



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

BLUETOOTH KOMUNIKAČNÍ SADA

BLUETOOTH COMMUNICATION SET

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Tomáš Plášek

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Vladimír Levek, Ph.D.

BRNO 2022

Diplomová práce

magisterský navazující studijní program **Mikroelektronika**

Ústav mikroelektroniky

Student: Bc. Tomáš Plášek

ID: 186450

Ročník: 2

Akademický rok: 2021/22

NÁZEV TÉMATU:

Bluetooth komunikační sada

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte modul na bázi mikrokontroléru, umožňující komunikaci prostřednictvím sítě Bluetooth v 5.0. Ten bude umožňovat vzájemnou komunikaci mezi moduly a také komunikaci s mobilním telefonem. Modul bude ovládán prostřednictvím sériové sběrnice a bude disponovat vybranými možnostmi komunikačního protokolu Bluetooth.

V rámci diplomové práce prostudujte protokol Bluetooth, provedte rešerši dostupných typů mikrokontrolérů a také prostudujte zásady návrhu vysokofrekvenčních obvodů užitých v modulu. Dále provedte návrh modulu a jeho realizaci. Závěrem bude provedeno testování modulu. To odborně popište a také kvantifikujte v textové části diplomové práce.

DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce

Termín zadání: 7.2.2022

Termín odevzdání: 24.5.2022

Vedoucí práce: Ing. Vladimír Levek, Ph.D.

doc. Ing. Lukáš Fucík, Ph.D.

předseda rady studijního programu

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

Obsahem této práce je popis vlastností a možností komunikačního rozhraní Bluetooth 5 a Bluetooth Low Energy. Následně jsou popsány mikroprocesory, které tímto rozhraním disponují, přičemž hlavní důraz je kladen na mikrokontroléry BlueNRG od společnosti STMicroelectronics. V další části jsou popsány zásady návrhu desek pro vysokofrekvenční aplikace a návrh desky plošných spojů, která disponuje zmíněným mikrokontrolerem.

Další část pojednává o vlastním navrhnutém zapojení a výsledném motivu desky plošných spojů. Jsou vysvětleny vlastnosti a význam jednotlivých funkčních bloků ve schématu a poté i na desce.

V závěru práce je popsán zdrojový kód pro mikrokontroler, jeho funkční bloky a struktura a jsou předvedeny ukázky povelování modulu a měření jeho parametrů.

Klíčová slova

Bluetooth 5, Bluetooth Low Energy, BlueNRG-232, STMicroelectronics, komunikační sada, UART, SPI, I²C

Abstract

This thesis describes features and capabilities of the Bluetooth 5 and Bluetooth Low Energy communication interface. Subsequently, microcontrollers with this interface are described with the main emphasis on BlueNRG microcontrollers from STMicroelectronics company. The next part is about designing PCBs for high-frequency applications and also board containing BlueNRG microcontroller is designed.

The next part is about design of final circuit schematic and motive of printed circuit board. Every functional block of printed circuit board is here also described and it's purpose explained.

Final chapters of this thesis describe how does the program for MCU works, it's functional parts and also example of sending commands to modules and verifying it's parameters is present.

Keywords

Bluetooth 5, Bluetooth Low Energy, BlueNRG-232, STMicroelectronics, communication set, UART, SPI, I²C

Bibliografická citace

PLÁŠEK, Tomáš. Bluetooth komunikační sada. Brno, 2022. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/142428>. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav mikroelektroniky. Vedoucí práce Vladimír Levek.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta: *Tomáš Plášek*

VUT ID studenta: *186450*

Typ práce: *Diplomová práce*

Akademický rok: *2021/22*

Téma závěrečné práce: *Bluetooth komunikační sada*

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucího závěrečné 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 závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné 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: 24. května 2022

podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Vladimírovi Levkovi, PhD. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

Dále děkuji Ing. Lukášovi Novákovi za pomoc při opravách a osazování desek plošných spojů součástkami.

V Brně dne: 24. května 2022

podpis autora

Obsah

SEZNAM OBRÁZKŮ	9
SEZNAM TABULEK.....	10
ÚVOD	11
1. STANDARD BLUETOOTH	12
1.1 HISTORIE BLUETOOTH	12
1.2 PROTOKOLY TVOŘÍCÍ BLUETOOTH	13
1.3 VRSTVA HOST	14
1.4 VRSTVA CONTROLLER	15
1.4.1 LL (Link Layer).....	15
1.4.2 BLE PHY.....	16
1.5 PROFILY APLIKACÍ A SLUŽEB.....	17
1.6 PRŮBĚH VZÁJEMNÉ KOMUNIKACE ZAŘÍZENÍ	17
2. MIKROKONTROLER BLUENRG-2.....	21
2.1 REŽIMY ZAŘÍZENÍ BLUETOOTH	22
2.1.1 Aktivní mód	22
2.1.2 Sniff mode (čihací mód)	22
2.1.3 Hold mode (neaktivní mód).....	22
2.1.4 Park mode (parkovací mód).....	22
2.2 ŘÍZENÍ SPOTŘEBY	22
2.3 PERIFÉRIE A ROZHRAŇÍ	23
2.3.1 ADC	24
2.3.2 DMA.....	24
2.3.3 SPI.....	24
2.3.4 UART	25
2.3.5 I ² C.....	25
2.3.6 GPIO.....	26
2.4 POPIS POUZDRA OBVODU BLUENRG-2	26
3. CHARAKTERISTIKA KOMUNIKAČNÍHO MODULU	29
3.1 PŘEVODNÍK Z USB NA UART.....	29
3.2 OCHRANA USB PORTU PROTI ESD.....	29
3.3 OLED DISPLEJ	30
3.4 TEPLOTNÍ SENZOR MCP9701T-E/LT	30
4. NÁVRH SCHÉMATU ZAPOJENÍ A PLOŠNÉHO SPOJE KOMUNIKAČNÍ SADY.....	31
4.1 BLUENRG-232 – HLAVNÍ ČÁST ZAPOJENÍ	31
4.2 PŘEVODNÍK ROZHRAŇÍ USB - UART	34
4.3 PERIFERIE A DALŠÍ PRVKY MODULU	35
5. PROGRAMOVÁNÍ MIKROKONTROLERU	38
5.1 STRUKTURA PROGRAMU.....	38
5.2 ZPOŽDĚNÍ.....	39
5.3 OVLÁDÁNÍ LED POMOCÍ TLAČÍTEK.....	39

5.4	PŘÍJEM POVELŮ A DAT POMOCÍ ROZHRAŇÍ UART	40
5.4.1	<i>Odeslání příkazu a dat z rozhraní UART</i>	41
5.4.2	<i>Odesílání příkazů a dat z rozhraní Bluetooth</i>	45
5.5	PŘÍJEM POVELŮ A DAT POMOCÍ ROZHRAŇÍ BLUETOOTH	47
5.6	ZÍSKÁNÍ DAT Z ADC PŘEVODNÍKU	49
5.7	NAMĚŘENÉ HODNOTY TEPLŮT.....	50
6.	MĚŘENÍ DOSAHU MODULU	53
7.	SEZNAM PŘÍKAZŮ PRO UART TERMINÁL A BLUETOOTH TERMINÁL	54
8.	ZÁVĚR.....	56
	LITERATURA.....	57
	SEZNAM SYMBOLŮ A ZKRATEK	61
	SEZNAM PŘÍLOH.....	62

SEZNAM OBRÁZKŮ

1.1	Přehled vrstev protokolů Bluetooth, převzato z [17].....	14
1.2	Přehled částí datového paketu, převzato z [18].....	16
1.3	Ukázka sítě pikonet	18
1.4	Princip inicializace zařízení typu klient a server	19
1.5	Princip komunikace Bluetooth majáků (Bluetooth Beaconů) [28].....	20
2.1	Mikrokontrolér BlueNRG-2 [19]	21
2.2	Stavy, ve kterých se může BlueNRG 2 nacházet, převzato z [19]	23
2.3	Sběrnice SPI s jedním nadřazeným zařízením typu Master a 3 podřízenými zařízeními typu Slave, převzato z [24].....	25
2.4	Sběrnice I ² C s jedním nadřazeným zařízením typu Master a 3 podřízenými zařízeními typu Slave, převzato z [26].....	26
2.5	Mikrokontrolér BlueNRG-2 v provedení QFN32, převzato z [19]	27
2.6	Mikrokontrolér BlueNRG-2 v provedení QFN48, převzato z [19]	27
2.7	Mikrokontrolér BlueNRG-2 v provedení WCSP34, převzato z [19]	28
3.1	Vnitřní zapojení obvodu TPD2E001 [30]	30
4.1	Zapojení mikroprocesoru BlueNRG-232	32
4.2	Návrh šířky přívodní cesty k anténě podle požadované impedance	33
4.3	Porovnání výsledného návrhu VF částí desky (nalevo) s doporučením výrobce (napravo).....	34
4.4	Zapojení rozhraní USB a převodníku FT230XQ	35
4.5	Diferenciální páry u USB konektoru	35
4.6	Zapojení tlačítek a ostatních periférií	36
4.7	Vrchní strana navržené desky plošných spojů.....	37
4.8	Spodní strana navržené desky plošných spojů (zrcadlově převrácená)	37
5.1	Vývojový diagram průběhu zpracování a vyhodnocování dat z protokolu UART	41
5.2	Ukázka odesílání dat z UART do rozhraní UART i Bluetooth zároveň, přijatá data z programu Hercules, který emuluje terminál	42
5.3	Ukázka odesílání dat z UART do rozhraní UART i Bluetooth zároveň, přijatá data v aplikaci Bluetooth serial terminal na mobilním telefonu s operačním systémem Android (modul je inicializován s názvem BLNRG232).....	43
5.4	Ukázka odesílání dat do Bluetooth terminálu, výstup z programu Hercules.....	44
5.5	Ukázka odesílání dat do Bluetooth terminálu, výstup na terminálu mobilního telefonu.....	44
5.6	Ukázka odesílání dat do UART terminálu, výstup z programu Hercules.....	45
5.7	Ukázka odesílání dat do UART terminálu, výstup z programu Bluetooth serial terminal na mobilním telefonu	45
5.8	Ukázka odesílání dat z mobilu do UART i Bluetooth terminálu, výstup v programu Hercules.....	46
5.9	Ukázka odesílání dat z mobilu do UART i Bluetooth terminálu, výstup na terminálu mobilního telefonu.....	47
5.10	Vývojový diagram popisující přijetí dat z protokolu Bluetooth	48
5.11	Ukázka hodnot teploty naměřené při konfiguraci AdcPin1 a AdcPin12	51
5.12	Ukázka hodnot teploty naměřené při konfiguraci TempSensor	52
6.1	Grafické znázornění naměřených hodnot intenzity výstupního signálu.....	53

SEZNAM TABULEK

5.1	Naměřené hodnoty teploty při různých senzorech	50
6.1	Naměřené hodnoty vysílaného signálu v závislosti na vzdálenosti od modulu	53
7.1	Přehled příkazů pro modul	54

ÚVOD

S postupným rozvojem moderních technologií v minulosti vznikla potřeba komunikovat bezdrátově i na malé vzdálenosti. Hlavním požadavkem bylo vytvořit univerzální komunikační standard, který bude schopen zajistit spolehlivou a bezpečnou komunikaci mezi dvěma a více zařízeními, které spolu mohou v jeden okamžik komunikovat.

V 80. letech 20. století, kdy začal tento standard vznikat, bylo nejrozšířenějším rozhraním sériové rozhraní RS-232. Proto z něj kvůli své poměrně velké rozšířenosti tento standard vychází a lze jej tak chápat jako jeho bezdrátovou variantu.

Bluetooth je v současné době široce rozšířený komunikační standard, který využívá většina moderních chytrých zařízení, jako jsou mobilní telefony, notebooky, přenosná rádia a podobně. Lze s ním přenášet data, zvuk nebo dálkově ovládat různá zařízení. Proto se také hojně využívá u zařízení chytré domácnosti k bezdrátovému propojení jednotlivých komponent, které tak lze bezdrátově ovládat, např. regulovat intenzitu a barvu osvětlení, ovládat žaluzie na oknech a podobně.

Nejčastěji se ale používá při bezdrátovém přenosu dat na krátké vzdálenosti. Běžně se proto setkáváme s bezdrátovými počítačovými myši a klávesnicemi, sluchátky pro poslech hudby nebo těmi, které slouží jako hands-free sada pro telefonování bez nutnosti držet mobilní telefon v ruce. U všech těchto zařízení Bluetooth šetří nutnost použití propojovacích kabelů.

Hlavním požadavkem na Bluetooth je nízká spotřeba a co nejvyšší dosah. Toho je dosaženo vhodnou volbou výstupního výkonu, který je přímo závislý na spotřebě daného zařízení. Tyto a další klíčové vlastnosti by bylo možné ověřit ve výuce pomocí sady ukázkových modulů, které by simulovaly reálná zařízení. Bylo by tak možné vyzkoušet nastavení parametrů zařízení, odesílání dat v podobě textu z jednoho zařízení do druhého a sledovat jejich vzájemnou komunikaci pomocí terminálu na PC nebo v mobilním telefonu.

Semestrální práce je rozdělena do tří částí. V první kapitole je popsána historie a princip technologie Bluetooth, jsou popsány jednotlivé vývojové verze a u každé je zmíněno, co přinesla nového. Následuje popis částí, které tvoří standard Bluetooth a je vysvětlena jejich úloha při procesu komunikace.

Druhá kapitola popisuje mikrokontroler BlueNRG od firmy STMicroelectronics, který byl vybrán, protože disponuje standardem Bluetooth 5. Zaměřuje se na jeho periferie, rozhraní, popis pouzdra a funkce jednotlivých pinů.

Třetí část se zaměřuje na principy a zásady navrhování desek plošných spojů pro vysokofrekvenční aplikace, schéma zapojení navrženého modulu a motiv desky plošných spojů.

1. STANDARD BLUETOOTH

Hlavním z důvodů vzniku tohoto standardu byla snaha o vytvoření univerzálního komunikačního rozhraní, které bude kompatibilní ideálně s většinou zařízení dostupných na trhu a bude umožňovat bezdrátové propojení těchto zařízení a přenos dat mezi nimi. V době svého vzniku bylo nejvíce používaným standardem pro propojování zařízení rozhraní RS232, které v dnešní době v podobě kabelového propojení dvou zařízení nahradilo rozhraní USB (Universal Service Bus – univerzální rozhraní služeb). Pro svoji rozšířenost z něj později bylo odvozeno i rozhraní Bluetooth. Proto mají tato dvě rozhraní podobnou topologii a lze Bluetooth v podstatě považovat za bezdrátovou variantu rozhraní RS232.

V této kapitole je popsáno, jakým způsobem probíhá komunikace prostřednictvím standardu Bluetooth, z jakých vrstev se tento standard skládá a jaký je jejich význam při komunikaci. V části o historii je popsáno, s jakými přínosy jednotlivé verze postupně přicházely, jaké přinášely novinky. Technicky jde o velmi rozsáhlý komunikační standard, který vyvíjela řada firem mnoho let. Proto tato kapitola zachycuje základní přehled těchto vlastností.

1.1 Historie Bluetooth

Technologii bezdrátového přenosu dat na krátké vzdálenosti vytvořili roku 1994 dva tehdejší zaměstnanci firmy Ericsson – Jaap Haartsen a Sven Mattinson. Název této technologie odkazuje na anglický překlad jména dánského krále Haralda Modrozuba, který byl známý svými diplomatickými schopnostmi, když dokázal sjednotit rozhádané kmeny. Z toho vyplývá analogie, že by Bluetooth měl být společným prostředkem vzájemné komunikace, který pomáhá sjednotit komunikaci mezi zařízeními. Později roku 1999 vznikla skupina Bluetooth Special Interest Group, která dohlíží na aktualizaci a rozšiřování funkcí tohoto standardu [1] [2].

První verzí, která se objevila na trhu, byl Bluetooth 1.0 a 1.0b. V době svého uvedení na trh, v roce 1999, však ještě obsahovala řadu technických problémů, které musely být před rozšířením této technologie vyřešeny. Jednalo se o bezdrátovou variantu sériového rozhraní RS-232 a problémy byly především s párováním a zabezpečením připojení. Další verze, Bluetooth 1.1, která vyšla v roce 2002, přinesla kromě opravy předchozích problémů i nové funkce, např. indikátor síly signálu **RSSI** (Received Signal Strength Indicator) [4]. Tato verze byla standardizována a popsána standardem IEEE 802.15.1 [5] a dočkala se prvního širšího průmyslového využití.

O rok později vyšla verze 1.2, která přinesla zvýšení přenosové rychlosti, metodu přeskokování frekvence **FHSS** (Frequency Hopping Spread Spectrum) [3], kdy dochází v okolí frekvence 2,4 GHz k pravidelnému přeladování vysílané frekvence v určité sekvenci po předem stanoveném počtu kroků, které jsou velké přibližně 1 MHz. Posílání

dat tedy probíhá cyklicky po definovaných částech, a když dojde k chybnému přenosu některé z nich, je v dalším kroku odeslána znovu po jiné frekvenci, což celkově zvyšuje spolehlivost přenosu. Byla také přidána vrstva **HCI** (Host Controller Interface) [6], standardizované rozhraní mezi hostitelem a řadičem, které popisuje komunikaci mezi hostitelským zařízením (např. mobilní telefon) a řadičem, což je mikrokontroler s rozhraním Bluetooth.

V roce 2007 vyšla nová verze 2.1 + EDR, kde EDR znamená Enhanced Data Rate (zvýšená rychlost přenosu dat), přinesla možnost bezpečného jednoduchého párování (Secure Simple Pairing), párování s použitím párovacího PIN kódu a tím došlo ke zvýšení bezpečnosti připojení.

O dva roky později, roku 2009, přinesla verze Bluetooth 3 možnost použití alternativní vrstvy MAC PHY. Při vyhledávání zařízení, připojování k nim a konfiguraci se používá klasický nízkenergetický mód, ale při posílání velkého objemu dat se použije vysokorychlostní vrstva MAC PHY 802.11. Další novinkou byla možnost jednosměrného posílání dat. Při něm je sice možné posílat pouze malé objemy dat, ale je možné jej využít v aplikacích, kde je kladen důraz na krátkou odezvu zařízení na povelů uživatele, tedy např. při vysílání povelů zařízení.

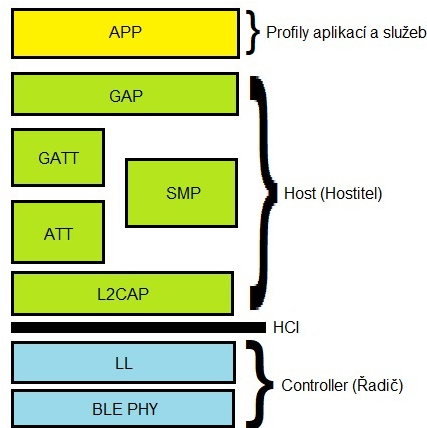
V současnosti stále používá mnoho zařízení Bluetooth 4.2 z roku 2014, který obsahuje protokol 6LoWPAN, který kombinuje protokol IPv6 s osobní bezdrátovou sítí (Wireless Personal Area Network) a umožňuje tak propojit zařízení s různými sítěmi, včetně internetu.

Nejnovější verze Bluetooth 5 z roku 2016 nabízí několikanásobně vyšší dosah, než předchozí verze a možnost přepínání rychlostí přenosů.

1.2 Protokoly tvořící Bluetooth

Celá technologie je tvořena mnoha protokoly, které tvoří sady instrukcí a postupů, pomocí kterých fungují jednotlivé činnosti v rámci celého procesu komunikace. Jde tedy o knihovny obsahující funkce, které zajišťují všechny činnosti potřebné pro fungování spojení.

Jednotlivé vrstvy ukazuje obrázek 1.1. Jedná se o tři hlavní části, kterými jsou **APP** (Application Profiles and Services), tedy profily, které se starají o přenos konkrétních druhů informací či dat, dále následuje vrstva Host, která obsahuje protokoly pro řízení vysílání a zpracování dat a jejich následné vysílání v podobě datových paketů. Vrstva Controller přijímá a zpracovává data, která poté vysílá do vrstvy Host v podobě čistých dat.



Obrázek 1.1 Přehled vrstev protokolů Bluetooth, převzato z [17]

1.3 Vrstva Host

Mezi významné protokoly patří **LMP** (Link Management Protocol), který se stará o vytvoření a ukončení spojení a také o jeho řízení.

Další vrstvou je **L2CAP** (Logical Link Control and Adaptation Protocol), která poskytuje multiplexní předávání dat mezi vyššími vrstvami technologie, kam patří třeba různé řídicí příkazy nebo povely, do nižších vrstev, které přímo zajišťují vykonávání požadovaných činností [9]. Stará se také o kontrolu, zda byly obdrženy správně instrukce pomocí **CRC** (Cyclic Redundancy Check) [10], v doslovném překladu „stále se opakující vynechatelný součet“. Při každém přenosu dat je z nich pomocí matematického algoritmu vypočítán kontrolní součet, který je odeslán společně s daty. Po jejich přijetí je znovu tento součet vypočítán, a pokud je jeho nová podoba stejná s tou původní, tak byla data přijata v pořádku. Pro samotný přenos dat však není kriticky důležitý a je možné jej uskutečnit i bez jeho použití (avšak s rizikem přijetí neúplných nebo poškozených dat). Odtud tedy plyne význam slova redundantní.

