.3 Komunikace s rozhraními

První podkapitola popisuje, jakým způsobem předávají knihovny jednotlivých rozhraní přijatá data do terminálu, a jak terminál odesílá data na jednotlivá rozhraní. Ostatní podkapitoly se zabývají komunikací s moduly, které zajišťují spojení přes rozhraní s počítačem.

#### 4.3.1 Výměna dat mezi rozhraním a terminálem

Terminál je naprogramován tak, že mu nezáleží na tom, přes jaké rozhraní se zrovna komunikuje. Podle módu, ve kterém terminál pracuje, se ke komunikaci využívá jedno nebo dvě rozhraní. Rozhraní předávají data do terminálu pomocí dvou zásobníků na příchozí data.

Obrázek 4-9 zobrazuje případ, kdy je zvoleno jako rozhraní č. 1 rozhraní USB a jako rozhraní č. 2 je připojeno WiFi. Terminál je v módu MOST (pouze tento mód využívá dvě rozhraní). Spojení mezi zásobníky na příchozí data a terminálem (znázorněno černými šipkami) je pevně dané. Spojení mezi rozhraními (znázorněno červenými šipkami) je konfigurovatelné. Jedno rozhraní není možné připojit na oba zásobníky najednou.



Obrázek 4-9: Blokové schéma komunikace mezi rozhraními a terminálem

Každý zásobník se skládá z bufferu pro příchozí zprávy a z proměnné, která uchovává údaj o počtu přijatých bajtů. Příjem dat všech rozhraní se provádí v přerušení, z toho důvodu je potřeba zajistit, aby k bufferu s přijatými daty nemohlo přistupovat zároveň rozhraní i terminál. O to se stará proměnná s počtem přijatých bajtů, která tak slouží zároveň jako zámek výše zmíněného bufferu. Pokud je počet přijatých bajtů roven 0, může s bufferem nakládat rozhraní. To do bufferu zapisuje přijatá data a nastavuje počet přijatých bajtů. Pokud je počet přijatých bajtů větší než 0, může s bufferem nakládat terminál. Ten z bufferu přečte přijatá data a po jejich zpracování nastaví počet přijatých bajtů na 0, aby rozhraní mohlo do bufferu opět zapsat další přijatá data.

Odesílání dat funguje tak, že terminál zavolá funkci zásobníku. Ten pak podle toho, které rozhraní je na něj připojeno, zavolá funkci konkrétního rozhraní. Každé rozhraní pak řeší odesílání dat jinak. UART například odesílaná data přemístí do vlastního bufferu a dále je odesílá v přerušení. Ethernet posílá data v rámci hlavní smyčky programu. Rozdíl je potom v tom, jak moc zdržuje odesílání dat hlavní smyčku programu. Odesílání přes UART hlavní smyčku programu téměř nezdrží, zatímco odesílání přes Ethernet si vyžádá pozdržení vykonávání hlavní smyčky, dokud nejsou všechna data odeslána přes rozhraní SPI do bufferu modulu W5500.

### 4.3.2 Komunikace mikrokontroléru s modulem ESP32

Mikrokontrolér komunikuje s modulem ESP32 pomocí rozhraní UART. Rychlost rozhraní je 115200 baud, posílá se 8 datových bitů, bez parity. Modul přes toto rozhraní přijímá AT příkazy. Bylo potřeba vytvořit knihovnu, která dokáže modul pomocí těchto

příkazů ovládat. Tabulka 4-2 obsahuje všechny AT příkazy, které vývojový kit používá ke komunikaci s modulem ESP32.

Tabulka 4-2: AT příkazy, použité pro komunikaci s modulem ESP32

Příkaz	Popis
AT+CWMODE	Nastaví WiFi mód. Modul může být nastaven buď jako stanice, to znamená, že se může připojit k existujícím sítím. Další možností je mód Soft AP, kdy může modul založit vlastní WiFi. Modul také může fungovat zároveň jako stanice a zároveň jako Soft AP.
AT+CWLAP	Dotáže se na seznam dostupných WiFi. Údaje o WiFi se skládají z metody šifrování, názvu sítě (SSID), úrovně signálu, fyzické adresy routeru a čísla kanálu.
AT+CWJAP	Připojí se k WiFi síti. Je potřeba zadat název sítě a heslo.
AT+CWQAP	Odpojí se od WiFi sítě.
AT+CWSAP	Nakonfiguruje WiFi síť pro Soft AP. Konfigurace se skládá z názvu sítě, hesla k připojení, vysílacího kanálu, metody šifrování, maximálního počtu připojení a viditelnosti sítě.
AT+CIFSR	Dotáže se na IP a MAC adresu stanice a Soft AP. Příkaz vrátí adresy pouze v případě, že jsou dané součásti nastavené.
AT+CIPMUX	Povolí nebo zakáže vícenásobné připojení. Vícenásobné připojení musí být povoleno v případě, že je modul potřeba nastavit jako TCP server. V ostatních případech musí být vícenásobné připojení zakázáno.
AT+CIPSERVER	Založí nebo odstraní TCP server. Při zakládání je potřeba zadat port, na kterém má server naslouchat.
AT+CIPSTART	Založí UDP připojení nebo TCP klienta. Je potřeba zadat protokol připojení (UDP nebo TCP), vzdálený port a v případě UDP také lokální port.
AT+CIPSEND	Odešle data přes nastavené připojení. Při volání příkazu je potřeba specifikovat, kolik bajtů se má odeslat.
AT+CIPCLOSE	Ukončí UDP připojení nebo TCP klienta.
AT+BTINIT	Zapne nebo vypne Bluetooth zařízení.
AT+BTNAME	Nastaví jméno Bluetooth zařízení.
AT+BTSCANMODE	Určí, zda má být zařízení vyhledatelné a připojitelné.
AT+BTSPINIT	Nastaví Bluetooth SPP (Seriál Port Profile). Může být master nebo slave. Pro připojení k počítači se používá pouze slave.
AT+BTSPDISCONN	Ukončí Bluetooth SPP.

AT+BTSPSEND	Odešle data přes Bluetooth SPP.
AT+BTSPSTART	Zahájí BT SPP.

První část příkazů se týká rozhraní WiFi, druhá část, počínaje příkazem AT+BTINIT, se týká Bluetooth. Příchozí data z modulu se vyhodnocují v přerušení. Data se řadí do bufferu, odkud jsou vyčítána pomocí funkcí pro čtení bufferu, nebo pokud se jedná o příchozí zprávu, jsou data zkopírována do bufferu nakonfigurovaného zásobníku.

### 4.3.3 Komunikace mikrokontroléru s USB převodníkem

Mikrokontrolér komunikuje s převodníkem FT230 pomocí rozhraní UART. Toto rozhraní je nastaveno na rychlost 115200 baud, odesílá se vždy 8 bitů, bez parity. Odesílání i příjem dat jsou řešeny v přerušení. Pro odesílání a příjem dat má mikrokontrolér vytvořeny dva buffery, každý o velikosti 64 bajtů. Najednou je tedy možné odeslat nebo přijmout maximálně 64 bajtů.

Při požadavku na odeslání zprávy se data zkopírují do TX bufferu. Následně se odešle první bajt zprávy a povolí se přerušení TXE (*Transmit data register empty*). Přerušení se zavolá, pokud je registr pro odeslání dat prázdný. V tomto přerušení se postupně odešlou všechny bajty zprávy. Při odesílání poslední zprávy je přerušení TXE vypnuto.

Přerušení RXNE (*Receive data not empty*) a RTO (*Receiver timeout*) slouží pro příjem dat. První přerušení se zavolá v případě, že byl přijat nový bajt. V přerušení je tento bajt přečten a uložen do RX bufferu. Pokud je RX buffer plný (obsazeno všech 64 bajtů), překopíruje se do RX bufferu univerzálního rozhraní, na které je UART nastaven a počet přijatých bajtů univerzálního rozhraní se nastaví na 64. V případě, že UART nemá nastaveno žádné univerzální rozhraní nebo buffer univerzálního rozhraní obsahuje nezpracovaná data, je RX buffer rozhraní UART smazán a připraven přijímat další data.

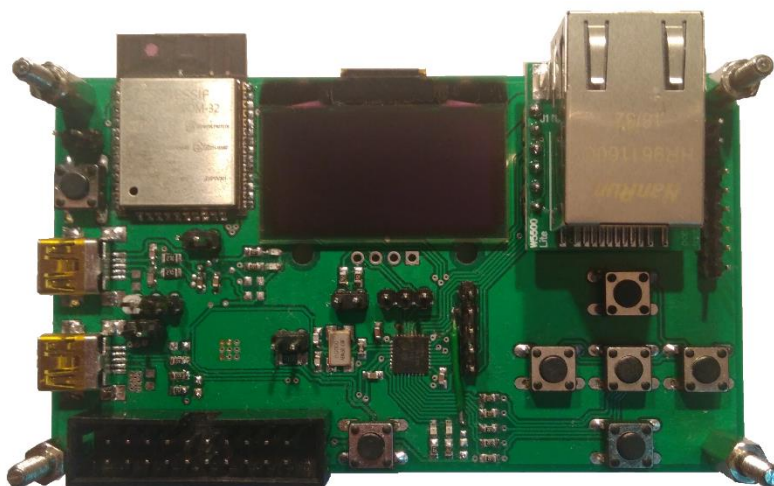
Přerušení RTO se zavolá, pokud byl ukončen příjem. Příjem je považován za ukončený, pokud není po určitou dobu přijat žádný nový bajt. V tom případě je RX buffer zkopírován do bufferu univerzálního rozhraní a počet přijatých bajtů je nastaven podle toho, kolik bajtů skutečně dorazilo.

