

Návrh a konstrukce testovacího produktu pro výrobní linku

Design and Construction of a Test Product for a Production Line

Bc. Daniel Kajzar

Diplomová práce

Vedoucí práce: Ing. Martin Stankuš, Ph.D.

Ostrava, 2021

Abstrakt

Cílem práce je návrh a konstrukce testovacího produktu s trojí funkcionalitou. Jedná se o funkce měření tepové frekvence, počtu kroků a teploty. Naměřená data jsou přeposílána pomocí bezdrátové komunikace Bluetooth Low Energy. Celková funkce zařízení je řízena pomocí mikrokontroléru. Pro řešení byly prostudovány principy zvolených snímačů, architektura bezdrátové technologie Bluetooth Low Energy a architektura zvoleného mikrokontroléru. Byly prostudovány funkce vývojových nástrojů, ve kterých byly následně realizovány návrhy. Programové vybavení zvoleného mikrokontroléru je narženo tak, aby mohlo být použito pro různá variantní osazení. Z navrženého elektrického schématu je vytvořen motiv plošného spoje. Podle těchto návrhů je zařízení vyrobeno a otestováno.

Klíčová slova

Altium Designer, Cortex-M0+, ARM, DPS, Bluetooth 4.0 Low Energy, multiplexované segmentové LCD, teploměr, krokoměr, pulzometr

Abstract

The aim of the thesis is the design and construction of a test product with triple functionality. These functions are the measuring of heart rate, counting of steps and measuring of the temperature. Furthermore, the measured data are forwarded by using Bluetooth Low Energy wireless communication. The overall function of the device is driven by a microcontroller. The principles of the selected sensors, the architecture of Bluetooth Low Energy wireless technology and the architecture of the selected microcontroller were studied. The function of development tools, in which designs were implemented, were studied. The software of the microcontroller is designed so that it can be used for various variants. The printed circuit board is created from the designed wiring diagram. Based on these designs, the device is manufactured and tested.

Keywords

Altium Designer, Cortex-M0+, ARM, PCB, Bluetooth 4.0 Low Energy, multiplex driven segmental LCD, thermometer, pedometer, heart rate monitor

Poděkování

Rád bych na tomto místě poděkoval svému vedoucímu práce doktorovi Martinu Stankušovi, za shovívavost a odborné rady při tvorbě této práce.

Obsah

Seznam použitých symbolů a zkratek	7
Seznam obrázků	9
Seznam tabulek	12
1 Úvod	13
2 Měření termodynamické teploty	16
2.1 Snímače pro měření termodynamické teploty	16
2.2 Polovodičové monolitické PN senzory termodynamické teploty	17
3 Snímače pro měření kroků	20
4 Měření tepové frekvence	22
4.1 Detekční algoritmus QRS komplexu Pan–Tompkins	23
5 Jádra ARM Cortex–M	25
5.1 Jádro ARM Cortex–M0+	25
5.2 Architektura Von Neumann	26
5.3 Zřetěžené zpracování instrukcí	26
5.4 Instrukční sady	27
6 Komunikační rozhraní	28
6.1 I ² C	28
6.2 SPI	30
6.3 UART	31
7 Segmentové LCD	33

8 Bluetooth Low Energy	35
8.1 GATT - Generic Attribut Profile	36
8.2 GAP - Generic Access Profile	37
9 Návrh DPS pomocí Altium Designer	38
10 Nástroje pro tvorbu programu	42
11 Zařízení Testbed	44
11.1 Mechanický návrh řešení	44
11.2 Návrh funkce programového vybavení	45
12 Návrh elektroniky	48
12.1 Zdroj	49
12.2 Displej	50
12.3 Integrovaný obvod pro měření teploty	51
12.4 Integrovaný obvod pro detekci počtu kroků	52
12.5 Integrovaný obvod pro detekci tepové frekvence	53
12.6 BLE modul	54
12.7 Tlačítka	56
12.8 Mikrokontrolér	56
12.9 Návrh elektrického zapojení	57
12.10 Návrh motivu plošného spoje	58
13 Návrh programového vybavení	60
13.1 Časovače	61
13.2 Detekce zdroje napájení	63
13.3 Ovladač pro LCD	64
13.4 Výběrové menu	66
13.5 Autodetekce osazených snímačů	68
13.6 Měření teploty	69
13.7 Detekce počtu kroků	71
13.8 Měření tepové frekvence	76
13.9 Identifikační číslo zařízení	79
13.10 Komunikace mezi BLE modulem a mikrokontrolérem	80
13.11 Programové vybavení BLE112	82
13.12 Komunikační protokol pro testovací stanici	84

14 Testování navrženého produktu	87
14.1 Měření tepové frekvence generovaného EKG signálu	88
14.2 Testování komunikačního protokolu pro testovací stanici	89
15 Závěr	90
Literatura	93
Přílohy	96
A Schéma zapojení	98

Seznam použitých zkratek a symbolů

3D	– trojdimenzionální
ASCII	– American Standard Code for Information Interchange
ATT	– Attribute Profile
BAL	– BPM_ARRAY_LENGTH
BLE	– Bluetooth 4.0 Low Energy
BMAV	– BPM_MAX_VALUE
BMIV	– BPM_MIN_VALUE
BMME	– BPM_MAX_MEAS_ERROR
BMS	– BPM_MAX_SPAN
CISC	– Complex Instruction Set Computer
CPU	– central processing unit
CTS	– Clear to send
DPS	– Deska plošného spoje
EDA	– Electronic design automation
EKG	– elektrokardiografie
GAP	– Generic Access Profile
GATT	– Generic Attribute Profile
GCC	– GNU Compiler Collection
GNU	– GNU's Not Unix
GPIO	– General-purpose input/output
H.A.S.L.	– Hot Air Solder Leveling
HexId	– Identifikační číslo zařízení Testbed
IBM	– International Business Machines Corporation
JTAG	– Joint Test Action Group
LCD	– Liquid crystal display
LDO	– low-dropout
LSB	– least significant bit
MCU	– Microcontroller

MEMS	– Micro Electro Mechanical Systems
MISO	– Master in Slave out
MOSI	– Master out Slave in
MSB	– Most Significant Bit
MSOP	– Mini small-outline package
NI	– National Instruments
ODR	– Output Data Rate
PIT	– Periodic Interrupt Timer
RISC	– Reduced Instruction Set Computer
RST	– Request to send
SI	– Mezinárodní systém jednotek
SMD	– Surface Mount Device
SWD	– Serial Wire Debug
UART	– univerzální asynchronní přijímač/vysílač
UUID	– Univerzální unikátní identifikátor
VLPR	– Very Low Power Run
WFE	– Wait For Event
WFI	– Wait For Interrupt

Seznam obrázků

2.1	Princip uspořádání polovodičového PN teplotního senzoru - proudy $I_1 \neq I_2$ [1]	17
2.2	Princip uspořádání polovodičového PN teplotního senzoru - proudy $I_1 = I_2$ [1]	18
3.1	Princip kapacitního akcelerometru [4]	20
3.2	Princip MEMS akcelerometru pro jeden rozměr[4]	21
4.1	Převodní srdeční systém [5]	22
4.2	Tvar normální EKG křivky [5]	23
4.3	Řetězec algoritmu Pan-Tompkins	23
4.4	Průběhy EKG signálu za jednotlivými bloky [7]	24
5.1	Von Neumann architektura	26
5.2	Řetěžené zpracování instrukcí s dvěma etapami [11]	27
6.1	Demonstrativní zapojení uzlů sběrnice I ² C [16]	29
6.2	Časový průběh Start bitu [16]	29
6.3	Sekvence komunikačního protokolu I ² C [16]	30
6.4	Demonstrativní zapojení uzlů SPI sběrnice [18]	30
6.5	Spojení dvou zařízení pomocí sběrnice UART	31
6.6	Průběh přenosu dat po asynchronní sběrnici UART [12]	31
6.7	Spojení dvou zařízení pomocí sběrnice UART s řídicími signály CTS a RTS [19]	31
7.1	Sedmisegment propojení pinů [20]	33
7.2	Multiplexovaný LCD se čtyřmi digity zapojení Frontplane	34
7.3	Multiplexovaný LCD se čtyřmi digity zapojení Backplane	34
8.1	Vztahy mezi profily, servisy a charakteristikami	36
9.1	Ukázka návrhu součástky pro SMD kondenzátor	38
9.2	Okno <i>Library Component Properties</i>	40

10.1	Propojení programovací sondy s debuggerem a mikrokontrolérem	43
11.1	Rozměry DPS	45
11.2	Nápisy na LCD	46
12.1	Blokové schéma elektroniky Testbed	48
12.2	Schéma zdroje	49
12.3	Graf výstupního napětí LDO regulátoru v závislosti na vstupním napětí [32]	50
12.4	Rozměry LCD v mm[inch] [33]	50
12.5	Schéma zapojení integrovaného obvodu AT30TS74	52
12.6	Vnitřní zapojení přepínače [34]	52
12.7	Schéma zapojení integrovaného obvodu KX126-1063	53
12.8	Schéma zapojení integrovaného obvodu MAX30003	54
12.9	Schéma zapojení modulu BLE112	55
12.10	Modul BLE112 a jeho motiv plošného spoje [37]	55
12.11	Schéma zapojení mechanických tlačítek	56
12.12	Oscilátor pro mikrokontrolér	57
12.13	Umístění komponent	59
12.14	Fotka reálného zařízení	59
13.1	Diagram aktivit zařízení	60
13.2	Diagram aktivit obslužné rutiny PIT_SLOW	62
13.3	Principiální schéma zapojení D/A převodníku [39]	64
13.4	Příklad zobrazení čísla dvě na prvním digitu	66
13.5	Příklad zobrazení hodnot	66
13.6	Stavový diagram funkce pravého a levého tlačítka	67
13.7	Diagram aktivit pro obslužnou rutinu pravého tlačítka	67
13.8	Diagram aktivit algoritmu zápisu na LCD	68
13.9	Hodnoty teploty zobrazené na LCD	69
13.10	Diagram aktivit algoritmu měření teploty	70
13.11	Průběh absolutní hodnoty tří rozměrů akcelerometru a okamžitá hodnota špičky signálu [40]	72
13.12	Časový interval špičky signálu [40]	73
13.13	Prahová hodnota STP_TH [40]	74
13.14	Diagram aktivit algoritmu čtení počtu kroků	75
13.15	Stavový diagram pro smazání počtu kroků uživatelem	76
13.16	Diagram aktivit algoritmu měření tepové frekvence	78
13.17	Grafické rozhraní pro editaci sériového čísla	79
13.18	Diagram aktivit algoritmu měření zápisu hodnoty do BLE modulu	81

13.19	Struktura vytvořeného servisu a jeho charakteristiky	83
14.1	Snímek z aplikace nRF Connect for Mobile	87
14.2	Schéma propojení měřicí karty a zařízení Testbed	88
14.3	Fotografie pracoviště pro měření generovaného signálu EKG	89
14.4	Snímek z programu Hercules při testování komunikačního protokolu	89

Seznam tabulek

5.1	Přehled obsahu jednotlivých typů jader Cortex-M	25
8.1	Rozdíl mezi klasickým Bluetooth a BLE	35
12.1	Tabulka pinů LCD-S401M16KF [33]	51
12.2	Definice adresy integrovaného obvodu teplotního snímače [34]	51
13.1	Přehled obsahu jednotlivých bitů proměnné PitFastProcessStart	62
13.2	Soupis pinů využité pro LCD	65
13.3	Registry a očekávané hodnoty při automatické detekci osazení	68
13.4	Výpis všech registrů teplotního snímače [34]	69
13.5	Soupis nastavitelných parametrů a jejich příslušných registrů	73
13.6	Datový rámeček zprávy pro zaslání hodnoty do BLE modulu	80
13.7	Datový rámeček zprávy pro zaslání dat z BLE modulu	80
13.8	Typ zprávy	80
13.9	Přehled obsahu jednotlivých bitů proměnné BleSendProcess	82
13.10	Struktura příkazu bez parametru	84
13.11	Soupis příkazů bez parametru	84
13.12	Struktura příkazu s parametrem	85
13.13	Soupis příkazů s parametrem	85
13.14	Struktura odpovědi	86
13.15	Soupis odpovědí	86

Kapitola 1

Úvod

V první kapitole teoretické části práce je nastíněné téma měření termodynamické teploty. Je zde osvětlena fyzikální podstata termodynamické teploty a její nejpoužívanější stupnice. Dále jsou uvedeny senzory, které se využívají pro měření termodynamické teploty a jsou uvedena kritéria jejich dělení. Poté je popsán princip polovodičového monolitického PN senzoru, který je použit ve zvoleném snímači. V kapitole zabývající se akcelerometry, jsou uvedeny a vysvětleny základní principy snímání zrychlení. Hlavní část je věnována akcelerometrům s proměnlivou kapacitou, jelikož na tomto principu je zkonstruován použitý akcelerometr v praktické realizaci zařízení. V následující kapitole je popsán EKG signál a je objasněn původ jeho vzniku. Následně jsou vysvětleny důležité pojmy a představen detekční algoritmus QRS komplexu Pan–Tompkins. Po úvodu principů použitých snímačů, je popsána architektura jádra použitého mikrokontroléru. Jedná se o architekturu Cortex–M0+, kde jsou uvedeny její základní vlastnosti. Je zde vysvětlena architektura MCU Von Neumann, uveden pojem zřetězené zpracování instrukcí a vysvětlena jeho funkce. V neposlední řadě je osvětlena využívána instrukční sada pro Cortex–M0+. V další kapitole teoretické části této práce jsou uvedeny komunikačními rozhraní I²C, SPI a UART. U těchto komunikačních rozhraní je popsána komunikace a jejich základní vlastnosti. Následně jsou popsány segmentové LCD. Jsou zde uvedeny využívané piny pro staticky řízené LCD, ale také i pro multiplexovaně řízené LCD. Dále pomocí obrázku je vysvětleno řízení multiplexovaných displejů. Následující kapitola je zaměřena na bezdrátovou technologii Bluetooth Low Energy. Jak je již z názvu zřetelné, jedná se o technologii s nízkou spotřebou elektrické energie, což je blíže popsáno v první části kapitoly. Poté jsou osvětleny pojmy GATT a GAP. V posledních dvou kapitolách teoretické části jsou uvedeny softwarové vývojové nástroje používané pro praktickou realizaci práce. První z kapitol je věnována EDA programu Altium Designer. Zde jsou popsány základní nástroje pro tvorbu knihovny součástek, kreslení elektrických schémat, návrh motivu plošného spoje a tvorba výrobních dat. Druhá z kapitol je věnována nástrojům pro tvorbu programového vybavení mikrokontroléru. Zde je blíže popsán program Atollic®TrueSTUDIO® a program Ozone od firmy SEGGER Microcontroller. Následně je osvětleno SWD rozhraní pro nahrání programu do mikrokontroléru.

V první kapitole praktické části je uvedena koncepce návrhu zařízení Testbed a jsou popsány požadavky zadavatele. Dále je stanoven typ bezdrátové komunikace a komunikace s testovací stanicí. Následně je popsán návrh elektroniky. Jsou zde uvedeny parametry, které ovlivnily výběr jednotlivých komponent a je uvedeno blokové schéma elektroniky. Poté jsou v jednotlivých kapitolách popsány vybrané komponenty. V první řadě je uveden návrh napájecího zdroje. Je uvedena napájecí hladina a zvolený LDO regulátor. Dále je zobrazeno navržené schéma zapojení napájecího zdroje, kde jsou mj. uvedeny dva možné zdroje elektrické energie. Posléze je uvedena vybraná komponenta displeje a uvedeny základní vlastnosti a způsob montáže displeje na DPS. Následně bylo přistoupeno k popisu vybraných snímačů. V první řadě je popsán snímač teploty a uvedeny důvody výběru snímače, typ použitého pouzdra, způsob vytvoření adresy pro komunikaci pomocí komunikačního rozhraní I²C. Jsou uvedeny teplotní rozsahy snímače s měřícím rozlišením a uvedeno navržené schéma zapojení. Druhý popsán snímač je akcelerometr. Opět jsou uvedeny základní parametry snímače, včetně způsobu komunikace, typu pouzdra a je zobrazeno navržené schéma zapojení. Jako poslední ze snímačů je uveden integrovaný obvod MAX3003, který je využit pro měření tepové frekvence. Jsou uvedeny základní vlastnosti snímače a navržené schéma zapojení. Dále je přistoupeno k popisu vybraného BLE modulu. Pomocí obrázku je zobrazeno navržené schéma zapojení s popisem podstatných částí návrhu. Dále je zobrazen doporučený motiv plošného spoje, který je uveden v katalogovém listu výrobce. V neposlední řadě je zobrazen návrh elektrického zapojení tlačítek a zvoleného mikrokontroléru. Po uvedení použitých komponent je přistoupeno k popisu návrhu elektrického zapojení a motivu plošného spoje. Je zde zmíněný použitý program pro návrh DPS a popsáno blokové schéma elektroniky. V předposlední kapitole praktické části je uveden návrh programového vybavení. V první části je uveden diagram aktivit programového vybavení zařízení. Jednotlivé části diagramu jsou dále rozebrány v následujících kapitolách. Jako první jsou popsány oba použité časovače. Posléze je popsána detekce zdroje napájení, ke které byla využita integrovaná periferie komparátoru v mikrokontroléru. Následně je uveden ovladač pro multiplexované LCD, popsán způsob jeho implementace a pomocí obrázku objasněna funkce navrženého ovladače. V další kapitole je popsán návrh cyklického menu a uveden stavový diagram pro pravé a levé tlačítko, pomocí kterých lze menu ovládat. Jsou uvedeny diagramy aktivit funkce těchto tlačítek. Poté je popsán princip autodetekce osazených snímačů. Je vysvětleno jakým způsobem dochází k detekci přítomnosti snímačů na daném zařízení. Dále následují tři kapitoly, ve kterých je popsán návrh měřících algoritmů pro jednotlivá měření. Obdobně jako při výběru komponent je jako první uveden algoritmus měření teploty, který je popsán pomocí diagramu aktivit. Poté je popsána integrovaná periferie krokoměru ve zvoleném akcelerometru. Následně je popsán algoritmus sběru dat z akcelerometru pomocí diagramu aktivit a možnost nulování počtu kroků. Jako poslední je uveden algoritmus pro měření tepové frekvence. Funkce algoritmu je opět popsána pomocí diagramu aktivit, a jsou vysvětleny jednotlivé navržené parametry ovlivňující průběh měření tepové frekvence. V následující kapitole je popsán způsob uložení identifikačního čísla (HexId), které slouží k identifikaci zařízení při párování zařízení s chytrým telefonem. Další dvě kapitoly jsou zaměřeny na BLE modul. První z kapitol je věnována návrhu

komunikace mezi BLE module a zvoleným mikrokontrolérem. S tímto úzce souvisí druhá z kapitol, ve které je popsáno programové vybavení BLE modulu. Poslední část věnována programovému vybavení, je návrh komunikačního protokolu mezi zařízením a testovací stanicí. Jsou uvedeny možné příkazy a odpovědi s definovanou strukturou.

V poslední části práce je provedeno testování navrženého produktu. Zde byly uvedeny způsoby ověření navržených algoritmů, otestování přenosu dat pomocí bezdrátové komunikace a komunikace mezi zařízením a testovací stanicí.