Dalším významným protokolem je **SDP** (Service Discovery Protocol) [11], který zajišťuje to, že se mohou jednotlivá zařízení vzájemně vyhledávat a být vyhledávána. Zajišťuje přiřazení zařízení **UUID**, jedinečný identifikátor, který je důležitý při procesu propojování zařízení.

GATT (Generic Attribute Profile) umožňuje přenos dat mezi dvěma zařízeními. Komunikace vždy probíhá mezi klientem a serverem, kde klient je řídicí zařízení a ovládá server, který mu posílá odpovědi. Klient může být připojen vždy pouze k jednomu serveru, ale server může být spojen s několika klienty najednou. Součástí tohoto protokolu jsou jednotlivé profily, které popisují jednotlivé druhy komunikací a jsou blíže popsány v samostatné kapitole. Dále jsou zde obsaženy charakteristiky a služby, které slouží k přenosu vlastních informací [12].

GAP (Generic Access Profile) má na starosti určení role zařízení (jestli bude slave nebo master) a vysílání dat v módu, kde je zařízení viditelné pro ostatní a vysílá advertising pakety obsahující informace o zařízení. Umožňuje určit, jaké informace o zařízení budou vysílány a jak často. Čím větší bude prodleva mezi odesláním advertujících (nabízejících) paketů, tím bude menší spotřeba zařízení, ale prodlouží se doba potřebná pro vzájemné nalezení zařízení [13].

HCI (Host Controller Interface) umožňuje vzájemnou komunikaci mezi klientem a serverem. Obsahuje funkce, které softwarově ovládají daný mikrokontroler, mezi které patří ovládání komunikačních rozhraní (RS232, USB a UART), příkazy pro reset zařízení, zahájení připojování, odpojení, nastavení advertising parametrů, testovací příkazy pro otestování komunikace a mnoho dalších.

RFCOMM je sada transportních protokolů sloužících k emulaci sériového rozhraní RS232. Díky tomu lze komunikovat se zařízením Bluetooth pomocí sériového portu. Hlavním důvodem, proč se tento standard používá, je jeho široká podpora většinou operačních systémů.

ATT (Attribute Protocol) popisuje, v jaké podobě se ukládají a přenášejí data. Zároveň slouží k přiřazování atributů (vlastností) těmto datům proto, aby bylo možné rozeznat, jaká data daná sekvence hodnot přenáší. Data jsou tedy přenášena v podobě atributů v tomto pořadí: typový atribut (popsán podle UUID – unikátního identifikátoru dané služby), atribut hlavička (Attribute Handle), což je číselná hodnota jedinečná pro daný typ atributu, atribut oprávnění (Attribute Permissions), který udává, jestli přenášené informace klient může jen číst nebo i upravovat a nakonec Attribute Value, tedy atribut se samotnou hodnotou přenášené informace [16].

SMP (Security Manager Protocol) zabezpečuje proces párování a ověřování identity zařízení. Využívá několika bezpečnostních klíčů, mezi které patří např. **STK** (Short Term Key), krátkodobý klíč, který si zařízení vyměňují při prvním párování, **LTK** (Long Term Key), dlouhodobý klíč pro šifrování dat a ověřování identity zařízení v linkové vrstvě **LL**, **CSRK** (Connection Signature Resolving Key), klíč sloužící k podepisování dat ve vrstvě **ATT** a **IRK** (Identity Resolving Key), který slouží k vytvoření soukromé adresy) [15].

1.4 Vrstva Controller

1.4.1 LL (Link Layer)

Link Layer, odkazová vrstva, se stará o správu vysílacích kanálů, příjem a odesílání paketů a vyhledávací a připojovací procesy. Zahrnuje **LMP** (Link Management Protocol), protokol správy odkazů, který zajišťují odesílání dat mezi klientem a severem, stará se o jejich autentizaci, šifrování posílaných zpráv a posílání informací o velikosti paketů. Do odkazové vrstvy patří také protokol **L2CAP**, díky které je zařízení schopno reagovat na povely z vyšších vrstev odesláním příslušných informací.

1.4.2 BLE PHY

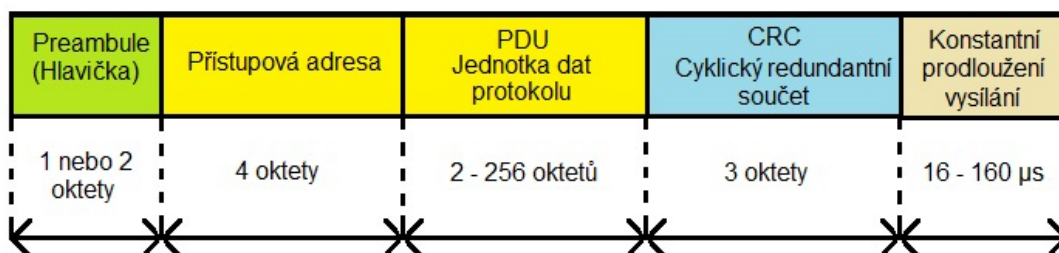
Bluetooth Low Energy Physical Layer, tedy fyzická vrstva této technologie, zajišťuje fyzicky samotné vysílání a příjem dat. Jde o bezdrátové rozhraní, které pracuje v rozmezí frekvencí 2,402 – 2,4835 GHz. Vysílání dat probíhá vysíláním paketů na jednotlivých kanálech, které se v pravidelných intervalech střídají. Šířka kanálu je 2 MHz a frekvence každého kanálu je dána vztahem (1.1):

$$f_k = 2402 + 2n \text{ [MHz]}, \quad (1.1)$$

kde f_k je frekvence kanálu v MHz a n je číslo kanálu (0, 1, 2, ..., 39).

Vysílání advertising paketů probíhá na kanálech 37, 38 a 39. Uživatelské pakety jsou vysílány na kanálech 0 – 36. Tím je zamezeno rušení s ostatními zařízeními, která vysílají ve stejném pásmu. Pokud některý z paketů přijde porušen nebo nedorazí vůbec, přijímající zařízení vyšle požadavek o jeho opakované zaslání.

Tvar paketu je znázorněn na obrázku 1.2, délka každé části je vyjádřena v oktetech, což je osmibitové číslo v binárním tvaru (např. „0011 0101“ atd.)



Obrázek 1.2 Přehled částí datového paketu, převzato z [18]

Hlavička slouží k synchronizaci frekvence vysílajícího a přijímajícího zařízení, k automatickému řízení zisku (citlivosti) přijímače a pro odhad časování, kdy data přišla. Jde o fixní sekvenci střídajících se nul a jedniček.

Přístupovou adresu náhodně generuje vysílající zařízení pokaždé, když jsou vysílána data nebo advertising (nabízeč) paket. Platí pro ni určitá pravidla – nesmí se v ní vyskytovat více než 6 po sobě jdoucích nul nebo jedniček, musí být unikátní vůči všem ostatním probíhajícím přenosům, všechny 4 oktety musí být různé atd.

PDU (Protocol Data Unit) nese informaci o tom, jestli daný paket prošel přes advertising kanály a je tedy advertising (vysílající zařízení vysílá informaci o tom, že je viditelné pro ostatní zařízení) a nebo že prošel přes některý z datových kanálů a zařízení tedy pouze odesílá data.

Cyklický redundantní součet (CRC) slouží ke kontrole, jestli je přijatý paket stejný jako ten odeslaný. Při odesílání paketu se z jeho obsahu vypočítá kontrolní součet, který se poté stejným algoritmem spočítá i z přijatého paketu. Pokud jsou oba součty stejné, jsou přijatá data v pořádku. V opačném případě je vyslán požadavek na opakované zaslání daného paketu.

Poslední část tvořena sekvencí signálu logické úrovně 1 modulovaného na nosnou frekvenci slouží k zajištění stále stejné fáze signálu. Jeho frekvence se neustále mění díky Gaussově frekvenční modulaci (GFSK – Gaussian Frequency Shift-Keying), kdy dochází k namodulování digitálních dat na nosný signál. V místě přenosu logické úrovně 1 dochází ke zvýšení frekvence a v místě přenosu logické nuly k jejímu snížení. Kdyby měly signály paketů různé fáze, nešlo by spolehlivě určit začátek jejich vysílání a docházelo by při přenosech k chybám [18].

1.5 Profily aplikací a služeb

Pro každý druh připojení nebo přenosu se používá specifický profil, který definuje, pomocí jakých instrukcí spolu mají zařízení komunikovat. Jde o sadu příkazů, které zajišťují vykonávání dané činnosti. Některé profily spolu mohou spolupracovat, např. **HFP** zajišťující ovládání hands-free sad obsahuje taky **A2DP** sloužící k přenosu hudby.

Mezi nejpoužívanější patří **A2DP** (Advanced Audio Distribution Profile), pokročilý profil pro šíření hudby, **AVRCP** (Audio/Video Remote Control Profile), který slouží k posílání příkazů, které ovládají zvuková zařízení (např. změnu hlasitosti nebo skladby) nebo k vysílání informací o autorovi skladby nebo jejím názvu např. do autorádia.

BIP (Basic Imaging Profile) umožňuje odesílat fotografie, dálkově ovládat fotoaparát nebo dálkově promítat snímky do dataprojektoru.

BPP (Basic Printer Profile) slouží k odesílání dokumentů do tiskárny k tisku

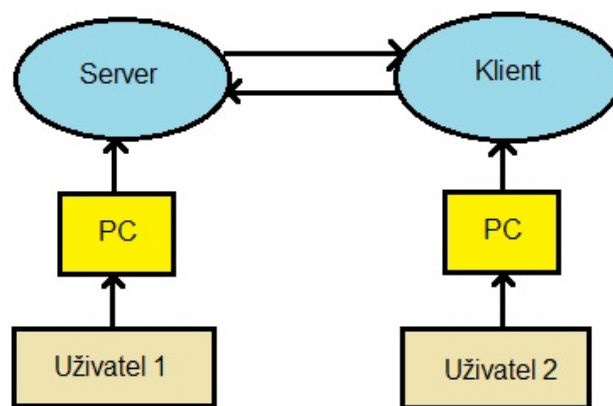
GAVDP (Generic Audio/Video Distribution Profile) má za úkol zajistit vytvoření živého přenosu obrazu a zvuku přes Bluetooth. Každý přenos je tvořen iniciátorem (**INT**), zařízením, které zahájí signalizační proces a akceptorem (**ACP**), který by měl na tento požadavek odpovědět.

HFP (Hands-Free Profile) zajišťuje komunikaci hands-free sad v automobilech s mobilními telefony. Jde tedy o přenos povelů k přijetí či odmítnutí hovoru, přenos čísla volajícího z telefonu do rádia automobilu a jeho zobrazení na displeji rádia apod.

HID (Human Interface Device Profile) umožňuje komunikaci vstupních uživatelských periférií, jako je třeba myš, klávesnice nebo joystick (herní ovládání) s počítačem [14].

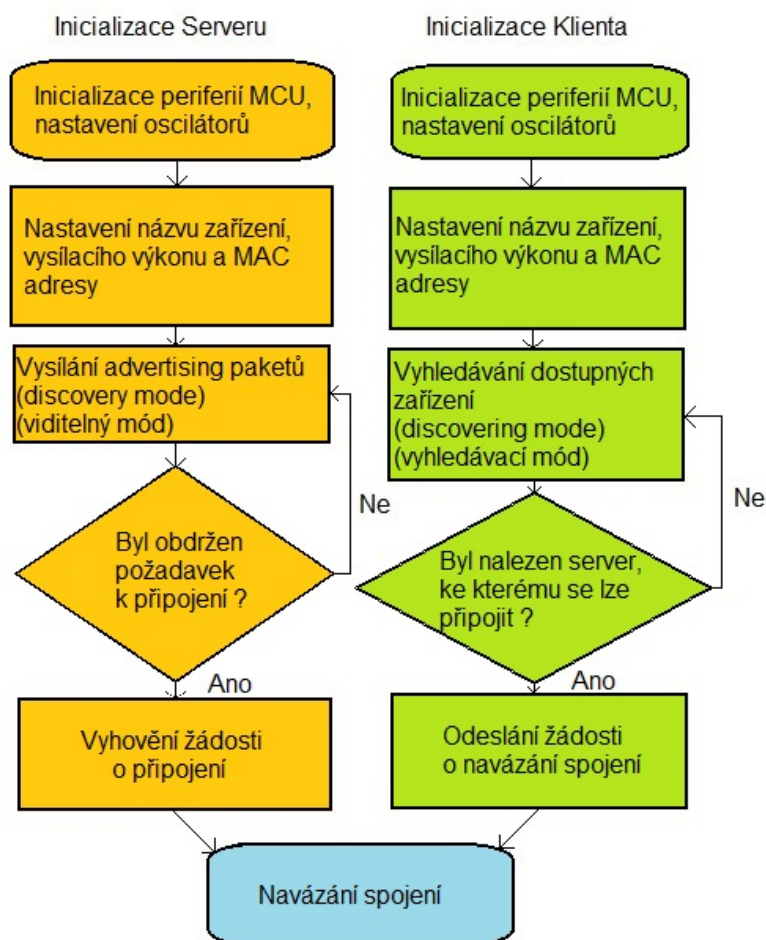
1.6 Průběh vzájemné komunikace zařízení

Existují dva způsoby komunikace – dvoubodová, kdy je o klasickou vzájemnou komunikaci dvou zařízení a vícebodová – v ní má jedno zařízení status slave (server), tedy podřízený, a k němu se může připojit až 7 zařízení s označením master (klient), nadřízený. Taková síť se nazývá pikonet. Tyto pikosítě však mohou být provázány a některá zařízení v nich mohou fungovat jako master i slave zároveň. Takovým sítím se říká scatternet.



Obrázek 1.3 Ukázka sítě piconet

Komunikace začíná inicializací zařízení typu server. Při inicializaci se nastaví název serveru, vysílací výkon a MAC adresa. Poté se na server přepne do viditelného (discoverable) módu a začne vysílat advertising pakety. To jsou řetězce informací (číselných hodnot v hexadecimálním tvaru) tvořené informací o počtu bitů paketu (jeho délce), typem informace a hodnotou dané veličiny [8]. Tak získá klient potřebné informace k připojení, kam patří např. název zařízení, UUID (Universally unique identifier – Univerzální jedinečný identifikátor), který slouží k identifikaci typu přenosu (který typ/profil služby se pro dané připojení používá) apod. [7]. Následuje inicializace klienta, kterému je také přiřazena MAC adresa a jméno. Ten poté začne vyhledávat dostupná zařízení, ke kterým se může připojit. Při požadavku na připojení mu server předá informace o sobě, jako je jeho název, druh zařízení, MAC adresa apod. a dojde k navázání spojení. Po zapnutí notifikace na klientovi může probíhat obousměrné posílání dat mezi klientem a serverem.

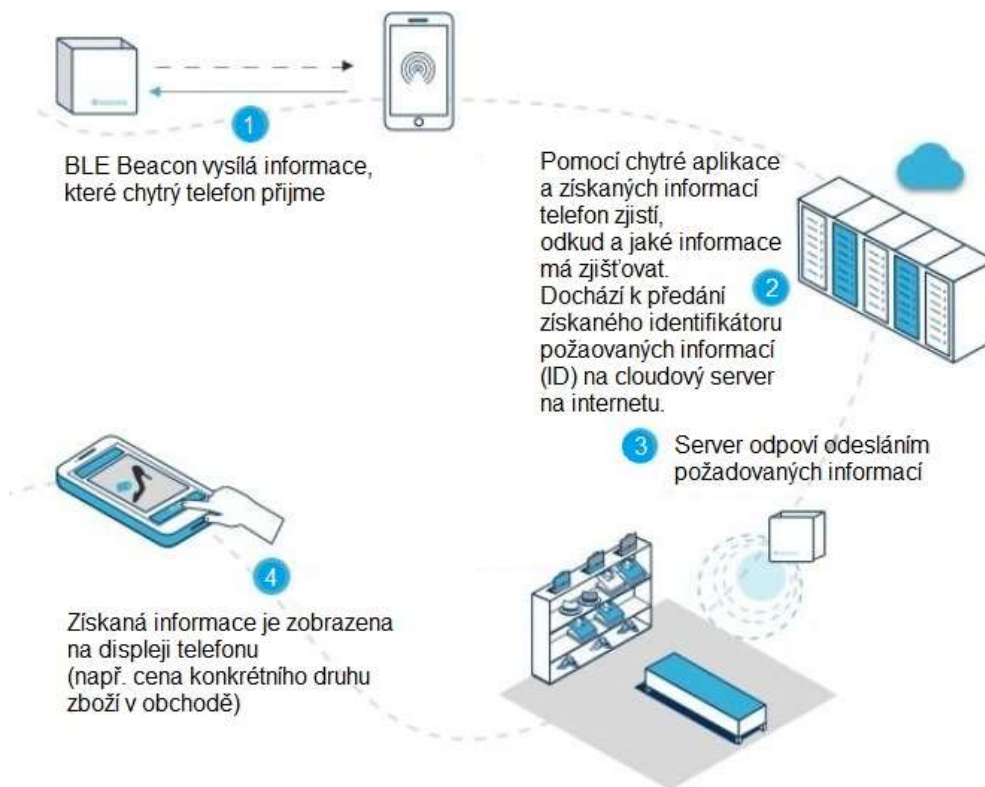


Obrázek 1.4 Princip inicializace zařízení typu klient a server

Takto by spolu teoreticky mohla komunikovat libovolná dvě zařízení, což představuje nemalé bezpečnostní riziko např. u zařízení ovládaných dálkově. Proto je pro posílení bezpečnosti možné využít možnost párování. Díky němu je možné zajistit, že se dvě spárovaná zařízení mohou automaticky spojit, aniž by bylo nutné je pokaždé manuálně k sobě připojovat. O párování může žádat klient, nebo může být iniciováno pokaždé automaticky ze strany serveru. Může být doprovázeno pouze potvrzením od protistrany nebo zadáním až 15-místného PIN kódu (zpravidla se používá 4-místný), který je předem stanoven. Každé takové spárování je možné v budoucnu zrušit.

Zařízení může také periodicky vysílat informace a chovat se tak jako jakýsi pomyslný maják (Beacon). Ten vysílá v pravidelných intervalech daný identifikátor (UUID), na který může např. aplikace v mobilním telefonu se zapnutým Bluetooth zareagovat a následně odeslat požadavek na informace pomocí předání ID (identifikátoru) daného beaconu cloudovému internetovému serveru, ke kterému se připojí pomocí Wi-Fi, což je server, na který je možné v reálném čase neustále zapisovat data a číst je z něj, tedy jakési online úložiště. Ten mu odpoví a odešle požadovaná data, která jsou poté v zařízení zobrazena. Tím dochází propojení obou těchto technologií (Bluetooth a Wi-Fi).

Taková zařízení by v budoucnu mohla sloužit k upozorňování zákazníků na nové atraktivní produkty, když např. projdou kolem obchodu nebo mohou usnadnit nalezení drobných ztracených věcí, třeba klíčů. Jde tak o interaktivní variantu klasických informativních cedulí a zároveň o technologii, která má vysoký potenciál ke svému rozšíření do mnoha odvětví průmyslu [27].



Obrázek 1.5 Princip komunikace Bluetooth majáků (Bluetooth Beaconů) [28]

Jedním z dalších zajímavých využití by do budoucna mohlo být i využití beaconů v restauraci, kdy by číšník při zadávání objednávky do pokladního systému pomocí tabletu připojeného k internetu přes Wi-Fi nemusel pokaždé zadávat číslo stolu, ale tablet by obdržel od Bluetooth beaconu namontovaného např. pod stolem informaci o čísle stolu, která by byla automaticky přiřazena k objednávce a ta tak byla spárována se správným stolem, což by přineslo jisté zjednodušení například ve velkých podnicích s mnoha hosty. Tento princip by bylo možné využít i obráceně, místo odesílání informací o navštíveném místě je možné odesílat informace o návštěvníkovi daného místa. V přeneseném významu by tedy bylo možné například touto technologií sledovat počet návštěvníků dané restaurace nebo zaměstnanců na školení v kanceláři apod.

Jako značné usnadnění by mohla tato technologie sloužit i nevidomým lidem k orientaci na veřejných místech. Po propojení mobilního telefonu s bezdrátovým sluchátkem by tak mohli třeba nevidomí cestující při průchodu na nádraží dostávat do sluchátka informaci o tom, u jakého nástupiště se právě nacházejí a informace o odjezdu nejbližšího spoje.

2. MIKROKONTROLER BLUENRG-2

Pro realizaci zadání této práce byl vybrán mikrokontrolér s označením BlueNRG-2 od firmy STMicroelectronics. Mezi jeho základní vlastnosti patří široký rozsah pracovních teplot od -40°C až do $+105^{\circ}\text{C}$, jádro Cortex-M0 založené na 32 bitové architektuře s nízkou spotřebou energie. To umožňuje použít tento obvod v zařízeních napájených bateriemi v mnoha odvětvích elektroniky, kam patří např. prvky chytrých domácností (barvu a intenzitu osvětlení, žaluzie, termostaty topení a další věci lze ovládat pomocí telefonu na dálku), chytré hodinky synchronizované s mobilním telefonem (díky čemuž lze pomocí nich např. přijímat sms zprávy), zabezpečovací technika (otevírání brány nebo vrat na dálku pomocí telefonu) a spousta dalších.