#### **4.3.4 Komunikace mikrokontroléru s modulem W5500**

S tímto modulem se komunikuje prostřednictvím rozhraní SPI. Knihovna, která komunikaci zajišťuje, je k dispozici přímo od výrobce. Ve zdrojovém kódu knihovny bylo pouze potřeba vybrat správný typ modulu, protože knihovna jich podporuje několik. Dále bylo potřeba nadefinovat funkce pro odeslání a příjem dat přes rozhraní SPI.

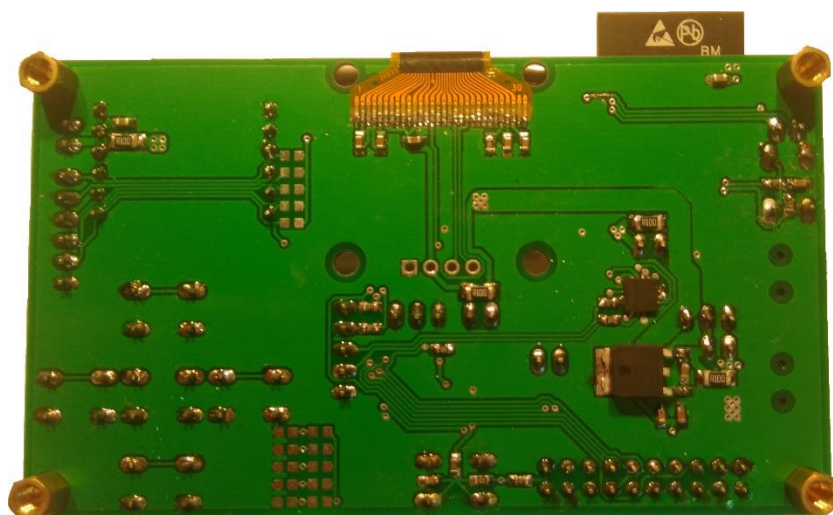
## 5 REALIZACE A TESTOVÁNÍ

Dříve než bylo možné naprogramovat FW, bylo potřeba osadit DPS součástkami a zapájet. Následně probíhalo programování a ladění FW. Nakonec bylo provedeno měření spotřeby, měření osciloskopem a měření datové propustnosti.



Obrázek 5-1: Pohled na osazenou DPS zeshora

Obrázky zobrazují DPS vývojového kitu, osazenou součástkami. Obrázek 5-1 je pohled shora, Obrázek 5-2 zase zespod.



Obrázek 5-2: Pohled na osazenou DPS zespod

Téměř všechny součástky, použité pro realizaci vývojového kitu, bylo možné připájet k DPS pomocí hrotové mikropáječky. Pouze mikrokontrolér STM32 a převodník z rozhraní USB na UART FT230 bylo potřeba připájet pomocí horkovzdušné páječky. Tyto součástky mají pouzdra QFN, která nejsou vhodná pro pájení hrotovou páječkou.

## 5.1 Měření osciloskopem

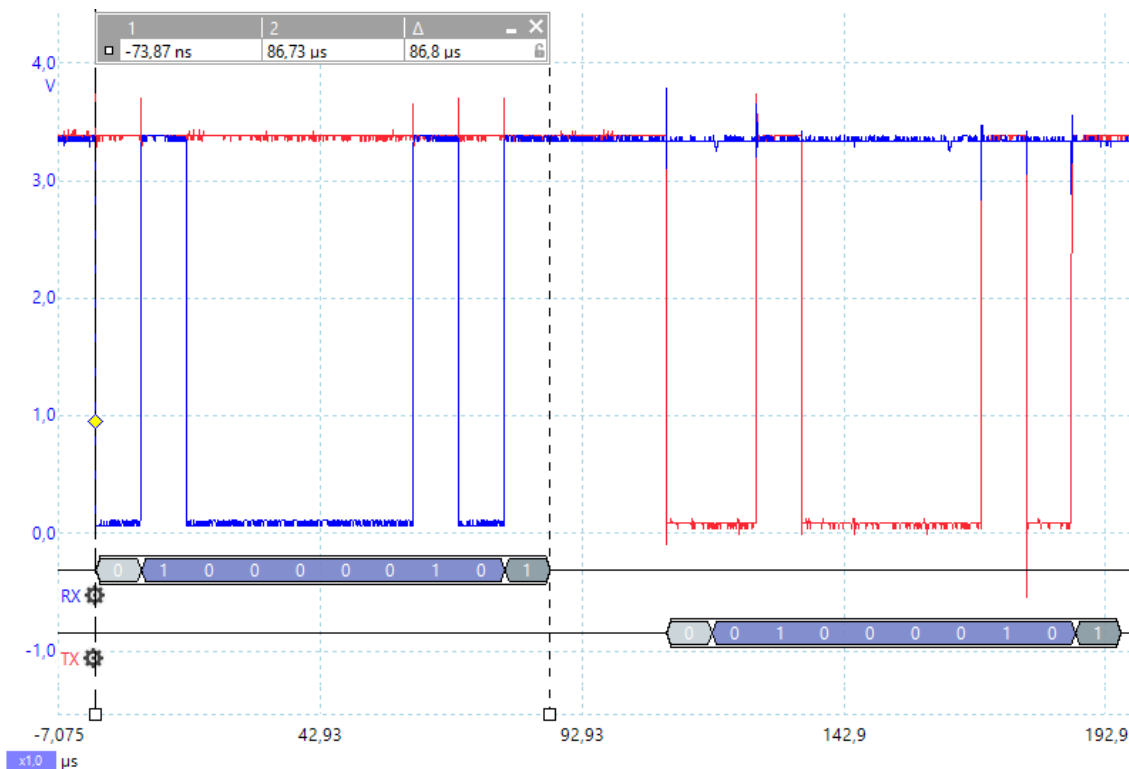
Na testovací piny rozhraní UART, I2C a SPI byl připojen USB osciloskop Picoscope S2205A. Pomocí něj byly změřeny následující průběhy. Cílem bylo změřit, zda rychlost přenosu odpovídá předpokládané rychlosti podle nastavení rozhraní. Dále bylo potřeba zjistit, zda je přenos dat při tomto nastavení spolehlivý. Aby tomu tak bylo, musí být náběžné a sestupné hrany všech signálů ostré a průběhy musí odpovídat předpokladům.

### 5.1.1 Komunikace UART

Pro otestování komunikace UART byl napsán firmware, který čeká na příchozí bajt po RX lince. V přerušení inkrementuje hodnotu tohoto bajtu o 1 a následně posílá upravený bajt zpět po TX lince. Při měření byl z aplikace v počítači odeslán znak 'A', který má hexadecimální hodnotu 0x41, mikrokontrolér odpověděl odesláním znaku 'B', který má hexadecimální hodnotu 0x42.

Záznam této komunikace zobrazuje obrázek Obrázek 5-3. Modrý průběh je změřen na RX pinu mikrokontroléru a zobrazuje příchozí bajt z počítače. Červený průběh je změřen na TX pinu a zobrazuje vysílání z mikrokontroléru. Nastavení komunikace je 115200 baud, 8 datových bitů, 1 stop bit. První bit je vždy LSB.

$$t_{byte} = \frac{1}{\text{baud}} * (\text{start} + \text{data} + \text{stop}) = \frac{1}{115200} * (1 + 8 + 1) = \underline{\underline{86,8 \text{ us}}} \quad (5-1)$$