Kapitola 2

Měření termodynamické teploty

Teplota je stavovou veličinou určující stav termodynamické rovnováhy. Jedná se o stav, kdy nedochází k žádným makroskopickým změnám v izolované soustavě těles od okolního prostředí. Dále pak všechny fyzikální veličiny, kterými je stav soustavy popsán, nejsou závislé na čase. Teplota je jedna z veličin, kterou nelze měřit napřímo, ale pouze prostřednictvím jiných fyzikálních veličin. Takové měření se nazývá nepřímé měření. Teplota se vyjadřuje pomocí tzv. termodynamické stupnice. Mezi tři nejpoužívanější stupnice patří: [1]

- **Kelvinova teplotní stupnice** - Počátkem této stupnice je tzv. absolutní nula. Jedná se o termodynamickou teplotu, při které dochází k zastavení tepelného pohybu všech částic látky. Tedy teplota 0 K odpovídá absolutní nule. Jednotka této stupnice se nazývá kelvin (značeno K).
- **Celsiova teplotní stupnice** - Jednotka této stupnice je stupeň Celsia (značeno °C). Počátek této stupnice odpovídá hodnotě 273,15 K. To znamená, že 0 °C odpovídá 273,15 K.
- **Fahrenheitova teplotní stupnice** - Tato stupnice se využívá v některých státech. Jednotka této stupnice je označována jako stupeň Fahrenheita (značeno °F). Převod mezi teplotou vyjádřenou ve stupních Celsia a teplotou ve stupních Fahrenheita je: [1]

$$t(^{\circ}F) = \frac{9}{5} \cdot t(^{\circ}C) - 32 \quad (2.1)$$

2.1 Snímače pro měření termodynamické teploty

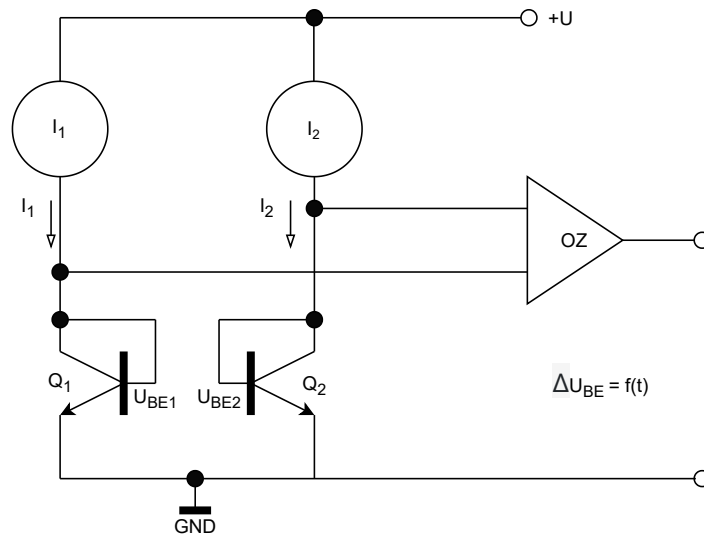
Pomocí těchto snímačů je možné snímat termodynamickou teplotu. Jak již bylo zmíněno, měření termodynamické teploty neprobíhá na přímo, ale prostřednictvím jiných fyzikálních veličin. Může se například jednat o fyzikální veličiny jako je elektrické napětí, elektrický proud, objem látky apod. Snímač je složen ze dvou základních částí, a to čidla a obvodů pro zpracování měřeného signálu. Čidlo je v přímém styku s měřeným prostředím a převádí termodynamickou teplotu na vhodnou

měřitelnou fyzikální veličinu, podle níž mohou být snímače děleny na odporové, termoelektrické, polovodičové s PN přechodem, radiační, akustické, optické, magnetické, kapacitní a mnoho dalších. [1]

Samozřejmě existují i jiná kritéria pro dělení teplotních snímačů. Například mohou být děleny na aktivní nebo pasivní snímače. Aktivním teplotním snímačem je (při změně termodynamické teploty) generovaná elektrická energie. Aktivní teplotní snímače tedy nepotřebují zdroj elektrické energie. Opakem jsou pasivní snímače termodynamické teploty, které pro svou funkci potřebují zdroj elektrické energie. Dále jsou snímače děleny podle styku s měřeným prostředím na dotykové a bezdotykové. [1]

2.2 Polovodičové monolitické PN senzory termodynamické teploty

Jedná se o jeden z nejpoužívanějších principů teplotních senzorů. Jsou založeny na teplotní závislosti PN přechodu v propustném směru. Teplotní rozsah se pohybuje v rozmezí $-55\text{ }^{\circ}\text{C}$ až $150\text{ }^{\circ}\text{C}$. Výhodou těchto senzorů je snadná integrace v čipech. Nevýhodou může být malý teplotní rozsah, nicméně v aplikacích jako je například pokojový teploměr je tento rozsah dostačující. [1] [2]

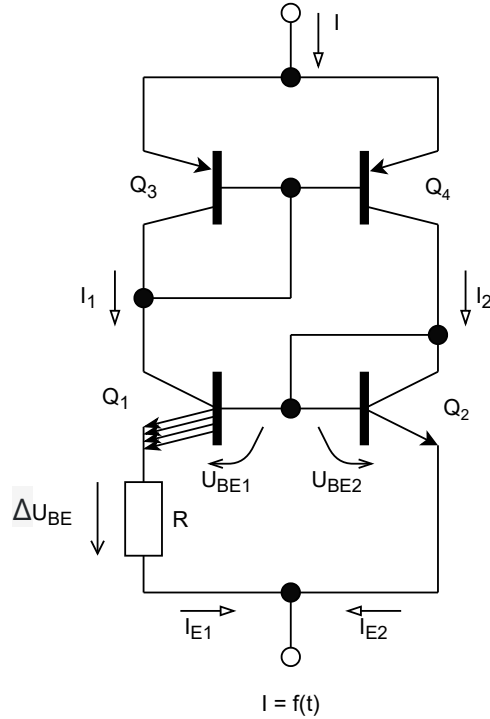


Obrázek 2.1: Princip uspořádání polovodičového PN teplotního senzoru - proudy $I_1 \neq I_2$ [1]

Monolitické (integrované) snímače jsou založeny na dvojici bipolárních tranzistorů napájených z proudových zrcadel. Na obrázku 2.1 je uvedeno zapojení, pro které platí, že proudy I_1 a I_2 se nerovnají. Vychází se tedy z rovnice:

$$\Delta U_{BE} = \frac{k}{e} \cdot \ln \left(\frac{I_1}{I_2} \right) \cdot T, \quad (2.2)$$

kdy T je termodynamická teplota v kelvinech, e je elementární náboj, k je Boltzmannova konstanta ($k = 1,38 \cdot 10^{-23} \text{ J} \cdot \text{K}^{-1}$). [1]



Obrázek 2.2: Princip uspořádání polovodičového PN teplotního senzoru - proudy $I_1 = I_2$ [1]

Další varianta je zobrazena na obrázku 2.2. V tomto případě se proudy I_1 a I_2 rovnají. Tranzistor Q_1 je nyní složen z několika paralelně zapojených tranzistorů s tím, že počet (značený r) těchto paralelně spojených tranzistorů, musí být menší než dvacet. Každý z těchto tranzistorů, má stejnou plochu emitorového přechodu S_E , jako tranzistor Q_2 . Platí tedy, že $S_{E1} = r \cdot S_{E2}$. Dále je nutné zmínit, že saturační proud PN přechodu v závěrném směru je závislý na ploše emitorového přechodu. Díky této skutečnosti lze pro poměr proudů I_1 a I_2 napsat rovnici:

$$\frac{I_2}{I_1} = \frac{S_{E2} \cdot e^{\frac{U_{BE2}}{U_T}}}{S_{E1} \cdot e^{\frac{U_{BE1}}{U_T}}}, \quad (2.3)$$

kde U_{BE1} a U_{BE2} je napětí mezi bází a emitorem, U_T je teplotní napětí, které má tvar:

$$U_T = \frac{k \cdot T}{e}. \quad (2.4)$$

Při vyjádření proudových hustot J_1 a J_2 (za předpokladu, že T_1 a T_2 jsou si rovny) dále vyplývá:

$$\frac{J_2}{J_1} = \frac{I_2}{r \cdot S_{E2}} \cdot \frac{I_1}{S_{E1}} \quad (2.5)$$

$$\frac{J_2}{J_1} = e^{\frac{U_{BE2} - U_{BE1}}{U_T}} \cdot \frac{1}{r} \quad (2.6)$$

$$r = e^{\frac{\Delta U_{BE}}{U_T}} \quad (2.7)$$

A nyní již můžeme vyjádřit ΔU_{BE} :

$$\Delta U_{BE} = U_T \cdot \ln r = \frac{k \cdot T}{e} \cdot \ln r = K \cdot T. \quad (2.8)$$

Ze zapojení na obrázku 2.2 je zřejmé, že $\Delta U_{BE} = I_{E1} \cdot R$. Dále dle uvedené rovnice 2.8 je ΔU_{BE} závislé na termodynamické teplotě. Taktéž tomu je u proudu I_{E1} . [1]

Tyto senzory se využívají převážně na měření běžných teplot. Tím je myšleno měření pokojové teploty, měření teploty v elektrických zařízeních apod. Většinou se jedná o čip, ve kterém je integrován samostatný senzor, ale také obvody pro zpracování a úpravu signálu. [2]

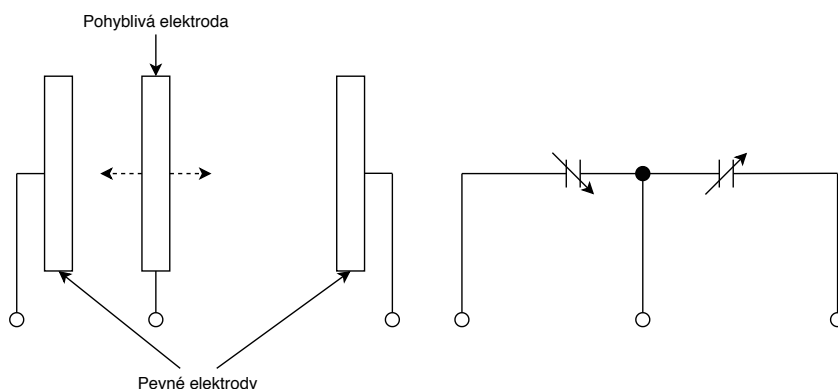
Kapitola 3

Snímače pro měření kroků

Pro snímání počtu kroků se využívají akcelerometry, které snímají statické nebo dynamické zrychlení. Pomocí akcelerometru je možné měřit odstředivé a setrvační síly, pozici či naklonění, případně vibrace snímače. Úkolem těchto snímačů je registrovat zrychlení snímače a převést tuto změnu na měřený elektrický signál. Pro snímání jsou využívány tři základní technologie akcelerometrů: [3]

- Akcelerometry s proměnnou kapacitou
- Piezoelektrické akcelerometry
- Piezoresistivní akcelerometry

Princip akcelerometru s proměnnou kapacitou založený na změně kapacity tříelektrodového vzduchového kondenzátoru, který je zobrazen na obrázku 3.1. Využívají se zde dvě pevné a jedna

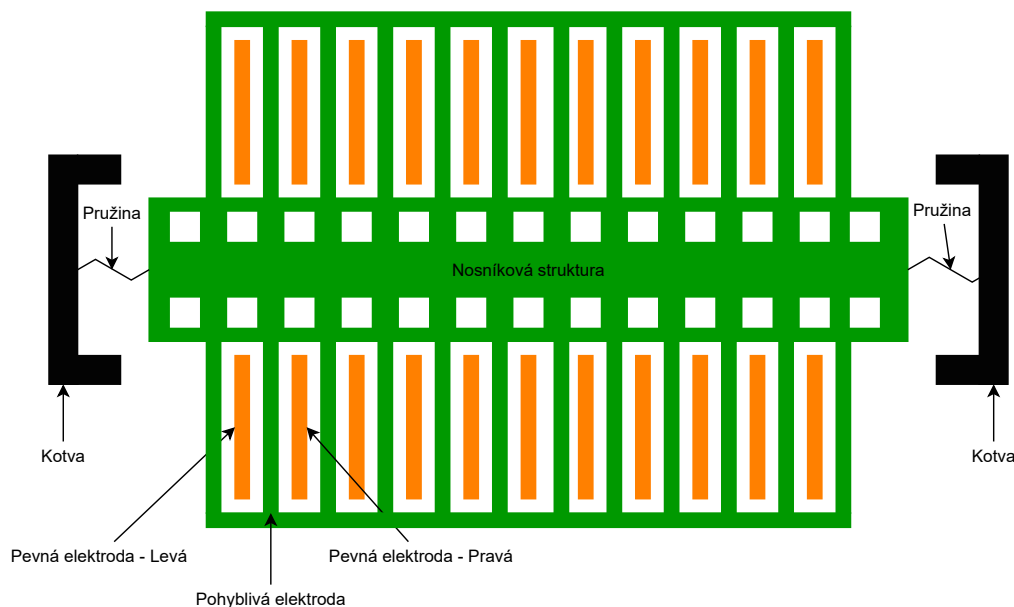


Obrázek 3.1: Princip kapacitního akcelerometru [4]

pohyblivá elektroda. Při pohybu snímače dochází ke změně polohy pohyblivé elektrody, a tudíž také ke změně kapacity, která je definována vzorcem:

$$C = \varepsilon \cdot \frac{S}{l} \tag{3.1}$$

kdy C je kapacita, ε je absolutní permitivita, l je vzdálenost mezi elektrodami a S je překrývající se plocha elektrod. Z uvedeného vzorce 3.1 vyplývá, že je zde využito nelineární závislosti mezi kapacitou a vzdáleností elektrod. Kapacitní akcelerometry je možné realizovat pomocí MEMS technologie, spojením více kapacitních akcelerometrů. Tím je vytvořen MEMS akcelerometr, který je principiálně zobrazen pro jeden rozměr na obrázku 3.2. Vychází se zde z druhého pohybového



Obrázek 3.2: Princip MEMS akcelerometru pro jeden rozměr[4]

Newtonova zákona, který má tvar:

$$F = m \cdot a, \quad (3.2)$$

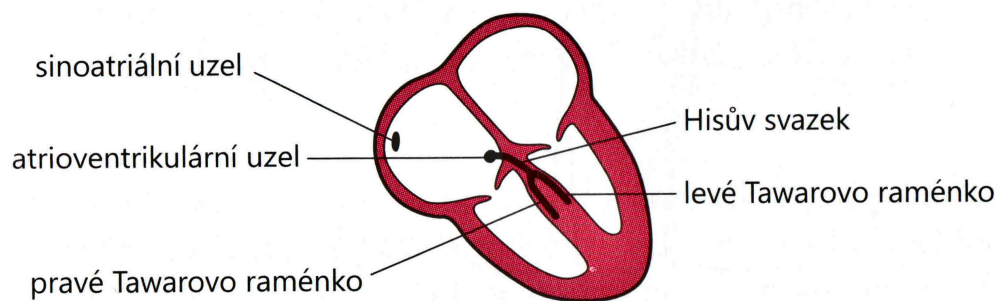
kdy F je síla, která vzniká působením zrychlení na těleso o hmotnosti m . Vzniklá síla působí na nosníkovou strukturu, která se díky pružinám po obou stranách pohybuje. Na ní jsou umístěny pohyblivé elektrody, které se pohybují mezi pevnými elektrodami. Tím dochází ke změně kapacity. Na uvedeném příkladu je zobrazen pouze akcelerometr pro jeden směr a to kolmý k nosníkové struktuře. Pro akcelerometry, pomocí kterých lze snímat změnu zrychlení ve třech prostorových osách, je potřeba přidat další dvě struktury. Tyto struktury musí být orientovány tak, aby byly citlivé ve zbylých dvou prostorových osách. [3] [4]

Piezoelektrické akcelerometry využívají tzv. piezoelektrický krystal. Pomocí něho je generován náboj při mechanickém namáhání krystalu kvůli působící síle (z druhého pohybového Newtonova zákona) na daný snímač. U piezorezistivních akcelerometrů se využívá piezorezistivního materiálu. Opět při mechanickém namáhání kvůli působící síle dochází ke změně odporu materiálu. [5]

Kapitola 4

Měření tepové frekvence

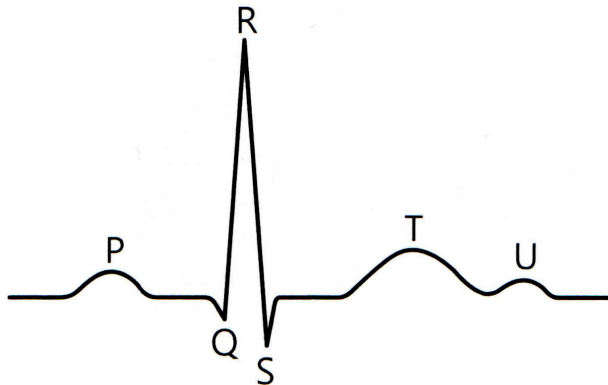
Při jakékoli kontrakci svalu dochází k tzv. depolarizaci. Jedná se o elektrické změny, které lze snímat pomocí elektrod umístěných na povrchu těla. Nicméně nesmí být opomenuto, že elektrodami jsou snímány všechny svalové kontrakce. Proto je nutné, aby osoba, na které probíhá měření, byla v klidu. Tím jsou minimalizovány kontrakce kosterních svalů. Poté jsou pak kontrakce srdečního svalu zřetelné. [5]



Obrázek 4.1: Převodní srdeční systém [5]

Na obrázku 4.1 je zobrazen převodní srdeční systém. Srdce je sice složeno ze čtyřech částí (dvou komor a dvou síní), nicméně z pohledu elektrické aktivace jej lze rozdělit pouze na dvě části. Důvodem je to, že obě síně a následně komory jsou stahovány současně. Elektrický výboj vychází (za normálních okolností) z tzv. sinoatriálního uzlu, který je umístěn v pravé síni. Následně se pak depolarizace šíří síňovými svalovými vlákny. Dále pak v atrioventrikulárním uzlu dochází ke zpoždění depolarizace. Vzniklý elektrický výboj se pak nadále velmi rychle šíří pomocí vodivé tkáně - Hisův svazek. Ten je dělen v komorovém septu na levé a pravé Tawarovo raménko. Následně se poté levé raménko dělí na dva svazky. Vzruch se dále ve svalovině komor šíří pomaleji pomocí tkáně zvané Purkyňova vlákna. [5]

Tvar měřeného signálu je zobrazen na obrázku 4.2. Tento signál je složen z několika částí. První část je tzv. vlna P, která je spojena s kontrakcemi obou síní. Dále pak následuje velký překmit



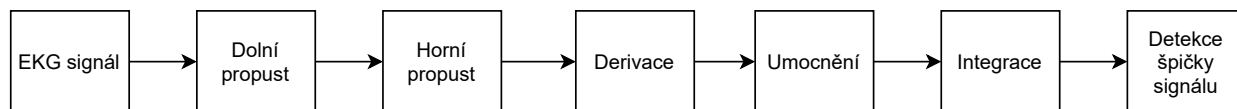
Obrázek 4.2: Tvar normální EKG křivky [5]

nazývaný QRS komplex, který vzniká korekcemi komor. Svalová hmota komor je větší než svalovina síní. Z toho plyne, že elektrická změna způsobena kontrakcemi komor je větší než u kontrakcí síní. Po QRS komplexu následuje vlna T, která je způsobená repolarizací (návrat komorového myokardu do elektricky klidového stavu). Jako poslední následuje vlna U, která není již tak běžná. Původ této vlny není zcela jasný, ale může se například jednat o repolarizaci papilárních svalů. Rozkmit EKG signálu je velice malý, typicky dosahuje 1 mV. [5]

Další důležitý pojem pro snímání EKG křivky je tzv. svod. Nejedná se o vodič připojený na pacienta, jak by se mohlo na první pohled zdát. Pojem svod naopak vyjadřuje elektrický obraz srdce. Typicky se využívají dvanáctisvodové EKG, kdy každý svod poskytne odlišný pohled na srdeční elektrickou aktivitu. V práci byl využit svod I, kde je snímáno napětí mezi elektrodami umístěných na levé a pravé horní končetině. Tímto je získán jeden EKG obraz, ze kterého jde nadále pomocí detekčních algoritmu detekovat srdeční tep. [5]

4.1 Detekční algoritmus QRS komplexu Pan–Tompkins

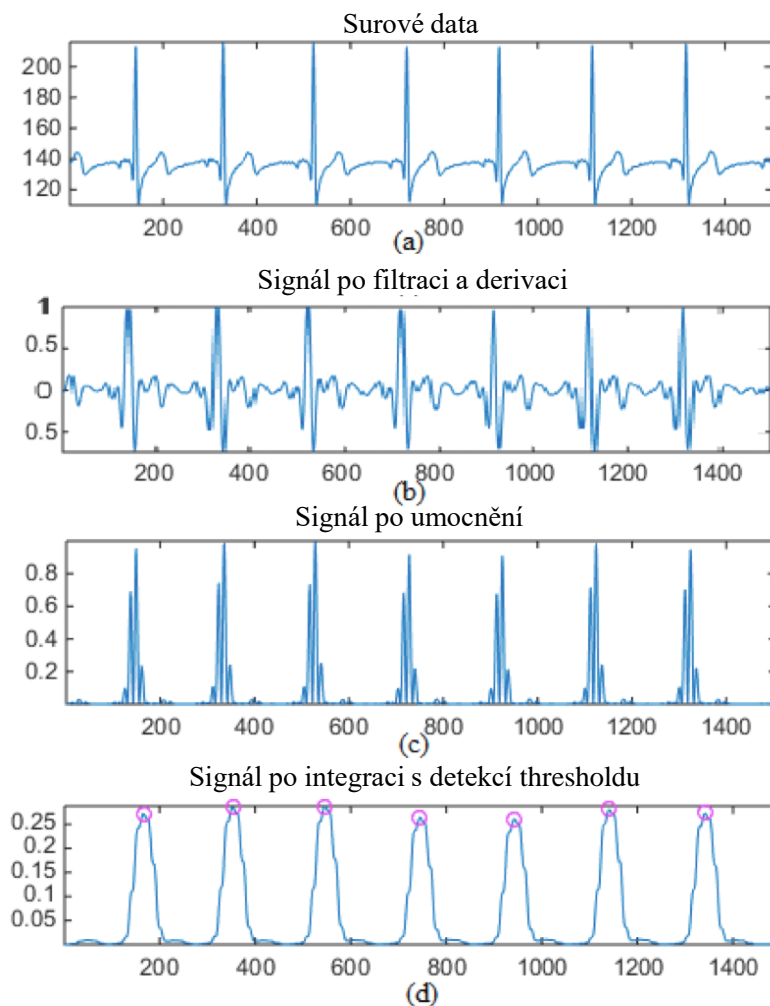
Detekční algoritmus Pan–Tompkins je jedním z nejpoužívanějších algoritmu pro detekci QRS komplexu. Tento algoritmus je složen z několika částí, které jsou graficky zobrazeny na obrázku 4.3. [6]



Obrázek 4.3: Řetězec algoritmu Pan-Tompkins

V první části je provedena filtrace signálu pomocí horní a dolní propusti. Jedná se o pásmový filtr, pomocí kterého jsou redukovány různé vzruchy (např. od svalstva). Požadované frekvenční pásmo pro detekci QRS komplexu je od 5 Hz až do 15 Hz. Následně po filtraci signálu je provedena derivace.