Obrázek 2.1 Mikrokontrolér BlueNRG-2 [19]

Každé zařízení s tímto mikrokontrolérem může mít 4 role:

- Central – Zařízení typu klient, které může vyhledávat ostatní zařízení a připojovat se k nim. Může data vysílat i přijímat. Jako příklad lze uvést mobilní telefon nebo notebook.
- Periferial – Zařízení typu server, vysílající advertising pakety (tedy může být nalezeno ostatními, kteří se k němu mohou připojit). Umožňuje odesílání i příjem dat.
- Broadcaster – Zařízení, které data pouze vysílá. Ostatní zařízení se k němu nemůžou připojit.
- Observer – Zařízení, které pouze data přijímá. Stejně jako broadcaster neumožňuje k sobě připojit ostatní zařízení [22].

Každý mikrokontrolér může mít roli Master nebo Slave a obsahuje taky programovatelnou flash paměť o velikosti 256 kB s 24 kB RAM paměti, řízení spotřeby, vlastní oscilátor, generátor náhodných čísel (RNG – Random Number Generator) a PKC (Public Key Cryptography), tedy šifrování přenášených dat pomocí veřejného klíče.

2.1 Režimy zařízení Bluetooth

Po nalezení a prozkoumání okolních dostupných zařízení a zjištění informací o nich (při procesu Discovery), může být připojení provedeno ve 4 režimech (módech). Ty jsou programovatelné a lze je v mikrokontroleru softwarově přepínat. Jsou seřazeny sestupně podle spotřeby elektrické energie.

2.1.1 Aktivní mód

V tomto módu zařízení aktivně přijímá a vysílá datové pakety. Jeho spotřeba je na maximální hodnotě. Pokud ovšem delší dobu nepřichází žádná data, může zařízení samovolně přecházet do dalších energeticky úspornějších módů.

2.1.2 Sniff mode (číhací mód)

Zařízení v daných časových úsecích, jejichž délka je nastavitelná, hlídá, jestli nepřišla nějaká data (např. každých 100 ms) a jinak je ve Sleep módu. Po přijetí dat přechází do aktivního módu. Ze všech módů je tento nejméně energeticky úsporný.

2.1.3 Hold mode (neaktivní mód)

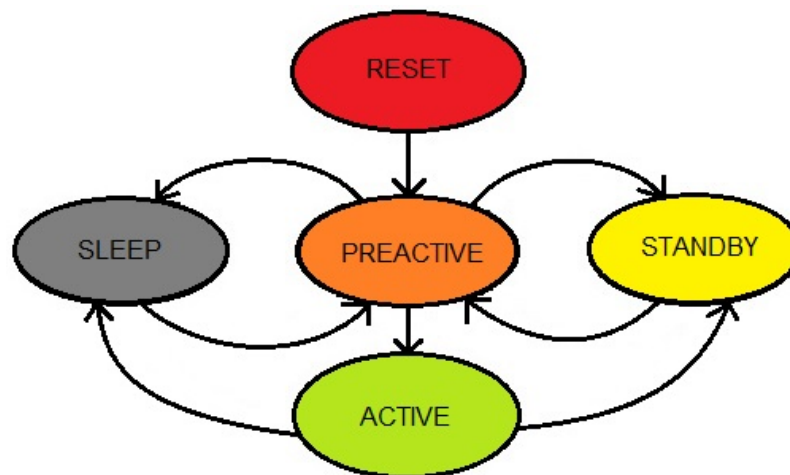
Zařízení je po zadanou dobu neaktivní a nepřijímá žádná data. Po uplynutí této doby přechází automaticky do aktivního stavu. Jedná se o energeticky úspornější mód, než předchozí jmenovaný Sniff mód.

2.1.4 Park mode (parkovací mód)

Jedná se o stav, kdy má zařízení nejmenší možnou spotřebu. Master (klient) může poslat na Slave (server) povel, aby zůstal vypnutý do té doby, než mu zase pošle příkaz k tomu, aby přešel do aktivního režimu [21]. V tomto módu je prodleva mezi kontrolami přijatých dat ze všech 4 módů největší, zařízení má tedy nejdelší duty cycle (pracovní cyklus).

2.2 Řízení spotřeby

Zařízení se může nacházet v několika různých stavech, které popisuje obrázek 2.2. Jedná se o aktuální stavy popisující chování obvodu. Přechodem mezi těmito stavy je možné ovlivňovat výslednou spotřebu energie MCU a tím i spotřebu celého zařízení..



Obrázek 2.2 Stavy, ve kterých se může BlueNRG 2 nacházet, převzato z [19]

Po připojení napájecího napětí se zařízení nachází ve stavu **Reset**, kdy je zajištěno, že nemůže být vykonána žádná instrukce ze strojového kódu. Až se napájecí napětí po určité době ustálí, dojde ke zrušení stavu Reset a zařízení přejde do stavu Preactive. Účelem tohoto procesu je zajistit, aby se všechny periferie mikrokontroleru inicializovali správně, aby všechny registry obsahovaly správné a nikoliv náhodné hodnoty, a aby se všechny vnitřní oscilátory inicializovaly přesně. Tento proces nazýváme Power on Reset (PoR) [20].

Ve stavu **Preactive** jsou k dispozici oba vnitřní oscilátory s frekvencí 16 MHz a 32 kHz a napájecí napětí všech periférií je stabilní. Zařízení poté přechází do režimu Active.

V aktivním režimu zajišťuje provoz vysokofrekvenčního 16 MHz oscilátoru externí oscilátor řízený krystalem a interní oscilátor je vypnut (interní oscilátor 32 kHz je pořád aktivní). Zároveň dochází k vyvolání a nastavení uložené konfigurace GPIO pinů (General Purpose Input Output pins – Vstupně výstupní piny pro všeobecné použití, piny mikrokontroleru, pomocí nichž lze s mikroprocesorem komunikovat nebo on může komunikovat s ostatními zařízeními).

Ve **Standby** módu je zařízení neaktivní a čeká na připojení jiného zařízení. Po jeho připojení dojde k přechodu do stavu Preactive a následně do stavu Active, kdy je již zařízení plně inicializováno a tím tedy dojde k jeho „zapnutí pro přenos dat“.

Při přechodu do **Sleep** módu dochází k vypnutí interního nízkofrekvenčního (32 kHz) oscilátoru a slouží hlavně ke snížení spotřeby zařízení, když právě není aktivní žádná uživatelská aplikace a zařízení čeká na příjem povelu.

2.3 Periférie a rozhraní

Pro správnou komunikaci mezi samotným mikrokontrolerem a ostatními perifériemi slouží různá rozhraní, která se starají o samotný přenos informací. V následujících kapitolách jsou jednotlivá rozhraní popsána podrobněji.

2.3.1 ADC

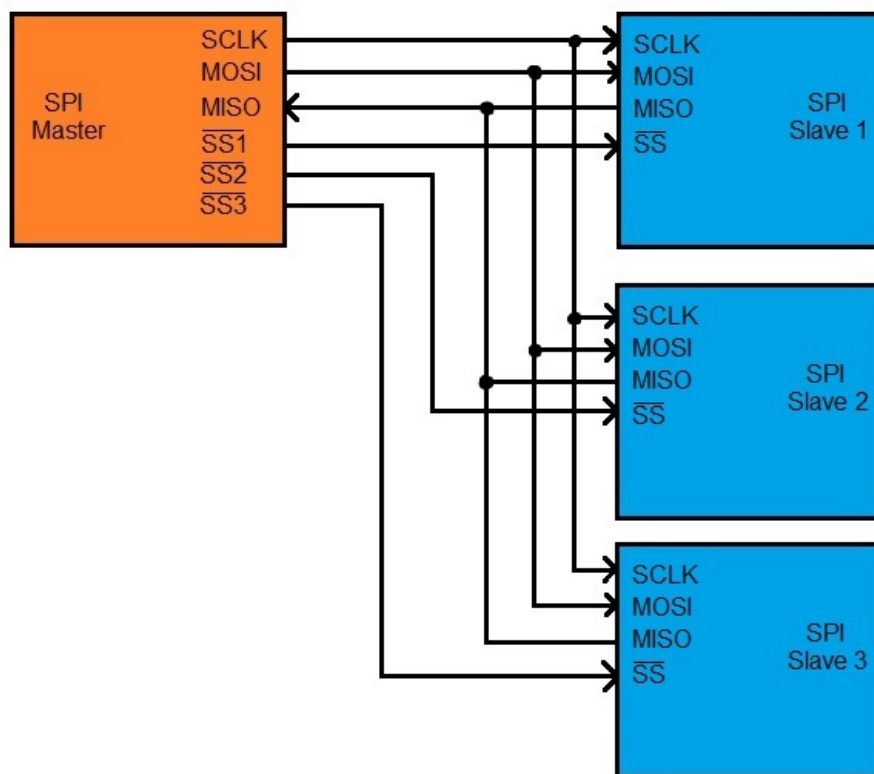
Analogově digitální převodník sloužící k převodu analogového signálu (napětí) na digitální signál (sekvenci nul a jedniček). Může sloužit např. k zjištění hodnoty senzorů napětí a teploty baterie, která napájí dané zařízení.

2.3.2 DMA

Direct Memory Access (Přímý přístup k paměti) je proces, při kterém jsou data přenášena z paměti RAM (Random Access Memory – Paměť s náhodným přístupem), která slouží jako dočasná paměť, přímo k danému rozhraní, které konkrétní data vyžaduje. Výhodou je, že data nemusí procházet přes CPU (Central Processor Unit – centrální výpočetní jednotku, procesor mikrokontroleru) a tím lze jejich přenos urychlit. Data jsou přenášena v 8 vyhrazených programovatelných kanálech [23].

2.3.3 SPI

Serial Peripheral Interface (Sériové periferní rozhraní) je rozhraní, díky kterému spolu může komunikovat mikroprocesor a ostatní periférie, případně další obvody. Je tvořeno čtyřmi důležitými vodiči – SCLK (seriál clock), který vysílá hodinový signál do všech zařízení na sběrnici, SS (slave select, někdy taky CS – chip select), který přenáší informaci o tom, se kterým zařízením na sběrnici bude Master (klient) komunikovat a dva vodiče, které přenáší data – MOSI (Master Output Slave Input) a MISO (Master Input Slave Output). Master má tedy tolik výstupů typu SS, kolik je k němu připojeno zařízení. V okamžiku, kdy chce zahájit s daným zařízením komunikaci, mu nastaví na jeho pinu SS logickou 0 a začne vysílat hodinový signál. Poté může docházet k přenosu dat přes piny MOSI a MISO [24].



Obrázek 2.3 Sběrnice SPI s jedním nadřazeným zařízením typu Master a 3 podřízenými zařízenými typu Slave, převzato z [24]

2.3.4 UART

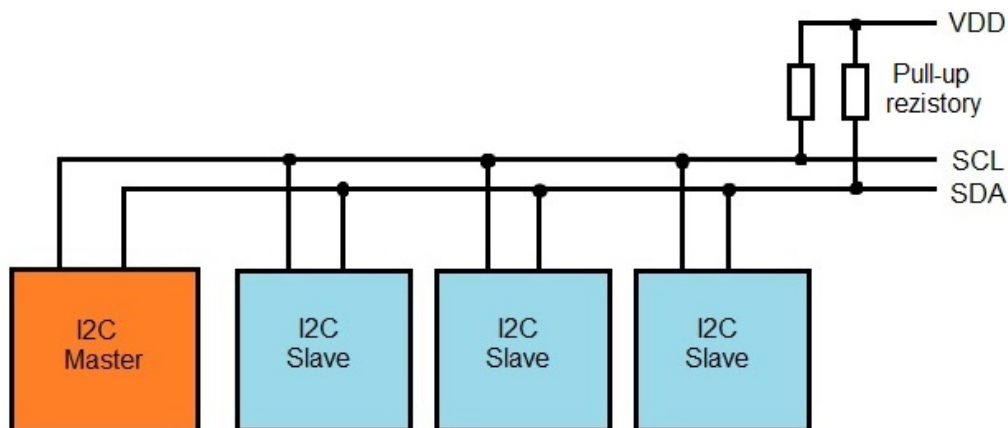
Universal Asynchronous Receiver-Transmitter (univerzální asynchronní vysílač-přijímač) slouží k přenosu dat pomocí asynchronní linky. To znamená, že data nejsou synchronizována pomocí hodinového signálu, ale pro označení jejich začátku a konce je na začátek a konec paketu přidán bit s informací, o kterou část paketu se jedná. Při vzájemné komunikaci je důležité křížové propojení pinů RX a TX, které jsou propojeny střídavě (RX pin prvního zařízení k TX pinu druhého zařízení a opačně). Rozhraní tedy využívá piny VCC (+5V), GND (ground – zem), RX a TX [25].

2.3.5 I²C

Komunikační rozhraní, které je tvořeno čtyřmi piny – +5V, GND, SDA (Serial Data) a SCL (Serial Clock, tedy sériový hodinový signál). Hodinový signál slouží k zahájení, případně ukončení vysílání informací. Platí, že v jednom okamžiku může vysílat data po této sběrnici jen jedno zařízení. Každé zařízení má svoji adresu, pomocí které je poté určeno, s kým komunikace probíhá.

Přenos dat začíná posláním startovacího bitu, kdy jsou na počátku oba piny SCL i SDA v úrovni logická 1 a pin SDA přejde do stavu logické 0. Poté dochází k přenosu dat při každém průchodu nástupné hrany hodinového signálu SCL. Jinými slovy – v okamžik,

kdy přechází pin SCL z log. 0 na log. 1, dojde k přenosu stavu pinu SDA. Přenos dat končí nastavením úrovně log. 1 na obou hlavních pinech (ukončením vysílání hodinového signálu). Kvůli zamezení nežádoucích stavů na pinech SDA a SCL je na nich pomocí pull-up rezistorů udržován kladný potenciál [26].



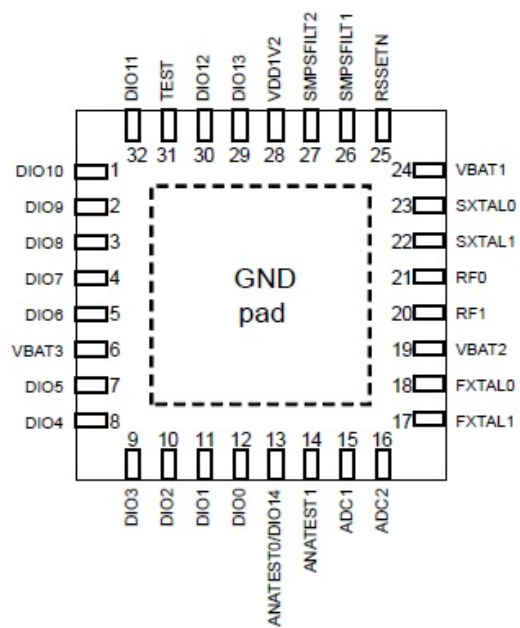
Obrázek 2.4 Sběrnice I²C s jedním nadřazeným zařízením typu Master a 3 podřazenými zařízením typu Slave, převzato z [26]

2.3.6 GPIO

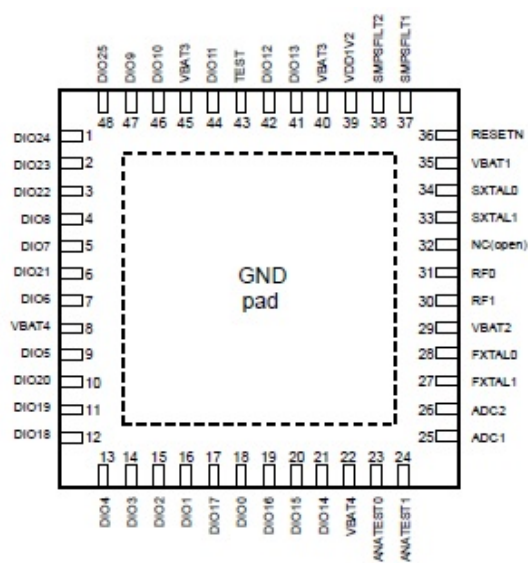
General Purpose Input Output – Vstupně výstupní piny pro všeobecné použití. Mohou sloužit jako vstupní piny (lze k nim připojit tlačítko nebo jiný řídicí obvod, třeba mikroprocesor) a ovládat s nimi chod mikrokontroleru BlueNRG-2, nebo mohou být výstupní a lze s nimi poté dále ovládat další externí periferie (LED diody, relé, zobrazovací displej a jiné).

2.4 Popis pouzdra obvodu BlueNRG-2

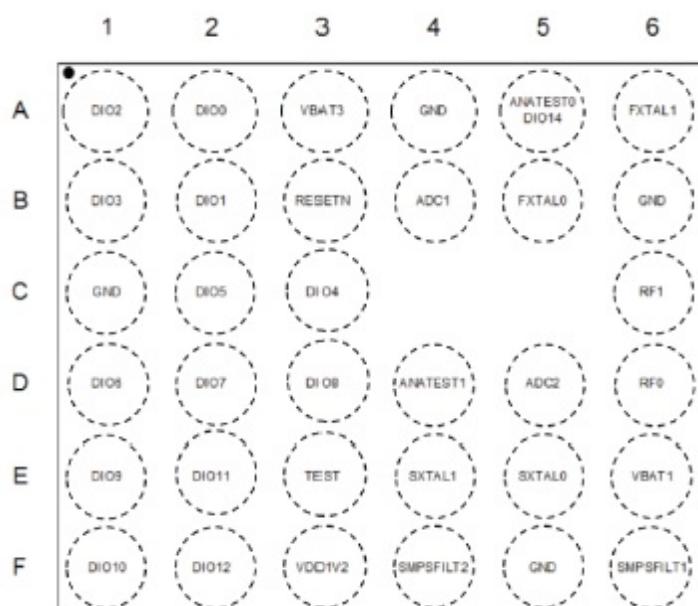
Mikrokontrolér BlueNRG-2 se vyrábí v pouzdech QFN32, QFN48 (má více pinů GPIO oproti provedení v pouzdru QFN32) a WCSP34, které je typu BGA (Ball Grid Array - Plocha pokrytá kruhovými pájecími ploškami uspořádanými do mřížky), k jeho pájení je tedy nutné nejdříve na plošný spoj natisknout tavidlo, do kterého jsou poté pomocí šablony rozmístěny kuličky pájky a na ty se poté obvod opatrně a přesně umístí. Následuje jeho zapájení přetavením. Číslo v označení typu pouzdra vyjadřuje počet vývodů. Na následujících obrázcích jsou jednotlivé varianty zobrazeny v pohledu na pouzdro shora, tedy tak, jak jsou jejich vývody rozmístěny na desce plošných spojů.



Obrázek 2.5 Mikrokontrolér BlueNRG-2 v provedení QFN32, převzato z [19]



Obrázek 2.6 Mikrokontrolér BlueNRG-2 v provedení QFN48, převzato z [19]



Obrázek 2.7 Mikrokontrolér BlueNRG-2 v provedení WCSP34, převzato z [19]

Vysvětlení významu označení jednotlivých pinů je jednoduché, DIO znamená Digital Input Output, tedy jde o programovatelný vstupně výstupní pin. Případné číslo za touto zkratkou označuje jeho konkrétní pořadí výskytu na daném pouzdru. Piny ADC1 a ADC2 jsou analogové vstupy pro analogově-digitální převodník. Piny ANATEST1 a ANATEST2 datasheet označuje jako analogové výstupní piny, ve většině zapojení se však nepoužívají a zůstávají nezapojeny (floating). Následují vstupní piny pro připojení krystalu pro rychlý oscilátor 16/32 MHz (FXTAL1 a FXTAL0) a pomalý 32 kHz oscilátor (SXTAL 1 a SXTAL0). Posledními piny jsou RF1 a RF0, které slouží k připojení antény a přizpůsobovacího obvodu, který uzpůsobuje impedanci vysílacího obvodu tak, aby zajistil maximální přenášený výkon. Na piny VBAT1, VBAT2, VBAT3 a VBAT4 se připojí napájecí napětí, pin GND označuje zem napájení.

3. CHARAKTERISTIKA KOMUNIKAČNÍHO MODULU

Navrhovaný modul je tvořen několika částmi, které zajišťují jednotlivé požadované funkce. Jádrem modulu tvoří mikroprocesor BlueNRG-2 a převodník z USB na UART FT230XQ. Ten poskytuje napájecí pin s napětím 3,3 V, jeho výstupní proud je maximálně 50 mA. Ochranu USB portu proti riziku poškození MCU zajišťuje ochranný obvod TPD2E001DZDR. Přes rozhraní I²C pak hlavní mikrokontroler ovládá OLED displej řízený odvodem SSD1306. K ADC převodníku mikrokontroleru BlueNRG je připojeno čidlo teploty MCP9701T-E/LT.

Moduly lze po jejich inicializaci a připojení modulu inicializovaného jako klient k modulu inicializovanému jako server ovládat povely přes terminál Windows. Lze mezi nimi posílat data ve formě textu, zobrazit je na displeji. Je možné vyčíst data z teplotního senzoru a odeslat je klientovi nebo zobrazit na displeji.