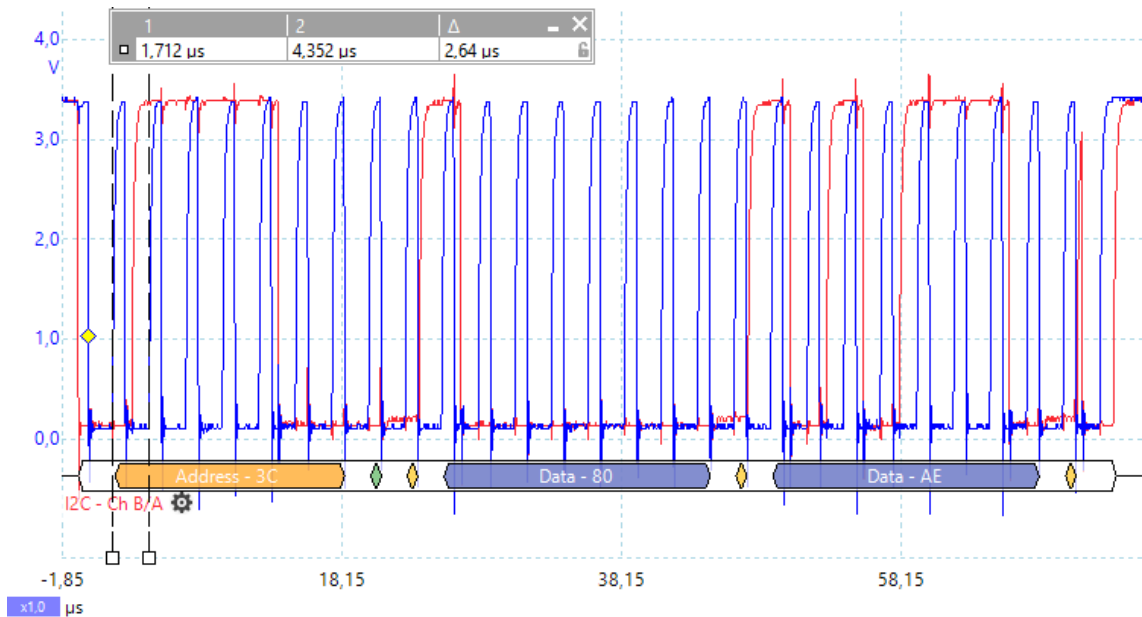
Obrázek 5-3: Průběhy komunikace UART

Podle vzorce 5-1 by doba odeslání jednoho bajtu přes takto nastavenou komunikaci UART měla být 86,8 us. Stejná doba byla změřena také pomocí osciloskopu.

## 5.1.2 Komunikace I<sup>2</sup>C

Rozhraní I<sup>2</sup>C se používá pro komunikaci s OLED displejem. Pomocí osciloskopu bylo zachyceno odesílání příkazu pro vypnutí displeje (viz. Obrázek 5-4). Modrý průběh na obrázku zobrazuje hodinový signál a červený zobrazuje datový signál. Jedná se o první příkaz, který je přes I<sup>2</sup>C odeslán po zapnutí vývojového kitu. V případě I<sup>2</sup>C komunikace se z bajtu nejprve odesílá bit MSB. Příkaz se skládá ze tří bajtů. První bajt obsahuje adresu zařízení, pro které je příkaz určen, a také je zde obsaženo, zda se bude jednat o zápis (hodnota log. 0) nebo čtení (hodnota log. 1). Adresa se nachází na 1. až 7. bitu, údaj o čtení / zápisu je na 0. bitu. V tomto případě lze z prvního bajtu vyčíst, že se jedná o zápis na adresu 0x3C. Tato adresa patří řadiči OLED displeje.

Druhý bajt je řídicí. Jeho 7. bit rozhoduje o tom, jestli bude po následujícím datovém bajtu poslán další řídicí bajt (hodnota log. 1), nebo budou všechny následující bajty pouze datové (hodnota log. 0). Šestý bit řídicího bajtu udává, zda je následující datový bajt určen jako příkaz (hodnota log. 0), nebo se jedná o data pro zápis do paměti RAM (hodnota log. 1). V tomto případě druhý bajt přenáší informaci o tom, že po příštím datovém bajtu bude následovat opět řídicí bajt, a že následující datový bajt je určen jako příkaz.



Obrázek 5-4: Průběhy komunikace I2C

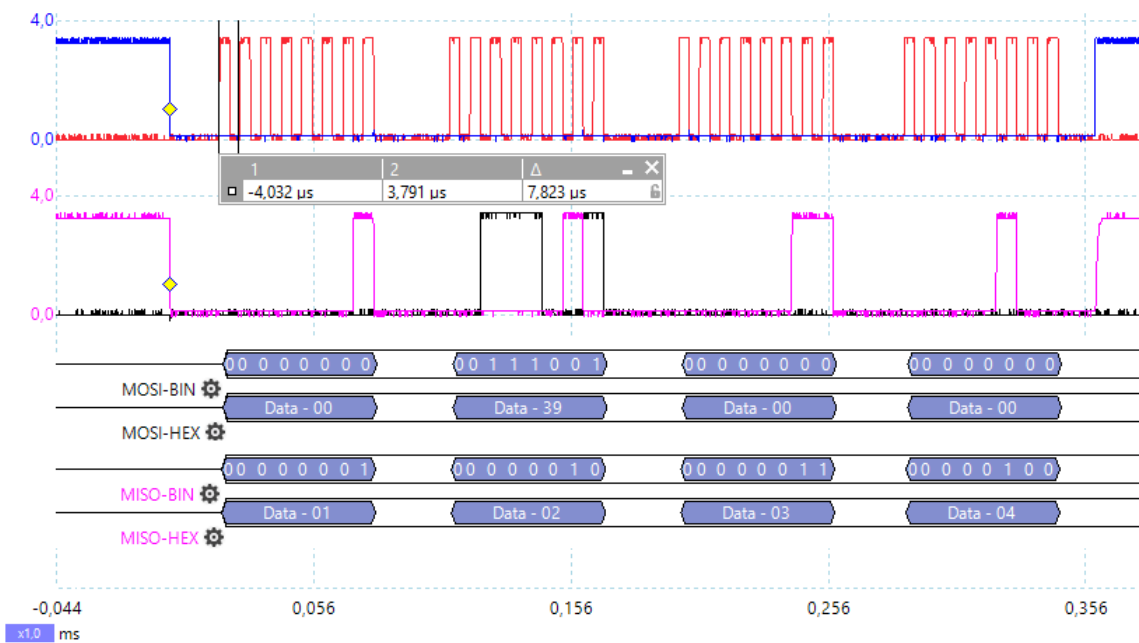
Třetí bajt v tomto případě slouží jako příkaz. Jeho hodnota 0xAE znamená příkaz „Set Display ON/OFF“, kde nulový bit rozhoduje o tom, zda se má displej zapnout, nebo vypnout. V tomto případě má nulový bit hodnotu log. 0, takže se má displej vypnout.

$$f_{I2C} = \frac{1}{T_{I2C}} = \frac{1}{2,64 * 10^{-6}} = 378,8 \text{ kHz} \quad (5-2)$$

Komunikace je nastavena na 400 kHz, což je maximální frekvence, kterou displej podporuje. Z měření na osciloskopu a vzorce 5-2 vyplývá, že ve skutečnosti je frekvence přenosu pouze 378,8 kHz. Aby byla frekvence 400 kHz, musela by perioda jednoho hodinového cyklu být 2,5 us.

### 5.1.3 Komunikace SPI

Obrázek 5-5 zobrazuje průběhy jednotlivých vodičů, které se podílejí na komunikaci SPI. Modrý průběh je vodič CS (*Chip select*), který slouží pro aktivaci slave zařízení, v tomto případě čip W5500. Když má vodič CS hodnotu log. 1, znamená to, že slave zařízení nepřijímá, ani nevysílá žádná data po rozhraní SPI. K aktivaci zařízení dochází až při sepnutí vodiče CS do log. 0. Červený průběh zobrazuje hodinový signál CLK. Ten je generován master zařízením, v tomto případě mikrokontrolérem. Z obrázku je patrné, že se data odesílají po 8 bitech a při komunikaci dochází k přenosu celkem 4 bajtů. Růžový průběh zobrazuje vodič MISO. Tento vodič slouží pro přenos dat ze slave zařízení do master zařízení, v tomto případě z čipu W5500 do mikrokontroléru. Černý průběh zobrazuje vodič MOSI, který zprostředkovává přenos dat z mikrokontroléru do čipu W5500.



Obrázek 5-5: Průběhy komunikace SPI

Obrázek 5-5 zobrazuje vyčtení registru VERSIONR na adrese 0x0039. Registr je pouze pro čtení a vždy vrací hodnotu 0x04. První dva bajty přenosu slouží pro zápis adresy, Třetí bajt udává, ve kterém bloku se požadovaný registr nachází a zda se z něj má číst, nebo do něj zapisovat. Poslední bajt slouží pro vyčtení hodnoty registru.

$$f_{SPI} = \frac{1}{T_{SPI}} = \frac{1}{7,823 * 10^{-6}} = 127,8 \text{ KHz} \quad (5-3)$$

Při měření tohoto průběhu byl pro taktování mikrokontroléru použit interní 8 MHz RC oscilátor bez násobičky. Kmitočet byl pro potřeby taktování SPI vydělen 64. Předpokládaná frekvence SPI byla 125 KHz. Z průběhu byla pomocí vzorce 5-3 vypočítána frekvence 127,8 KHz. Skutečná frekvence přenosu je tedy vyšší, než se předpokládalo.

## 5.2 Měření spotřeby

V rámci testování bylo provedeno měření spotřeby OLED displeje a modulů ESP32 a W5500. U displeje se měřila závislost spotřeby na zaplnění displeje (podíl rozsvícených pixelů k celkovému počtu pixelů). U modulů se měřila spotřeba v různých režimech. Měření bylo realizováno pomocí multimetru Extech EX530. Multimetr byl přepnut do režimu mA a byl připojen na konektory JP6, JP7 a J8. Pokud se neprovádí měření spotřeby na konektorech JP1, JP6, JP7, JP8 nebo JP9, musí být na tyto konektory připojena zkratovací propojka.