QRS komplex má strmější nárůst a pokles oproti vlnám P, T a U. Tím pádem pomocí derivace dojde k potlačení těchto nízkofrekvenčních vln a naopak zvýraznění QRS komplexu. Za derivačním blokem následuje blok umocnění. Pomocí tohoto bloku, signál již bude nabývat pouze kladných hodnot, jsou zesíleny vyšší frekvence signálu a naopak potlačeny nižší frekvence signálu. V posledním bloku je signál vyhlazen. Průběhy signálu za jednotlivými bloky jsou zobrazeny na obrázku 4.4.



Obrázek 4.4: Průběhy EKG signálu za jednotlivými bloky [7]

Poslední částí detekčního algoritmu je detekce špičky integrovaného signálu. Komplex QRS odpovídá vzestupné hraně integrovaného signálu. Taky doba trvání vzestupné hrany je stejná jako QRS komplex. Špička signálu je lokální maximum, kdy signál změnil svůj směr z rostoucího na klesající. [6]

Kapitola 5

Jádra ARM Cortex–M

Jednou z produktových řad společnosti Arm Ltd. Je právě řada Cortex–M, která obsahuje převážně jádra používána v mikrokontrolérech s náročnými požadavky na výkon a spotřebu elektrické energie. V tabulce 5.1 jsou uvedeny jednotlivé typy jader v produktové řadě Cortex–M. Jak lze vidět, tyto jádra jsou založena na architektuře ARMv6-M, ARMv7-M a ARMv7E-M. Dále jednotka MPU je jednotkou pro ochranu vybraných oblastí paměti. [8]

Tabulka 5.1: Přehled obsahu jednotlivých typů jader Cortex-M

Jádro	Architektura ARM	Architektura MCU	MPU
Cortex-M0	ARMv6-M	Von Neumann	ne
Cortex-M0+	ARMv6-M	Von Neumann	volitelné
Cortex-M1	ARMv6-M	Von Neumann	ne
Cortex-M3	ARMv7-M	Harvardská	volitelné
Cortex-M4	ARMv7E-M	Harvardská	volitelné
Cortex-M7	ARMv7E-M	Harvardská	volitelné

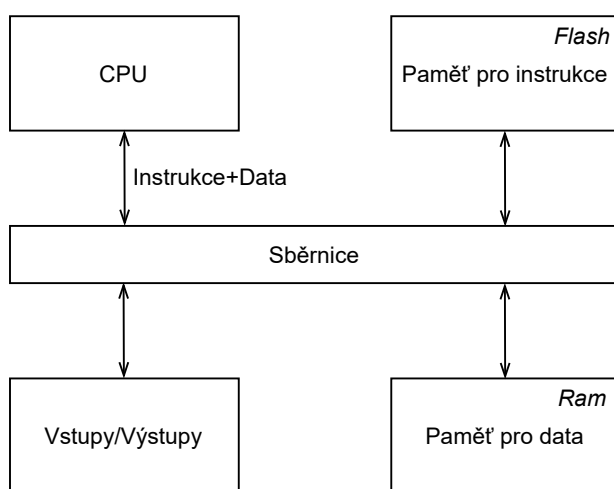
5.1 Jádro ARM Cortex–M0+

Cortex–M0+ má 32b jádro a jedná se o optimalizovanou verzi původních jader Cortex–M0. Jádra Cortex–M0+ vykazují nízkou spotřebou elektrické energie, která může dosahovat hodnot až 9,8 μ W na MHz. Mikrokontroléry s těmito jádry umožňují několik režimů s omezenou spotřebou elektrické energie. Z tohoto důvodu mají mikrokontroléry s touto architekturou velké uplatnění v zařízeních napájených z baterií. Nicméně při přechodu do některého z režimů s omezenou spotřebou elektrické energie dochází k určitým omezením. Například při režimu *sleep*, dochází ke snížení taktu mikrokontroléru na 0 Hz. Pro práci s režimy s omezenou spotřebou elektrické energie slouží instrukce WFI a WFE. Použitím instrukce WFI mikrokontrolér přestane vykonávat instrukce, ale pokud je generováno přerušení (případně i na signál od debuggeru), mikrokontrolér přejde do režimu chodu.

Instrukce WFE je v podstatě podobná s tím rozdílem, že mikrokontrolér nevykonává instrukce do doby, než vznikne událost od některého z koprocesorů, případně opět od debuggeru. [8] [9]

5.2 Architektura Von Neumann

Cortex-M0+ obsahuje architekturu MCU zvanou Von Neumann. Tato architektura je zobrazena na obrázku 5.1. Tento typ architektury využívá jednu sběrnici a adresy paměti pro instrukce a data se nepřekrývají. [10]



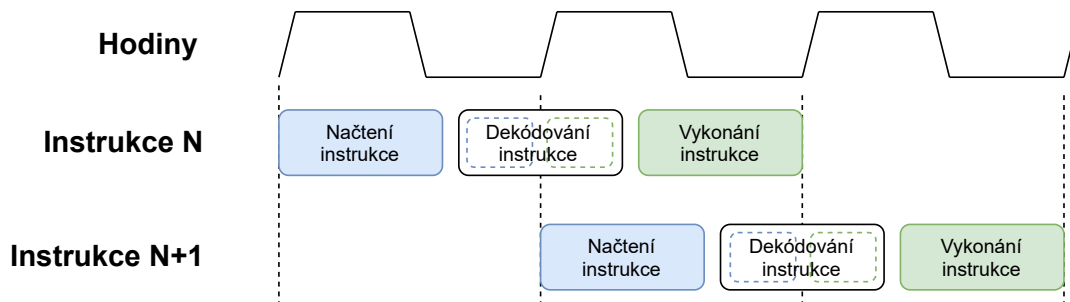
Obrázek 5.1: Von Neumann architektura

5.3 Zřetěžené zpracování instrukcí

Architektura Cortex-M0+ obsahuje pouze dvě etapy zřetěženého zpracování instrukcí. To je jeden z rozdílů od architektury Cortex-M0, která obsahuje celkem tři etapy zřetěženého zpracování instrukcí. Snížení počtu etap má samozřejmě kladný vliv na spotřebu elektrické energie díky snížení přístupu k flash paměti. K zpracování instrukce tedy dochází ve dvou etapách. Jedná se o tyto etapy:

- načtení instrukce a před dekódování,
- hlavní dekódování a vykonání instrukce.

Tyto etapy zpracování instrukce jsou zobrazeny na obrázku 5.2. [11]



Obrázek 5.2: Řetěžené zpracování instrukcí s dvěma etapami [11]

5.4 Instrukční sady

Mikrokontroléry s architekturou Cortex-M0+ obsahují instrukční sadu zvanou RISC (angl. Reduced Instruction Set Computer). Jak již z anglického názvu vyplývá, jedná se o mikrokontroléry s redukováným souborem instrukcí. To znamená, že soubor instrukcí je zúžen pouze na často využívané instrukce, přičemž délky všech instrukcí jsou totožné. Další používanou instrukční sadou je CISC (angl. Complex Instruction Set Computer), ve kterých se využívá složitý soubor instrukcí. Tato instrukční sada je využívána v mikrokontrolérech, pomocí kterých jsou prováděny složité operace. Pokud by u takových aplikací byla využita zúžená řada instrukcí, prováděné operace by se musely skládat z několika instrukcí. Naopak při použití rozšířené instrukční sady je možné využít mnohonásobně méně instrukcí. Tato skutečnost má pozitivní vliv na kompilaci programu z jazyka C, mikrokontroléry ale naopak musí obsahovat velice složitý řadič. Kvůli tomu je celý mikrokontrolér pomalejší. Tento neduh je možné odstranit právě využitím zúžené instrukční řady. Řadič takových mikrokontrolérů je konstruován jako klasický sekvenční obvod. Program, který je složen z jednoduchých instrukcí RISC, je mnohem delší než při použití rozšířené sady instrukcí. Nicméně celková doba provedení je kratší s ohledem na menší časovou náročnost použitých jednoduchých instrukcí. RISC mikrokontroléry mají tyto vlastnosti:

- jednotná délka instrukcí,
- malé množství instrukcí,
- uplatněno zřetěžené zpracování,
- load-store architektura. [12]

Dále mikrokontroléry s architekturou Cortex-M0+ nepodporují původní 32 bitovou architekturu instrukční sady ARM, ale instrukční sadu tzv. Thumb-2. Hlavní myšlenkou této instrukční sady je redukce délky často využívaných instrukcí z 32 bitů na pouhých 16 bitů. Tímto je ušetřena paměť, jelikož již v paměti neukládáme pouze 32 bitové instrukce, ale pouze 16 bitové. Tedy v instrukční sadě Thumb jsou obsaženy instrukce délky 16 bitů, ale také s délkou 32 bitů. [13] [8]

Kapitola 6

Komunikační rozhraní

Pro přenos dat mezi subsystémy slouží komunikační rozhraní. Jedná se o souhrn signálů s různými funkcemi. Podle funkce jsou tyto signály děleny:

- adresové signály,
- datové signály,
- řídicí signály.

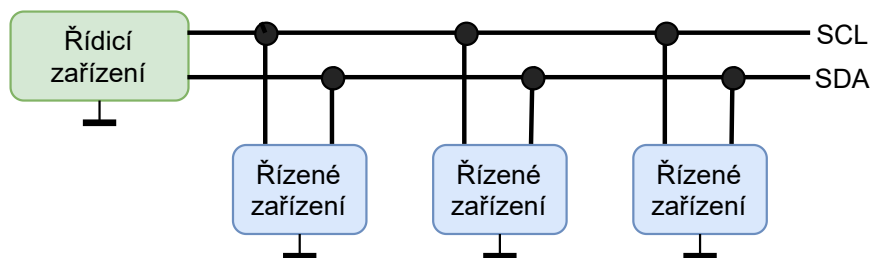
Komunikační rozhraní jsou dělena na sériová a paralelní. V současnosti jsou převážně používána sériová rozhraní, která odstraňují některé nevýhody paralelních rozhraní. U sériového rozhraní jsou adresy, data i řídicí signály přenášeny pomocí sériového protokolu. Sériové rozhraní je dále děleno podle směru přenosu dat (simplexní a duplexní) a podle typu časování (synchronní a asynchronní). [14]

V rozhraních se synchronním přenosem je obsažen jeden synchronizační signál. Pomocí toho signálu je mezi zařízeními přenášén hodinový signál. Data jsou pak přenášena s tímto hodinovým signálem. Naopak asynchronní rozhraní neobsahuje žádný synchronizační signál. Naopak jsou přesně definovány začátky a konce přenesených rámců. Dále je pro přijímací a odesílací zařízení definována stejná přenosová rychlost, se kterou jsou přenášena data. [15]

6.1 I²C

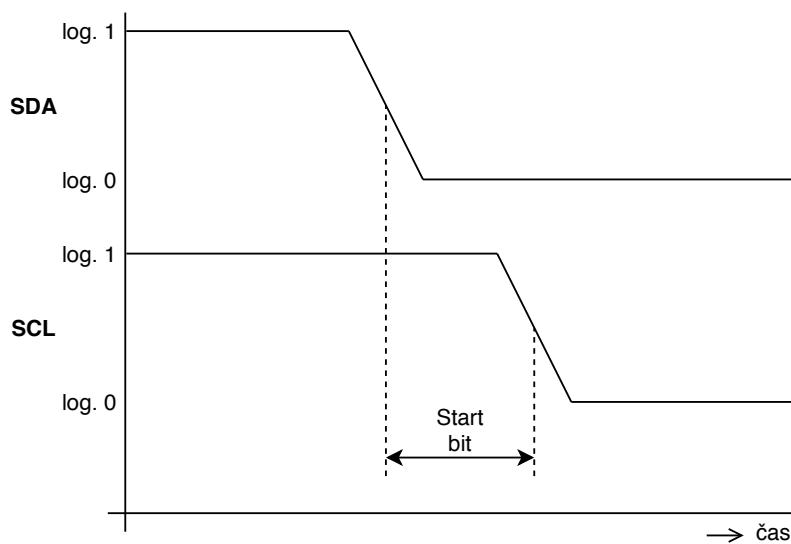
Jedná se o sériové synchronní komunikační rozhraní s poloduplexním přenosem. To znamená, že v danou chvíli je možné komunikovat pouze v jednom směru. Na obrázku 6.1 je zobrazeno vzorové zapojení komunikačních uzlů. Komunikační rozhraní se skládá ze dvou signálů:

- **SDA** - slouží pro přenos dat,
- **SCL** - slouží pro přenos hodinového signálu.



Obrázek 6.1: Demonstrativní zapojení uzlů sběrnice I²C [16]

V komunikačním rozhraní se objevují dvě role zařízení. Jednou z rolí je řídicí zařízení, pomocí kterého je řízena veškerá komunikace. V danou chvíli se na komunikačním rozhraní může nacházet pouze jedno řídicí zařízení. Tím je zamezeno vzniku kolize vysílání dvou či více řídicích zařízení najednou. Pomocí řídicího zařízení je generován hodinový signál a je inicializována komunikace. Druhým typem zařízení je tzv. řízené zařízení, které je řízeno řídicím zařízením a vykonává jeho požadavky. [16]



Obrázek 6.2: Časový průběh Start bitu [16]

Pokud na komunikačním rozhraní nedochází ke komunikaci, nachází se v tzv. klidovém stavu. V tomto stavu jsou oba signály na úrovni log. 1, což je zajištěno pull-up rezistory. Komunikace je zahájena řídicím zařízením pomocí start bitu, který je zobrazen na obrázku 6.2. Na začátku je logická úroveň SDA signálu stažena na úroveň log. 0 a logická úroveň signál SCL setrvá určitou dobu na úrovni log. 1. Start bit přijímají všechna řízená zařízení připojená na komunikační rozhraní. Po start bitu následuje sekvence bitů určující adresu řízeného zařízení, se kterým je navázaná komunikace. Délka adresy může nabývat sedmi bitů nebo deseti bitů. Za adresou řízeného

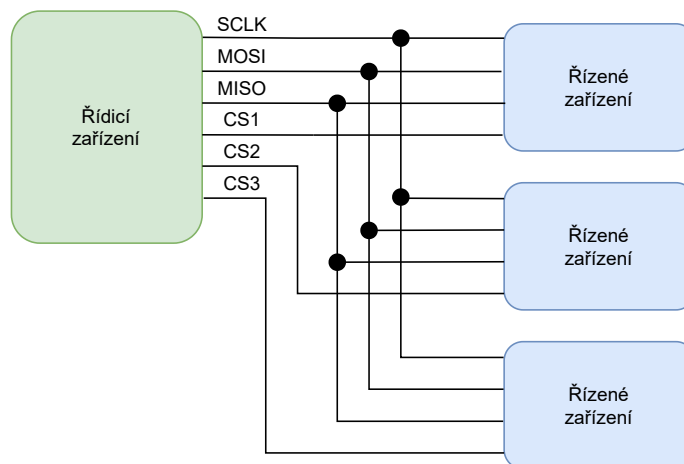
zařízení následuje bit, pomocí kterého je určeno, zda se jedná o čtení nebo zápis. Posléze následují samostatná data, která jsou potvrzována bitem ACK. Tento bit je generován jako potvrzení příjmu datového paketu. Pokud přenos proběhl v pořádku, úroveň ACK bitu je log. 0. Nicméně pokud se přenos nezdařil, tak ACK bit je na úrovni log. 1. Po přenosu veškerých dat je komunikace ukončena stop bitem, kdy probíhá změna napěťové úrovně z log. 0 na log. 1 signálu SDA zatímco napěťová úroveň signálu SCL je na úrovni log. 1.[16], [17]



Obrázek 6.3: Sekvence komunikačního protokolu I²C [16]

6.2 SPI

Dalším komunikačním rozhraním používaným v mikrokontrolérových aplikacích je rozhraní SPI. Jedná se opět o sériové synchronní rozhraní, které tvoří strukturu řídicího a řízeného zařízení. Vzorové zapojení komunikačních uzlů je zobrazeno na obrázku 6.4. [12] [18]

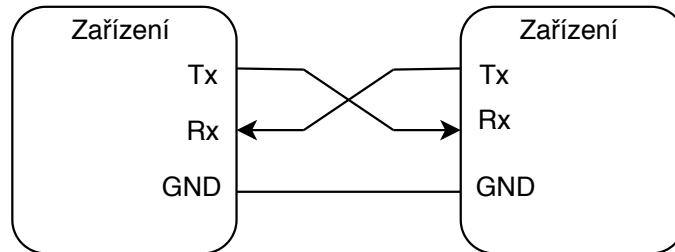


Obrázek 6.4: Demonstrativní zapojení uzlů SPI sběrnice [18]

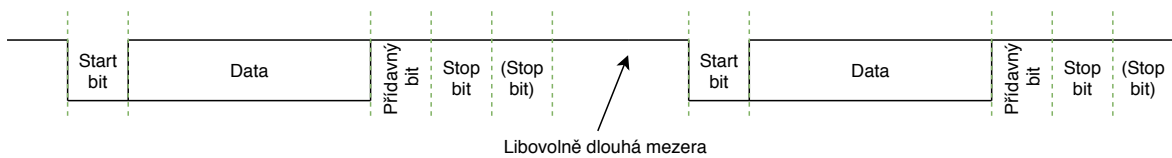
V SPI rozhraní jsou obsaženy čtyři signály. Pomocí signálu zvaného SCLK je přenášén hodinový signál, který je opět generován řídicím zařízením. Poté následují dva signály MOSI a MISO. Signálem MOSI jsou přenášena data ve směru z řídicího zařízení do řízeného zařízení. Naopak signálem MISO jsou data přenášena z řízeného zařízení do řídicího zařízení. V SPI rozhraní nejsou adresována jednotlivá řízená zařízení. Výběr řízeného zařízení je prováděno řídicím zařízením pomocí signálu CS. [12] [18]

6.3 UART

Rozhraní UART (univerzální asynchronní přijímač/vysílač) je asynchronní komunikační rozhraní umožňující duplexní přenos. Na obrázku 6.5 je zobrazeno základní tři zapojení. [12]

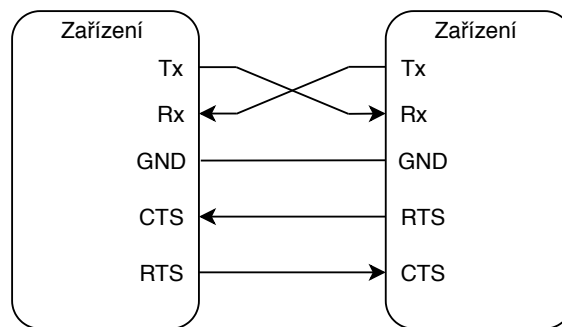


Obrázek 6.5: Spojení dvou zařízení pomocí sběrnice UART



Obrázek 6.6: Průběh přenosu dat po asynchronní sběrnici UART [12]

V klidovém stavu jsou oba signály na úrovni log. 1. Přenos dat je zahájen pomocí tzv. start bitu, kdy dochází k přechodu na úroveň log. 0. Poté následují datové pakety a za nimi volitelný přídavný bit, který může nabývat významu například paritního bit, případně může úplně chybět. Poté je přenos dat ukončen stop bitem, u kterého dochází ke změně úrovně na log. 1. To umožní bezprostřední poslání následujícího rámce, protože na začátku přenosu rámce dat dochází vždy ke změně z úrovně log. 1 na log. 0. [12]