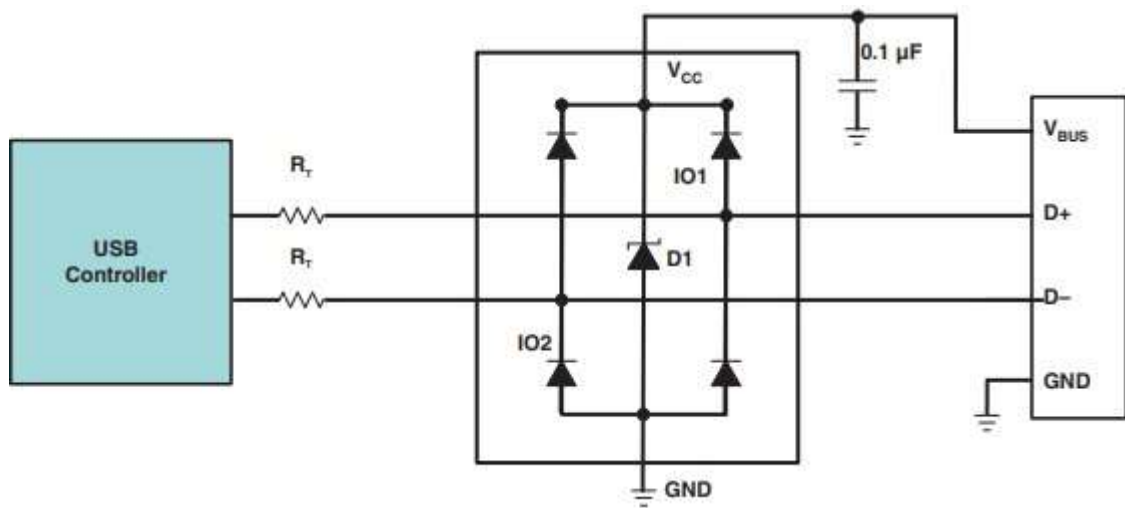
3.1 Převodník z USB na UART

Obvod FT230XQ je převodník z rozhraní USB na rozhraní UART. V modulu má za úkol umožnit ovládání zařízení přes terminál PC pomocí povelů. Terminál je aplikace, která zobrazuje veškerou komunikaci přes daný port, lze tedy sledovat jednotlivé přenášené zprávy a odpovědi na ně. Obvod není nutné nijak programovat, je připraven k použití v základním provedení. Obsahuje také programovatelné výstupy pro LED diody signalizující přenos a příjem dat přes toto rozhraní. Ty jsou na desce osazeny jako TX LED D1 – zelená LED signalizující vysílání dat převodníkem a RX LED D2 – červená, která signalizuje přijetí dat převodníkem.

3.2 Ochrana USB portu proti ESD

Při manipulaci s deskami plošných spojů, které obsahují součástky citlivé na elektrostatický výboj (označujeme je zkratkou ESD – Electrostatic Sensitive Device) je nutné zajistit, aby nedošlo k poškození těchto součástek působením vysokonapěťového elektrostatického impulzu. Statická elektřina vzniká přirozeně všude kolem nás jako důsledek tření různých povrchů, kdy dochází k přenosu volných elektronů z jednoho tělesa na druhé. Tato tělesa poté získávají rozdílné náboje a při dostatečně velkém rozdílu nábojů může po jejich přiblížení nebo doteku dojít k přeskočení výboje statické elektřiny, jehož napětí se pohybuje v řádech kilovoltů. Proto je nutné použít ochranný obvod, který tyto impulzy odfiltruje.

V zapojení je použit obvod TPD2E001, který je principiálně tvořen diodami, které jsou zapojeny antiparalelně mezi jednotlivými vývody USB konektoru (V_{BUS} , D-, D+, GND). Ty případné napěťové pulzy odvádí na pin GND.



Obrázek 3.1 Vnitřní zapojení obvodu TPD2E001 [30]

3.3 OLED displej

Pro ukázkou komunikace přes rozhraní I²C byl vybrán OLED (Organic Light Emitting Diode) displej o úhlopříčce 0,96“ (24,4 mm) s rozlišením 128×32 bodů. Rozhraní lze použít i pro digitální komunikaci s jinými periferiemi, třeba senzory různých fyzikálních veličin, toho je však využito u ADC převodníku. Displej může zobrazovat text, který mu byl odeslán pomocí terminálu PC, případně může zobrazovat aktuální teplotu, kterou měří senzor MCP9701T-E/LT, což je popsáno v následující kapitole.

3.4 Teplotní senzor MCP9701T-E/LT

Senzor MCP9701T-E/LT slouží pro měření teploty. Díky svojí přesnosti ± 4 °C na rozsahu 0 - 70 °C se hodí spíše pro orientační měření teploty např. baterií nebo chladicích ploch komponentů např. v počítačích, serverech a jiných zařízeních. Pro účely výukového kitu slouží jako praktická ukáзка využití ADC převodníku mikrokontroleru BlueNRG. Výhodou senzoru je také jeho nízká vlastní spotřeba (okolo 6 μ A) a proto je jeho vlastní ohřev zanedbatelný, protože je mnohem menší než přesnost senzoru. [31]

4. NÁVRH SCHÉMATU ZAPOJENÍ A PLOŠNÉHO SPOJE KOMUNIKAČNÍ SADY

V této kapitole je představeno navržené schéma modulu spolu s motivem obou stran desky plošných spojů. Kompletní schéma celého zapojení je uvedeno v příloze A. Pasivní součástky, tedy rezistory, kondenzátory a diody, jsou v pouzdru o velikosti 0402, cívky v pouzdru 0603. Uvedené označení představuje rozměry pouzdra v setinách palce, tedy 0402 znamená $0,04 \times 0,02$ palce (anglicky inch), což odpovídá rozměrům $1 \times 0,5$ mm, protože 1 palec odpovídá 25,4 mm (pouzdro 0603 má tedy rozměry $1,6 \times 0,8$ mm). [32]

4.1 BlueNRG-232 – hlavní část zapojení

Hlavní jádro celého zapojení tvoří mikrokontroler BlueNRG-232 v pouzdru QFN32 se součástkami, které jsou nutné k jeho správnému fungování. Jeho zapojení je převzato z datasheetu od výrobce STMicroelectronics [19] a je znázorněno na obrázku 4.1.

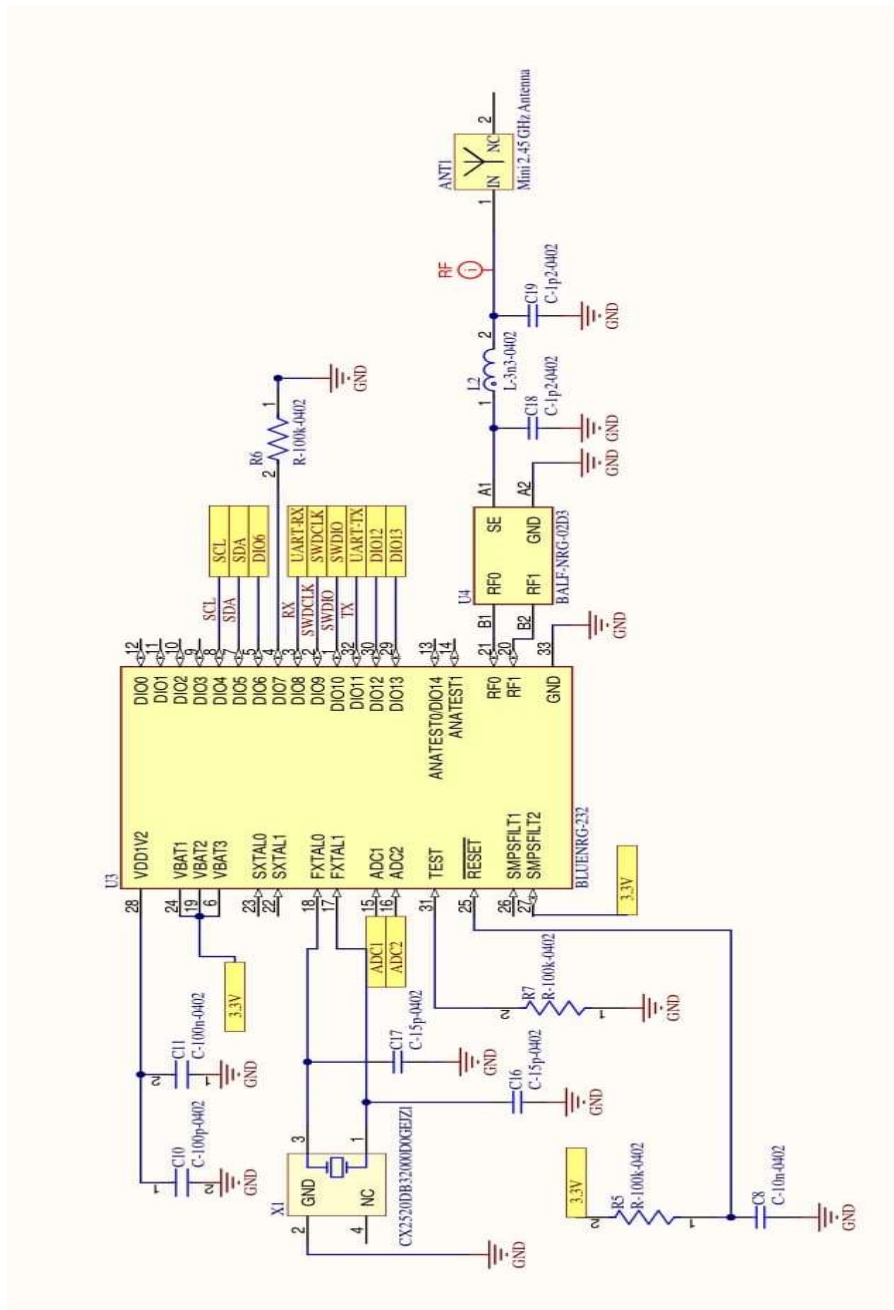
Napětí na napájecích vývodech jsou filtrována kondenzátory, aby nedocházelo k přenosu nežádoucího zvlnění z napájecího zdroje, který napájí modul, do MCU a nedošlo tak k ovlivnění frekvence jeho vnitřních oscilátorů a výstupního signálu. U pinu VDD1V2 jsou to kondenzátory C10 a C11 a u napájecí větve s napětím 3,3V (piny 6, 19 a 24) jsou to kondenzátory C9, C12, C13, C14, C15 a C22, které jsou znázorněny na obrázku 4.6. Všechny jsou umístěny co nejbližší MCU k potlačení parazitních vlastností (impedancí) případných dlouhých přívodních cest.

K pinům 17 a 18 je připojen externí krystal o frekvenci 32 MHz, který je zdrojem hlavního hodinového kmitočtu pro MCU.

Vysokofrekvenční vysílací část je tvořena VF balunem BALF-NRG-D203, který představuje impedanční přizpůsobení pro anténu. Zajišťuje tedy, že impedance výstupu MCU je komplexně sdružená s impedancí antény a nedochází tedy k útlumu signálu na výstupu MCU a celý obvod je v rezonanci. Zároveň balun slouží k potlačení výskytu vyšších harmonických kmitočtů, jejichž přítomnost by snižovala vysílaný výkon modulu. Za balunem se nachází pí-článek tvořený kondenzátory C18, C19 a cívkou L2. Ten slouží ke zvýšení stability kmitočtu vysílaného signálu. Poté již následuje samotná anténa. Její označení je 2450AT18A100 a jde o anténu určenou pro frekvence 2400 – 2500 MHz. Její impedance je 50 Ohmů. Na tuto impedanci byla také pomocí programu Altium Designer vypočítána šířka přívodní cesty a tato hodnota byla použita v návrhu. Přívod k anténě je tedy veden z uzlu spojujícího kondenzátor C19 a cívkou L2 cestičkou odpovídající šířce pouzdra těchto součástek s mírným přesahem (0,7 mm) a poté je ihned rozšířena na šířku vypočtenou programem (1,65 mm). Ta musí samozřejmě být větší než šířka pájecí plošky antény, aby bylo možné anténu v podobě SMD součástky spolehlivě zapájet. V datasheetu antény je uvedena hodnota šířky pájecí plošky 1,65 mm, takže tento

požadavek je taktéž splněn. Tím je dosaženo co nejmenšího útlumu na přívodu VF signálu. Ostatní použité součástky jsou vybrány podle doporučení výrobce obvodu.

Prokovy (vias) umístěné v prostoru u antény slouží ke zmenšení parazitní kapacity měděné plochy (rozlité mědi) zapojené k zemnímu potenciálu, která se nachází kolem všech součástek na horní i spodní straně. Jenom v prostoru kolem antény je tato ploška vynechána, aby nedocházelo k nechtěnému pohlcování vyzařovaného signálu právě touto plochou.



Obrázek 4.1 Zapojení mikroprocesoru BlueNRG-232

Na obrázku 4.2 je ukázka výpočtu šířky cestičky na plošném spoji v závislosti na její požadované impedanci. Program Altium Designer má přímo nástroj pro tento výpočet, který nalezneme v panelu nástrojů pod záložkou Design – Layer Stack Manager. Vstupními parametry jsou počty vodivých vrstev a dielektrik, tomto případě tedy 2 vodivé vrstvy a 1 dielektrikum mezi nimi. Tloušťky a počty jednotlivých vrstev si program nastaví sám, případně je možné hodnoty upravit (parametr Dk je relativní permitivita dielektrika). Po přepnutí na záložku Impedance se zobrazí další tabulka s vypočtenými tloušťkami vodivých cest v závislosti na jejich požadované impedanci. Zadáním tloušťky cesty a potvrzením klávesou ENTER je možné zjistit impedanci dané cestičky nebo po zadání požadované impedance a kliknutí na tlačítko f_x v pravém postranním panelu lze nechat program dopočítat přesnou šířku a jí odpovídající teoretickou impedanci včetně odchylky od požadované hodnoty.

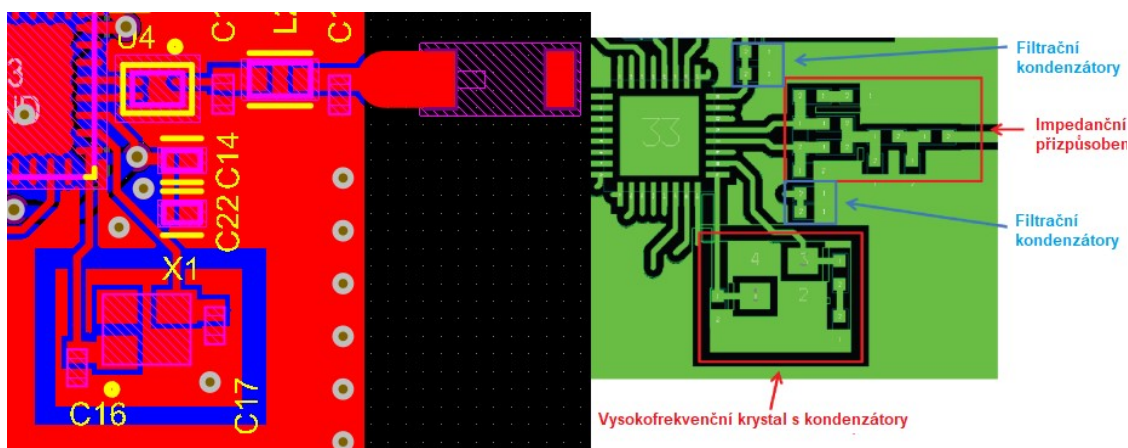
#	Name	Material	Type	Weight	Thickness	Dk	Copper Orientation
	Top Overlay		Overlay				
	Top Solder	Solder Resist	Solder Mask		0.01016mm	3.5	
1	Top Layer		Signal	1oz	0.03556mm		Above
	Dielectric 1	FR-4	Dielectric		0.9398mm	4.8	
2	Bottom Layer		Signal	1oz	0.03556mm		Below
	Bottom Solder	Solder Resist	Solder Mask		0.01016mm	3.5	
	Bottom Overlay		Overlay				

Top Ref	Bottom Ref	Width (W1)	Impedance (Z0)	Deviation	Delay (Tp)
✓	2 - Bottom Lay...	1.63618mm	49.98	0.03%	6.30898ns/m
✓	1 - Top Layer	1.63618mm	49.98	0.03%	6.30898ns/m

Impedance Profile	
Description	RF
Type	Single
Target Impedance	50
Target Tolerance	10%
Use Solder Mask	<input checked="" type="checkbox"/>
Trace Inverted	<input type="checkbox"/>
Etch (?)	0
Width (W1)	1.63619mm
Width (W2)	1.63619mm f_x
Covering (C1)	0.01016mm
Covering (C2)	0.01016mm
Impedance (Zo)	49.98
Deviation	0.03%
Delay (Tp)	6.30897ns/m
Inductance	315.34578nH/m
Capacitance	126.22009pF/m

Obrázek 4.2 Návrh šířky přívodní cesty k anténě podle požadované impedance

V dalším obrázku 4.3 je znázorněno porovnání motivu navržené desky VF části s motivem doporučeným výrobcem, který je uveden v dokumentu AN4819 (Application note) dostupném z [35]. Je patrné, že byl zachován odstup stínění od plochy s krystalem a jeho dvěma kondenzátory a že uspořádání součástek ve výstupní vysílací části MCU kopíruje rozložení součástek podle doporučení v AN4819.



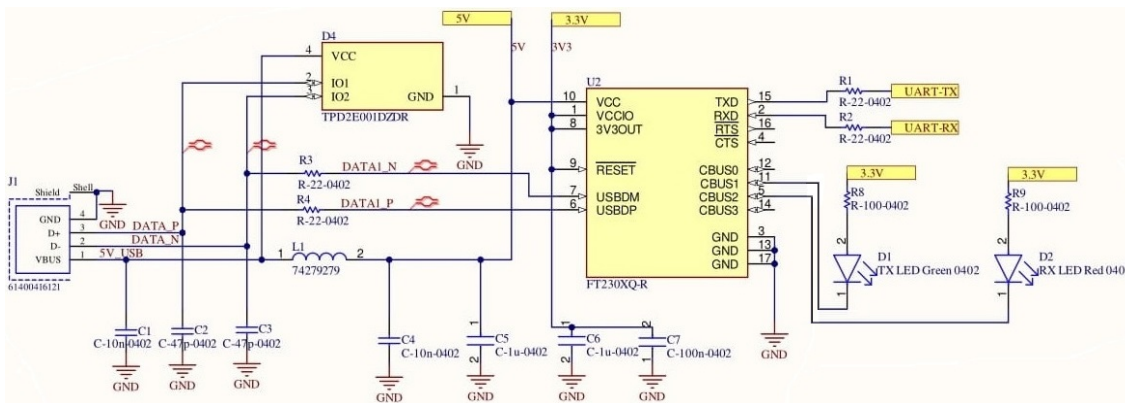
Obrázek 4.3 Porovnání výsledného návrhu VF částí desky (nalevo) s doporučením výrobce (napravo)

4.2 Převodník rozhraní USB - UART

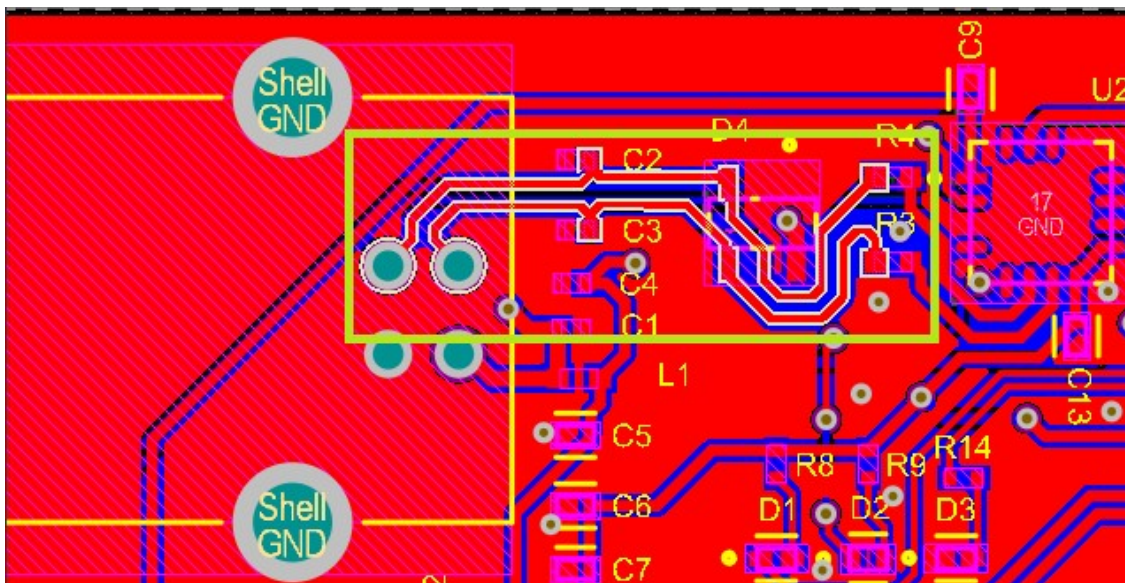
Aby bylo možné s mikrokontrolerem komunikovat a naprogramovat jej pomocí PC, je nutné použít převodník z rozhraní USB na rozhraní UART. Tuto úlohu obstarává integrovaný obvod FT230XQ/R. Jeho zapojení bylo taky inspirováno doporučeným zapojením od výrobce, které lze najít v jeho datasheetu [29]. LED s označením D1_TX (zelená) a D2_RX (červená) slouží k indikaci přenosu (TX – transmitting – vysílání) povelu z převodníku do MCU a k indikaci příjmu povelu z MCU do převodníku (RX – receiving – příjem).