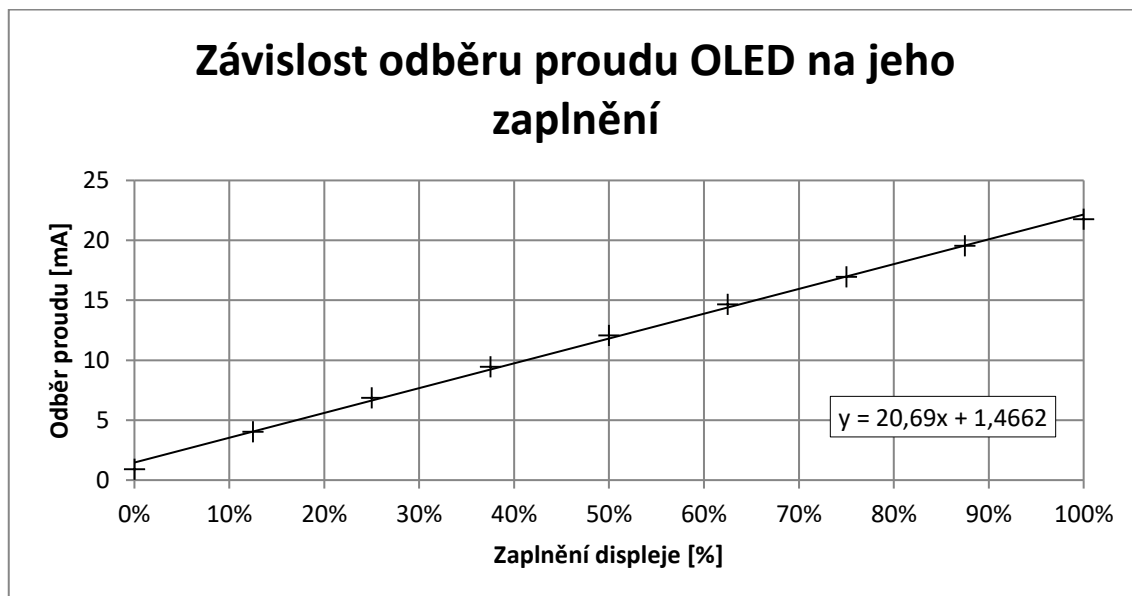
### 5.2.1 Spotřeba OLED displeje

Pro účely měření byl napsán firmware, který postupně po 8 rádcích zaplňoval celý displej. Vykonávání programu bylo vždy po překreslení pozastaveno, aby bylo možné změřit, jaký proud displej odebírá. Tabulka 5-1 obsahuje naměřené hodnoty proudu v závislosti na zaplnění displeje. Obrázek 5-6 zobrazuje graf proudové závislosti.

Tabulka 5-1: Závislost odběru proudu OLED na jeho zaplnění

Zaplnění displeje [%]	Odběr proudu I [mA]	Zaplnění displeje [%]	Odběr proudu I [mA]
0,0	0,923	62,5	14,652
12,5	4,048	75,0	16,960
25,0	6,871	87,5	19,541
37,5	9,469	100,0	21,756
50,0	12,079		

Před inicializací je proudový odběr displeje 0,8 uA. Po inicializaci stoupne jeho spotřeba až na 0,923 mA. To je způsobeno tím, že se při inicializaci zapne nábojová pumpa, která vytváří pro displej potřebné napětí 8 V.



Obrázek 5-6: Závislost odběru proudu OLED na jeho zaplnění

Měřením byl potvrzen předpoklad, že spotřeba displeje stoupá s téměř lineární závislostí na zaplnění displeje.

### 5.2.2 Spotřeba modulů ESP32 a W5500

Spotřeba v režimu inicializováno byla změřena po zapnutí vývojového kitu v úvodní obrazovce. U režimů WiFi stanice, WiFi AP a Bluetooth bylo měření provedeno po navázání spojení s počítačem. Režimy, které zahrnují komunikaci, byly nasimulovány tak, že byl na vývojovém kitu terminál nastaven do módu ECHO a na počítači byla spuštěna aplikace pro měření datové propustnosti. Tato aplikace je podrobně popsána v kapitole 5.3. Moduly odebírají krátkodobě velké proudové špičky, v tabulce Tabulka 5-2 jsou zaznamenány pouze odběry mimo proudové špičky.

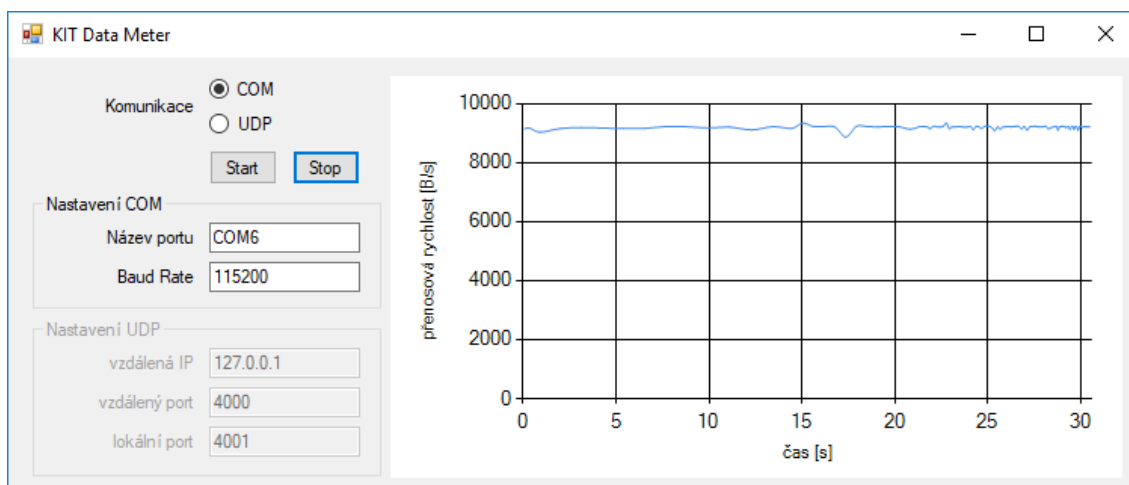
Tabulka 5-2: Odběr proudu modulů ESP32 a W5500 v různých režimech

Modul	Režim	Odběr proudu [mA]
ESP32	Inicializováno	22
	WiFi stanice	24
	WiFi stanice + komunikace	94
	WiFi AP	97
	WiFi AP + komunikace	104
	Bluetooth	23
	Bluetooth + komunikace	90
W5500	Inicializováno	62,9
	Spojení navázáno	121
	Komunikace	121,5

Z měření vyplývá, že nejmenší spotřebu má rozhraní Bluetooth. U rozhraní WiFi záleží na tom, zda je modul nakonfigurován jako stanice (připojuje se k WiFi), nebo jako AP (zakládá WiFi). Vyšší spotřeba je podle předpokladu v režimu AP, kdy modul kromě případné komunikace periodicky vysílá také zprávu s informacemi o síti. Modul W5500 má spotřebu okolo 60 mA, dokud se k němu nepřipojí ethernetový kabel. V okamžiku připojení kabelu stoupne spotřeba na 120 mA a při komunikaci už se významně nemění.

### 5.3 Měření datové propustnosti

Za účelem porovnání datové propustnosti jednotlivých rozhraní byla naprogramována aplikace pro počítač. Ta dokáže poslat na sériový port nebo přes UDP klienta zprávu o délce 64 bajtů. Následně čeká, až se mu zpráva od příjemce vrátí zpět. Na sériový port nebo přes UDP je připojen vývojový kit, na kterém je nastaven terminál v módu ECHO, takže posílá příchozí zprávy po stejném rozhraní nazpět. Z doby od okamžiku odeslání zprávy po její opětovné přijetí se pak spočítá datová propustnost rozhraní. Ihned po přijetí dat odešle aplikace dalších 64 bajtů. Tato aplikace byla využita kromě měření datové propustnosti také k měření spotřeby modulů při komunikaci.



Obrázek 5-7: Okno aplikace pro měření datové propustnosti

Závislost datové propustnosti na čase je zobrazována v grafu. Měření se spustí stiskem tlačítka Start, ukončí se tlačítkem Stop. Vzhled aplikace ukazuje Obrázek 5-7. Tabulka 5-3 obsahuje změřené datové propustnosti jednotlivých rozhraní.

Tabulka 5-3: Datové propustnosti jednotlivých rozhraní

Rozhraní	Datová propustnost [KB/s]
USB	20000
Ethernet	11000
WiFi	2750
Bluetooth	2750
UART	2500

Největší datová propustnost byla změřena u rozhraní USB. To je způsobeno tím, že je toto rozhraní jako jediné implementováno přímo v mikrokontroléru a data tak putují pouze po jednom rozhraní. Ostatní rozhraní jsou pomalejší hlavně kvůli zprostředkování přes rozhraní UART a SPI. Přes rozhraní UART jsou k mikrokontroléru připojena rozhraní WiFi a Bluetooth, proto je jejich propustnost srovnatelná. Rozhraní Ethernet je zprostředkováno přes rozhraní SPI, které bylo při měření propustnosti natakto这件áno na 9 MHz. Mikrokontrolér při měření využíval externí krystalový 8 MHz oscilátor. Jeho frekvence byla po vynásobení 72 MHz.