Obrázek 6.7: Spojení dvou zařízení pomocí sběrnice UART s řídicími signály CTS a RTS [19]

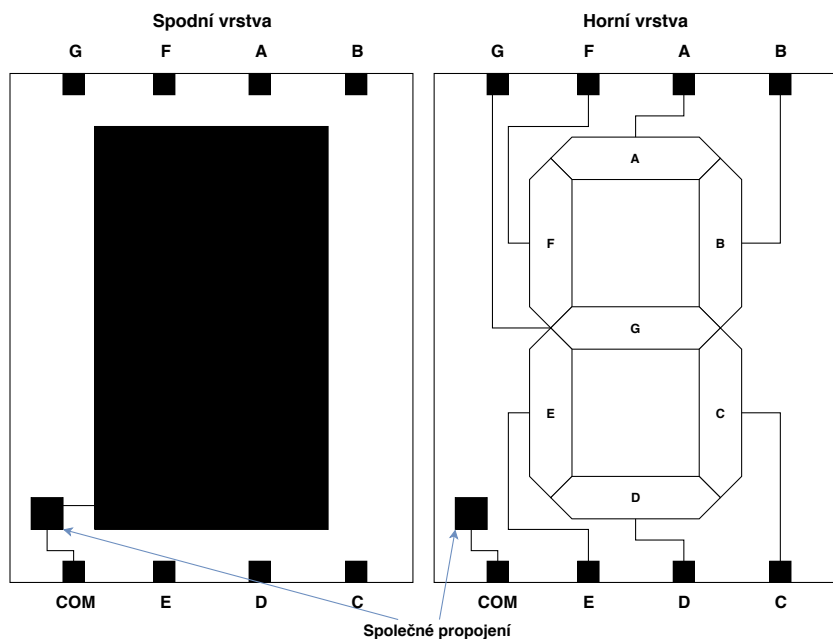
Komunikační rozhraní UART je většinou využito v rámci normy RS 232 C, kde kromě výše zmíněných signálů jsou využívány další řídicí a stavové signály. Jedny z těchto signálů jsou signály CTS a RTS. Na obrázku 6.7 je zobrazeno zapojení těchto řídicích signálů. Pin CTS na zařízení je pinem vstupním a pin RTS je naopak pinem výstupním. Pomocí těchto signálů je přijímacím zařízením podána informace vysílacímu zařízení, zda je přijímací zařízení schopné přijímat data. V případě, kdy přijímací zařízení není schopné přijímat data, je nastaven řídicí signál RTS přijímacího zařízení na úroveň log. 1. Tím je nastaven signál CTS na vysílacím zařízení a komunikace pozastavena. [12] [19]

Kapitola 7

Segmentové LCD

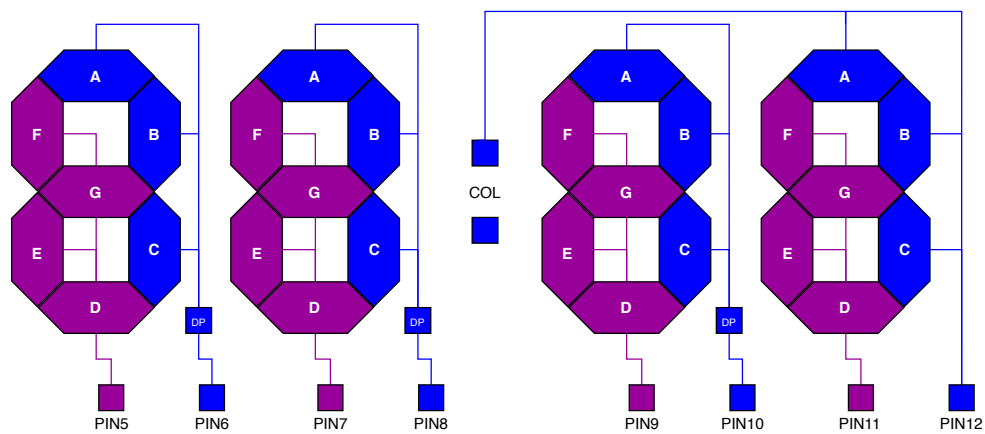
Displeje jsou určeny pro komunikaci mezi zařízením a uživatelem. Existuje několik typu displejů, ale nyní budou blíže popsány sedmisegmentové displeje s multiplexovaným řízením.

Sedmisegmentový displej je složen ze sedmi segmentů uspořádaných do číslice osm. Sedmisegmentové LCD mají dva druhy pinů. Jedny se nazývají backplane piny (jedná se o pin s názvem COM na obrázku 7.1) a druhé frontplane piny (piny A až G). Oba typy pinů jsou připojeny k odlišné straně tekutého krystalu. Zatímco backplane pin je spojen se spodní vrstvou zobrazenou na obrázku 7.1, frontplane piny jsou připojeny na jednotlivé segmenty LCD. Toto zapojení platí pro tzv. staticky řízené LCD, které mají obrovskou nevýhodu a to počet řídicích pinů. Z těchto důvodů byly sestrojeny sedmisegmentové LCD s multiplexovaným řízením. [20]

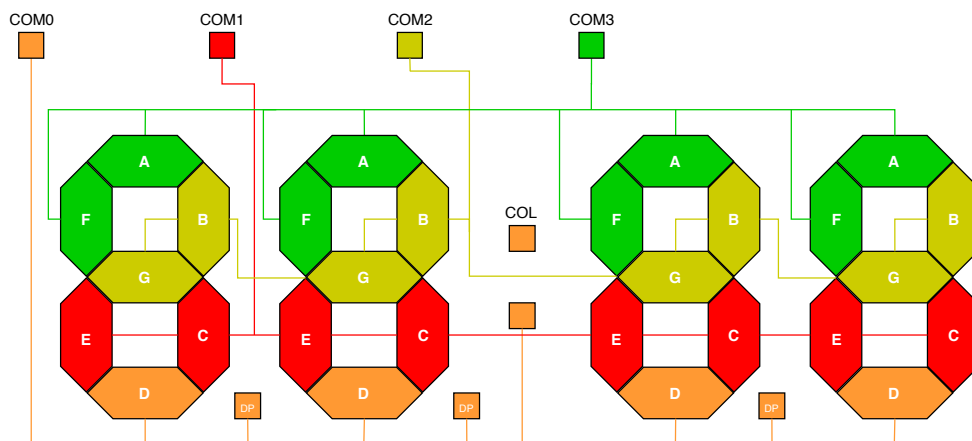


Obrázek 7.1: Sedmisegment propojení pinů [20]

Displeje s multiplexovaným řízením je možné vidět na obrázcích 7.2 (zde je zobrazeno propojení pinu frontplane) a na obrázku 7.3 (zobrazeno propojení pinů backplane). Nyní už neplatí to, že každý frontplane pin je připojen pouze k jednomu segmentu, ale pomocí jednoho frontplane pinu je možné řídit více segmentů. V uvedeném příkladě pomocí jednoho frontplane pinu je možné řídit až čtyři segmenty. Dále pak jsou obsaženy celkem čtyři backplane piny, kdy každý backplane pin je připojen k odlišným segmentům připojeným na jeden frontplane pin. Funkce multiplexovaného LCD je založena na rychlém přepínání jednotlivých kombinací frontplane a backplane pinů. Díky této skutečnosti může být využito méně pinů, ale vzrůstá zde náročnost řízení těchto displejů. [21]



Obrázek 7.2: Multiplexovaný LCD se čtyřmi digity zapojení Frontplane



Obrázek 7.3: Multiplexovaný LCD se čtyřmi digity zapojení Backplane

Kapitola 8

Bluetooth Low Energy

V poslední době jsou stále častěji využívána bateriová zařízení s bezdrátovou technologií pro komunikaci s ostatními zařízeními. Tímto jsou ale kladeny nové nároky na bezdrátovou technologii a to právě v oblasti energetické náročnosti dané technologie. Existuje několik technologií jako například Ant+, Bluetooth, ZigBee atd. V práci bylo využita technologie Bluetooth, respektive Bluetooth 4.0 Low Energy (dále již BLE). [22]

Pro komunikaci s chytrými mobilními telefony je využíváno komunikačního standartu BLE. Oproti klasickému Bluetooth (nižší než Bluetooth 4.0) má BLE nižší spotřebu energie, tím pádem je možné využívat tuto technologii v zařízeních napájených například knoflíkovými bateriemi. Nízká spotřeba je zajištěna tím, že komunikační moduly jsou po většinu doby udržovány v úsporném režimu. V tomto režimu nejsou posílána ani přijímána data. Aby mohla být data posílána, musí komunikační modul opustit úsporný režim a přejít do režimu chodu. Režim chodu musí trvat pouze několik milisekund, jelikož proudový odběr v tomto režimu stoupá na desítky miliampér (obvykle 15 mA). BLE je sice méně náročnější na spotřebu elektrické energie oproti klasickému Bluetooth, ale s tím přicházejí určitá omezení. Například je možné přenášet menší objemy dat než u klasického Bluetooth. Pro porovnání klasického Bluetooth a BLE je uvedena tabulka 8.1. [22]

Tabulka 8.1: Rozdíl mezi klasickým Bluetooth a BLE

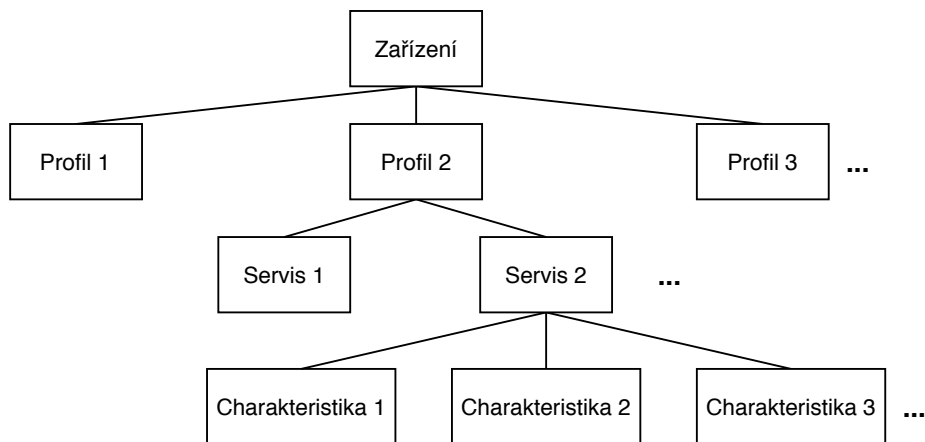
	Klasický Bluetooth	BLE
Dosah	100 m	50 m
Datová propustnost kanálu	1 až 3 Mb/s	1 Mb/s
Skutečná přenosová rychlost	0,7 až 2,1 Mb/s	0,27 Mb/s
Počet připojených zařízení (slave)	max. 7	ideálně neomezeno
Přenos hlasu	ano	ne
Topologie sítě	Scatternet	hvězda
Maximální proudový odběr	30 mA	15 mA

Standart BLE obsahuje sadu profilů, založených na tzv. GATT. Tyto profily určují chování Bluetooth zařízení při komunikaci s ostatními prvky sítě. Pomocí GATT specifikace jsou definovány atributy, skupiny služeb, deklarace a popisy zařízení. Další části definující komunikaci mezi zařízeními je tzv. GAP. Zde je definován přehled zařízení, spojení a vazeb. Oba tyto profily budou dále popsány v dalších podkapitolách. [22]

8.1 GATT - Generic Attribute Profile

GATT je založen na tzv. ATT protokolu, který definuje, jakým způsobem bude docházet k výměně dat mezi zařazeními. Na základě této skutečnosti jsou pro komunikaci definovány dvě role. Jedná se o roli serveru a klienta. Roli serveru zastupují zařízení, která obsahují zmíněný ATT protokol a umožňují přístup k datům připojenému klientovi. Klient může data číst a zapisovat do příslušných atributů. Pomocí serveru mohou být zaslány notifikace klientovi o změně v daných attributech. Jako příklad rolí serveru a klienta může být chytrý telefon a chytrý náramek. V tomto případě chytrý telefon zastává roli klienta a chytrý náramek roli serveru. [23]

GATT definuje koncept tzv. profilů, servisů a charakteristik, který je znázorněn na obrázku 8.1. Každý profil může obsahovat několik servisů a každý servis může obsahovat několik charakteristik. Na oficiálních stránkách Bluetooth SIG (Special Interest Group) je uveden seznam předdefinovaných servisů a charakteristik, které může programátor při tvorbě firmware využít. Každý servis je definován pomocí tzv. UUID. Pokud programátor nevyužije žádný z předdefinovaných servisů, může si vytvořit své vlastní. Stěžejní částí při vytváření vlastního servisu je právě definování UUID. K tomu jsou určeny UUID generátory, které vygenerují 128 bitové číslo. [23]



Obrázek 8.1: Vztahy mezi profily, servisy a charakteristikami

8.2 GAP - Generic Access Profile

Pomocí specifikace GAP je určeno, jakým způsobem se má zařízení chovat vůči vnějššímu prostředí. Například jeho viditelnost, jak mají být data odesílána atd. Pomocí BLE zařízení je možné komunikovat dvěma způsoby. Jedná se o komunikace vysíláním nebo připojením. Při způsobu vysílání se pro přenos dat nemusí explicitně vytvářet spojení. Naopak tomu je u způsobu připojení, u kterého musí být mezi zařízeními navázané spojení. [23] [24] [25]

Pro komunikaci vysíláním, je možné zařízení dělit na dvě role:

- **Hlasatel (angl. Broadcaster)** - Zde jsou vysílány veřejné advertising pakety.
- **Pozorovatel (angl. Observer)** - Tímto zařízením jsou přijímány advertising pakety, které jsou vysílány Broadcastem.

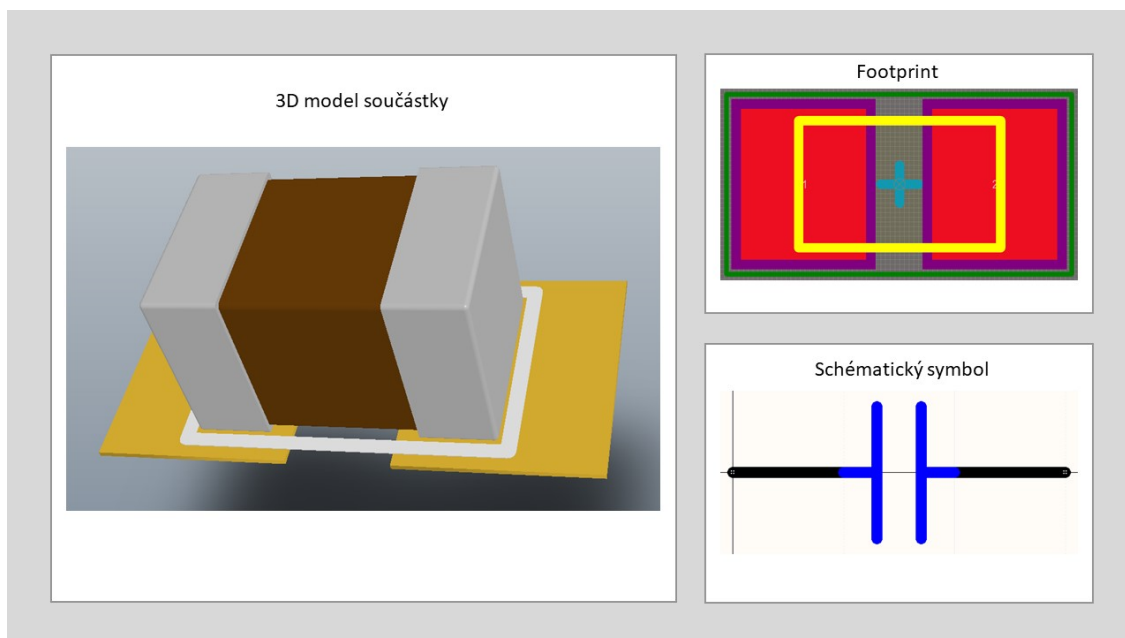
Podle způsobu připojení, je možné zařízení dělit na dvě role:

- **Periferní zařízení** - Zařízením, které zastává tuto roli, jsou inzerovány informace o jejich přítomnosti v okolí (pomocí advertising paketů). Díky této skutečnosti může být poté centrálním zařízením navázáno spojení s periferními zařízeními. Poté co jsou zařízení spojeny, periferním zařízením již nejsou dále inzerovány informace o jeho přítomnosti okolí.
- **Centrální zařízení** - Pomocí centrálního zařízení je zahájeno připojení k perifernímu zařízení sledováním advertising paketů. Centrální zařízení může být připojeno k více periferním zařízením. V případě, že centrální zařízení má požadavek na spojení se s periferním zařízením, je centrálním zařízením vyslán datový paket s požadavky na připojení k perifernímu zařízení. Pokud periferním zařízením je přijat požadavek, je spojení navázáno. [23] [24] [25]

Kapitola 9

Návrh DPS pomocí Altium Designer

EDA programy pro tvorbu desek plošných spojů jsou v poslední době stále zdokonalovány a vylepšovány. Jedním z představitelů je právě Altium Designer, který nabízí několik možností. Ať už návrh desek plošných spojů ve třech rozměrech, ale také návrh programů pro obecné CPU, konfigurace FPGA a CPLD obvodů. [26]



Obrázek 9.1: Ukázka návrhu součástky pro SMD kondenzátor

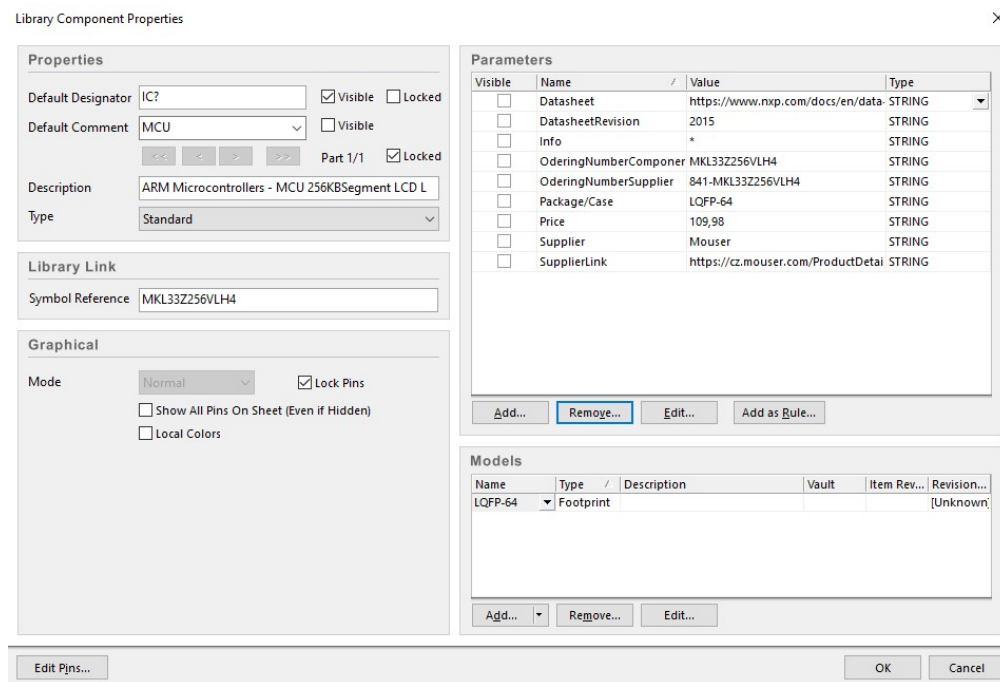
Před tvorbou samostatného elektrického schématu zařízení je nutné vytvořit knihovnu všech použitých součástek. Na obrázku 9.1 je uvedena ukázka pro součástku kondenzátoru v SMD pouzdře. Tvorba knihovny je rozdělena na dvě části. V první části se jedná o tvorbu footprintu a v druhé části o schematické značky dané součástky. Tyto části jsou rozděleny do dvou odlišných souborů. Pro footprinty má soubor příponu *.PcbLib* a přípona souboru pro schematické značky je *.SchLib*.