Napájecí napětí z USB filtrují kondenzátory C1 – C7, tlumivka L1 slouží k odfiltrování napěťových špiček při připojení modulu ke zdroji napětí a k odfiltrování nežádoucích vyšších harmonických kmitočtů, které by do modulu mohly pronikat z napájecího zdroje (jedná se tedy o dolní propust). Rezistory R1 – R4 jsou vybrány podle doporučení výrobce [36]. Ochranu proti poškození součástek citlivých na statickou elektřinu (hlavně MCU) zajišťuje obvod TPD2E001, který byl blíže popsán v kapitole 3.2. Zapojení této části modulu popisuje obrázek 4.4.

Datový signál z USB konektoru z pinů DATA+ a DATA- je k převodníku přiveden pomocí vodivých cest, které vzájemně tvoří diferenciální pár. To znamená, že jsou obě cesty vzájemně tvarovány tak, aby byla jejich délka stejná. Tím se na každé z nich indukuje rušivé napětí stejné velikosti, ale opačného znaménka (kladné na cestičce přenášející signál kladné polarity, záporné na spoji se záporným signálem). Tím dojde k potlačení jejich vzájemných účinků a zařízení je tak odolnější vůči okolnímu elektromagnetickému rušení. Provedení těchto signálových cest je znázorněno na obrázku 4.5, kde jsou označeny obdélníkem.



Obrázek 4.4 Zapojení rozhraní USB a převodníku FT230XQ



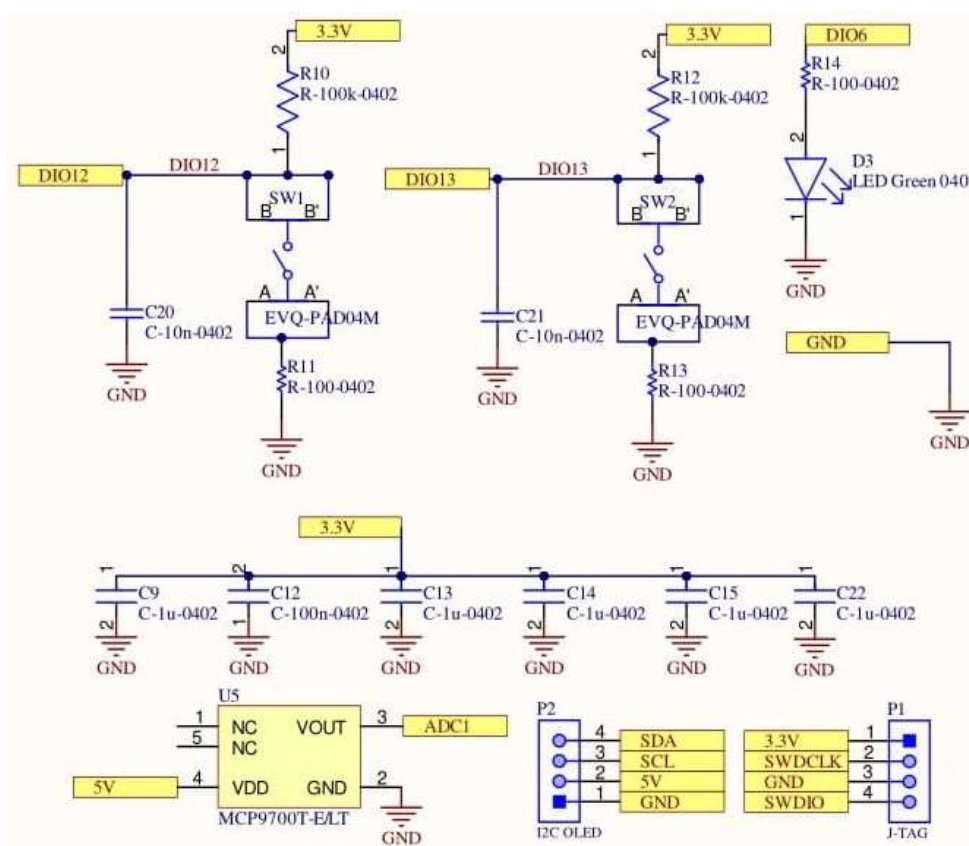
Obrázek 4.5 Diferenciální páry u USB konektoru

4.3 Periferie a další prvky modulu

Abychom mohli využívat i ostatní periferie popisovaného mikrokontroleru, je nutné je také zahrnout do schématu a osadit na desku. Ze všech možností, čím obvod BlueNRG-232 disponuje, bylo vybráno několik základních rozhraní. Mezi ně lze zařadit rozhraní GPIO, které se stará o určování funkcí a úloh jednotlivých pinů MCU. Ty totiž mohou sloužit k mnoha účelům. V datasheetu k BlueNRG-232 [19] se můžeme na straně 91 dočíst, že existují 4 varianty, jak mohou být piny nakonfigurovány. V každé takové variantě má každý pin jinou funkci a kromě toho, že může sloužit jako GPIO (General Purpose Input Output – Vstupně-výstupní pin pro všeobecné použití), tak může sloužit také jako vstupní nebo výstupní pin pro rozhraní některé z periferií MCU (UART, I2C, SPI, výstup hodinového signálu z oscilátoru atd.).

V rámci této kapitoly bude popsáno pouze použití pinu jako „univerzálně vstupně-výstupního“. Popis funkce zapojení bude vztažen k pinu DIO12, pro pin DIO13 platí stejný princip. V takovém případě, jak je znázorněno na obrázku 4.6, je pin inicializován jako vstupní a je k němu připojeno tlačítko SW1. Za normálního stavu, jak je zakresleno ve schématu, je na pinu udržována hodnota logická 1 (přibližně 3,3 V) pomocí pull-up rezistoru R10. V okamžiku stisku tlačítka dojde k připojení pinu DIO12 k zemnímu potenciálu přes odpor R11. Dojde tedy k překlopení napěťové úrovně na logickou 0. Tím dochází ke změně stavu, na kterou MCU zareaguje. Kondenzátor C20 slouží k omezení zákmitů při stisku tlačítka. Při něm totiž vznikají nežádoucí zákmity, které by mohl MCU vyhodnotit jako několikanásobný stisk tlačítka a provést naprogramovanou operaci několikrát po sobě.

Jako reakce na stisk tlačítka je poté rozsvícena dioda D3, ta je uživatelsky programovatelná a její chování ovlivňuje kód nahraný v MCU. Chování tlačítek a diody lze samozřejmě libovolně upravovat.



Obrázek 4.6 Zapojení tlačítek a ostatních periférií

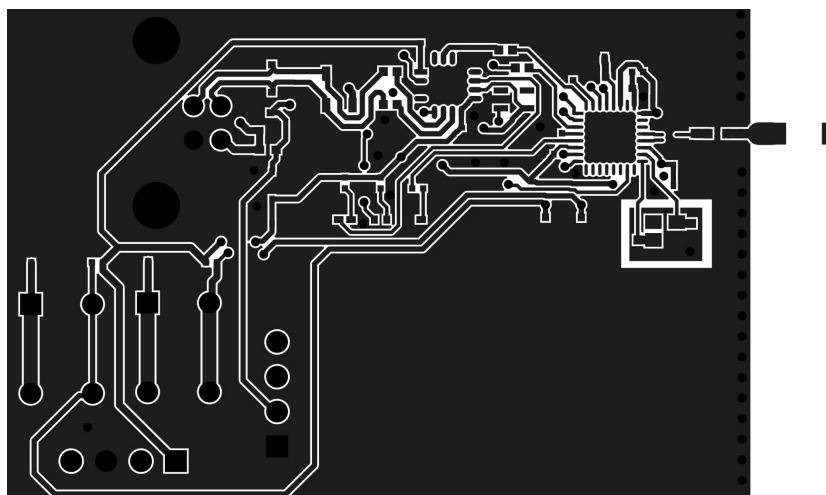
V neposlední řadě zde najdeme také převodník teploty na napětí, senzor MCP9701T-E/LT od společnosti Microchip, který je na pouzdru označen textem AVIS. Jeho přesnost je spíše orientační ($\pm 4\text{ }^{\circ}\text{C}$ na rozsahu $0 - 70\text{ }^{\circ}\text{C}$), slouží tedy k měření relativní změny

teploty v aplikacích, kde není přesnost naměřené hodnoty kritickým parametrem. Jeho offset, tedy napětí při teplotě 0 °C je 400 mV a rozlišení (teplotní koeficient) 19,5 mV/°C. Z toho vyplývá, že pro převod naměřeného napětí na teplotu platí:

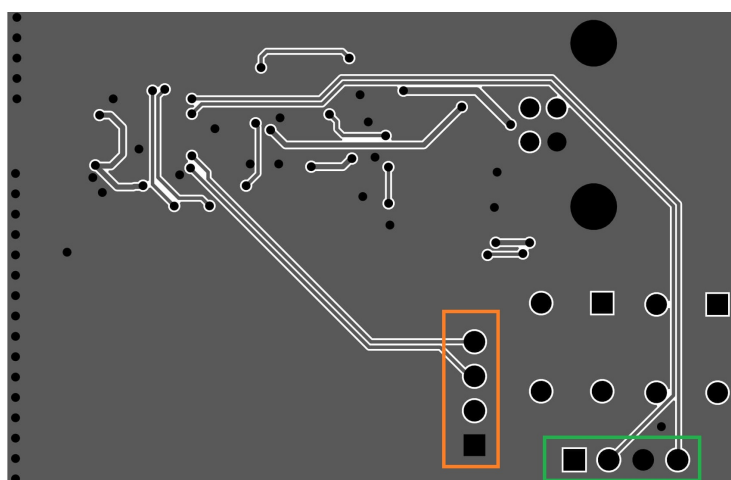
$$T = \frac{U_{OUT} - U_{0\text{ }^{\circ}\text{C}}}{T_C} \quad (4.1)$$

T je okolní teplota ve °C, U_{OUT} je výstupní napětí senzoru [mV], $U_{0\text{ }^{\circ}\text{C}}$ je napěťový offset [mV] a T_C je teplotní koeficient [mV/°C]. Pomocí této rovnice poté v programu dochází k převodu napětí na okolní teplotu.

Následující obrázky zobrazují celkový motiv obou stran desky, včetně konektoru pro programátor (na obrázku 4.8 vpravo dole vodorovná řada pinů, označen zeleně) a displej, který se nachází na stejném místě, je orientován svisle a označen oranžově.



Obrázek 4.7 Vrchní strana navržené desky plošných spojů



Obrázek 4.8 Spodní strana navržené desky plošných spojů (zrcadlově převrácená)

5. PROGRAMOVÁNÍ MIKROKONTROLERU

K tomu, aby modul fungoval podle zadaných požadavků, je nutné jej správně naprogramovat. Pro tento účel byl zvolen programátor Segger ST-Link od výrobce MCU. Vlastní úpravy a kompilování programu, tedy překlad zdrojového kódu v jazyce C do strojového kódu pro MCU, probíhá v programu Atollic TrueStudio, který je volně dostupný na internetu [33]. Na stránkách výrobce se také nacházejí ukázkové kódy pro vývojové desky s mikrokontroléry řady BlueNRG navržené výrobcem. Tyto zdrojové kódy jsou volně dostupné ke stažení z [34] a některé z nich sloužily jako inspirace k vytvoření výsledné podoby programu pro moduly popisované v této práci.

Po stažení instalátoru a nainstalování aplikace STSW-BLUENRG1-DK se v instalační složce Project nachází složka BLE_Examples s ukázkami programů pro vývojové desky od STMicroelectronics a složka BlueNRG1_Periph_Examples, kde jsou ukázky kódů k inicializaci a používání jednotlivých periférií obvodů řady BlueNRG.

5.1 Struktura programu

Ke správné funkci modulu je zapotřebí do něj nahrát program, který představuje soubor instrukcí, které jej ovládají. V tomto případě je napsán v jazyce C a skládá se z několika souborů.

Hlavním je soubor main.c, který se skládá z úvodní části, ve které dochází ke vkládání (inkluzi) hlavičkových souborů knihoven, které odkazují na definice funkcí, které se dále v programu používají.

Poté následuje funkce int main (void), ve které jsou napsány instrukce, které se provedou jednorázově při připojení modulu k napájecímu napětí. Jsou to zpravidla definice konstant, inicializační funkce a nastavení parametrů pro následné použití v dalších funkcích apod.

Po této části následuje nekonečná smyčka programu while(1), která obsahuje instrukce, které jsou vykonávány neustále při provozu zařízení. Je důležité, aby nedošlo k přerušení této nekonečné smyčky. Modul by přestal reagovat na příkazy a „zaseknul by se“.

V této části tedy najdeme funkci, která kontroluje, zda došlo k úspěšné inicializaci rozhraní Bluetooth a pokud ano, dojde ke kontrole, zda přišla na toto rozhraní nějaká data. Ta přichází v podobě paketu, což je blok dat o definované velikosti a kromě části nazývané hlavička, která obsahuje informaci o typu příchozí informace a části s informací o počtech bytů příchozí informace, zde najdeme hlavní část, kterou nazýváme payload. Ta představuje vlastní přenášená data (v doslovném překladu „náklad“). Pokud tedy bude zaznamenána hodnota, na kterou má podle instrukcí MCU zareagovat, dojde ke skoku do odpovídající části programu a vykonají se zde nastavené instrukce. Zpravidla jde o volání

dalších funkcí, které jsou definovány v navazujících souborech. Stejným způsobem se kontroluje a vyhodnocuje přijetí dat přes rozhraní UART.

5.2 Zpoždění

Instrukce jsou vykonávány rychlostí, která závisí na kmitočtu použitého oscilátoru. V popisovaném zapojení se využívá externí oscilátor tvořený krystalem o frekvenci 32 MHz, vykonání jedné instrukce tedy trvá dobu, která je převrácenou hodnotou této frekvence, tedy 31,25 ns. K tomu, abychom mohli provádět všechny instrukce v časovém sledu, který je výrobcem požadován, je nutné použít zpoždění. MCU je tvořen setrvačnými obvody, které potřebují ke své funkci určité zpoždění, které je pro každý konkrétní prvek obvodu popsáno v datasheetu výrobce.

Mezi instrukcemi je tedy nutné nastavit zpoždění, během kterého program vyčkává a teprve poté začne vykonávat další instrukci. To však za předpokladu, že pořad běží nekonečná smyčka programu. K tomuto účelu jsou v souboru `mcu.c` definovány funkce, které umožňují počítat zpoždění v řádech 0,1 ms, 1 ms, 10 ms, 100 ms a 1000 ms, které lze dále využít. Systémová funkce `SysTick_Handler()`; počítá čas od spuštění programu a každých 100 μ s nastaví proměnnou (vlajku) `sysTickFlags.flag100us` (která je typu `int` – celé číslo) na hodnotu 1. Obdobně pak také další funkce počítající ostatní časové úseky.

Pokud tedy chceme, aby se nějaká instrukce provedla se zpožděním, je nutné ji umístit do podmínky `if`, ve které kontrolujeme, jestli je platná (má hodnotu 1) vlajka (flag) daného zpoždění. Když ano, je dané vlajce přiřazena hodnota 0 a dojde k provedení požadované instrukce (nebo skupiny instrukcí). Tak je zajištěno, že daným zpožděním nedojde k přerušení nekonečné smyčky programu.

Jako ukázkou využití zpoždění je možné uvést příklad nekonečného blikání LED.

```
// nekonecne blikani ledkou

if(sysTickFlags.flag1000ms)
{
    sysTickFlags.flag1000ms = 0;
    GPIO_ToggleBits(GPIO_Pin_6);
}
```

Program tedy v nekonečné smyčce kontroluje, jestli je hodnota proměnné `sysTickFlags.flag1000ms` 1 (uběhlo tedy 1000 ms = 1 sekunda) a poté tuto proměnnou vynuluje a překlápí stav pinu 6 z logické úrovně 1 na úroveň 0 a obráceně. Úplně stejně lze do této podmínky zadat jakýkoliv jiný příkaz, který chceme vykonávat jednou za sekundu. Vhodným příkladem je třeba vypisování aktuální teploty ze senzoru do terminálu, což bude popsáno dále.

5.3 Ovládání LED pomocí tlačítek

Po úspěšném ověření funkce programu a celého zapojení je na řadě ověření funkce tlačítek. K tomu je zapotřebí definovat a zavolat funkci `initGPIO()`, která se nachází

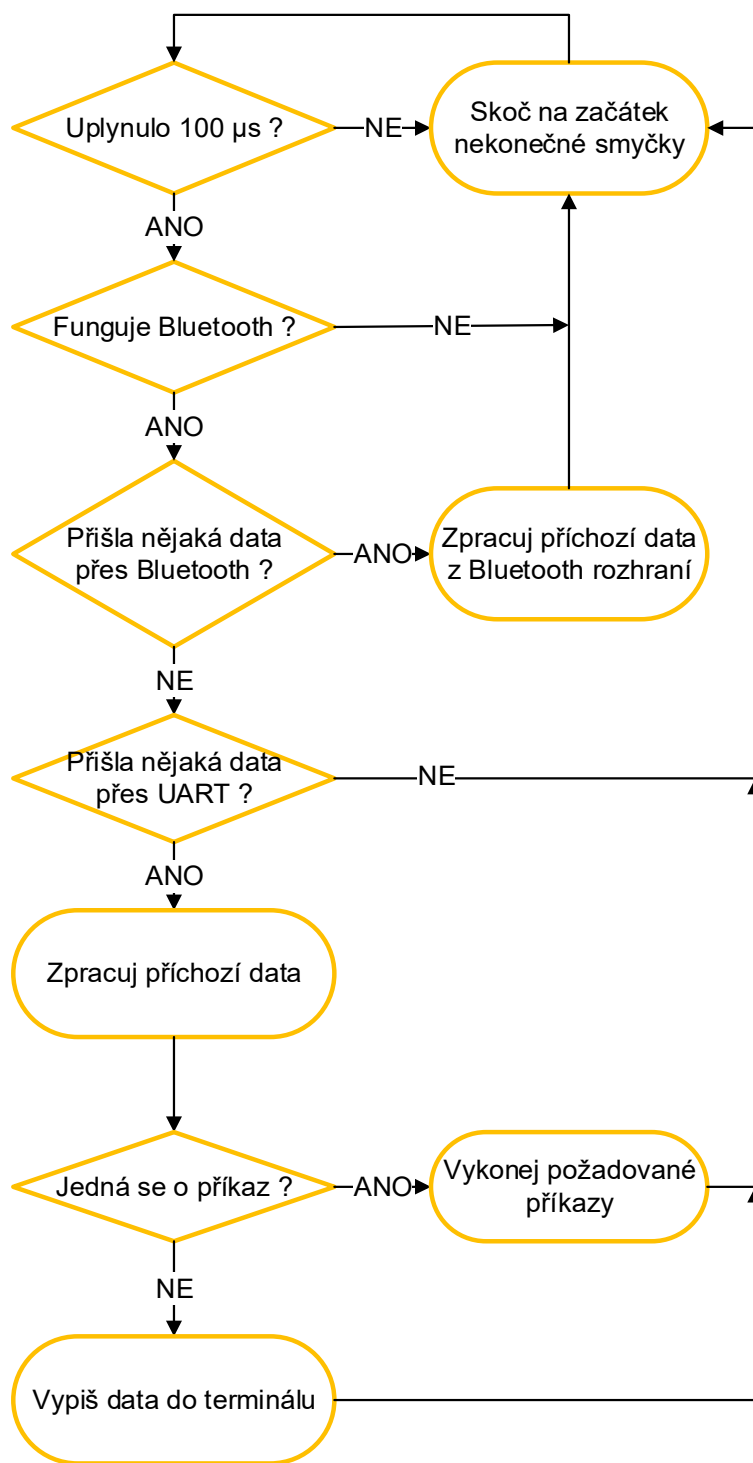
v souboru `drvBleServer.c`. Obsahuje instrukce, kterými nastavujeme, že pin GPIO 6, kde je připojena LED D3, bude výstupní a povolení oscilátoru pro rozhraní GPIO. Důležitá je také funkce `tlacitka()`, která nastavuje, že piny GPIO 12 a GPIO 13 budou nastaveny jako vstupní, jestli bude MCU reagovat na stisknutí nebo uvolnění tlačítka (tedy v časovém průběhu napětí na pinu ovládaném tlačítkem na hranu sestupnou nebo náběžnou). Obě tyto funkce jsou volány v hlavní funkci `int main(void)`.

V nekonečné smyčce programu je poté uvedena funkce, která zjišťuje stav logické úrovně pinu GPIO 12 (ovládaný tlačítkem SW1) a pokud je hodnota 0 (tlačítko je stisknuto), dojde k rozsvícení LED D3. Po uvolnění tlačítka je však opět zhasnuta. Pro druhé tlačítko funguje funkce velmi podobně, jen s tím rozdílem, že stisk tlačítka způsobí pouze překlopení stavu pinu, tedy po uvolnění tlačítka LED zůstává ve svém stavu a až další stisk tlačítka SW2 tento stav změní.