## 6 ZÁVĚR

Cílem této diplomové práce bylo navrhnout vývojový kit, který bude komunikovat pomocí čtyřech rozhraní s počítačem a bude sloužit pro výuku laboratorních cvičení. Dále bylo za úkol naprogramovat firmware pro mikrokontrolér a vytvořit návrh laboratorní úlohy pro seznámení studentů s vývojovým kitem.

Autor prostudoval jednotlivá rozhraní, pomocí kterých mikrokontrolér komunikuje s počítačem a s moduly, které jsou k mikrokontroléru připojeny. Dále bylo navrženo, jakým způsobem bude vývojový kit realizován. Byly vybrány a popsány součástky, použité k realizaci kitu. Následně bylo navrženo schéma zapojení a DPS. DPS byla osazena součástkami. K ověření komunikace mikrokontroléru s počítačem přes všechna vybraná rozhraní byl naprogramován FW pro mikrokontrolér. FW se stará o komunikaci s počítačem, jednotlivými moduly na DPS, a také o interakci s uživatelem. K té slouží OLED displej, tlačítka a dvě LED. Na displeji je zobrazeno grafické prostředí, které slouží k nastavení jednotlivých rozhraní a také k ověření funkčnosti komunikace. Bylo provedeno měření spotřeby modulů a displeje. Také byla změřena datová propustnost jednotlivých rozhraní v módu terminálu ECHO. Nakonec byla navržena laboratorní úloha pro seznámení studentů s kitem. V té si studenti mohou vyzkoušet spojení vývojového kitu nejen s počítačem, ale také s mobilním telefonem.

Vývojový kit je vhodný pro programátory, kteří s mikrokontroléry ARM teprve začínají. Vytvořený firmware obsahuje nastavení a použití různých rozhraní pro komunikaci mikrokontroléru s periferiemi nebo počítačem, které mohou posloužit k rychlejšímu pochopení a k následnému využití pro vlastní projekty. Díky pinovým konektorům je možné k vývojovému kitu připojit také jiné periferie, nebo sledovat průběhy komunikací na osciloskopu.

# LITERATURA

- [1] STM32F3DISCOVERY. In: *St.com* [online]. b.r. [cit. 2019-04-19]. Dostupné z: [https://www.st.com/content/st\\_com/en/products/evaluation-tools/product-evaluation-tools/mcu-mpu-eval-tools/stm32-mcu-mpu-eval-tools/stm32-discovery-kits/stm32f3discovery.html](https://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/mcu-mpu-eval-tools/stm32-mcu-mpu-eval-tools/stm32-discovery-kits/stm32f3discovery.html)
- [2] PETERKA, Jiří. Báječný svět počítačových sítí: Část XX.: Příběh Ethernetu. *Archiv článků a přednášek Jiřího Peterky* [online]. Praha: Jiří Peterka, 2006 [cit. 2018-11-16]. Dostupné z: <http://www.earchiv.cz/b06/b1200001.php3>
- [3] Ethernet connector. In: *Amazon.com* [online]. Amazon: Amazon, 2019 [cit. 2019-05-19]. Dostupné z: [https://images-na.ssl-images-amazon.com/images/I/61YHLW%2BswxL.\\_SL1500\\_.jpg](https://images-na.ssl-images-amazon.com/images/I/61YHLW%2BswxL._SL1500_.jpg)
- [4] PETERKA, Jiří. Báječný svět počítačových sítí: Část XXIV: Wi-Fi. *Archiv článků a přednášek Jiřího Peterky* [online]. Praha: Jiří Peterka, 2007 [cit. 2018-12-03]. Dostupné z: <http://www.earchiv.cz/b07/b0400001.php3>
- [5] *Microchip Developer Help: Universal Serial Bus (USB)* [online]. Chandler(Arizona): Microchip Technology, 2018 [cit. 2018-12-07]. Dostupné z: <http://microchipdeveloper.com/usb:start>
- [6] Radio Versions. *Bluetooth Technology Website* [online]. Kirkland(Washington): Bluetooth Technology, 2018 [cit. 2018-12-12]. Dostupné z: <https://www.bluetooth.com/bluetooth-technology/radio-versions>
- [7] *I2C-bus specification and user manual: UM10204*. Rev. 6. NXP Semiconductors, 2014. Dostupné také z: <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>
- [8] TEXAS INSTRUMENTS. *KeyStone Architecture Serial Peripheral Interface (SPI) User Guide*. Release A. Dallas, 2012. Dostupné také z: <http://www.ti.com/lit/ug/sprugp2a/sprugp2a.pdf>
- [9] STMICROELECTRONICS. *STM32F302x6 STM32F302x8*. Rev. 7. Amsterdam(Netherlands), 2017. Dostupné také z: <https://www.st.com/resource/en/datasheet/stm32f302k6.pdf>
- [10] *FT230X USB TO BASIC UART IC*. Version 1.4. FTDI, 2016. Dostupné také z: [https://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS\\_FT230X.pdf](https://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT230X.pdf)
- [11] Modules. In: *Espressif Systems* [online]. Pudong (Shanghai), 2018 [cit. 2018-12-13]. Dostupné z: <https://www.espressif.com/en/products/hardware/modules>
- [12] *ESP32-WROOM-32 Datasheet V2.7*. Espressif Systems, 2018. Dostupné také z: [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf)
- [13] Mini Ethernet Board W5500. In: *Makerfabs* [online]. Makerfabs, 2018 [cit. 2018-12-13].

Dostupné z: <https://www.makerfabs.com/Mini-Ethernet-Board-W5500.html>

- [14] *W5500 Datasheet*. Version 1.0.2. WIZnet, 2013. Dostupné také z: [https://cdn.sparkfun.com/datasheets/Dev/Arduino/Shields/W5500\\_datasheet\\_v1.0.2\\_1.pdf](https://cdn.sparkfun.com/datasheets/Dev/Arduino/Shields/W5500_datasheet_v1.0.2_1.pdf)
- [15] Vishay Dale O128O064LLAP3N0000. In: *DigiKey Electronics* [online]. 2018 [cit. 2018-12-13]. Dostupné z: <https://www.digikey.lv/product-detail/en/vishay-dale/O128O064LLAP3N0000/541-3506-ND/7041877>
- [16] *OLED-128O064D-BPP3N00000*. Vishay, 2016. Dostupné také z: <https://www.vishay.com/docs/37902/oled128o064dbpp3n00000.pdf>
- [17] *LD1086: 1.5 A adjustable and fixed low drop positive voltage regulator*. Rev 36. STMicroelectronics, 2017. Dostupné také z: <https://www.st.com/resource/en/datasheet/ld1086.pdf>
- [18] <https://www.tme.eu/Document/85698efaaaf9d79420d9482d2850b5ad/LL-S192UGC-1B.pdf>. Lucky Light Electronics, 2006. Dostupné také z: <https://www.tme.eu/Document/85698efaaaf9d79420d9482d2850b5ad/LL-S192UGC-1B.pdf>
- [19] USB differential pair length. In: *StackExchange* [online]. 2014 [cit. 2018-12-10]. Dostupné z: <https://electronics.stackexchange.com/questions/52851/usb-differential-pair-length>

# SEZNAM OBRÁZKŮ

Obrázek 1-1: Vývojový kit STM32F3 Discovery [1].....	13
Obrázek 1-2: Komunikace mezi počítačem, vývojovým kitem a uživatelem .....	15
Obrázek 2-1: Ethernetová přípojka RJ45 [3].....	17
Obrázek 2-2: Sdílení internetu pomocí WiFi.....	18
Obrázek 2-3: Rámec rozhraní UART .....	24
Obrázek 2-4: Rámec I <sup>2</sup> C, odeslání adresy a dat.....	25
Obrázek 2-5: Ukázka řízení CLK signálu v různých módech. [8]. .....	26
Obrázek 3-1: Blokové schéma realizace kitu .....	27
Obrázek 3-2: SOC ESP32-WROOM-32 [11].....	30
Obrázek 3-3: Modul s obvodem W5500 [13].....	32
Obrázek 3-4: OLED displej [15] .....	34
Obrázek 3-5: Zapojení tlačítka .....	36
Obrázek 3-6: Pinový konektor pro měření proudu .....	38
Obrázek 3-7: Rozložení součástek na DPS.....	40
Obrázek 4-1: Vývojový diagram, hlavní smyčka FW .....	42
Obrázek 4-2: Vývojový diagram nastavení rozhraní Wifi.....	43
Obrázek 4-3: Vývojový diagram nastavení rozhraní Ethernet .....	44
Obrázek 4-4: Vývojový diagram terminálu .....	46
Obrázek 4-5: Příklad obrazovky se seznamem.....	47
Obrázek 4-6: Různé módy terminálu.....	47
Obrázek 4-7: Informační okno.....	49
Obrázek 4-8: Různé obrazovky pro zadávání hodnot.....	49
Obrázek 4-9: Blokové schéma komunikace mezi rozhraními a terminálem .....	51