Jak již bylo dříve zmíněno Altium Designer podporuje návrh desek plošného spoje ve třech rozměrech. Z toho důvodu již při tvorbě knihovny musí být jednotlivým komponentům přiřazeny nebo vytvořeny 3D objekty. K tomu slouží nástroj *3D Body*, který umožní vytvořit nejen základní jednoduché objekty jako koule, válec atd., ale umožní také vložení souboru s příponou *.step*. Stejně jako v jiných návrhových programech, tak i v Altium Designeru se pracuje s tzv. vrstvami. Každá vrstva má definováno, jaké objekty bude obsahovat. Existují vrstvy, například pro vrchní motiv plošného spoje, spodní motiv plošného spoje, vrchní motiv nepájivé masky atd. Dále jsou zde obsaženy tzv. mechanické vrstvy, kdy jejich využití určuje sám návrhář. Nicméně například mechanická vrstva osm je ve většině případů použita pro 3D objekty. Při tvorbě knihovny lze využít komplexní nástroj pro tvorbu footprintu dané komponenty. Pomocí tohoto nástroje lze jednoduše vytvářet footprinty pro základní typy pouzder a jejich 3D modely. Po vytvoření footprintu je dále nutné vytvořit schématický symbol k dané součástce. Pro tvorbu schématických značek v EDA programu existuje mnoho nástrojů. Jednou z hlavních částí je objekt pin. V nastavení objektu je možné zadat označení daného pinu, pomocí kterého jsou pak dané piny schématické značky spojovány s pájecími ploškami footprintu. Dále je možné si zvolit elektrický typ pinu, zda se jedná o výstupní, napájecí, vstupně výstupní, s otevřeným kolektorem, pasivní atd. Toto nastavení pinu je pak bráno v potaz při kontrole elektrické správnosti navrženého schématu. Samozřejmě pro návrh schématické značky je možné využít hned několik základních grafických objektů, jako například obdélník, kružnici, elipsu, čáru a mnoho dalších. Dále je možné každé vytvořené komponentě přidat určité vlastnosti, které lze definovat v okně *Library Component Properties*. Okno *Library Component Properties* lze otevřít dvojklikem na daný schématický symbol, to je zobrazeno na obrázku 9.2. Zde samozřejmě, kromě definování názvu, označení, jde také definovat typ součástky, přidávat různé parametry a připojit ke schématické značce daný footprint. Dále je volen typ součástky, kdy má návrhář výběr z těchto šesti typů:

- **Standart:** Jedná se o běžnou součástku, přičemž je tato součástka uvedena v seznamu součástek.
- **Standart (No BOM):** Totožné jako typ Standart, až na to, že součástka není uvedena v seznamu součástek.
- **Mechanical:** Jedná se o mechanické části, které mohou i nemusí být zastoupeny na DPS. Může se jednat o šrouby, chladiče atd. Tyto komponenty jsou uvedeny v seznamu součástek.
- **Graphical:** Jedná se pouze o grafický objekt, který není obsažen v seznamu součástek.
- **Net Tie:** Jedná se o komponentu, která je tvořena motivem plošného spoje. Tyto komponenty nejsou uváděny v seznamu součástek. Může se jednat o antény, zkratovací propojky, proudové bočníky atd.

- **Net Tie (In BOM):** Totožné jako typ Net Tie, s tím rozdílem, že nyní jsou tyto komponenty uvedeny v seznamu materiálů.

Parametry si může návrhář vytvořit dle své potřeby. U každého parametru lze nastavit, zda bude viditelný, jeho jméno, hodnotu a datový typ hodnoty. Častým parametrem je objednávací číslo, odkaz na datasheet, jméno dodavatele, typ pouzdra atd. [27]



Obrázek 9.2: Okno *Library Component Properties*

Po zhotovení knihovny se již přistupuje k tvorbě schématu a DPS. Schéma a DPS jsou opět rozděleny do dvou souborů s jinou příponou. Pro soubor obsahující schéma je využívána přípona *.SchDoc* a pro DPS *.PcbDoc*. Pro vytvoření elektrického zapojení jsou používány komponenty z dostupných knihoven. K elektrickému spojení dvou komponent slouží nástroj *Wire*. Při spojování je možné pomocí kombinace Shift + mezerník volit mód propojení (přímé spojení, pravoúhlé či zakulacené). Při tvorbě elektrických schémat je důležité jednotlivé vodiče pojmenovat. Díky tomuto pojmenování nemusí být piny součástek na přímo spojeny, ale stačí jim pouze přiřadit stejné jméno. K tomu slouží nástroj *Place Net Label*. Každý prvek vložený do schématu má unikátní označení, které se skládá z písmene a číslice. Například kondenzátor umístěný ve schématu může mít označení C20. Samozřejmě že návrhář nemusí toto označení tvořit sám, ale může využít implementovaný nástroj v EDA programu. Pouze je důležité, aby do schématu vkládal komponenty s označením např. C?, kdy místo otazníku bude doplněno číslo. K tomuto slouží nástroj zvaný *Annotate Schematics*. [28]

Následně po vytvoření schématu může být přistoupeno k tvorbě motivu plošného spoje. V první řadě je nutné určit rozložení vrstev desky plošného spoje. K tomu slouží nástroj *Layer Stack Ma-*

nager, ve kterém je možné přidávat či ubírat vrstvy, určovat jejich velikost, volit materiál apod. Při nastavování je zobrazen interaktivní obrázek, který zobrazuje náhled DPS. Jednotlivé vrstvy si může návrhář pojmenovat. Nejdůležitější je ale nastavení návrhových pravidel. Zde jsou definovány jednotlivé parametry pro návrh motivu plošného spoje a rozložení součástek, jako je například minimální a maximální šířka vodivé cesty, minimální a maximální velikost průměru prokovených otvorů, izolační mezery mezi jednotlivými komponenty, atd. Po definování vrstev plošného spoje a nastavení pravidel, které korespondují s technickými možnostmi pro výrobu, může být přistoupeno k tvorbě motivu plošného spoje. Pro tvorbu motivu plošného spoje nabízí návrhový program řadu nástrojů. Například pro vodivé spojení se využívá nástroj zvaný *Interactively Route Connections*, pro vkládání prokovených otvorů nástroj *Place via*, pro vytváření polygonů *Polygon Pour* a mnoho dalších. Dále je možné importovat změny ze schématu elektrického zapojení výběrem kolonky *Design → Import Changes From* a s názvem projekt, ve kterém je dané schématické zapojení. Po dokončení motivu plošného spoje je nutné vygenerovat výrobní data. V Altium Designeru je implementováno několik nástrojů, pomocí kterých lze vygenerovat GERBER data, souřadnice umístěných komponent, souřadnice vrtaných prokovených otvorů apod. Tyto nástroje jsou v nabídce *File → Fabrications Outputs*. [28]

Kapitola 10

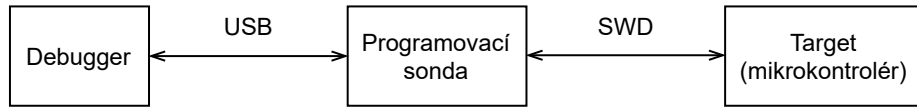
Nástroje pro tvorbu programu

Podstatnou částí pro vývoj zařízení je návrh programového vybavení pro zvolený mikrokontrolér. Pro návrh programového vybavení je potřeba sada prostředků pro tvorbu, kompilaci programu a nástroje pro debug vytvořeného programu. Jedním ze zástupců návrhových programů je Atollic® TrueSTUDIO® (dále jen Atollic). V práci byla využívána verze 8.1.0. Atollic je založen na otevřeném standartu ECLIPSE a GNU. Atollic poskytuje návrhářům řadu profesionálních nástrojů nejen pro tvorbu programového vybavení, ale také pro debug napsaného programu pomocí GNU projekt debugger. [29]

Eclipse je otevřené vývojové prostředí. Projekt Eclipse byl vytvořen v roce 2001 společností IBM. Dále pak nadace Eclipse byla již založena v lednu 2004 jako nezávislá nezisková společnost. Vývojové prostředí disponuje hlavně přidávanými pluginy, pomocí kterých je možné využívat více programových funkcí, než je uvedeno v základu.[30]

Součástí Atollic studia je tzv. GCC (GNU Compiler Collection). Jedná se o nástroje, které obsahují překladače pro několik programovacích jazyků jako jsou jazyky C, C++, Java apod. Podstatnou částí GCC je stejnojmenný GNU C Compiler. Pomocí tohoto nástroje jsou všechny c soubory překompilovány do objektových souborů. Dále je součástí návrhového prostředí GNU Binutils. Jedná se o kolekci nástrojů, kde hlavní součástí jsou nástroje **ld** (GNU linker) a **as** (GNU assembler). Pomocí nástroje GNU assembler jsou všechny assemblerské soubory převedeny do objektových souborů. Následně pak pomocí nástroje GNU linkeru dojde k sestavení souboru .elf z objektových souborů. Dalším nástrojem GNU Binutils je tzv. **objcopy**, pomocí kterého je umožněn převod souboru typu .elf na formát, který již lze nahrát do mikrokontrolérů. Jedná se například o soubory typu .bin a .hex.

Jak již bylo zmíněno, Atollic sice obsahuje GNU projekt debugger, ale v práci byl využit produkt firmy SEGGER Microcontroller. Produkt se nazývá Ozone a v práci byla využita verze 2.70. Dále byla od stejné firmy použita programovací sonda J-Link ve verzi 9.3, který mimo jiné obsahuje standart SWD. Propojení programovací sondy s debuggerem a mikrokontrolérem je zobrazeno na obrázku 10.1. Standart SWD je alternativou k JTAG standartu. Ve standartu SWD je využí-



Obrázek 10.1: Propojení programovací sondy s debuggerem a mikrokontrolérem

váno sběrnice DAP (Debug Access Port). Na této sběrnici mohou být připojeny dvě hlavní jednotky. Jedná se o port ladění (DP - Debug Port) a přístupový port (AP - Access Port). Tento standart obsahuje pouze dva signály, které jsou označovány:[31]

- **SWDCLK** - Pomocí tohoto signálu je přenášen hodinový signál. Výběr frekvence hodinového signálu je určen pomocí hostitelského rozhraní.
- **SWDIO** - Slouží k oboustrannému přenosu dat.

Kapitola 11

Zařízení Testbed

Předmětem práce je návrh elektroniky a programového vybavení pro bateriové zařízení Testbed, které bude využito ve výrobní lince umístěné v budově CPIT TL3. Výrobní linka bude obsahovat mj. testovací stanici, v níž bude probíhat testování funkcionality zařízení. Jedná se o zařízení Testbed, které má trojí funkcionalitu: měření tepové frekvence, krokoměr a teploměr. Zařízení může být vyrobeno v různých variantách. Bylo rozhodnuto, že se bude jednat o tyto varianty:

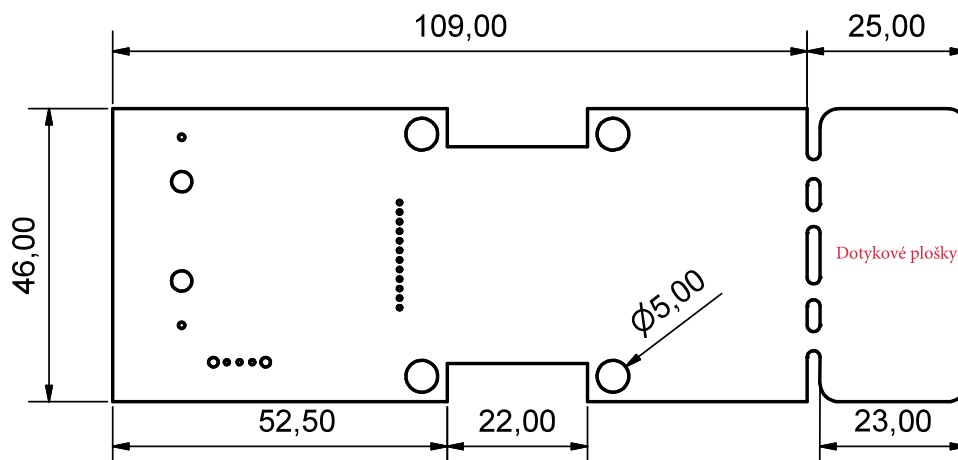
- dvě rozměrové varianty
 - s EKG ploškami
 - bez EKG plošek
- tři varianty osazení:
 - Plná verze (je osazen obvod pro měření tepové frekvence, krokoměr a teploměr)
 - Krokoměr (není osazen integrovaný obvod pro teploměr a obvod pro měření tepové frekvence)
 - Teploměr (není osazen integrovaný obvod pro krokoměr a obvod pro měření tepové frekvence)

Dále bylo rozhodnuto, že naměřená data zařízením budou posílána pomocí BLE komunikace do chytrého telefonu. Pro tuto komunikaci byly určeny vlastní servery a charakteristiky protokolu GATT. Dále bylo rozhodnuto, že další možností komunikace se zařízením je pomocí komunikačního rozhraní UART, která je využita při testování v testovací stanici.

11.1 Mechanický návrh řešení

Zařízení může být vyráběno ve dvou rozměrových variantách. V první variantě jsou obsaženy EKG plošky, které jsou zobrazeny na obrázku 11.1. V druhé variantě jsou EKG plošky odstra-

něny. Rozměry byly určeny zadavatelem na 134 mm na šířku a 46 mm na výšku. EKG plošky jsou upevněny se zbytkem DPS pomocí čtyřech můstků.



Obrázek 11.1: Rozměry DPS

Mimo rozměry desky nesmí přesahovat žádné součástky. Dle požadavků návrháře pouzdra, bylo rozhodnuto, že pouzdro pro baterii a přepínač bude umístěn na spodní stranu DPS. Uprostřed desky je na obou stranách výkroj, který slouží k uchopení desky v lince pomocí robotického ramena. Dále po stranách těchto výkrojů jsou celkem čtyři díry použité k upevnění DPS v obalu případně v Lego domku, který je sestavován výrobní linkou.

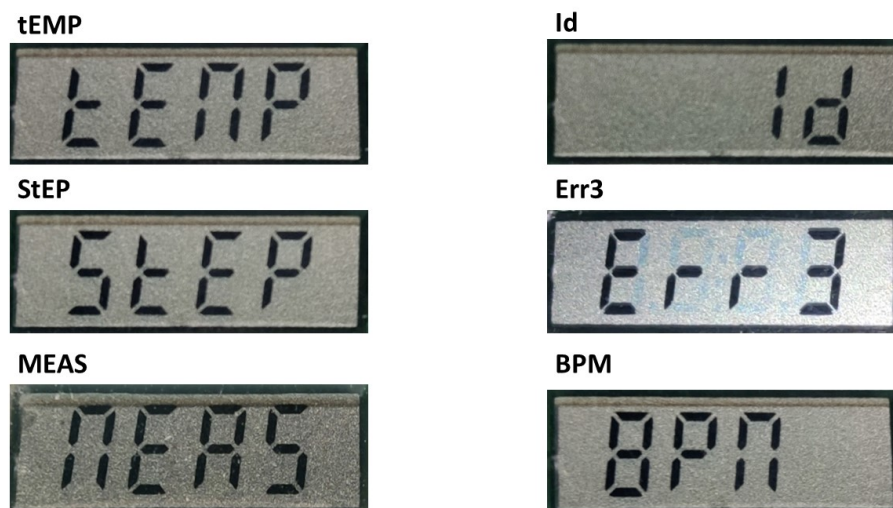
11.2 Návrh funkce programového vybavení

Po zapnutí zařízení je na displeji zobrazeno čtyřmístné unikátní identifikační číslo (dále již hexID) na deset sekund. Po uplynutí času je místo hodnoty HexID zobrazen text **Id**, který je na obrázku 11.2.

Pokud nedojde k žádné akci (ze strany uživatele), text **Id** bude trvale zobrazován. Jedná se o jednu z položek výběrového menu. Pomocí levého tlačítka zařízení je uživateli umožněno přepínat jednotlivé položky výběrového menu. Stisknutím pravého tlačítka je možné danou položku zvolit. V cyklickém menu jsou obsaženy tyto položky:

- Teploměr (na displeji **tEMP**),
- Krokoměr (na displeji **StEP**),
- Měření tepové frekvence (na displeji **BPM**),
- hexID (na displeji **Id**).

Jednotlivé položky menu jsou zobrazeny na obrázku 11.2. Počet položek cyklického menu je závislý na verzi osazení DPS. Pokud např. není osazený integrovaný obvod pro měření teploty, tak ve výběrovém menu není zobrazena položka s nápisem **tEMP**. Pomocí programového vybavení je automaticky detekováno, které obvody jsou dostupné na daném zařízení. Je tedy využita jedna verze programového vybavení pro všechny verze osazení.



Obrázek 11.2: Nápisý na LCD

11.2.1 Teploměr

Pokud je na displeji zobrazen nápis **tEMP**, tak po stisknutí pravého tlačítka je zobrazena aktuální naměřená teplota. Teplota je aktualizována každých deset sekund v případě, že došlo ke změně od poslední naměřené hodnoty. Měření teploty probíhá, i když není v cyklickém menu zvolena položka s nápisem **tEMP**.

Když je na displeji zobrazena hodnota teploty, pravé tlačítko pak slouží k okamžité aktualizaci teploty. Hodnota vyvolaná okamžitou aktualizací ale není přenesena přes bezdrátovou komunikace BLE.

11.2.2 Krokoměr

Pokud je na displeji zobrazen text **StEP**, tak po stisknutí pravého tlačítka bude zobrazen počet vykonaných kroků. Kroky budou na displeji aktualizovány minimálně po dvanácti krocích. Měření počtu kroků probíhá, i když není v cyklickém menu zvolena položka s nápisem **StEP**.

Ve stavu, kdy je na displeji zobrazen počet vykonaných kroků, lze pomocí pravé tlačítka nulovat počet kroků. Pro nulování kroků je nutné, aby bylo pravé tlačítko třikrát po sobě rychle stisknuto. Tři kliky musí být provedeny do 500 ms.

11.2.3 Měření tepové frekvence

Pokud je na displeji zobrazeno **BPM**, tak po stisknutí pravého tlačítka je zobrazena poslední naměřená hodnota tepové frekvence. Pokud by do té doby nebyla naměřena žádná tepová frekvence (např. zařízení bylo zapnuto a ještě neproběhlo první měření), bude zobrazena hodnota nula. Nyní při stisku pravého tlačítka je zobrazen nápis **MEAS** a tím je spuštěn algoritmus pro měření tepové frekvence. Měření nebude prováděno periodicky, ale na vyžádání uživatelem. Po spuštění měření musí být EKG plošky uchopeny do levé a pravé ruky. Poté na displeji bude blikat nápis **MEAS**. Interval blikání nápisu **MEAS** je totožný s R to R intervalem. Pokud dojde k úspěšnému dokončení měření, je na displeji zobrazena hodnota změřené tepové frekvence a měření je ukončeno. Pokud by došlo k neúspěšnému měření, bude na displeji zobrazena hláška **Err3**. Jako neúspěšné měření jsou brány tyto případy:

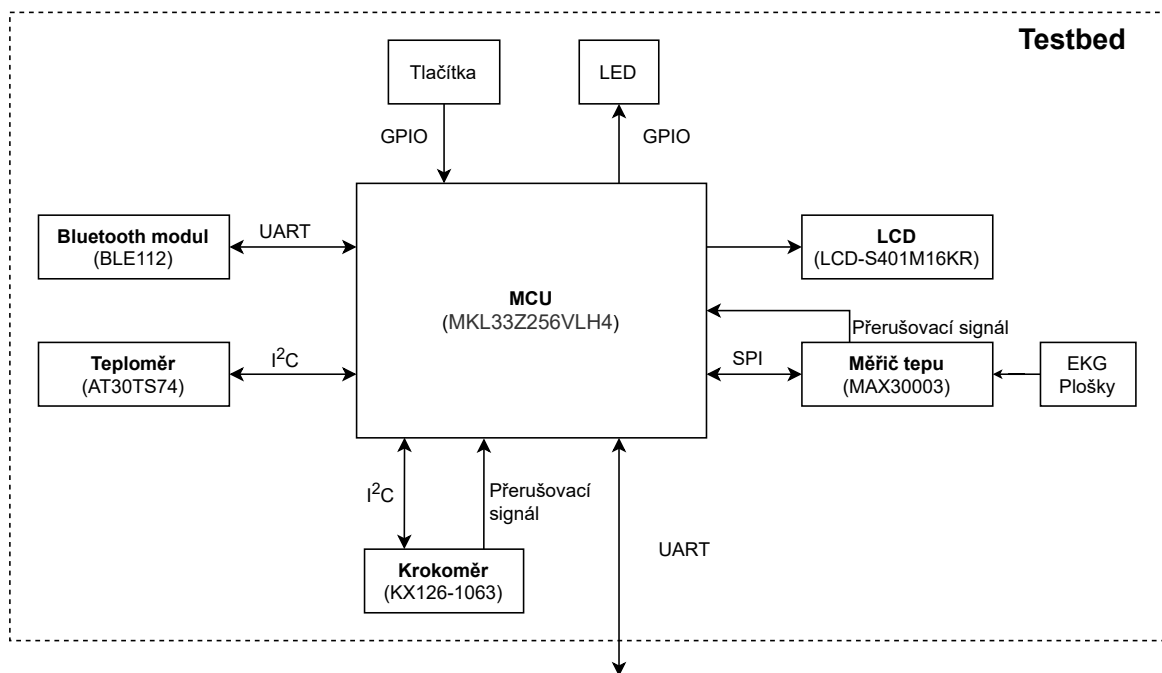
- Pokud nedojde v intervalu patnácti sekund od stisku pravého tlačítka (resp. spuštění měřícího algoritmu) k úspěšnému měření.
- Pokud změřená tepová frekvence nebude ustálena do sedmi sérií měření (v jedné sérii měření jsou tři R to R intervaly), pak je to měření bráno jako chybné.