5.4 Příjem povelů a dat pomocí rozhraní UART

Při prvotním odlaďování funkcí programu je nezbytné, aby modul dokázal reagovat na povely, které mu jsou zaslány prostřednictvím rozhraní UART. To funguje na principu posílání dat po jednotlivých bitech v pořadí od nejméně důležitého (posledního v řadě) bitu (LSB – Least Significant Bit) do nejvíce důležitého (prvního v řadě) bitu (MSB – Most Significant Bit). Přijaté bity z pinu RX jsou tedy pomocí posunového registru (shift register), který je řízen hodinovým signálem z interního oscilátoru převodníku, přemístěny do registru přijatých dat (receive register – v programu pro tento účel slouží proměnná typu celočíselné pole `uint8_t rxUart.data[2100]`). Odesílání dat probíhá obdobně – z registru odesílaných dat (transmit register) jsou pomocí posunového registru řízeným kmitočtem data odeslána na pin TX.

Opakování kontroly příchozích dat musí probíhat s určitým zpožděním. Program tedy zjišťuje, jestli už uběhlo 100 μ s a pokud ano, zjistí, jestli je rozhraní Bluetooth úspěšně inicializováno (tedy jestli má proměnná `uint8_t bleWorking` hodnotu 1, kterou získá na konci úspěšného inicializačního cyklu, kdy se nastavují parametry rozhraní Bluetooth) a pokud i toto platí, nastane volání funkce `checkBle()`, která zjistí, jestli nedošla nějaká data přes Bluetooth. Pokud ne, zavolá se funkce `checkUart()`, která kontroluje, zda nepřišla nějaká data přes UART. Pokud ano, v závislosti na hlavičce těchto dat, tedy hodnotě proměnné `uint8_t rxUart.hdr` se skočí do daného případu ve smyčce typu switch – case a provedou se zadané instrukce. Celý postup ukazuje vývojový diagram na obrázku 5.1:

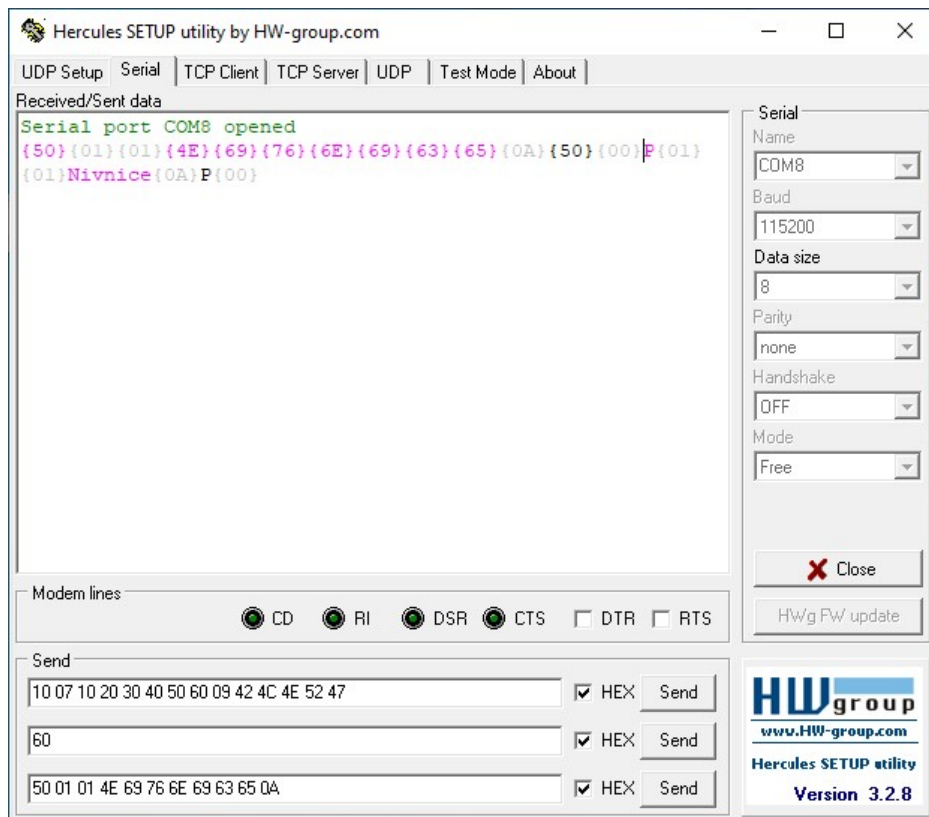


Obrázek 5.1 Vývojový diagram průběhu zpracování a vyhodnocování dat z protokolu UART

5.4.1 Odeslání příkazu a dat z rozhraní UART

Na následujícím obrázku 5.2 je znázorněna ukázka přijímaných a vysílaných dat

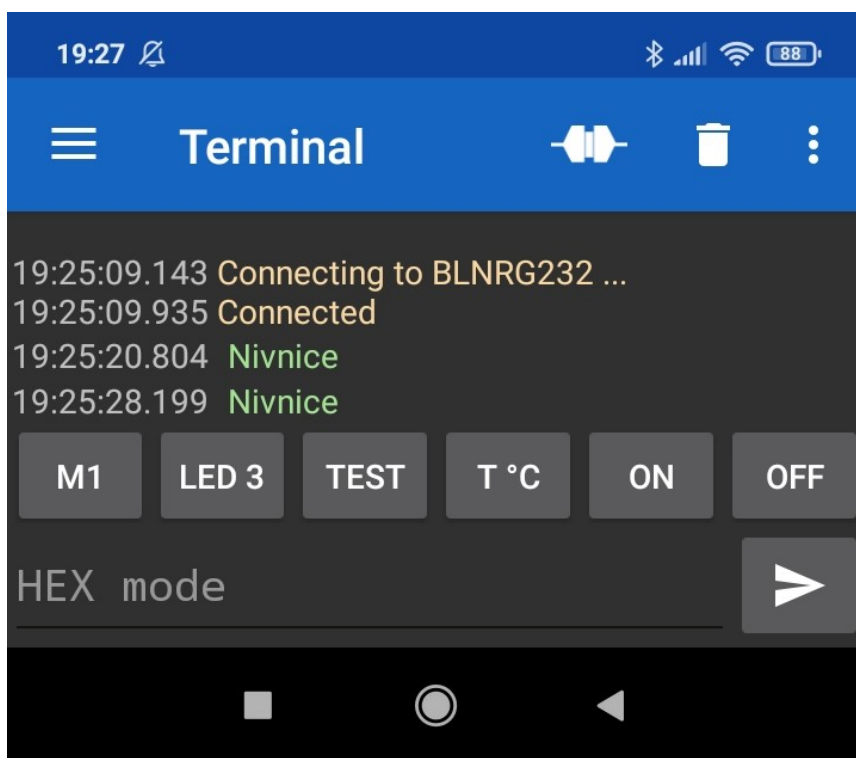
z rozhraní UART v podobě textového výpisu v okně programu Hercules. Ten je volně ke stažení na internetu a mimo jiné slouží k monitorování a zaznamenávání komunikace na sériovém portu PC.



Obrázek 5.2 Ukázka odesílání dat z UART do rozhraní UART i Bluetooth zároveň, přijatá data z programu Hercules, který emuluje terminál

Na obrázku vidíme, že po odeslání dat v posledním řádku u spodního okraje okna terminálu se nejdříve vypíše posílaná data (v tomto případě prvních 11 bitů) a potom nám modul odpoví odesláním bitu typu header o hodnotě 50 (definovaný jako konstanta UCOMM_Hello_Key, který naznačuje, že přišla data z funkce sendPacketData();, poté následuje bit 00 (příkaz byl vykonán úspěšně). Všechny hodnoty bitů jsou hexadecimální a zároveň hodnota každého bitu odpovídá danému znaku ASCII tabulky, který je v ní obsažen pod stejnou hodnotou. Za kurzorem vidíme odpověď na stejný příkaz ale po zapnutí zobrazování v hexadecimálním módu. Bity jsou tedy vypsány jako znaky v ASCII tabulce.

Zároveň dojde k vypsání odesílaných dat do Bluetooth terminálu.



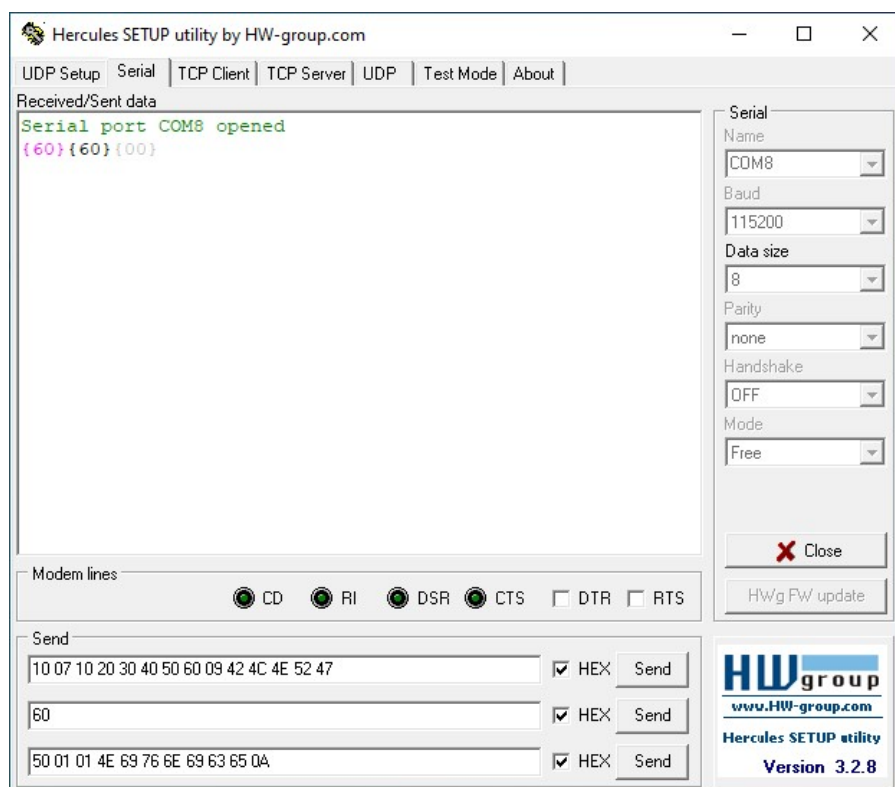
Obrázek 5.3 Ukázka odesílání dat z UART do rozhraní UART i Bluetooth zároveň, přijatá data v aplikaci Bluetooth serial terminal na mobilním telefonu s operačním systémem Android (modul je inicializován s názvem BLNRG232)

Funkce `sendPacketData()`; tedy posílá data jak přes UART, tak přes Bluetooth. Je možné také posílat data pouze do Bluetooth terminálu. To zajišťuje funkce `setDataToSendBle()`; Ta může přímo vypisovat řetězec znaků uložený v programu (což bude předvedeno na následujícím příkladu) nebo vypisovat znaky, které jí předá nějaká funkce (bude popsáno v příští kapitole o posílání teploty přes Bluetooth).

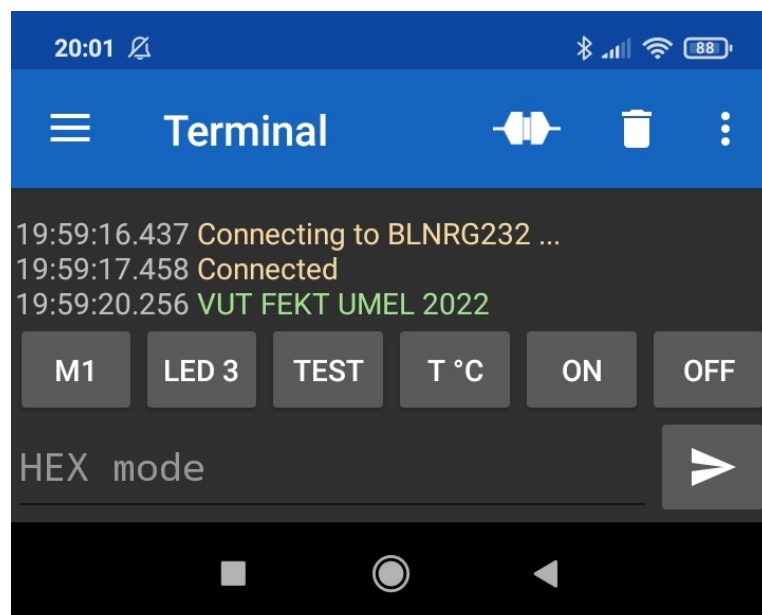
Volání funkce vypadá následovně:

```
setDataToSendBle((uint8_t*)"VUT FEKT UMEL 2022\n", 19, UCOMM_SENDBLE_TEST);
```

Po jejím zavolání získáme tento výsledek:



Obrázek 5.4 Ukázka odesílání dat do Bluetooth terminálu, výstup z programu Hercules

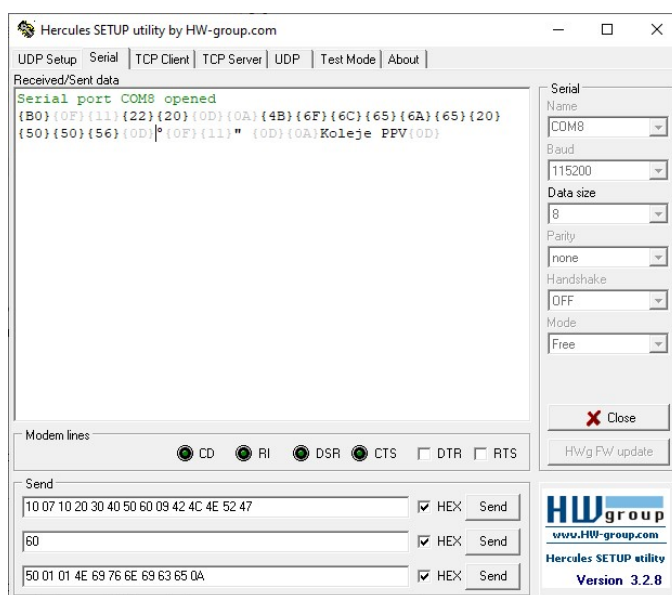


Obrázek 5.5 Ukázka odesílání dat do Bluetooth terminálu, výstup na terminálu mobilního telefonu

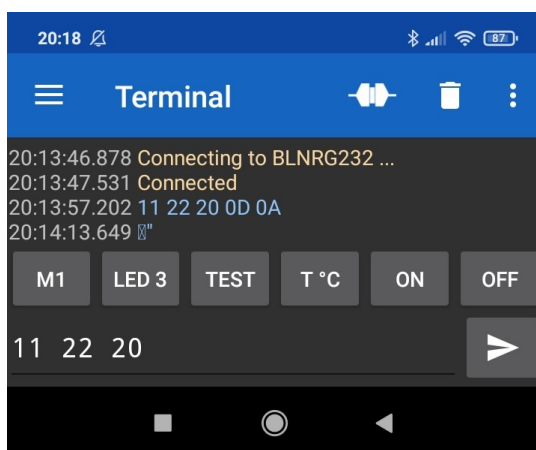
Vidíme tedy, že došlo k odeslání informací pouze do terminálu Bluetooth, protože na rozhraní UART bylo zaznamenáno pouze přijetí bitu 60, který značí volání funkce setDataToSendBle(); a na to odeslání odpovědi 60 00, tedy že příkaz pod hlavičkou 60 byl vykonán úspěšně.

5.4.2 Odesílání příkazů a dat z rozhraní Bluetooth

Následující snímky popisují reakci UART terminálu na příjem dat z Bluetooth. V textovém výstupu je za kurzorem uvedena odpověď na stejný příkaz s vypnutým zobrazením v hexadecimálním módu.



Obrázek 5.6 Ukázka odesílání dat do UART terminálu, výstup z programu Hercules

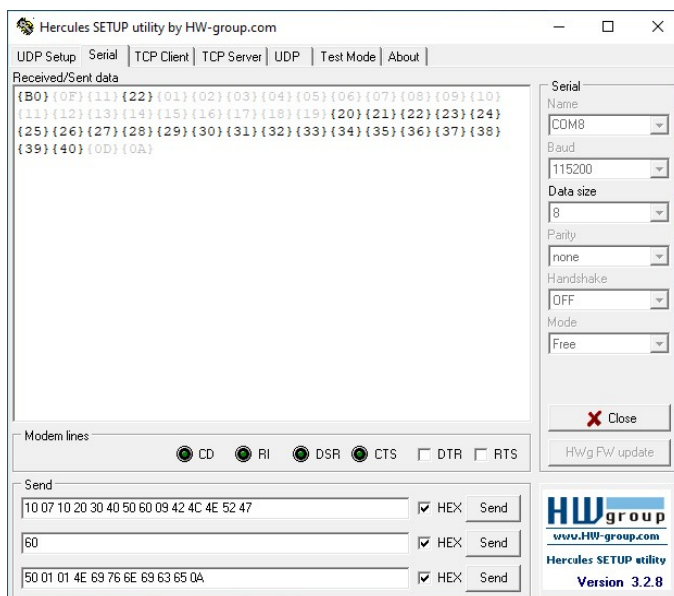


Obrázek 5.7 Ukázka odesílání dat do UART terminálu, výstup z programu Bluetooth serial terminal na mobilním telefonu

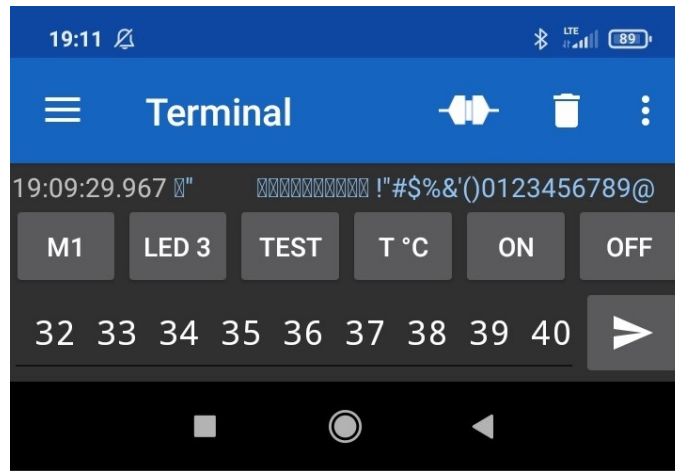
Na obrázcích výše je možné vidět, že při odesílání dat z Bluetooth terminálu se na mobilní telefonu zobrazí pouze odeslaná data, tedy 11 22 20 a odpověď 0D 0A znamenající, že byl příkaz úspěšně dokončen. Vlastní výpis dat však proběhne pouze v terminálu UART. Tam můžeme na obrázku 5.6 vidět, že nejdříve obdržíme hlavičku B0, která znamená přijetí dat z rozhraní Bluetooth, poté hodnotu 0F typu auxiliary a pak následují samotná odeslaná data, tedy 11 22 20 0D 0A. Na ta přijde odpověď v podobě znaků, které jsou naprogramovány ve funkci, která je volána pod hlavičkou 20, tedy sendHelloKey();.

Na následujících obrázcích 5.8 a 5.9 vidíme přijetí dlouhého množství dat, která nepředstavují žádný příkaz, který by měl být specificky vyhodnocen. Je zřejmé, že pokud posíláme příkaz z jednoho rozhraní, informace z něj získáme na rozhraní druhém a naopak. Při posílání dat z jednoho rozhraní do druhého je vidíme na obou těchto rozhraních.

Pro úspěšné zachycení dat přijímaných z rozhraní Bluetooth je nutné, aby paket začínal sekvencí 11 22. Tím dojde k nastavení vlajky uint8_t rxBle.flag do hodnoty 1, čímž dojde k zaznamenání příchodu dat z Bluetooth a případné další reakci na data. Z obrázků vidíme, že oproti datům z rozhraní UART, kdy je hlavička pokaždé jiná a lze toho využít při tvorbě programu, kdy vyhodnocujeme její hodnotu a podle ní definujeme další běh programu, u dat z Bluetooth se tohoto principu využít nedá. Příchozí data mají na tomto rozhraní totiž hlavičku pořád stejnou (B0). Následující část paketu, auxiliary (0F) je také pořád stejná, tak jakožto i data na pozici 0 (11) a 1 (22). Mění se tedy až data na pozici 2. Právě toho je využito při zaznamenávání příchodu dat z terminálu telefonu a jejich vyhodnocování.