Obrázek 5-1: Pohled na osazenou DPS zeshora .....	55
Obrázek 5-2: Pohled na osazenou DPS zespod .....	55
Obrázek 5-3: Průběhy komunikace UART.....	57
Obrázek 5-4: Průběhy komunikace I2C.....	58
Obrázek 5-5: Průběhy komunikace SPI.....	59
Obrázek 5-6: Závislost odběru proudu OLED na jeho zaplnění .....	61
Obrázek 5-7: Okno aplikace pro měření datové propustnosti .....	63

# SEZNAM TABULEK

Tabulka 2-1: Přenosová frekvence a maximální rychlost jednotlivých standardů [4] ...	18
Tabulka 2-2: Vývoj specifikace USB [5] .....	20
Tabulka 2-3: Vysílací výkon podle třídy zařízení [6].....	22
Tabulka 2-4: Maximální přenosové rychlosti sběrnice [7].....	25
Tabulka 3-1: Parametry mikrokontroléru STM32F302K6U8 [9] .....	28
Tabulka 3-2: Parametry převodníku FT230XQ [10] .....	29
Tabulka 3-3: Parametry modulu ESP32 [12].....	30
Tabulka 3-4: Přehled typů AT příkazů .....	31
Tabulka 3-5: Příklady AT příkazů .....	31
Tabulka 3-6: Parametry obvodu W5500 [14] .....	33
Tabulka 3-7: Parametry OLED displeje [16].....	34
Tabulka 3-8: Nastavení rozhraní OLED displeje [16].....	34
Tabulka 4-1: Volby a akce okna pro zadání hodnot .....	50
Tabulka 4-2: AT příkazy, použité pro komunikaci s modulem ESP32 .....	52
Tabulka 5-1: Závislost odběru proudu OLED na jeho zaplnění .....	60
Tabulka 5-2: Odběr proudu modulů ESP32 a W5500 v různých režimech .....	62
Tabulka 5-3: Datové propustnosti jednotlivých rozhraní .....	63

# SEZNAM VELIČIN A ZKRATEK

$f$  [Hz] Frekvence

$I$  [A] Proud

$P$  [W] Výkon

$R$  [ $\Omega$ ] Odpor

$U$  [V] Napětí

*ADC* Analog to digital converter

*AP* Access point

*BLE* Bluetooth low energy

*BR* Basic rate

*CLK* Clock

*DAC* Digital to analog converter

*DHCP* Dynamic host configuration protocol

*DPS* Deska plošného spoje

*EDR* Enhanced Data Rate

*FPGA* Field programmable gate array

*FW* Firmware

*GSM* Groupe spécial mobile

*GUI* Graphic user interface

*IEEE* Institute of electrical and electronics engineers

*I<sup>2</sup>C* Inter-integrated circuit

*IO* Input Output

*IP* Internet protocol

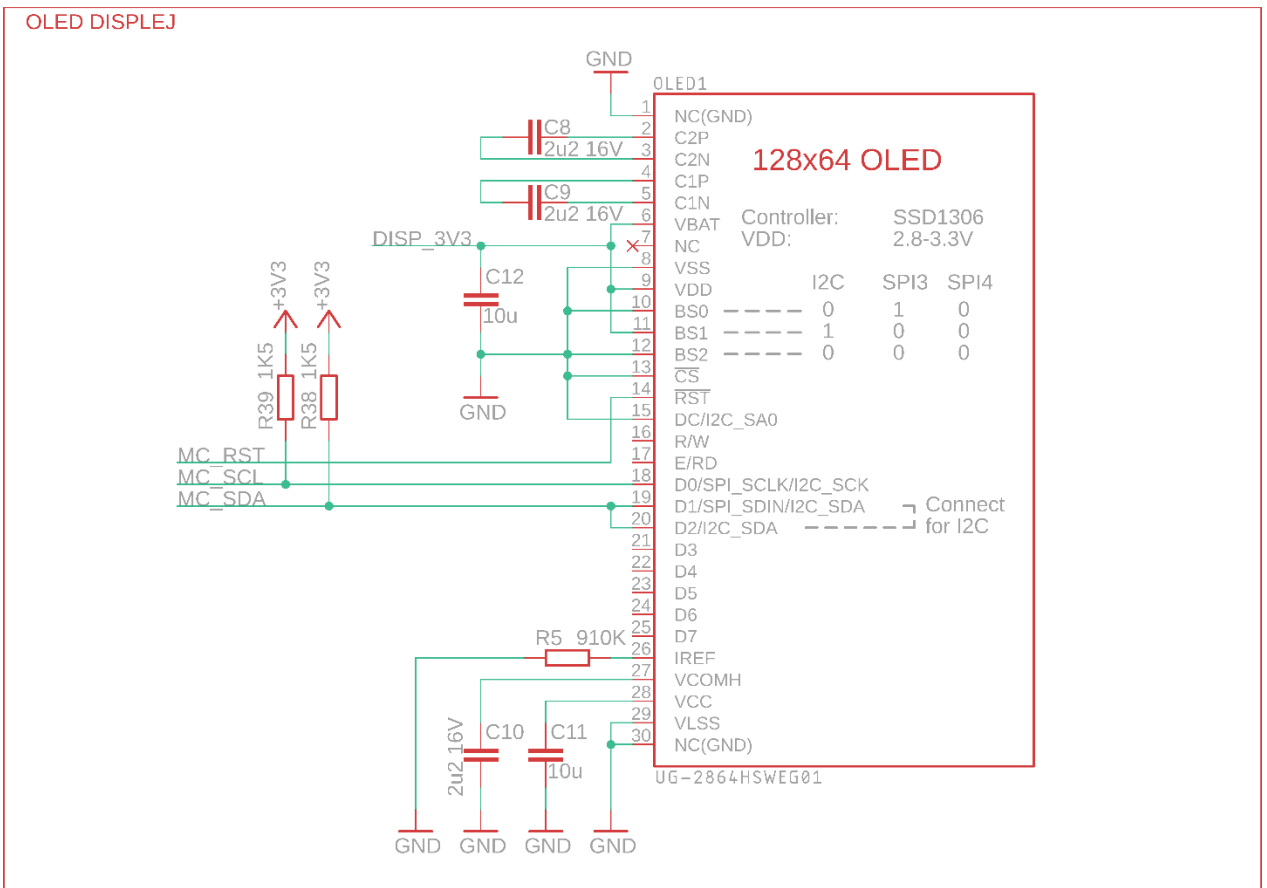
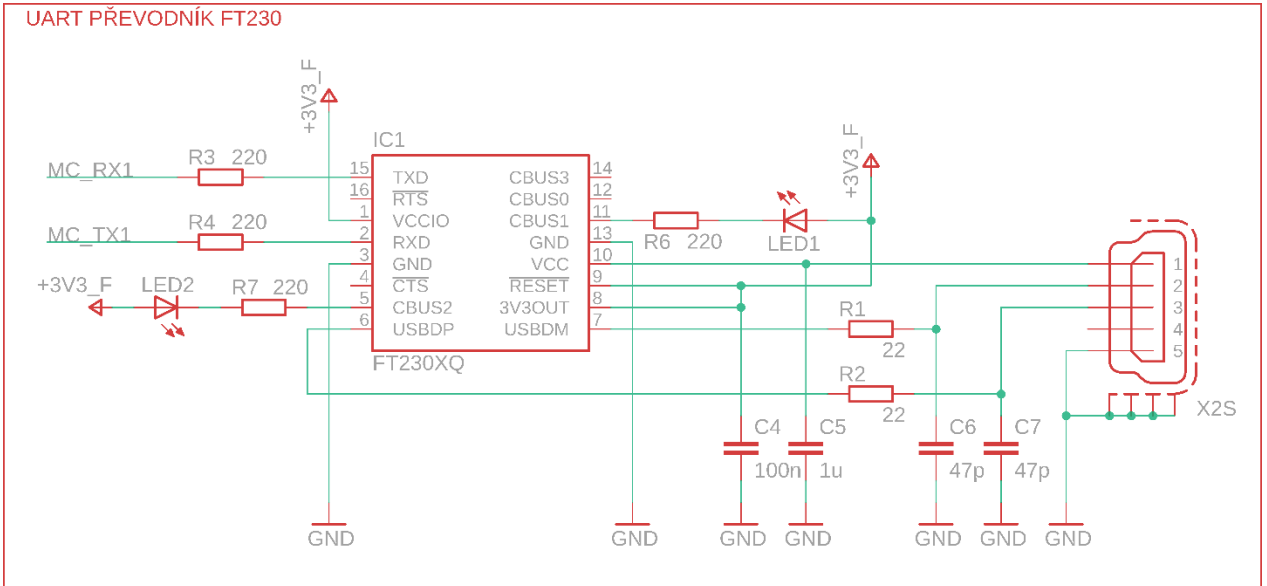
<i>JTAG</i>	Joint test action group
<i>LCD</i>	Liquid crystal display
<i>LED</i>	Light emitting diode
<i>LSB</i>	Least significant bit / byte
<i>MAC</i>	Media access control
<i>MDIX</i>	Medium dependent interface crossover
<i>MISO</i>	Master in slave out
<i>MOSI</i>	Master out slave in
<i>MSB</i>	Most significant bit / byte
<i>OC</i>	Open collector
<i>OD</i>	Open drain
<i>OLED</i>	Organic light emitting diode
<i>QFN</i>	Quad flat no-leads package
<i>RAM</i>	Random access memory
<i>RC</i>	Resistor capacitor
<i>RTC</i>	Real-time clock
<i>RTO</i>	Receiver timeout
<i>RXNE</i>	Receive data not empty
<i>SCL</i>	Synchronnous clock
<i>SDA</i>	Synchronnous data
<i>SOC</i>	System on chip
<i>SPI</i>	Serial peripheral interface
<i>SSID</i>	Service set identifier
<i>TCP</i>	Transmission control protocol
<i>TWI</i>	Two wire interface