Kapitola 12

Návrh elektroniky

Dle zadaných požadavků, ať už rozměrových nebo funkčních, byl proveden průzkum trhu. Při výběru komponent byly brány v úvahu převážně tyto parametry:

- nízká energetická náročnost komponenty,
- cena,
- součástky určené pro povrchovou montáž.



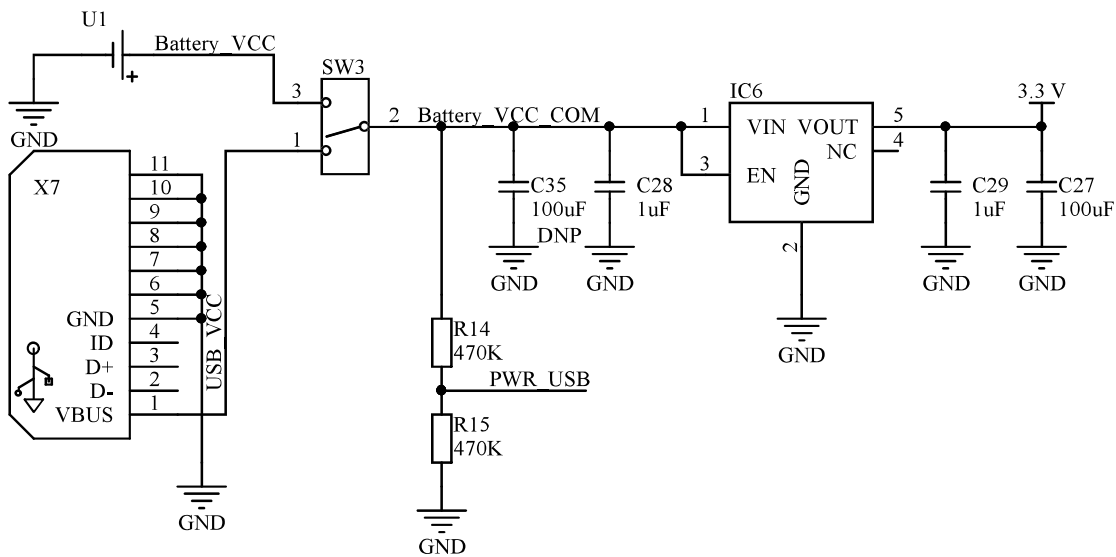
Obrázek 12.1: Blokové schéma elektroniky Testbed

V souladu s těmito parametry byly vybrány komponenty, které jsou popsány v následujících kapitolách. Na obrázku 12.1 je zobrazeno blokové schéma elektroniky. Po výběru komponent a následného návrhu elektrického zapojení bylo přistoupeno k návrhu motivu plošného spoje.

12.1 Zdroj

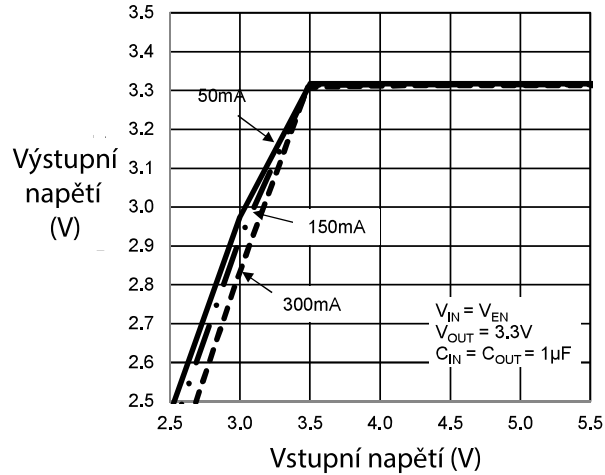
Zařízení Testbet je napájeno z 3,6 V lithiové baterie typu 1/2AA. Kapacita baterie je 1200 mAh. Druhou možností napájení zařízení je pomocí USB konektoru. USB je využito pouze k napájení, nikoli ke komunikaci se zařízením. Pro volbu typu napájení slouží přepínač na straně desky.

Napájecí hladina pro zařízení byla zvolena na 3,3 V vzhledem k napětí na baterii. Jako minimální hodnota napětí byla určena hladina 2,5 V. Při této hladině napětí je zařízení stále funkční. Vstupní napětí se tedy může pohybovat v rozmezí 2,5 V až 5,25 V. Schéma zdroje je zobrazeno na obrázku 12.2.



Obrázek 12.2: Schéma zdroje

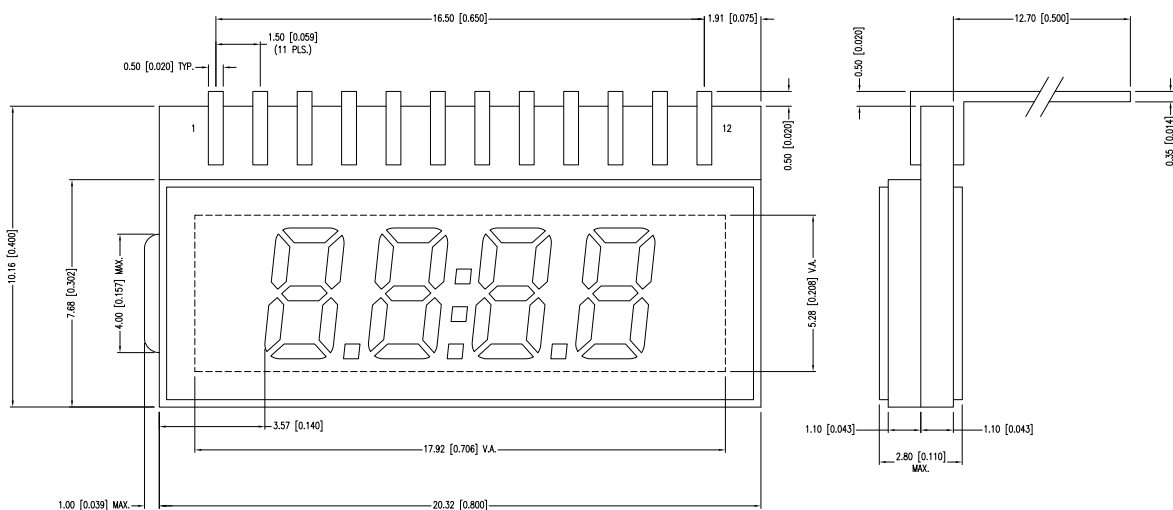
Pro změnu napětí z napěťové hladiny baterie (případně napěťové hladiny USB rozhraní) na pracovní hladinu napětí zařízení bylo využito LDO regulátoru. Jedná se o typ s označením MIC5504-3.3YM5-TR od výrobce Microchip Technology/Micrel. Vstupní napětí se může pohybovat od 2,5 V do 5,5 V, kdy napětí 6 V je určeno jako absolutní maximum. Po překročení napětí 6 V může dojít k destrukci regulátoru. Povolený výstupní proud je dle katalogu výrobce definován na 300 mA, což je v tomto případě dostačující. Výstupní napětí dosahuje maximálně 3,3 V. Pokud je ale vstupní napětí LDO regulátoru menší než 3,3 V, dochází k snížení napětí na výstupu LDO regulátoru podle následující křivky zobrazené na obrázku 12.3. LDO regulátor je vyráběn ve dvou pouzdrech. V této realizaci byla využita varianta SOT23 s 5 piny. [32]



Obrázek 12.3: Graf výstupního napětí LDO regulátoru v závislosti na vstupním napětí [32]

12.2 Displej

Výběr displeje byl velice limitován omezeními, ať už rozměrovými, nebo s ohledem na minimalizaci spotřeby elektrické energie. Na základě těchto důvodů byl zvolen sedmissegmentový displej bez podsvícení a s čtyřmi segmenty od výrobce Lumex s označením LCD-S401M16KF. Rozměry zvoleného LCD je možno vidět na obrázku 12.4. Je zřejmé, že tento typ displeje přinesl omezení v oblasti zobrazení textu a maximální zobrazované číselné hodnoty. Bylo nutné s těmito omezeními počítat při tvorbě programového vybavení.



Obrázek 12.4: Rozměry LCD v mm[inch] [33]

Dále se jedná o multiplexovaný displej. Výhodou těchto displejů je menší počet ovládacích pinů.

V tabulce na obrázku 12.1 jsou uvedeny jednotlivé piny se segmenty, které jsou připojeny při daném aktivním backplane vývodu.

Pod displej byla vložena 2 mm distance, díky čemuž byl usnadněn návrh krabičky pro zařízení. Displej bez této distance byl v průhledovém okénku krabičky výrazně zapuštěn. Distance byla provedena oboustrannou pěnovou lepicí páskou s požadovanou tloušťkou 2 mm.

Tabulka 12.1: Tabulka pinů LCD-S401M16KF [33]

Pin	1	2	3	4	5	6	7	8	9	10	11	12
COM0	COM0	/	/	/	1D	DP1	2D	DP2	3D	DP3	4D	COL
COM1	/	COM1	/	/	1E	1C	2E	2C	3E	3C	4E	4C
COM2	/	/	COM2	/	1G	1B	2G	2B	3G	3B	4G	4B
COM3	/	/	/	COM3	1F	1A	2F	2A	3F	3A	4F	4A

12.3 Integrovaný obvod pro měření teploty

Jednou z funkcí zařízení je měření teploty, pro kterou je využit integrovaný obvod od firmy Atmel Corporation s označením modelu AT30TS74. Důvody výběru byly následující:

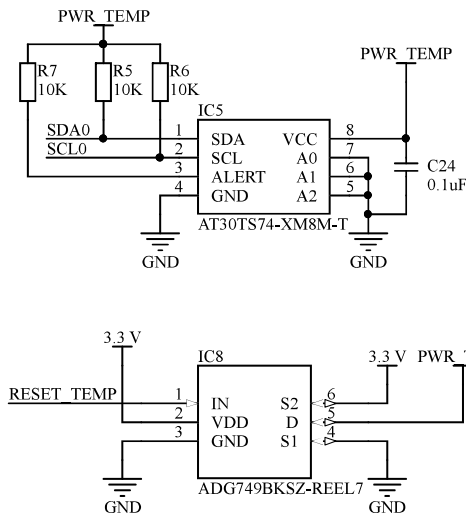
- integrovaný obvod je určen pro nízkovýkonné aplikace,
- obsahuje I²C rozhraní pro komunikaci s mikrokontroléry,
- pracovní napětí je v rozsahu 1,7 V až 5,5 V. (napěťová hladina zařízení, jak bylo zmíněno v podkapitole 12.1 je 3,3 V.)

Tabulka 12.2: Definice adresy integrovaného obvodu teplotního snímače [34]

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Identifikace typu zařízení				Adresa zařízení			Čtení/Zápis
1	0	0	1	A2	A1	A0	R/W

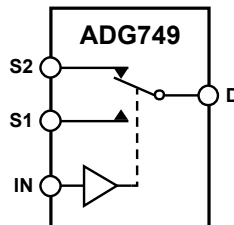
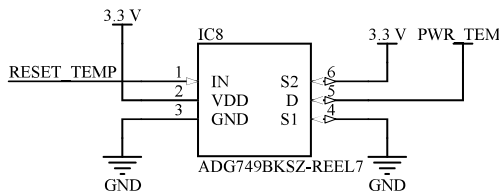
Dále byla využita verze v pouzdře s označením MSOP - 8. Adresa pro komunikaci po I²C rozhraní je definována pomocí tří adresových pinů. Jak je možno vidět v tabulce 12.2, adresní piny modifikují poslední tři bity adresy. Bylo rozhodnuto (viz schéma zapojení na obrázku 12.5), že všechny adresní piny budou připojeny na společnou zem. Adresa teplotního snímače je tedy v binární podobě 1001000_b. Teplotní rozsah snímače se pohybuje od -55 °C až po +125 °C. Teplotní snímač má dále definovanou přesnost ±1 °C v rozmezí teplot -20 °C až +100 °C. Rozlišení teplotního snímače je možné definovat programátorem. Může se pohybovat v rozmezí devíti až dvanácti bitů. To odpovídá rozlišení 0,5 °C až 0,0625 °C.[34]

Hodnota pull-up rezistorů byla zvolena na 10 KΩ, aby byl minimalizován únikový proud při komunikaci. Dále bylo rozhodnuto, že pin zvaný ALERT, který je určený pro signalizaci překročení



Obrázek 12.5: Schéma zapojení integrovaného obvodu AT30TS74

určených teplotních limitů, nebude využíván. Pin nesmí zůstat nepřípojený, proto byl přes rezistor 10 K Ω připojen ke kladnému pólu napětí. Dále byl pro teplotní snímač navrhnout hardwarový reset. K tomu byl využit přepínač s označením ADG749 od výrobce Analog Devices. Pomocí přepínače je možné hardwarově resetovat integrovaný obvod. Symbolické vnitřní zapojení je možné vidět na obrázku 12.6. [34]

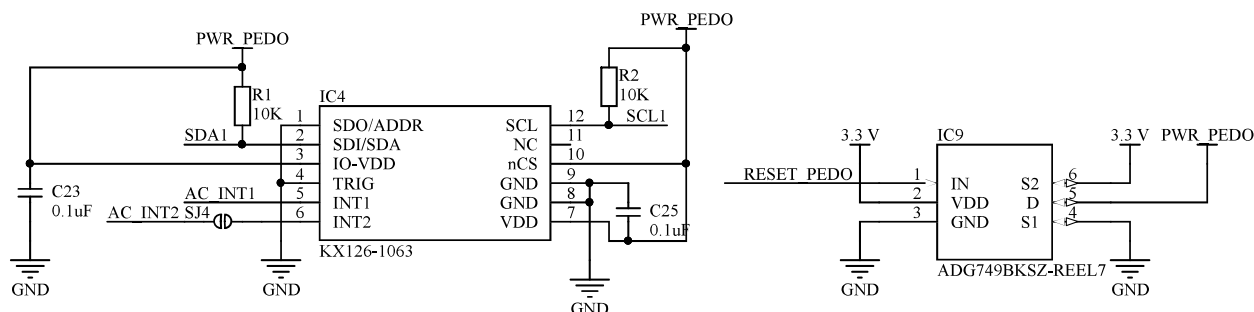


Obrázek 12.6: Vnitřní zapojení přepínače [34]

12.4 Integrovaný obvod pro detekci počtu kroků

Další funkcí zařízení Testbed je měření počtu kroků. Pro tuto funkci byl zvolený akcelerometr od výrobce Kionix, Inc. S označením KX126-1063. Jedná se o integrovaný obvod s nízkou spotřebou elektrické energie. Obsahuje několik funkcionalit, ať už detekci dvojkliku, volného pádu, ale také integrovaný krokoměr. Výhoda integrovaného krokoměru spočívá v úspoře spotřebované energie z důvodu jednoduchosti návrhu programového vybavení pro mikrokontrolér. Integrovaný obvod je vyráběn v pouzdře typu LGA s 12 piny. Dále jsou v integrovaném obvodu obsaženy dvě komunikační rozhraní, jedná se o I²C a SPI. Výběr použitého rozhraní pro danou aplikaci se pro-

vádí pomocí pinu nCS. V práci bylo využito rozhraní I²C, proto byl pin nCS připojen k napájecímu napětí, jak je uvedeno v katalogu od výrobce. Schéma zapojení integrovaného obvodu KX126-1063 je zobrazeno na obrázku 12.7. [35]



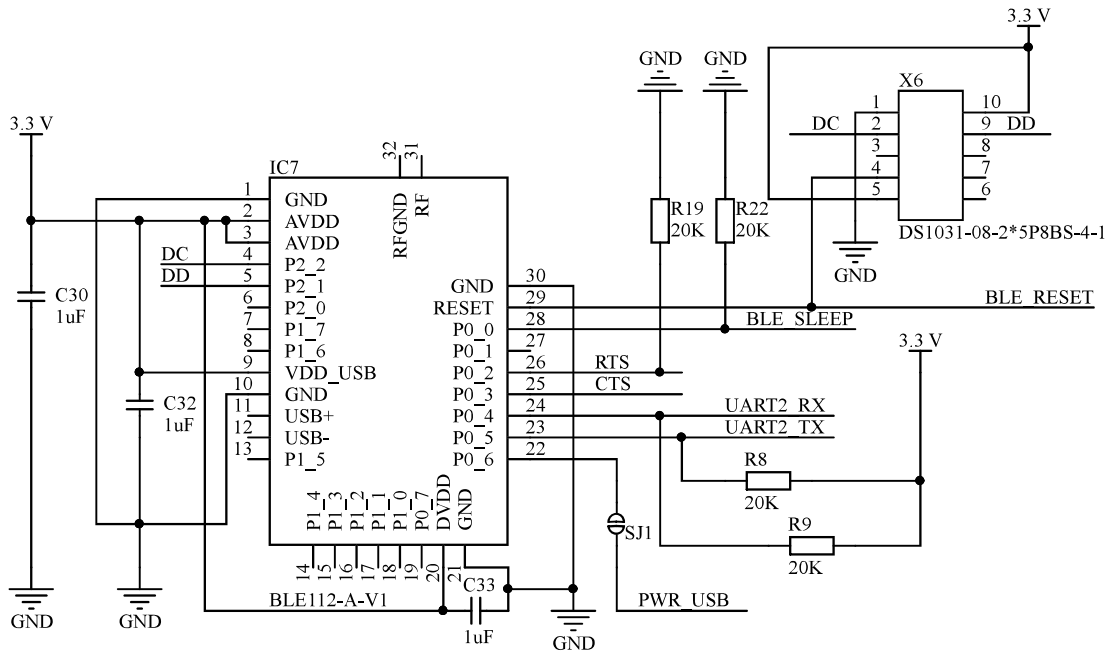
Obrázek 12.7: Schéma zapojení integrovaného obvodu KX126-1063

Stejně jako u obvodu pro měření teploty (viz kapitola 12.3) byla hodnota pull-up rezistorů zvolena na 10 K Ω a byl implementován hardwarový reset integrovaného obvodu pomocí přepínače s označením ADG749. U integrovaného obvodu je možné využít programovatelné přerušovací piny INT1 a INT2. V práci je využít pouze přerušovací pin INT1, kterému byla následně přiřazena funkce. [35]

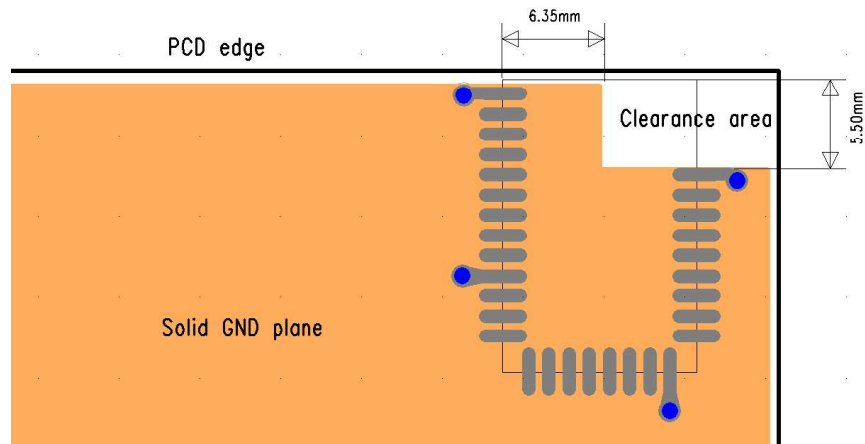
12.5 Integrovaný obvod pro detekci tepové frekvence

Poslední funkcí zařízení je měření tepové frekvence. Pro tuto funkci byl zvolen integrovaný obvod od firmy Maxim Integrated s označením MAX30003. Jedná se o profesionální analogový front - end, primárně určený pro zařízení s nízkou spotřebou elektrické energie. Komunikace s integrovaným obvodem je možná pomocí SPI komunikačního rozhraní. Integrovaný obvod je vyráběn v pouzdře typu TQFN s 28 piny. Teplotní rozsah součástky je výrobcem určen v rozmezí od 0 °C až po +100 °C. Teplota skladování je definována v rozsahu -65 °C až +150 °C. Při osazování součástky může samozřejmě teplota dosáhnout až na +300 °C. [36]

Na obrázku 12.8 je zobrazeno schéma zapojení senzoru. Jedná se o doporučené zapojení firmou Maxim Integrated. Integrovaný obvod je napájen napětím 1,8 V, které je sníženo z 3,3 V pomocí LDO regulátoru od firmy Microchip Technology Inc. Pro správnou funkci integrovaného obvodu je vyžadován externí oscilátor s frekvencí 32,768 kHz. Z důvodu úspory místa na DPS byla zvolena následující realizace. Použitý mikrokontrolér umožňuje generovat signál s touto frekvencí na jednom z jeho pinů. Takto vygenerovaný signál je přiveden na FCLK pin integrovaného obvodu MAX30003. Integrovaný obvod dále obsahuje dva přerušovací piny, kterými lze programově nastavit dané vlastnosti. V práci byl využít pouze jeden přerušovací pin, jehož funkce je osvětlena v kapitole 13.8.