Obrázek 5.8 Ukázka odesílání dat z mobilu do UART i Bluetooth terminálu, výstup v programu Hercules

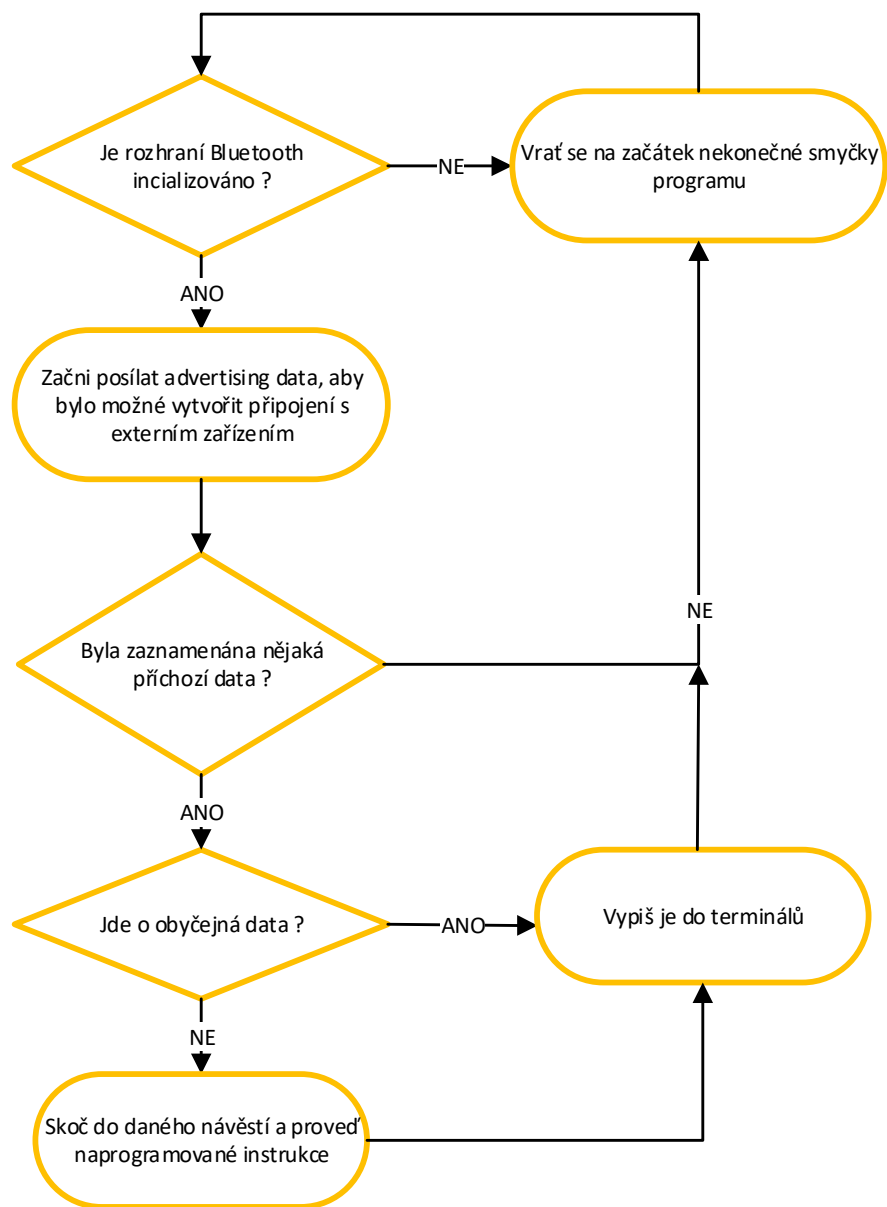


Obrázek 5.9 Ukázka odesílání dat z mobilu do UART i Bluetooth terminálu, výstup na terminálu mobilního telefonu

5.5 Příjem povelů a dat pomocí rozhraní Bluetooth

Příjem a vyhodnocování dat odeslaných přes Bluetooth probíhá velmi podobně jako u dat z rozhraní UART. Popis celého procesu zachycuje obrázek 5.10.

Jak již bylo zmíněno, hlavním rozdílem při vyhodnocování, jaký povel vlastně přišel přes dané rozhraní, je podmínka, kterou vyhodnocuje switch. U rozhraní UART je to hodnota proměnné `uint8_t rxUart.hdr` a u Bluetooth je to hodnota proměnné `uint8_t rxBle.data[2]`. Obě proměnné `rxUart` a `rxBle` představují struktury definované v souboru `mcu.h`, které slouží k záznamu příchozích dat a jejich dalšímu zpracování. O jejich naplnění příchozími informacemi se starají systémové funkce definované ve vlastním ovladači MCU. Obdobně fungují i proměnné `txBuffer` a `auxBuffer`, obě typu `uint8_t`, které slouží naopak k záznamu odesílaných dat a jejich dalšímu předávání funkcím, které je dále zpracovávají.



Obrázek 5.10 Vývojový diagram popisující přijetí dat z protokolu Bluetooth

5.6 Získání dat z ADC převodníku

K tomu, aby bylo možné pracovat s rozhraním ADC je nutné jej správně inicializovat. To zajišťuje funkce `ADC_configuration()`, která je volána hned při spuštění programu ve funkci `int main(void)`. Ta obsahuje instrukce, které zajistí povolení oscilátoru pro rozhraní ADC a upřesňující parametry inicializace, kam patří především, odkud bude toto rozhraní získávat data. MCU totiž obsahuje dva vstupy pro ADC převodník a to ADC1 na pinu 15 a ADC2 na pinu 16. Pin ADC1 je připojen k napětíovému výstupu senzoru MCP9701 a pin ADC2 není zapojen. Nabízí se tedy varianta inicializovat převodník v režimu, kdy vstupní informací bude napětí na pinu ADC1, tedy nastavit ve funkci `ADC_Configuration()`; v souboru `drvBleServer.c` parametr `xADC_InitType.ADC_Input = ADC_Input_AdcPin1`. Takto naměřená teplota však neodpovídá skutečné hodnotě okolní teploty s poměrně velkou odchylkou až $+4,6\text{ }^{\circ}\text{C}$.

Vlastní měření teploty probíhá při volání funkce `zjisti_teplotu()`. Nejdříve proběhne kontrola, jestli proběhlo uložení hodnoty napětí z daného senzoru do odpovídajícího registru. Poté je zavolána funkce `ADC_GetConvertedData()`, která podle zadaných parametrů zjistí hodnotu teploty. Rovnou je zavolána funkce `ADC_Cmd()`; pro zahájení nového zjištění napětí. Následně je teplota převedena pomocí tohoto příkazu:

```
float teplota = ((adc_value-0.448)/0.0195);
```

Převodní vztah je popsán rovnicí (4.1) která je uvedena v kapitole 4.3. Následuje funkce, která z hodnoty teploty vytvoří destinné číslo a všechny jeho číslice postupně nakopíruje na pozice v poli `teplota_cislice`.

```
teplota/=100;
//zkraceny zapis vyrazu teplota=teplota/100;;
uint8_t teplota_cislice[10];
teplota_cislice[0]=((int) (teplota*10)%10)+48;
teplota_cislice[1]=((int) (teplota*100)%10)+48;
teplota_cislice[2]='.';
teplota_cislice[3]=(((int) (teplota*1000)%100)%10)+48;

teplota_cislice[4]=((((int) (teplota*10000)%1000)%100)%10)+48;

teplota_cislice[5]=((((int) (teplota*100000)%10000)%1000)%100)%10)+48;
teplota_cislice[6]=' ';
teplota_cislice[7]=39;
//vypise apostrof misto znacky stupne, kterou BLNRG neumi
teplota_cislice[8]='C';
teplota_cislice[9]='\n';
//Carriage return - preskoci na novy radek (odradkovani)
```

V prvním kroku je proměnná teplota vynásobena 10 a její hodnota převedena na celočíselný typ `int` (integer). Poté je provedena operace modulo (značena znakem `%`), která vrací zbytek poceločíselném dělení deseti. Tím je vrácena pozice znaku v ASCII tabulce. Abychom se však pohybovali v rozsahu čísel 0 – 9, je nutné ještě přičíst hodnotu 48, což je pozice čísla 0 v ASCII tabulce. Ostatní cifry jsou převáděny obdobně, jen je potřeba zvyšovat řád násobení původní hodnoty a provádět modulo pro každý řád hodnoty.

Následuje funkce, která okopíruje pole získaných hodnot do proměnné rxUart.data[] a odešle je do rozhraní UART.

```
for (uint8_t i=0; i<10; i++)
{
    rxUart.data[i]=teplota_cislice[i];
}

setDataToSendBle(rxUart.data, 10, UCOMM_SENDBLE_DATA);
```

Při nastavování výstupních hodnot funkcí, které vypisují data do jednotlivých rozhraní, nebo je z nich získávají, je vhodné použít jako přehled všech tisknutelných znaků ASCII tabulku. ASCII znamená American Standard Code for Information Interchange (v překladu americký standardní kód pro výměnu informací) a obsahuje všechny znaky, které je možné přes Bluetooth posílat. Při vývoji programu byla použita tabulka dostupná z [37].

5.7 Naměřené hodnoty teplot

V tabulce naměřených hodnot jsou zaznamenány hodnoty teplot s různými konfiguracemi vstupu informací. Nejpřesnější je varianta ADC_Input_AdcPin12, kdy se měří diferenciální napětí mezi piny ADC1 a ADC2. Varianta, kdy se využívá vnitřní teplotní senzor MCU (ADC_Input_TempSensor) je nejméně přesná, dosahuje odchylky až +7,3 °C. Jako referenční měřidlo teploty byl použit digitální multimetr FY107C s teplotním senzorem.

Tabulka 5.1 Naměřené hodnoty teploty při různých senzorech

Typ senzoru	Naměřená hodnota teploty T [°C]	Odchylka od referenční hodnoty teploty naměřené multimetrem δT [%]
Vnitřní senzor multimetru FY107C (T_{ref})	23	0
Externí sonda multimetru FY107C	24	4,35
ADC_Input_AdcPin1	27,568	19,86
ADC_Input_AdcPin12	23,479	2,08
ADC_Input_TempSensor	30,327	31,86

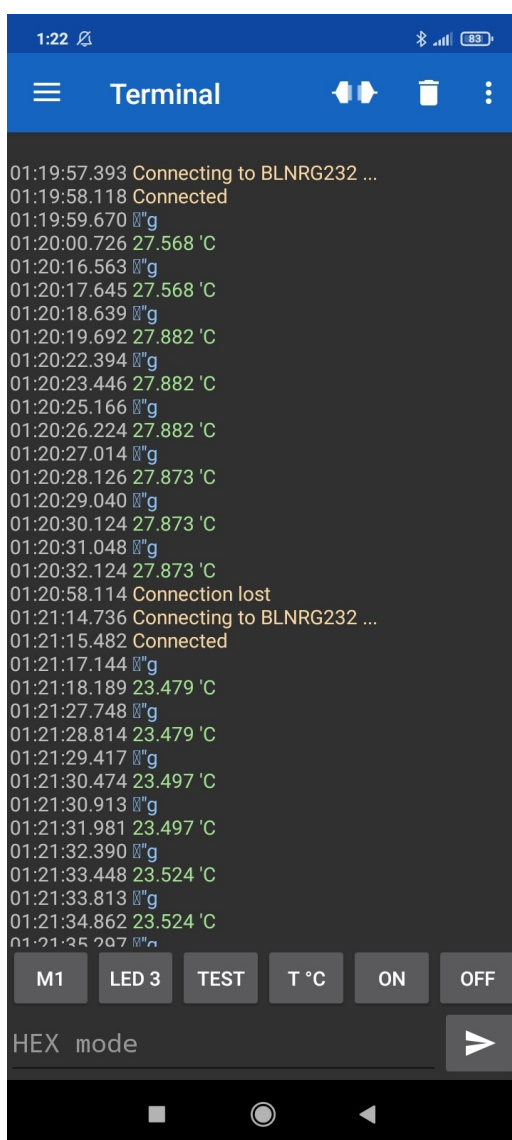
Odchylka naměřené teploty od referenční hodnoty teploty je spočítána podle rovnice:

$$\delta T = \frac{T_{naměřená} - T_{ref}}{T_{ref}} * 100 \quad (5.1)$$

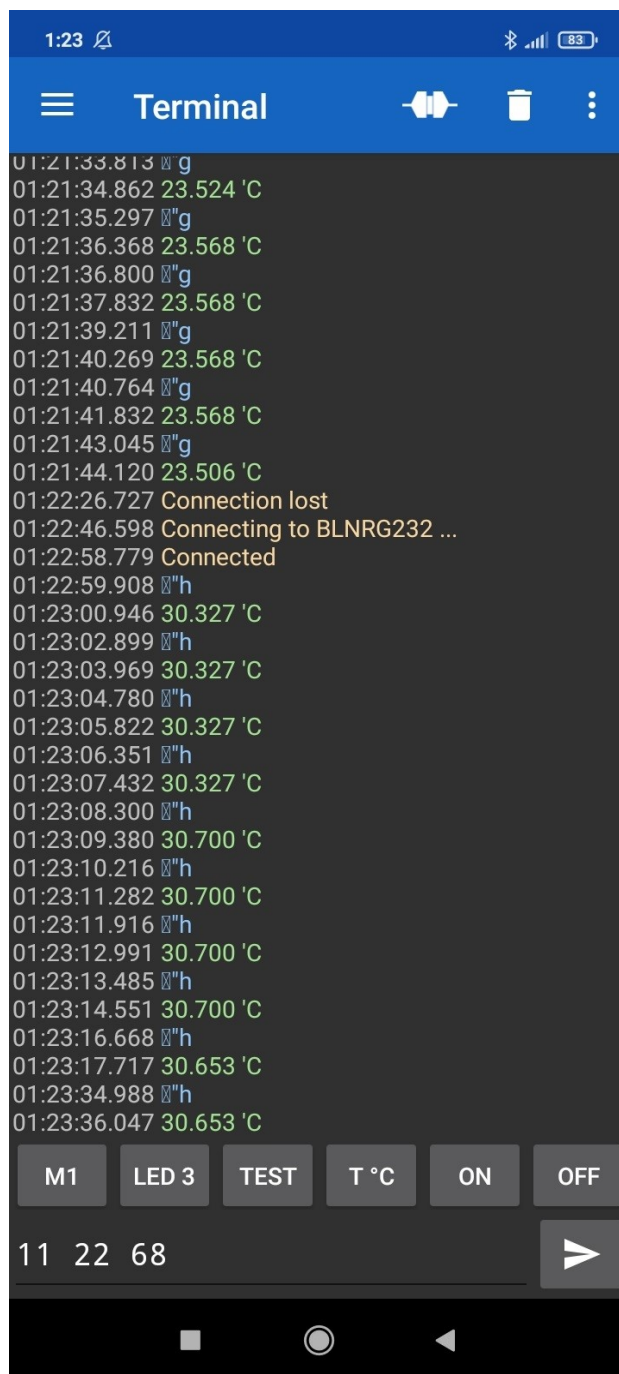
Příklad výpočtu:

$$\delta T = \frac{T_{\text{naměřená}} - T_{\text{ref}}}{T_{\text{ref}}} * 100 = \frac{23,479 - 23}{23} * 100 = 2,08 \% \quad (5.2)$$

δT je odchylka naměřené hodnoty teploty od skutečné hodnoty teploty v %, $T_{\text{naměřená}}$ je hodnota teploty z výstupu daného senzoru [°C] a T_{ref} je referenční teplota naměřená vnitřním senzorem multimetru FY107C.



Obrázek 5.11 Ukázka hodnot teploty naměřené při konfiguraci AdcPin1 a AdcPin12



Obrázek 5.12 Ukázka hodnot teploty naměřené při konfiguraci TempSensor

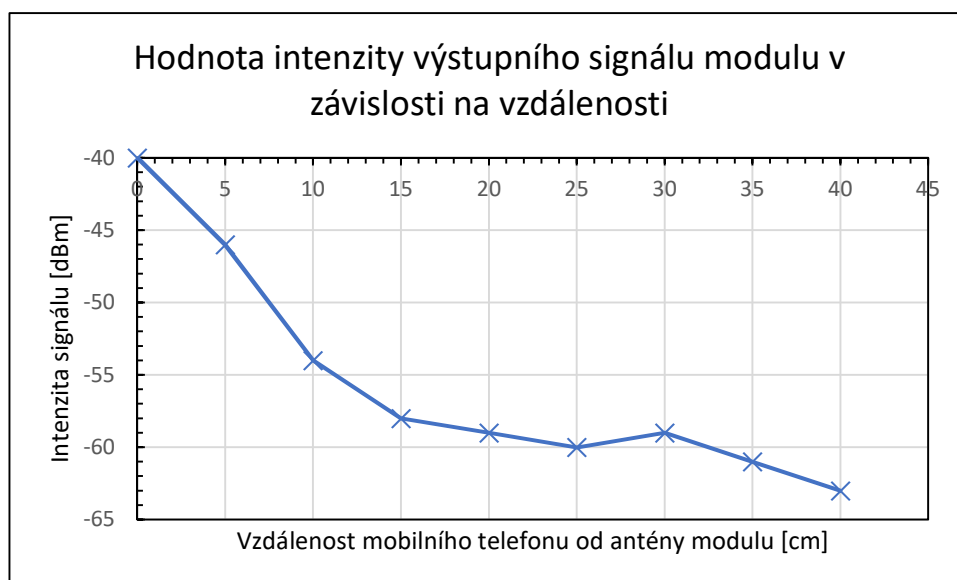
Na obrázcích výše jsou uvedeny naměřené hodnoty teplot v jednotlivých konfiguracích ADC převodníku. Pro výpočet vždy byla vybrána první naměřená hodnota, protože ostatní se od ní zpravidla lišily pouze minimálně, když vezmeme v úvahu relativně nízkou přesnost senzoru MCP9701. Proto nebyl stanoven průměr z naměřených hodnot pro daná měření.

6. MĚŘENÍ DOSAHU MODULU

Pro experimentální ověření dosahu modulu byly provedeny dva pokusy. Byla testována intenzita vysílaného signálu modulu v závislosti na vzdálenosti mobilního telefonu od antény na DPS ve stavu, kdy je modul inicializován s maximální intenzitou výstupního výkonu. Naměřené hodnoty jsou zobrazeny v tabulce 6.1 a grafu 6.1 .

Tabulka 6.1 Naměřené hodnoty vysílaného signálu v závislosti na vzdálenosti od modulu

Vzdálenost [cm]	Hodnota intenzity signálu [dBm]
0	-40
5	-46
10	-54
15	-58
20	-59
25	-60
30	-59
35	-61
40	-63



Obrázek 6.1 Grafické znázornění naměřených hodnot intenzity výstupního signálu

7. SEZNAM PŘÍKAZŮ PRO UART TERMINÁL A BLUETOOTH TERMINÁL

V této kapitole je uveden přehled všech příkazů, které je možné zadávat. U každého z nich je uvedeno odpovídající rozhraní, ve kterém se příkaz zadává a daná vykonaná událost. U příkazů pro Bluetooth se do terminálu zadává celý kód včetně začátku (11 22), který je uveden v závorce, protože je pro všechny tyto příkazy stejný. Zadává se bez závorek. Všechny znaky, ať už písmena nebo číslice se zadávají v podobě hexadecimální hodnoty, která odpovídá danému znaku podle ASCII tabulky.

*pozn. 1: U příkazu (11 22) 68 je nutné nejdříve v souboru drvServerBle.c změnit ve funkci ADC_Configuration(); parametr

```
xADC_InitType.ADC_Input = ADC_Input_AdcPin12;
```

za parametr

```
xADC_InitType.ADC_Input = ADC_Input_TempSensor;
```

*pozn. 2: Modul lze inicializovat již přímo při připojení k napájecímu napětí. Stačí do funkce int main(void) umístit následující kód:

```
                //nastavení MAC adresy modulu
rxUart.data[0]='p'; //v ASCII je to hexadecimálně 70
rxUart.data[1]='\`'; //v ASCII je to hexadecimálně 60
rxUart.data[2]='P'; //v ASCII je to hexadecimálně 50
rxUart.data[3]='@'; //v ASCII je to hexadecimálně 40
rxUart.data[4]='0'; //v ASCII je to hexadecimálně 30
rxUart.data[5]=' '; //v ASCII je to hexadecimálně 20
rxUart.data[6]=0x09; //oddelovací znak
rxUart.data[7]=0x42; //B
rxUart.data[8]=0x4C; //L
rxUart.data[9]=0x4E; //N
rxUart.data[10]=0x52; //R
rxUart.data[11]=0x47; //G
rxUart.data[12]=0x32; //2
rxUart.data[13]=0x33; //3
rxUart.data[14]=0x32; //2

initBle(rxUart.par); //inicializuj modul
```

Tabulka 7.1 Přehled příkazů pro modul

Rozhraní	Kód příkazu	Akce
Bluetooth	(11 22) 20	Vypíše do UARTU text „Koleje PPV“
Bluetooth	(11 22) 60	Vypíše do Bluetooth terminálu text „VUT FEKT UMEL 2022“
Bluetooth	(11 22) 65	Rozsvítí, případně zhasne LED D3

Bluetooth	(11 22) 67	Vypíše do Bluetooth terminálu teplotu z rozhraní AdcPin12
Bluetooth	(11 22) 68	Vypíše do Bluetooth terminálu teplotu z rozhraní TempSensor (*pozn. 1)
UART	50 01 01 xx xx	Vypíše do rozhraní UART a Bluetooth znaky na pozicích xx xx (těch může být více)
UART	10 07 xx xx xx xx xx xx 09 yy yy yy yy	Provede inicializaci modulu, kde 07 je úroveň výstupního výkonu (největší možná), xx představuje požadovanou MAC adresu modulu (6 čísel) a yy představuje požadovaný název modulu (až 11 písmen nebo čísel) (*pozn. 2)
UART	20	Vypíše do UARTU text „Koleje PPV“
UART	60	Vypíše do Bluetooth terminálu text „VUT FEKT UMEL 2022“
UART	65	Rozsvítí, případně zhasne LED D3
UART	67	Vypíše do Bluetooth terminálu teplotu z rozhraní AdcPin12

Součástí diplomové práce je i CD, které obsahuje návrhové soubory z programu Altium Designer (projekt, schéma zapojení, pcb design, schematic library a footprint library pro smd anténu), soubory pro výrobu desky plošných spojů a šablony pro nanášení pájecí pasty (soubory Gerber X2 a soubory s informacemi pro vrtání otvorů do DPS), kompletní program z programu Atollic TrueStudio a program Herkules, který slouží k monitorování komunikace přes COM port.