<i>TXE</i>	Transmit data register empty
<i>UART</i>	Universal asynchronous receiver and transmitter
<i>UDP</i>	User Datagram Protocol
<i>USB</i>	Universal serial bus
<i>WECA</i>	Wireless ethernet compatibility alliance
<i>WiFi</i>	Wireless fidelity

# SEZNAM PŘÍLOH

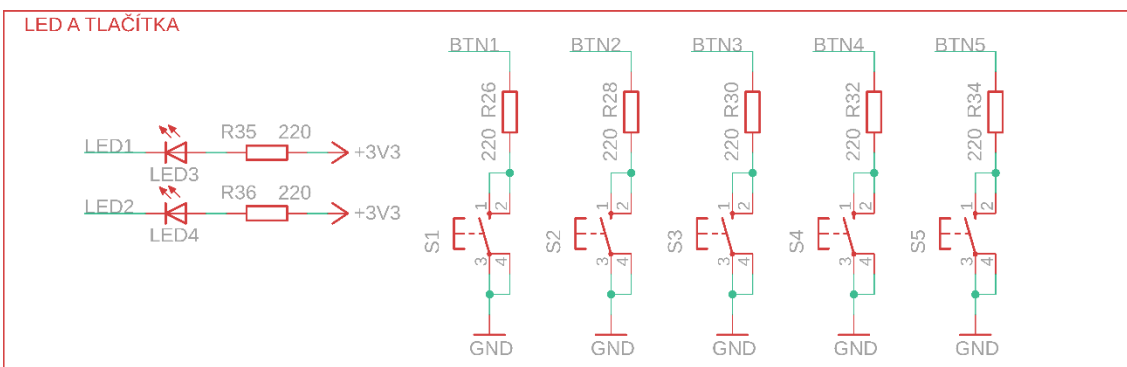
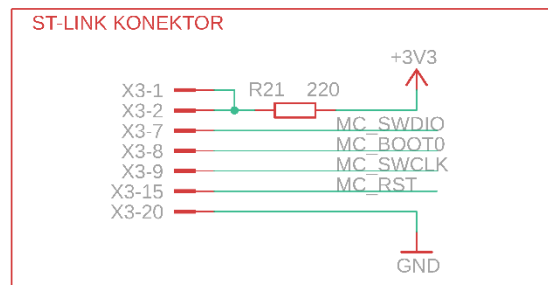
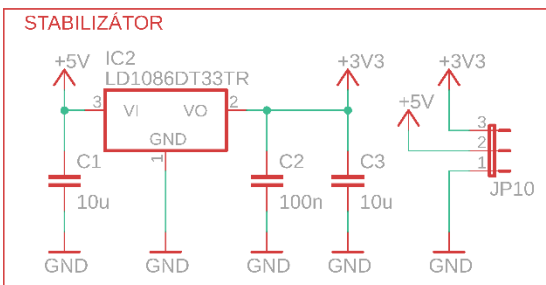
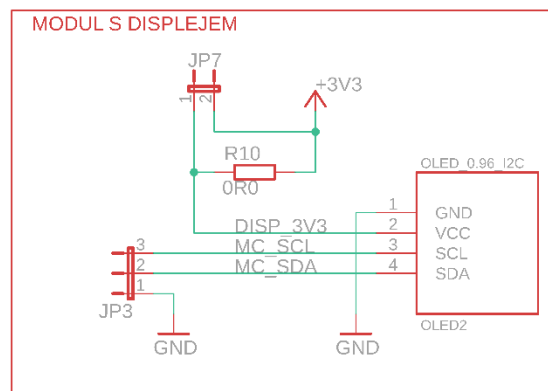
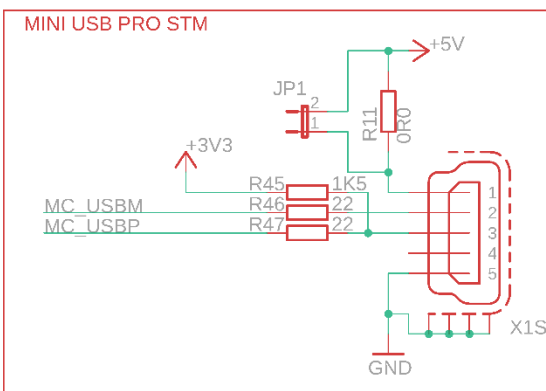
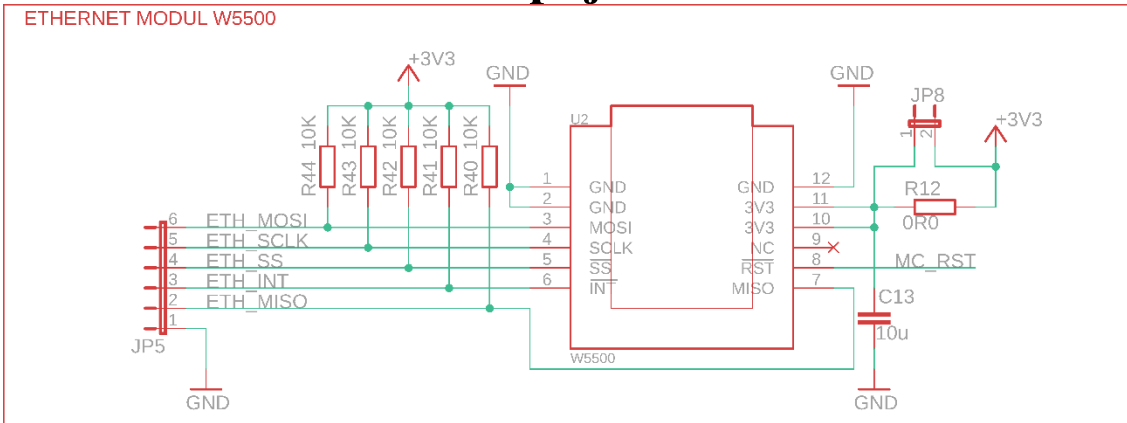
<b>Příloha 1</b>	<b>Schéma zapojení: 1. část</b>	<b>74</b>
<b>Příloha 2</b>	<b>Schéma zapojení: 2. část</b>	<b>75</b>
<b>Příloha 3</b>	<b>Schéma zapojení: 3. část</b>	<b>76</b>
<b>Příloha 4</b>	<b>DPS – TOP</b>	<b>77</b>
<b>Příloha 5</b>	<b>DPS – BOT</b>	<b>77</b>
<b>Příloha 6</b>	<b>Návrh laboratorní úlohy</b>	<b>78</b>

# Příloha 1 Schéma zapojení: 1. část

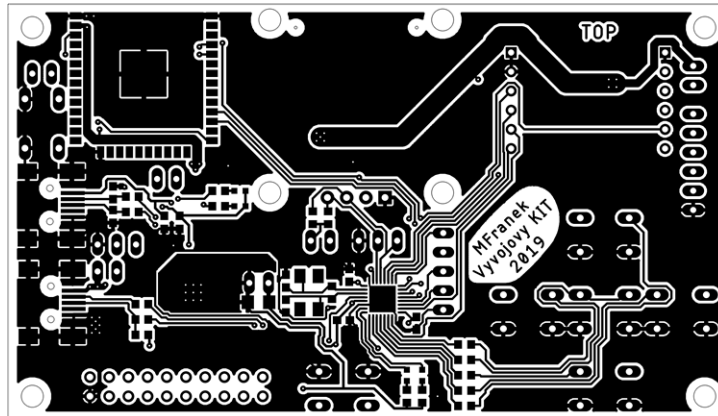




# Příloha 3 Schéma zapojení: 3. část

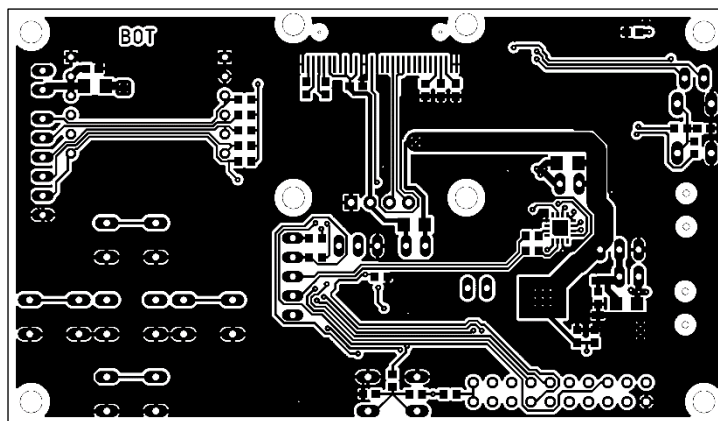


## Příloha 4 DPS – TOP



Rozměry DPS 95 x 55 mm, měřítko 1:1

## Příloha 5 DPS – BOT



Rozměry DPS 95 x 55 mm, měřítko 1:1

## Příloha 6 Návrh laboratorní úlohy

<b>Vysoké učení technické v Brně</b>	
<b>Fakulta elektrotechniky a komunikačních technologií</b>	
Ústav mikroelektroniky	
Seznámení s výukovým kitem pro připojení k počítači	
Jméno a příjmení, ID	
Datum	



### Cíle laboratorní úlohy

Tato laboratorní úloha slouží k seznámení studentů s vývojovým kitem pro komunikaci s počítačem. Vývojový kit může s počítačem komunikovat pomocí čtyř rozhraní: Ethernet, WiFi, USB a Bluetooth. Studenti se během laboratorní úlohy naučí, jak se pomocí aplikace na počítači nebo mobilním telefonu spojit s vývojovým kitem přes rozhraní Ethernet a Bluetooth.