Obrázek 12.9: Schéma zapojení modulu BLE112

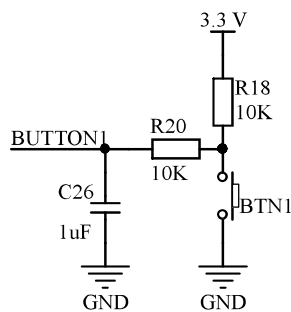


Obrázek 12.10: Modul BLE112 a jeho motiv plošného spoje [37]

a umístění modulu na DPS. Dle dostupných informací z katalogu od výrobce je vhodné modul umístit do rohu DPS. Následně v oblasti integrované antény nesmí být obsažena měď na DPS. Pro demonstraci je uveden obrázek 12.10 dostupný z katalogu od výrobce. Při dodržení těchto zásad bude BLE modul vynikat lepším dosahem. [37]

12.7 Tlačítka

Pro komunikaci mezi uživatelem a zařízením slouží dvě mechanická tlačítka. Jedná se o tlačítka určená pro povrchovou montáž se standardními rozměry 6 mm na šířku a 6 mm na délku. Celková výška tlačítka je 4,3 mm. Mechanické tlačítko spíná, případně rozezpíná eklektický obvod. Důsledkem sepnutí mechanického tlačítka je tekoucí proud přes toto tlačítko. Nicméně kontakty tlačítka vzhledem ke svým mechanickým vlastnostem mají jednu špatnou vlastnost. Při sepnutí tlačítka, případně rozepnutí dochází k odskokům kontaktů. Důsledkem těchto odskoků jsou tzv. zákmity, kdy pro jejich eliminaci byl navržen filtr tvořený dolní propustí, jak lze vidět na obrázku 12.11. Výstup z filtru označený BUTTON1 je pak spojen s GPIO pinem mikrokontroléru.



Obrázek 12.11: Schéma zapojení mechanických tlačítek

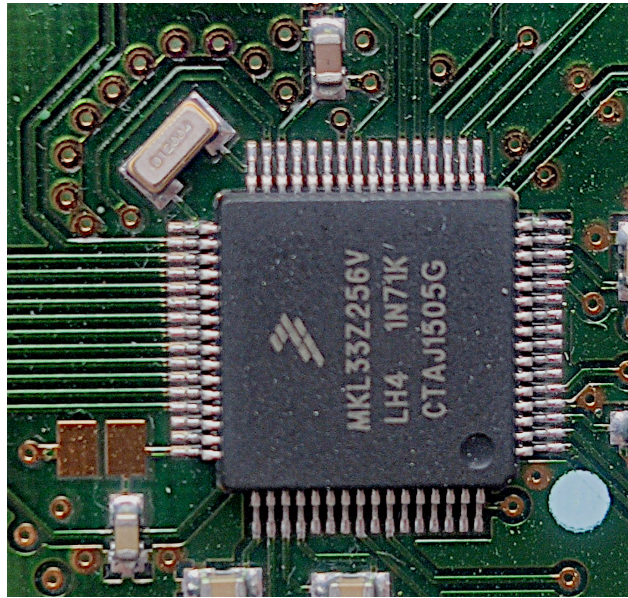
12.8 Mikrokontrolér

Ústředním členem zařízení je mikrokontrolér. Jak je z předešlých kapitol zřejmé, bylo zde několik požadavků, které musel daný mikrokontrolér splňovat:

- dvě komunikační rozhraní UART,
- řadič LCD,
- dva komunikační rozhraní I²C,
- komunikační rozhraní SPI,
- provozní napětí 3,3 V,

- nízká spotřeba.

Na základně vyjmenovaných požadavků byl vybrán vyhovující mikrokontrolér. Jedná se o Kinetis KL33 od výrobce NXP Semiconductors. Celkově mikrokontrolér obsahuje dvě 16 bitové SPI rozhraní, dvě I²C rozhraní a tři UART rozhraní. Dále je v mikrokontroléru integrovaná periferie LCD kontroléru, která podporuje multiplexovaný segmentový displej 28x8 (28 frontplane pinů a 8 backplane pinů) nebo 32x4. [38]



Obrázek 12.12: Oscilátor pro mikrokontrolér

Pro mikrokontrolér byl zvolen oscilátor s frekvencí 32,768 KHz. Jak lze vidět na obrázku 12.12, oscilátor byl přímo připojen k pinům mikrokontroléru s natočením o 45° (z důvodu minimalizace délky cest). V návrhu se neobjevují žádné externí kondenzátory, jelikož tyto kondenzátory je možné nastavit v programovém vybavení mikrokontroléru. To lze provést nastavením příslušných bitů v registru OSCx_CR.

V mikrokontroléru je implementováno programovací rozhraní SWD. Pro toto rozhraní byl následně vybrán příslušný konektor, který je totožný jako programovací konektor pro BLE modul. Ten byl popsán v kapitole 12.6.

Mikrokontrolér je vyráběn ve dvou typech pouzder, LQFP a BGA. Pro tuto realizaci byl zvolen typ pouzdra LQFP s 64 piny o rozměru 10 x 10 mm s výškou 0,5 mm.

12.9 Návrh elektrického zapojení

Po výběru komponent bylo nutné navrhnout schéma zapojení a motiv plošného spoje. Pro návrh byl zvolen již zmíněný EDA program Altium Designer (v14.3). Výhodou zvoleného programu

je možnost návrhu s vloženými, případně vytvořenými 3D modely jednotlivých součástek. Na základě toho je možné vytvořit 3D model celého zařízení. Tento model byl nesmírně důležitý, jelikož podle modelu byla navržena krabička a testovací stanice. Modely jednotlivých součástek byly vytvářeny v programu Autodesk Inventor Professional 2019 nebo staženy přímo z webu výrobce, pokud je výrobce poskytoval.

Pro návrh elektrického zapojení bylo nutné nakreslit blokové schéma elektroniky, které již bylo zobrazeno na obrázku 12.1. Ústřední člen celého zařízení je mikrokontrolér. K tomuto mikrokontroléru jsou hvězdicově připojeny ostatní komponenty potřebné pro funkci celého zařízení. Zařízení může být využíváno jako učební pomůcka, proto některé komponenty, případně vodivá spojení, nejsou využity pro základní funkci zařízení. Ale jejich využití je možné pro jiné vzorové úkoly.

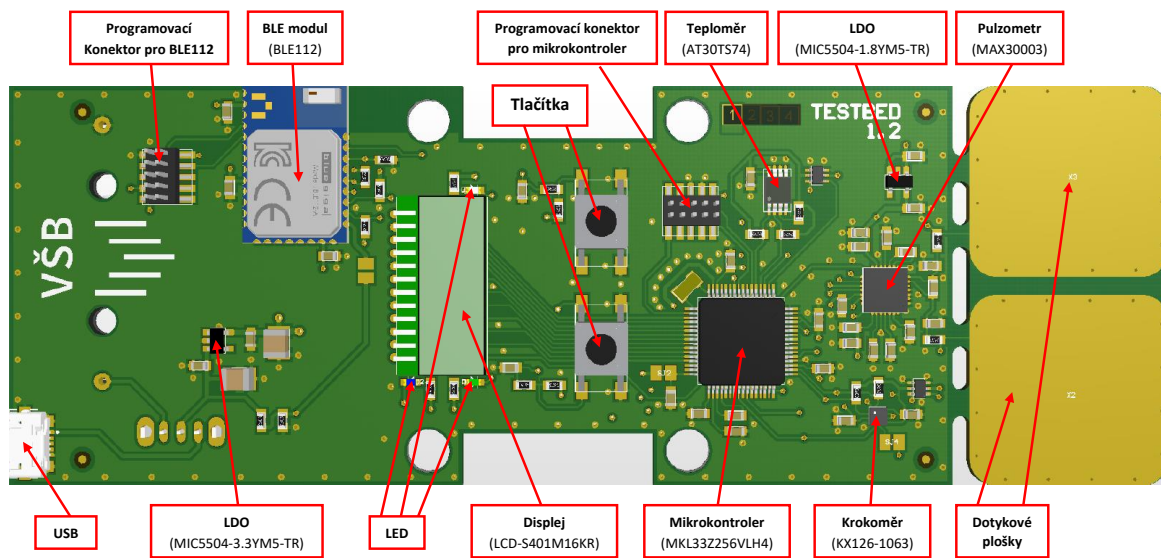
Jak bylo v předchozích kapitolách 12.3 a 12.4 zmíněno, tak u integrovaného obvodu pro detekci počtu kroků a teploměru je využito komunikační rozhraní I²C. Mikrokontrolér obsahuje dvě komunikační periferie I²C. Z důvodu jednodušší implementace byly využity obě periferie mikrokontroléru.

Po zhotovení blokového schématu zařízení Testbed bylo přistoupeno k návrhu v daném EDA programu. V první řadě bylo nutné vytvořit knihovnu jednotlivých komponent s jejich *footprintem*, schématickým symbolem a 3D modelem dané součástky. Po vytvoření knihovny v programu Altium Designer pak bylo nakresleno elektrické schéma zapojení.

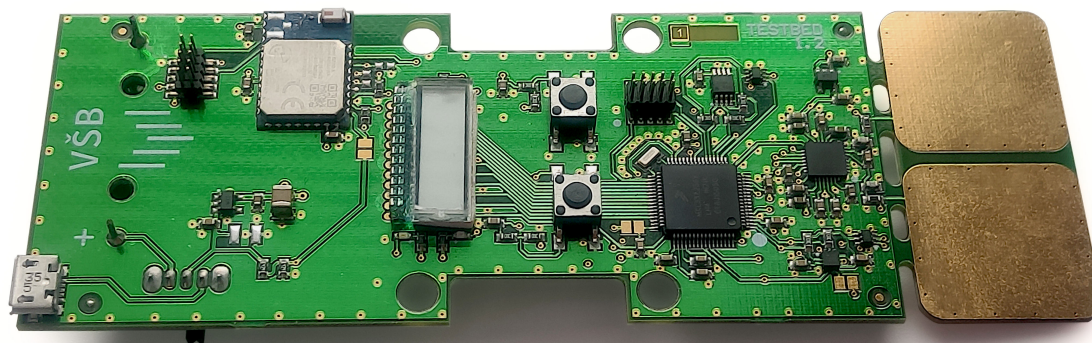
12.10 Návrh motivu plošného spoje

Po dokončení knihovny a schématu zařízení Testbed již bylo přistoupeno k návrhu motivu plošného spoje celého zařízení. Rozměry desky byly definovány zadavatelem. Tloušťka DPS byla zvolena na standardní tloušťku 1,5 mm. Počet vrstev DPS byl kvůli složitosti elektrického zapojení zvolen na čtyři vrstvy motivu. Druhá vrstva byla vyhrazena pro kladný pól napájení. Ve zbylých třech vrstvách byly rozvedeny signálové vodiče a zemnicí pól napětí ve formě polygonu. Rozmístění jednotlivých komponent s jejich popisem je možné vidět na obrázku 12.13 a fotka reálného zařízení je na obrázku 12.14. Byly zvoleny dva typy povrchových úprav pro dvě rozměrové varianty zařízení. Pro variantu s EKG ploškami byla zvolena povrchová úprava zvaná galvanické pozlacení, u níž nehrozí opotřebení a spoje jsou tvrdší. Nicméně pro rozměrovou variantu bez EKG plošek byla zvolena levnější varianta povrchové úpravy zvaná H.A.S.L.

Zařízení bude ve výrobní lince testováno v testovací stanici. Proto na spodní straně desky byly vytvořeny testovací body, které jsou kruhového tvaru s průměrem 2 mm. Na tyto body budou přivedeny testovací jehly.



Obrázek 12.13: Umístění komponent

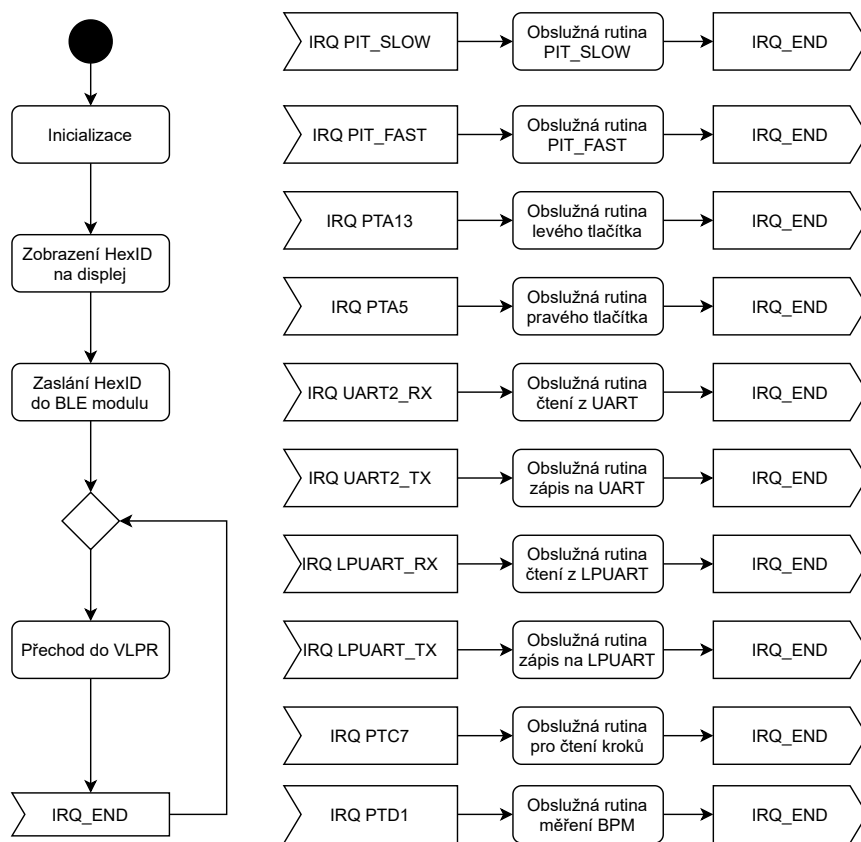


Obrázek 12.14: Fotka reálného zařízení

Kapitola 13

Návrh programového vybavení

V předchozí kapitole byly popsány zvolené komponenty a uvedeno blokové schéma elektroniky. Následně po návrhu elektrického zapojení a motivu plošného spoje, bylo přistoupeno k tvorbě programového vybavení pro mikrokontrolér. V první řadě byly navrženy knihovny, pomocí kterých je prováděna inicializace a obsluha jednotlivých periférií nebo integrovaných obvodů.



Obrázek 13.1: Diagram aktivit zařízení

Zvolený mikrokontrolér Kinetis KL33 od výrobce NXP Semiconductors obsahuje jádro AMR Cortex M0+. Přesné označení použité verze je MKL33Z256VLH4, která obsahuje flash paměti o velikosti 256 KB a 32 KB SRAM. Pro nahrání programového vybavení slouží SWD rozhraní.

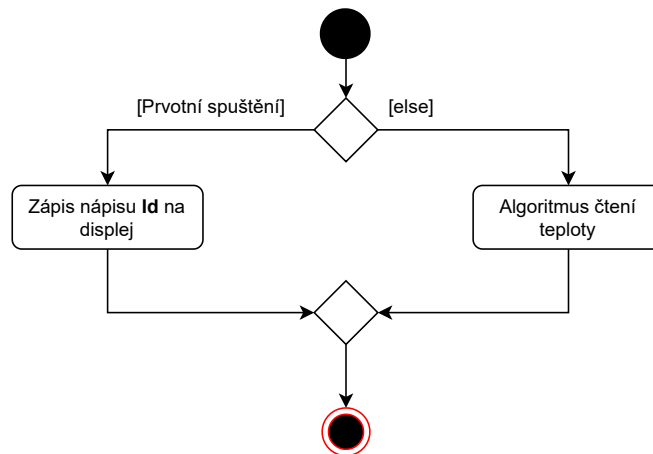
Mikrokontrolér je spuštěný v režimu VLPR. Jedná se o režim chodu s nízkou spotřebou elektrické energie, což má pozitivní dopad na životnost baterie. Návrh programového vybavení je popsán diagramem aktivit na obrázku 13.1. Při zapnutí zařízení je provedena prvotní inicializace. Zde jsou inicializovány všechny použité periferie, dále je provedena detekce osazených integrovaných obvodů a zobrazeno hexID na LCD. Následně je nastaven příznak SEND_ID a spuštěn časovač FAST_PIT (viz kapitola 13.1). Tím je zajištěno zaslání hexID s verzi osazení do BLE modulu. Poté je chod mikrokontroléru převeden do VLPR režimu. Přejed z režimu VLPR do provozního režimu je způsoben generovaným přerušením od jednotlivých integrovaných obvodů, tlačítek, vnitřní periferie časovače a případně komunikačních periférií. Po vykonání příslušného algoritmu je režim chodu mikrokontroléru opět přenesen do VLPR režimu. Jednotlivé algoritmy jsou blíže popsány v příslušných kapitolách.

13.1 Časovače

V práci byly využity dvě periferie časovače zvané PIT (Periodic Interrupt Timer), kde pro každý je implementována jiná funkce. Časovače byly pojmenovány PIT_SLOW a PIT_FAST. V prvé řadě byl implementován časovač PIT_SLOW, který má časovou periodu deset sekund. V obslužné rutině přerušení (zobrazená diagramem aktivit na obrázku 13.2) je prováděn algoritmus čtení hodnoty teploty z teplotního snímače a její další zpracování. Dále tento časovač slouží k časovanému zobrazení hexID při zapnutí zařízení. Po zapnutí se na deset sekund zobrazí hexID daného zařízení a po uplynutí deseti sekund, je spuštěna obslužná rutina přerušení časovače PIT_SLOW. Zde je zapsána na displeji první položka výběrového menu, tedy nápis **Id**.

Další použitý časovač PIT_FAST, byl vytvořen s časovou periodou 500 ms. Časovač PIT_FAST je využit v následujících případech:

1. Opětovné poslání hodnoty do BLE modulu
 - pokud vznikla chyba v komunikaci mezi mikrokontrolérem a BLE modulem,
 - data nemohli být poslány z mikrokontroléru do BLE modulu přes komunikační rozhraní, kvůli právě probíhající komunikaci po komunikačním rozhraní,
2. Postupné odeslání posledních naměřených hodnot při připojení chytrého telefonu k zařízení Testbed,
3. Ukončení měření tepové frekvence,
4. Ukončení časového rozmezí pro nulování počtu kroků,



Obrázek 13.2: Diagram aktivit obslužné rutiny PIT_SLOW

5. Detekce vibrací při testování v testovací stanici.

Jelikož jednotlivé případy jsou prováděny v jedné obslužné rutině, bylo nutné je navrhnout tak, aby nedocházelo k nedefinovaným stavům. V první řadě je nutné zmínit, že poslední případ může nastat jen v režimu při detekci vibrací, který je spouštěným pomocí testovací stanice. Tento režim je možné spustit pouze přes komunikační rozhraní UART určené pro komunikaci s testovací stanicí. Pokud je režim vibrací aktivní, nemohou probíhat ostatní zmíněné případy. Z toho důvodu při ošetřování jednotlivých stavů nebyl tento případ brán v potaz. Nicméně pro ostatní případy bylo nutné ošetřit vypnutí běhu časovače. Časovač je spuštěn pouze když nastane jeden z případů. Dále může být vypnut pouze v případě, když není aktivní ani jeden z výše vyjmenovaných případů. Z toho důvodu byla vytvořena proměnná, které byl v programu přiřazen název PitFastProcessStart. Tato proměnná obsahuje jednotlivé bity. Ty signalizují, který z případů je právě aktivní (daný bit je nastaven na hodnotu 1), či neaktivní (bit je nastaven na hodnotu 0). Dále byl vytvořen výčetový datový typ, ve kterém jsou obsaženy symbolické konstanty zastupující daný případ. V tabulce 13.1 je uvedena proměnná PitFastProcessStart i s popisem jednotlivých bitů (název koresponduje s názvy symbolických konstant vytvořeného výčetového datového typu).