8. ZÁVĚR

V této diplomové práci byl popsán princip fungování standardu Bluetooth, jeho parametry, jednotlivé části, vrstvy a bylo poukázáno na jeho vysoký potenciál v rozvoji komunikačních prostředků na malé vzdálenosti s velmi nízkou spotřebou energie.

Dále byl představen mikrokontroler BlueNRG-2, byly popsány jeho vlastnosti, periferie a jejich možné využití.

V další kapitole byl popsán návrh komunikačního modulu Bluetooth, bylo představeno jeho celkové schéma zapojení a motiv plošného spoje, jakožto i schémata zapojení jednotlivých funkčních bloků a byla popsána také konstrukční řešení motivů plošného spoje pro tyto bloky. Byla vysvětlena jejich funkce a možnosti jejich ovládání.

Dále byl popsán řídicí program, jeho funkční části, jakým způsobem probíhá spuštění a ovládání jednotlivých funkčních částí modulu. Podrobně byla představena komunikace modulu s rozhraním UART a Bluetooth a vzájemná výměna informací mezi těmito rozhraními. Byly vysvětleny použité algoritmy k získávání informací ze senzorů modulu.

Na závěr bylo provedeno experimentální ověření navrženého modulu a funkčnost řešení jednotlivých částí zadání. Bylo provedeno měření intenzity signálu v závislosti na vzdálenosti od modulu, měření teploty a porovnání přesností naměřených hodnot s referenční hodnotou nezávislého měřidla.

Podarilo se tedy navrhnout, vyrobit a oživit modul, který komunikuje se sběrnici UART, je možné jej povelovat přes Bluetooth např. pomocí mobilního telefonu. Lze ovládat jeho vstupně výstupní periferie – tlačítka, LED a je možné získávat hodnotu z ADC převodníku a posílat ji přes Bluetooth zpět do mobilního telefonu.

Oproti původnímu zadání se nepodařilo zprovoznit OLED displej komunikující přes rozhraní I2C. I přes veškerou snahu byl hlavní problém s optimalizací funkce, která se stará o posílání dat přes sběrnici I2C do cílového zařízení – displeje. Rozhraní bylo inicializováno a byla k dispozici knihovna obsahující inicializační sekvenci displeje a funkce k zápisu dat a znaků do paměti displeje. Hlavní problém byl v optimalizaci funkce, která zajišťuje zápis příkazů do paměti displeje. Z několika dostupných variant, které byly odvozeny z oficiálních příkladů programů od výrobce STMicroelectronics, nalezeny na internetu nebo doporučený vedoucím práce, se nakonec nepodařilo žádnou optimalizovat pro použití pro mikrokontroler BlueNRG-232.

I přesto se však u modulu nachází stále spousta možností, co by se dalo vylepšit nebo opravit. Moduly je však možné použít k praktické ukázce toho, jak funguje komunikace pomocí standardu Bluetooth a jako učební pomůcku k procvičování programovacích dovedností.

LITERATURA

- [1] KOVAŘÍK, David. Bluetooth – modrozub pod drobnohledem (vědecké okénko). *Mobilizujeme.cz* [online]. Praha: ArteMan, 2011 [cit. 2021-11-11]. Dostupné z: <https://mobilizujeme.cz/clanky/bluetooth-modrozub-pod-drobnohledem-vedecke-okenko>
- [2] Bluetooth. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-11-11]. Dostupné z: <https://cs.wikipedia.org/wiki/Bluetooth>
- [3] FHSS. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-11-11]. Dostupné z: <https://cs.wikipedia.org/wiki/FHSS>
- [4] Received Signal Strength Indication. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-11-11]. Dostupné z: https://cs.wikipedia.org/wiki/Received_Signal_Strength_Indication
- [5] IEEE 802.15. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-11-11]. Dostupné z: https://en.wikipedia.org/wiki/IEEE_802.15
- [6] List of Bluetooth protocols. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-11-11]. Dostupné z: https://en.wikipedia.org/wiki/List_of_Bluetooth_protocols#HCI
- [7] Univerzální unikátní identifikátor. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-11-24]. Dostupné z: https://cs.wikipedia.org/wiki/Univerz%C3%A1ln%C3%AD_unik%C3%A1tn%C3%AD_identifik%C3%A1tor
- [8] KBA_BT_0201: Bluetooth advertising data basics. *Silicon Labs* [online]. USA: Silicon Labs, 2021 [cit. 2021-11-24]. Dostupné z: https://community.silabs.com/s/article/kba-bt-0201-bluetooth-advertising-data-basics?language=en_US
- [9] Logical Link Control Adaptation Protocol (L2CAP). *Tutorialspoint.com* [online]. Indie: Tutorial Point, 2020 [cit. 2021-11-24]. Dostupné z: <https://www.tutorialspoint.com/logical-link-control-adaptation-protocol-l2cap>
- [10] Cyklický redundantní součet. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-11-24]. Dostupné z: https://cs.wikipedia.org/wiki/Cyklick%C3%BD_redundantn%C3%AD_sou%C4%8Det
- [11] Bluetooth Host - L2CAP, SDP & GAP. *Electronics notes* [online]. UK: Radio-electronics.com, 2021 [cit. 2021-11-24]. Dostupné z: <https://www.electronics-notes.com/articles/connectivity/bluetooth/host-l2cap-sdp-gap.php>

- [12] TOWNSEND, Kevin. GATT. *Learn.adafruit.com* [online]. New York, USA: Adafruit, 2021 [cit. 2021-11-24]. Dostupné z: <https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt>
- [13] TOWNSEND, Kevin. GAP. *Learn.adafruit.com* [online]. New York, USA: Adafruit, 2021 [cit. 2021-11-24]. Dostupné z: <https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gap>
- [14] List of Bluetooth profiles. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-11-24]. Dostupné z: https://en.wikipedia.org/wiki/List_of_Bluetooth_profiles
- [15] Pairing. *Everywhereyoulook* [online]. USA: Teledyne Lecroy, 2021 [cit. 2021-11-25]. Dostupné z: <https://fte.com/WebHelpII/BPA600/Content/Documentation/WhitePapers/BTLE/Pairing.htm>
- [16] The Attribute Protocol (ATT). *Http://lpccs-docs.dialog-semiconductor.com/* [online]. UK: Dialog Semiconductor, 2019 [cit. 2021-11-25]. Dostupné z: <http://lpccs-docs.dialog-semiconductor.com/tutorial-custom-profile-DA145xx/att.html>
- [17] Bluetooth Protocol Stack. *Mathworks.com* [online]. USA: MathWorks, 2021 [cit. 2021-11-25]. Dostupné z: <https://www.mathworks.com/help/comm/ug/bluetooth-protocol-stack.html>
- [18] Bluetooth Packet Structure. *Mathworks.com* [online]. USA: MathWorks, 2021 [cit. 2021-11-25]. Dostupné z: <https://www.mathworks.com/help/comm/ug/bluetooth-packet-structure.html>
- [19] *BlueNRG-2 datasheet: Bluetooth® Low Energy wireless system-on-chip*. Rev. 8. USA, 2021. Dostupné také z: <https://www.st.com/resource/en/datasheet/bluenrg-2.pdf>
- [20] Introduction to Power-on-Reset. *HardwareBee.com* [online]. Izrael: HardwareBee, 2021 [cit. 2021-11-30]. Dostupné z: <https://hardwarebee.com/introduction-to-power-on-reset-circuit/>
- [21] Bluetooth Basics. *SparkFun Electronics* [online]. Colorado, USA: SparkFun Electronics, 2021 [cit. 2021-11-30]. Dostupné z: <https://learn.sparkfun.com/tutorials/bluetooth-basics/how-bluetooth-works>
- [22] What is Broadcaster, Observer, Central and Peripheral about ble module. *Feasycom.net* [online]. Čína: Feasycom.net, 2020 [cit. 2021-11-30]. Dostupné z: <https://www.feasycom.net/info/what-is-broadcaster-observer-central-and-per-46198363.html>
- [23] THAKUR, Dinesh. Direct memory access (DMA). *Ecomputernotes.com* [online]. USA: Computer Notes, 2021 [cit. 2021-11-30]. Dostupné z: <https://ecomputernotes.com/fundamental/introduction-to-computer/direct-memory-access>

- [24] Serial Peripheral Interface. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-11-30]. Dostupné z: https://cs.wikipedia.org/wiki/Serial_Peripheral_Interface
- [25] CAMPBELL, Scott. Basics of UART communication. *Circuitbasics.com* [online]. USA: Circuit Basics, 2021 [cit. 2021-11-30]. Dostupné z: <https://www.circuitbasics.com/basics-uart-communication/>
- [26] I²C. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-12-01]. Dostupné z: <https://cs.wikipedia.org/wiki/I%C2%B2C>
- [27] Bluetooth® SMART Beacons: chytré elektronické majáky [online]. 2015. Liberec: CADware, 2015 [cit. 2021-12-08]. ISSN 1805-5044. Dostupné z: <https://www.dps-az.cz/soucastky/id:8572/bluetooth-smart-beacons-chytre-elektronicke-majaky>
- [28] Bluetooth Beacon Technology Advantages. *Acrylic Wi-Fi* [online]. Španělsko: Acrylic WiFi software by Tarlogic Research, 2021 [cit. 2021-12-08]. Dostupné z: <https://www.acrylicwifi.com/en/blog/bluetooth-beacon-technology-advantages/>
- [29] Future Technology Devices International FT230X (USB to BASIC UART IC): Datasheet obvodu FT230X. Velká Británie, 2021. Dostupné také z: https://ftdichip.com/wp-content/uploads/2021/10/DS_FT230X.pdf
- [30] TPD2E001 Low-Capacitance 2-Channel ESD-Protection for High-Speed Data Interfaces. Texas, USA, 2016. Dostupné také z: <https://www.ti.com/lit/ds/symlink/tpd2e001.pdf>
- [31] Datasheet k MCP9701T-E/LT: Low-Power Linear Active Thermistor ICs. Microchip Technology Inc., 2016. Dostupné také z: <https://cz.mouser.com/datasheet/2/268/20001942G-1820450.pdf>
- [32] Rozměry SMD součástek. *ISPShop.cz* [online]. Zlín: ISPshop, 2022 [cit. 2022-05-21]. Dostupné z: http://www.ispshop.cz/index.php?main_page=product_info&cPath=89&products_id=934
- [33] TrueSTUDIO - A powerful eclipse-based C/C++ integrated development tool for your STM32 projects - STMicroelectronics [online]. Ženeva: STMicroelectronics, 2022 [cit. 2022-05-22]. Dostupné z: <https://www.st.com/en/development-tools/truestudio.html>
- [34] STSW-BLUENRG1-DK - BlueNRG-1, BlueNRG-2 DK SW package - STMicroelectronics [online]. Ženeva: STMicroelectronics, 2022 [cit. 2022-05-22]. Dostupné z: https://www.st.com/content/st_com/en/products/embedded-software/evaluation-tool-software/stsw-bluenrg1-dk.html
- [35] PCB design guidelines for the BlueNRG-1 and BlueNRG-2 devices. STMicroelectronics, Ženeva, Švýcarsko, 2018. Dostupné také z:

https://www.st.com/resource/en/application_note/an4819-pcb-design-guidelines-for-the-bluenrg1-and-bluenrg2-devices-stmicroelectronics.pdf

- [36] *Future Technology Devices International FT230X (USB to BASIC UART IC) - datasheet*. Future Technology Devices International Limited, 2021. Dostupné také z: https://ftdichip.com/wp-content/uploads/2021/10/DS_FT230X.pdf
- [37] ASCII. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2022 [cit. 2022-05-23]. Dostupné z: <https://cs.wikipedia.org/wiki/ASCII>

SEZNAM SYMBOLŮ A ZKRATEK

Zkratky:

ADC	Analogově digitální převodník (Analog Digital Converter)
DPS	Deska plošných spojů (stejná jako PCB)
GND	Referenční napájecí pin s nulovým napětím (Ground)
IPv6	Internetový protokol verze 6
MAC	Jednoznačný identifikátor síťového zařízení (Media Access Control)
MCU	Mikroprocesor (MicroController Unit)
PC	Osobní počítač (Personal Computer)
PCB	Deska plošných spojů (Printed Circuit Board)
PIN (kód)	Osobní identifikační kód (Personal Identification Number) (pozn. používá se také ve významu pin – vývod integrovaného obvodu)
RAM	Paměť s náhodným přístupem (Random Access Memory)
RS232	Doporučený standard (Recommended Standard) 232, standard komunikace pomocí sériového portu
USB	Univerzální sériové rozhraní (Universal Service Bus)
VF	Vysoko-frekvenční (např. signál)
WI-FI	Zkratka označující bezdrátovou internetovou síť, která vznikla ze slovního spojení Wireless Fidelity (bezdrátová věrnost) jako analogie ke dříve zažité zkratce Hi-FI (High Fidelity – vysoká věrnost), která se používá pro označení kvalitních zvukových souprav.

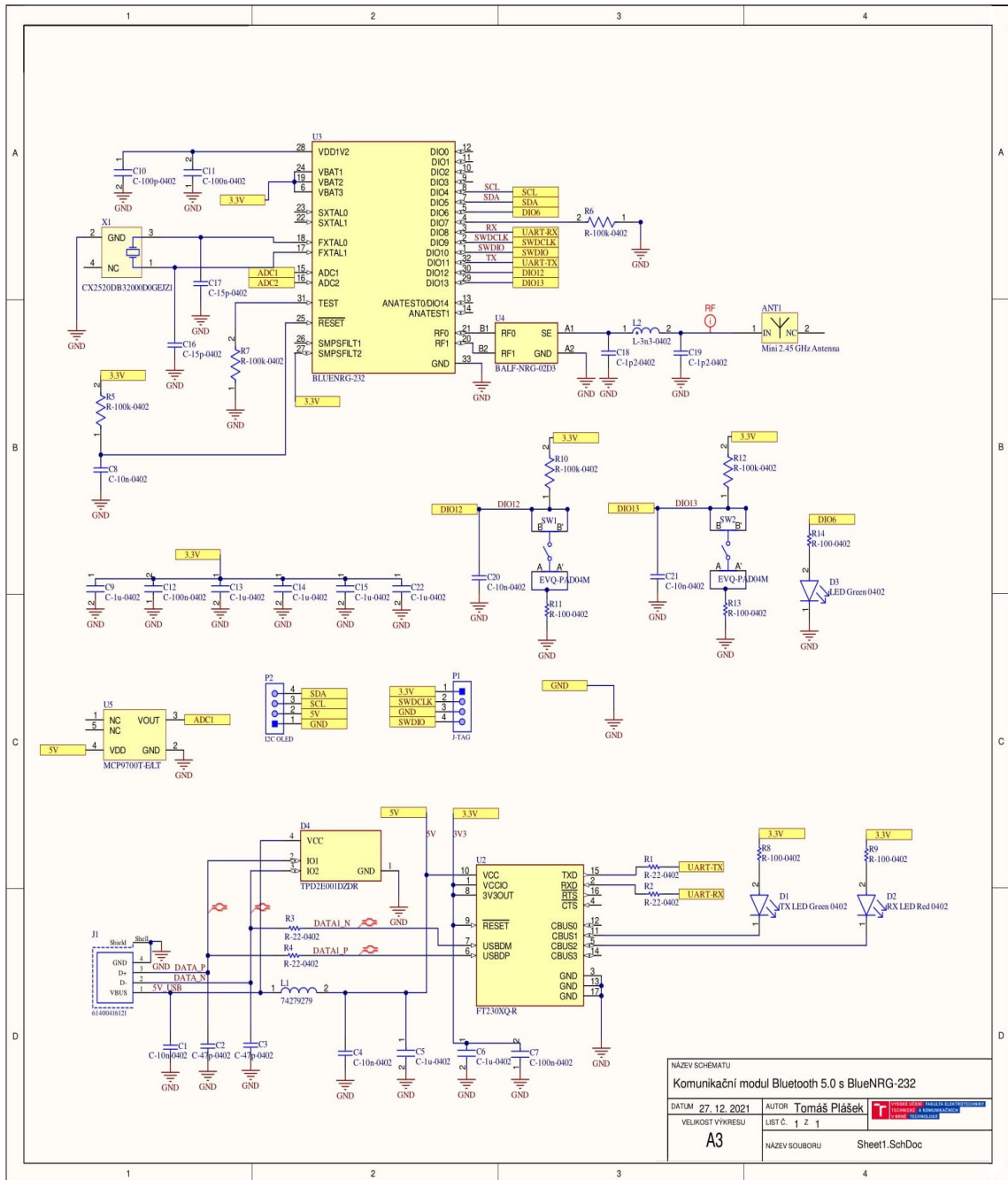
Symboly:

U	napětí	(V)
I	proud	(A)
R	rezistor	
C	kondenzátor	
L	cívka	

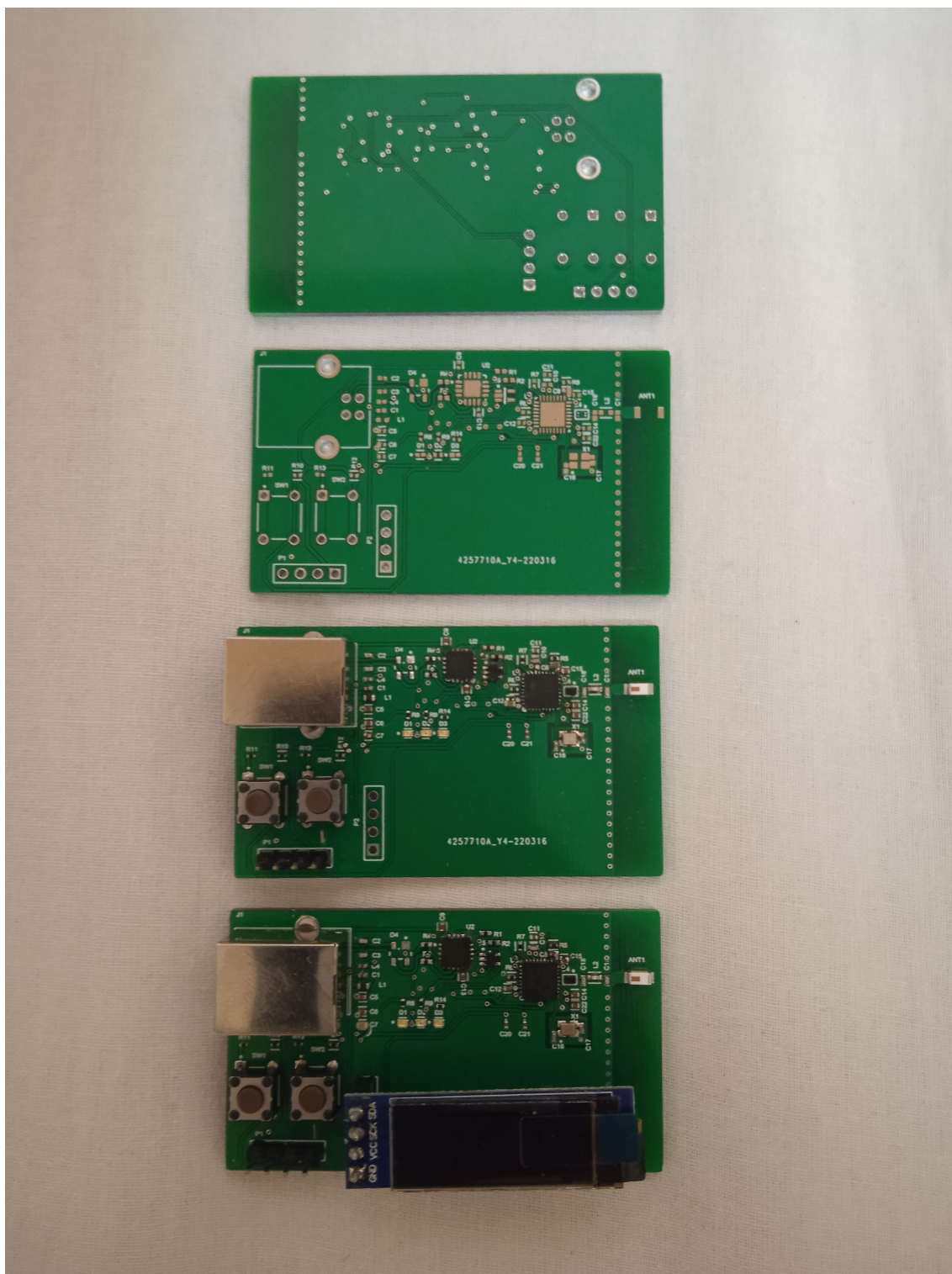
SEZNAM PŘÍLOH

PŘÍLOHA A - SCHÉMA ZAPOJENÍ.....	63
PŘÍLOHA B - FOTKY DESKY PŘED A PO OSAZENÍ SOUČÁSTKAMI.....	64
PŘÍLOHA C - USPOŘÁDÁNÍ PŘI MĚŘENÍ INTENZITY VÝSTUPNÍHO SIGNÁLU MODULU	65

Příloha A - Schéma zapojení



Příloha B - Fotky desky před a po osazení součástkami



Příloha C - Uspořádání při měření intenzity výstupního signálu modulu