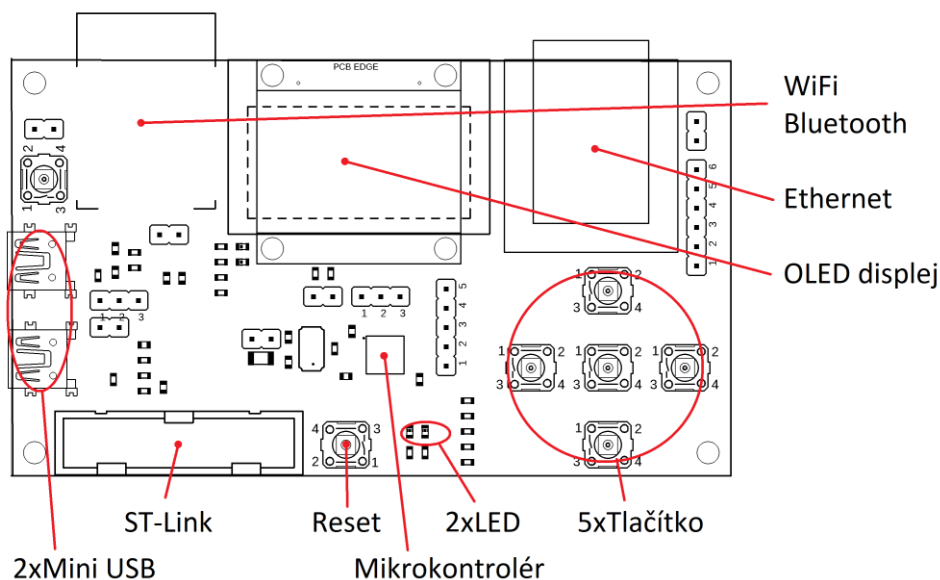
### Zadání

1. Seznamte se s vývojovým kitem, jeho součástmi a funkcí.
2. Vyzkoušejte komunikaci mezi počítačem a vývojovým kitem přes rozhraní Ethernet. Terminál vývojového kitu nastavte nejprve do módu ENDPOINT a poté do módu ECHO.
3. Otestujte připojení Bluetooth mezi mobilním telefonem a vývojovým kitem.
4. Vytvořte připojení Počítač – vývojový kit – Mobilní telefon. Počítač bude připojen přes Ethernet, mobilní telefon přes Bluetooth.

### Popis vývojového kitu

Jádrům vývojového kitu je mikrokontrolér STM32F302K8. Jedná se o 32bitový mikrokontrolér ARM. Mikrokontrolér může být k počítači připojen přímo přes rozhraní USB. To je vyvedeno na mini USB konektor, umístěný vedle konektoru pro připojení programátoru. Tento konektor slouží zároveň k napájení celého vývojového kitu, takže musí být vždy připojen.

Druhý konektor mini USB slouží ke zprostředkovanému připojení vývojového kitu přes rozhraní USB. Zprostředkované je z toho důvodu, že samotnou komunikaci USB zařizuje čip FT230. Jedná se o převodník z USB na rozhraní UART. Teprve UART je připojen do mikrokontroléru. V mikrokontroléru se o tuto komunikaci stará UART1.



Obr 1: Popis součástí vývojového kitu

UART2 mikrokontroléru je připojen k modulu ESP32. Tento modul zařizuje připojení mikrokontroléru k sítím WiFi a Bluetooth. Mikrokontrolér s ním komunikuje pomocí AT příkazů. Posledním rozhraním je Ethernet. Jeho připojení zprostředkovává modul s čipem W5500. Ten je připojen k mikrokontroléru pomocí rozhraní SPI3. V horní části uprostřed se nachází grafický černobílý OLED displej s rozlišením 128x64 pixelů.

## Firmware

Pro účely této úlohy byl vytvořen firmware, který obsahuje grafické prostředí, pomocí kterého je možné nastavit všechna rozhraní a terminál pro komunikaci. Terminál má tři různé módy nastavení: ENDPOINT, MOST a ECHO.

### ENDPOINT

V tomto módu je ke kitu připojeno pouze jedno rozhraní. Kit data přijímá a zobrazuje je na displeji. Při stisku tlačítka na kitu odesílá zprávu o stisku přes vybrané rozhraní.

### MOST

Ke kitu jsou připojeny dvě rozhraní. Kit přeposílá data z jednoho rozhraní na druhé a obráceně. Zobrazuje počet přeposlaných bajtů jednotlivými směry.

## **ECHO**

Zde je připojeno opět pouze jedno rozhraní. Přijatá data kit odesílá zpět po téže rozhraní. Na displeji je zobrazen počet přeposlaných bajtů.

## **Ovládání**

Grafické prostředí se ovládá 5 tlačítky na kitu. V seznamech se roluje tlačítky nahoru a dolů. Volby se potvrzují prostředním tlačítkem. Návrat zpět je možný tlačítkem vlevo. Pokud je na displeji zobrazena klávesnice, pro pohyb kurzoru se používají také tlačítka doleva a doprava.

## **Připojení kitu k počítači přes Ethernet**

Připojte kit mini USB kabelem k počítači. Kabel musí být zapojen do konektoru, který je blíže konektoru pro připojení ST-Link programátoru

1. Zkontrolujte, zda je v kitu nahrán správný firmware. Na displeji by mělo být zobrazeno "Vyberte mod terminalu". Případně naprogramujte do kitu správný firmware.
2. Připojte kit pomocí ethernetového kabelu do sítě, společné s počítačem. Pokud je kit připojen na switch, musí se použít přímý kabel, při připojení přímo k počítači je potřeba použít křížený kabel.
3. Pomocí grafického prostředí kitu vyberte mód ENDPOINT.
4. Vyberte rozhraní Ethernet. Zadejte IP adresu kitu, nebo ji nechte přiřadit DHCP serverem.
5. Zapište si přidělenou IP adresu.
6. Zjistěte IP adresu počítače.
7. Na počítači spusťte terminál Hercules.
8. Na kitu i terminálu počítače nastavte připojení UDP. IP adresu počítače zadejte do kitu a IP kitu naopak do počítače. Hodnota všech portů bude 4000.
9. Po úspěšném spojení vyzkoušejte odesílání dat z počítače do kitu. Stiskem

kteréhokoli z pěti ovládacích tlačítek se odešle zpráva z kitu do počítače.

10. Dlouhým stiskem prostředního tlačítka se ukončí terminál a zobrazí se opět původní obrazovka.
11. Dále zvolte mód ECHO.
12. Opakujte body 5 až 11 i pro tento mód. UDP spojení počítače zůstane vytvořeno, stačí znovu nastavit pouze připojení kitu.

### **Připojení kitu k mobilnímu telefonu přes Bluetooth**

1. Zprovozněte kit podle bodu 1 a 2 předchozího postupu.
2. Do mobilního telefonu si stáhněte aplikaci Bluetooth terminál z odkazu.
3. <https://play.google.com/store/apps/details?id=Qwerty.BluetoothTerminal&hl=cs>
4. Nastavte kit do módu ENDPOINT
5. Zvolte připojení Bluetooth a zadejte název zařízení.
6. Spárujte mobilní telefon s kitem.
7. Otevřete na telefonu Bluetooth terminál a navažte komunikaci s kitem.
8. Otestujte oboustrannou komunikaci mezi telefonem a kitem podle bodu 10 a 11 z předchozího postupu.
9. To samé pro mód ECHO.

### **Vytvoření spojení počítač – kit – mobilní telefon**

1. Pomocí znalostí z prvního a druhého postupu připojte kit k počítači přes rozhraní Ethernet a zároveň k mobilnímu telefonu přes Bluetooth. K tomu slouží mód terminálu MOST.
2. Odeslané zprávy z počítače by se měly zobrazovat na mobilu a naopak.
3. Vývojový kit bude zobrazovat počet přeposlaných bajtů.