Tabulka 13.1: Přehled obsahu jednotlivých bitů proměnné PitFastProcessStart

Bit	7..5	4	3	2	1	0
Název	X	StopMeasBPM	ResetStep	ResetBLE	ConnectDevice	SendError

Jednotlivé bity uvedené v tabulce 13.1 jsou nastaveny na hodnotu 1 v příslušných obslužných rutinách současně se spuštěním časovače PIT_FAST. Bylo ale nutné ošetřit stav, kdy může vzniknout požadavek na spuštění časovače, i když časovač již byl spuštěn jiným případem. Tento stav byl řešen kvůli případům 3 a 4, které jsou závislé na čase mezi spuštěním případu a vyvoláním obslužné

rutiny. Pokud již časovač běžel, tak čas mezi spuštěním případu a vyvoláním obslužné rutiny přerušeni bude menší než 500 ms. Proto byla vytvořena funkce, která opětovně spustí časovač s periodou 500 ms. Dále pokud byl nastaven bit pro příslušný případ, musí být opět vrácen do neaktivního stavu. To může být učiněno pouze, pokud by daný případ byl proveden. Ukončení případu, a tedy změna hodnoty bitu v proměnné `PitFastProcessStart` je provedena pro každý případ individuálně. Tedy díky proměnné `PitFastProcessStart`, může být příkaz pro vypnutí časovače `PIT_FAST` v kódu použit pouze jednou, a to v obslužné rutině časovače. Vypnutí tedy proběhne pouze, když není aktivní žádný z uvedených případů.

13.2 Detekce zdroje napájení

Zařízení je možné napájet z baterie nebo pomocí USB konektoru. Napětí baterie je 3,6 V a napětí USB standartu je 5 V. Jelikož napěťové hladiny obou zdrojů se liší, je možné podle napěťové hladiny na vstupu detekovat, z jakého zdroje je zařízení napájeno. K tomu je využita periferie komparátoru implementována v mikrokontroléru. Napětí z výstupu přepínače (viz obrázek 12.2) je sníženo pomocí napěťového děliče. Pro dělič byly voleny velké hodnoty elektrického odporu rezistorů tak, aby došlo k minimalizaci úniku proudu. Napětí za děličem je pak vstupem pro periferii komparátoru v mikrokontroléru. Hodnoty elektrického odporu rezistorů v děliči jsou stejné, tedy dle vztahu pro výpočet výstupního napětí napěťového děliče, platí:

$$U_{výst} = U_{vst} \cdot \frac{R_2}{R_1 + R_2}, \quad (13.1)$$

kdy rezistory $R_1 = R_2$:

$$U_{výst} = \frac{1}{2} \cdot U_{vst}. \quad (13.2)$$

Napětí $U_{výst}$ je výstupní napětí pro dělič napětí a současně vstupní napětí pro periferii komparátoru. Napětí U_{vst} je vstupní napětí pro dělič napětí. Na vstupu děliče mohou být přivedeny dvě úrovně napájecího napětí. Pro napájení pomocí USB konektoru je napěťová hladina na vstupu komparátoru přibližně 2,5 V, naopak v případě napájení zařízení z baterie je napěťová hladina 1,8 V a nižší, záleží na kapacitě baterie. Přivedené napětí na komparátoru je porovnáváno se zvoleným referenčním napětím.

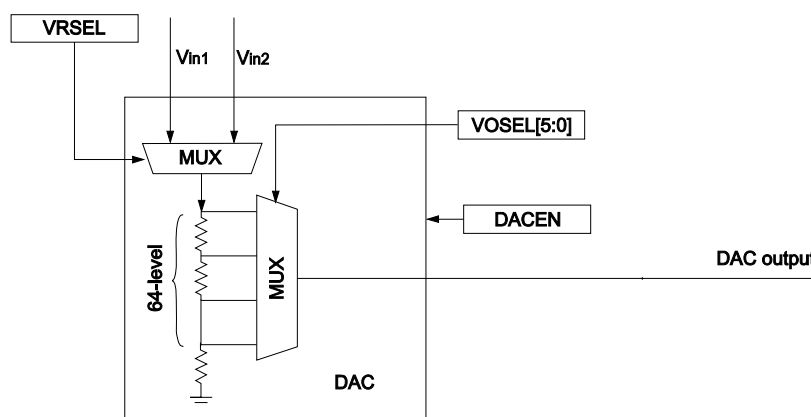
Pro obsluhu periferie komparátoru byla vytvořena knihovna. Pomocí knihovny je inicializována periferie komparátoru v mikrokontroléru a obsahuje funkci pro určení typu napájení.

V první části inicializace je nastavena funkce GPIO pinu jako jeden ze vstupů komparátoru. Poté v registru `CMP0_MUXCR` jsou definovány vstupy pro kladný a záporný pin komparátoru. Jako vstupem pro kladný pin komparátoru byl zvolen výstup z napěťového děliče. Naopak jako vstup pro záporný pin komparátoru byla využita napěťová reference, která je tvořena 6 bitovým D/A pře-

vodníkem. Inicializace D/A převodníku je součástí inicializační funkce komparátoru. Na obrázku 13.3 je zobrazeno principiální schéma pro inicializaci D/A převodníku. Inicializace probíhá zápisem do registru CMP0_DACCR. V této realizaci je využit vstup Vin2, který je přímo spojený s napájecím napětím 3,3 V. Tato napěťová hladina je upravena pomocí odporové sítě, která je nastavitelná bity zvaných VOSEL. Výstupní napětí D/A převodníku je poté definováno následujícím vztahem uvedeným v referenčním manuálu [39]:

$$U_{DAC} = \frac{U_{in2}}{64} \cdot (VOSEL[0 : 5] + 1) \quad (13.3)$$

kdy U_{DAC} je výstup D/A převodníku a U_{in2} je v této realizaci rovno 3,3 V. Registr VOSEL byl definován na hodnotu 37_d. Tedy výstupní napětí D/A převodníku je přibližně rovno 1,96 V. Napěťová úroveň 1,96 V je porovnávána s napěťovou úrovní na výstupu napěťového děliče. Výsledek porovnání komparátoru je zpracován pomocí funkce nazvané PowerSupply_Get, která neobsahuje žádné vstupní parametry. Návrátový parametr funkce je výčtový datový typ, který obsahuje dvě konstanty USB a Battery. Na základě typu zdroje napájení zařízení je funkcí vrácena příslušná konstanta. [39]



Obrázek 13.3: Principiální schéma zapojení D/A převodníku [39]

13.3 Ovladač pro LCD

Jak již bylo zmíněno v kapitole 12.2 byl zvolen displej s označením LCD-S401M16KR. Displej má dvanáct pinů, z toho jsou čtyři backplane a osm frontplane. Pro ovládnání displeje byla vytvořena knihovna, ve které je obsažena inicializační funkce a další funkce, které provádějí zápis čísel a znaků na displej.

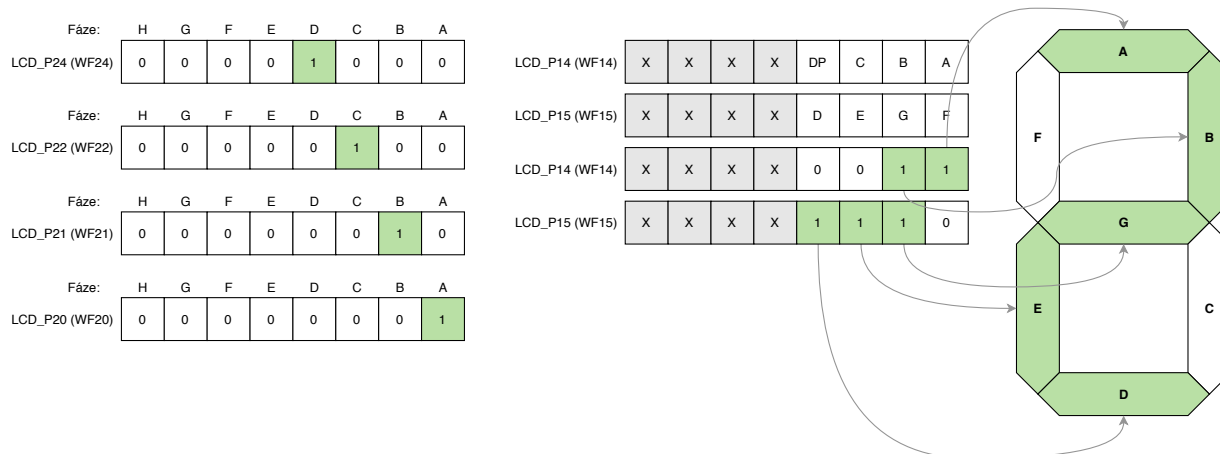
V inicializační funkci je v první řadě povolen zdroj hodin pro periférii LCD kontroléru. Byla zvolena frekvence 32,768 KHz. Následně byly nakonfigurovány příslušné vstupně/výstupní piny pro funkci LCD kontroléru. Využití pinů je možné vidět v tabulce 13.2. Dále byly nastaveny konfi-

gurační registry pro LCD kontrolér. Poté je na daných pinech povolena funkce pro LCD kontrolér a nastaveny backplanes piny.

Tabulka 13.2: Soupis pinů využitých pro LCD

Pin na LCD	Pin mikrokontroléru	Pin LCD kontroléru	Typ
1	PTC4	LCD_P24	Backplane
2	PTC2	LCD_P22	Backplane
3	PTC1	LCD_P21	Backplane
4	PTC0	LCD_P20	Backplane
5	PTB19	LCD_P15	Frontplane
6	PTB18	LCD_P14	Frontplane
7	PTB17	LCD_P13	Frontplane
8	PTB16	LCD_P12	Frontplane
9	PTB3	LCD_P3	Frontplane
10	PTB2	LCD_P2	Frontplane
11	PTB1	LCD_P1	Frontplane
12	PTB0	LCD_P0	Frontplane

V mikrokontroléru jsou obsaženy tzv. Waveform registry (LCD_WF). Jedná se o 32 bitové registry, kdy každý registr je rozdělen na čtyři 8 bitové sekce. Jedna sekce odpovídá danému LCD pinu. Dále u zvoleného displeje jsou pro daný digit využity dva frontplane piny. Pro lepší pochopení je uveden příklad na obrázku 13.4. Backplane piny (LCD_P24, LCD_P22, LCD_P21, LCD_P20) jsou nastaveny na příslušné fáze již v inicializační funkci. Dále piny LCD_P14 a LCD_P15 jsou frontplane piny pro první digit (viz tabulka 13.2 a tabulka 12.1). Na obrázku 13.4 jsou zobrazeny obě osmi bitové sekce registru LCD_WF s popisem, který bit přísluší danému segmentu. Pokud daný bit obsahuje logickou jedničku, tak daný segment je zapnutý. V opačném případě, kdy v daném bitu je zapsaná logická nula, tak daný segment je vypnutý. V uvedeném příkladu je na segmentu zobrazeno číslo dva. Tedy bity odpovídající segmentům A, B, D, E, G musí být nastaveny do logické jedničky (viz obrázek 13.4). Na základě této skutečnosti byla vytvořena funkce (LCD_set), pomocí které lze zobrazit osmi bitovou hodnotu v šestnáctkové soustavě na zvolený segment displeje. Tato funkce je dále využita pro zápis dekadické a hexadecimální hodnoty na displej. Jelikož zobrazení dekadických hodnot na LCD je omezeno pouze na čtyři digity, bylo nutné navrhnout styl označení pro hodnoty vyšší než 9999. Bylo tedy rozhodnuto, že tisíce budou odděleny tečkou. Tedy při zobrazení hodnoty o velikosti deseti tisíců na displeji nejsou zobrazeny jednotky. Následně při zobrazení statisíců na displeji nejsou navíc zobrazeny desítky. Tato skutečnost je zobrazena na obrázku 13.5, kde je uveden příklad.



Obrázek 13.4: Příklad zobrazení čísla dvě na prvním digitu

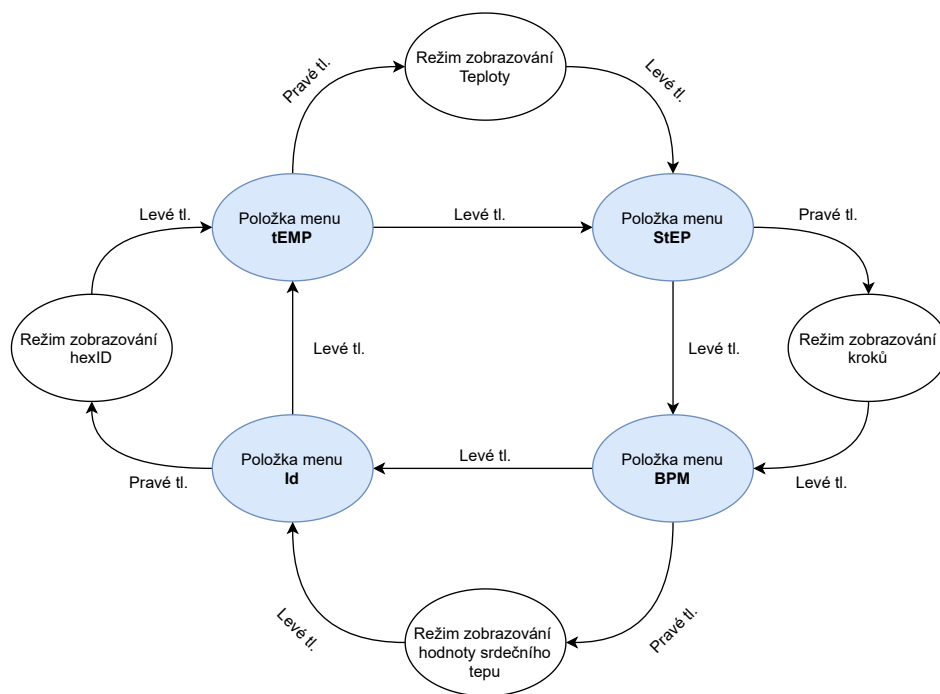


Obrázek 13.5: Příklad zobrazení hodnot

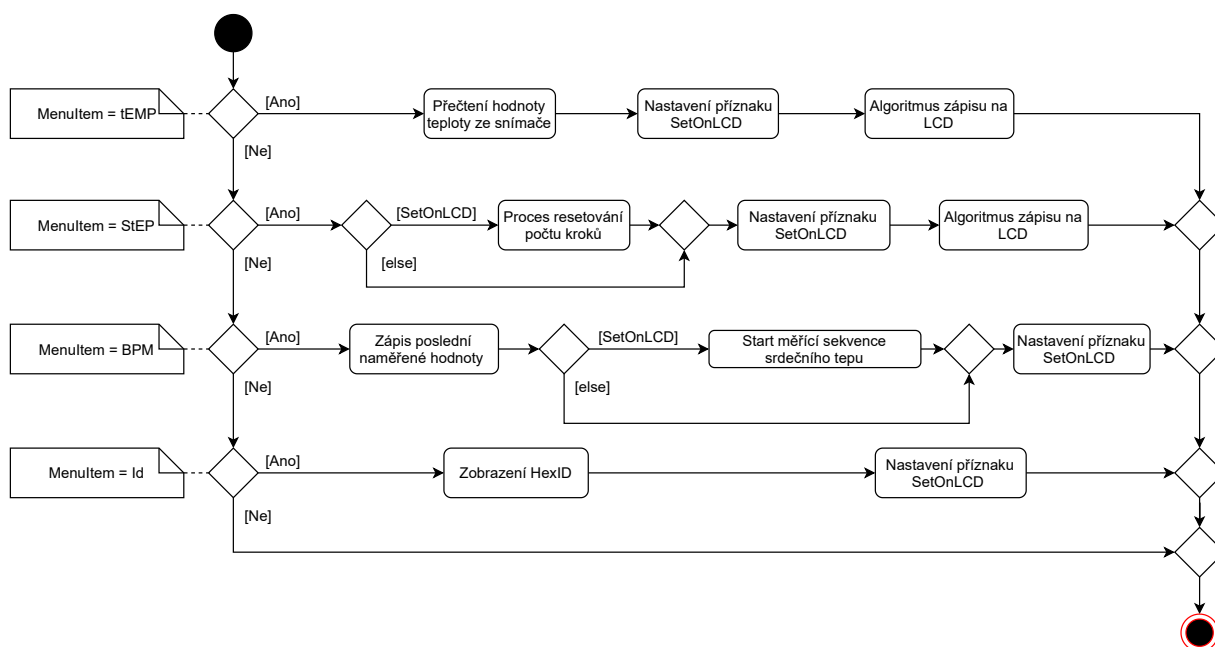
13.4 Výběrové menu

Pro výběr zobrazení hodnoty měřených veličin bylo vytvořeno výběrové menu, ve kterém jsou obsaženy celkem čtyři položky (tEMP, StEP, BPM, Id). Stisknutím levého tlačítka je možné tyto položky mezi sebou přepínat. Pomocí pravého tlačítka je možné zvolit danou položku ve výběrovém menu. Po zvolení položky je zařízení přepnuto do daného režimu zobrazování, ve kterém pak pravé tlačítko může nabývat jiné funkce. Na obrázku 13.6 je zobrazen stavový diagram pro navrženou funkci pravého a levého tlačítka. Na základě uvedeného diagramu byla vytvořena programová funkce. V diagramu je zobrazeno celkem osm stavů. Modře označené stavy signalizují stav, kde je na displeji zobrazen text jedné z položek výběrového menu. Pro tyto stavy byla vytvořena stavová proměnná s názvem MenuItem. Pokud dojde k stisku pravého tlačítka, je provedena obslužná rutina, která je zobrazena diagramem aktivit na obrázku 13.7. Zde, podle hodnoty stavové proměnné MenuItem, je provedena příslušná větev. V každé větvi je nastaven příznak SetOnLCD, který určuje, že daná položka menu byla zvolena. Tento příznak lze vynulovat pouze při stisku levého tlačítka.

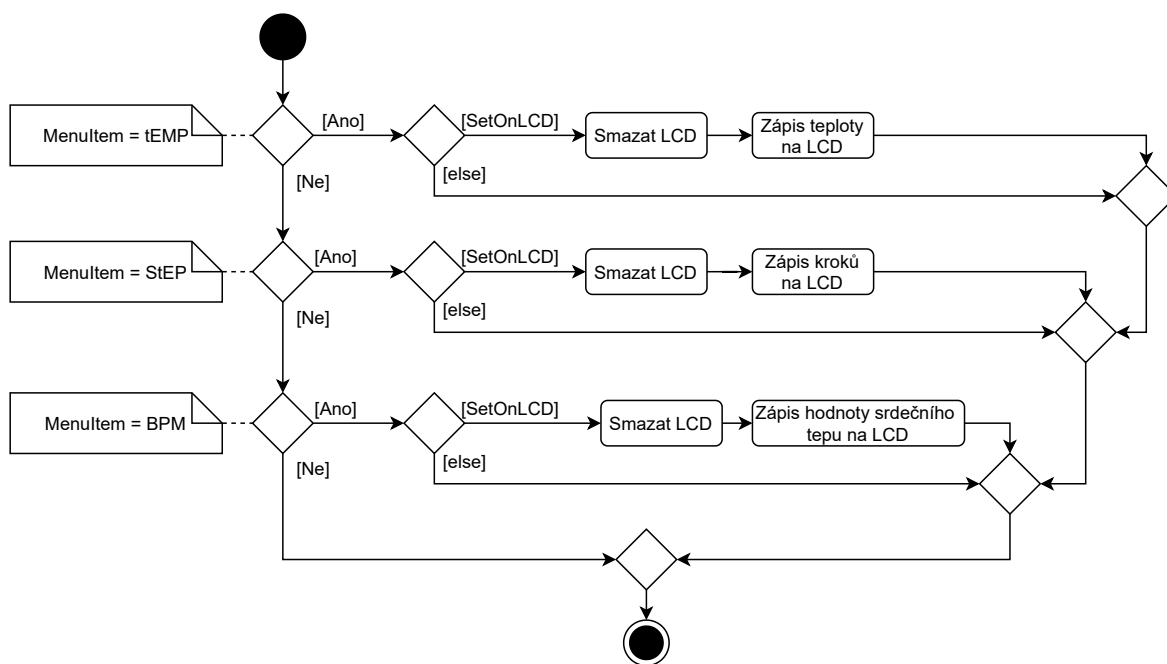
V diagramu aktivit na obrázku 13.7 je zobrazena aktivita "Algoritmus zápisu na LCD". Jedná se o algoritmus, který je využíván ve více obslužných rutinách a je dále rozveden diagramem aktivit na obrázku 13.8. V prvé řadě je zjištěno, která položka menu byla vybrána a pokud již byla nastaven příznak SetOnLCD, dojde k zápisu příslušné hodnoty.



Obrázek 13.6: Stavový diagram funkce pravého a levého tlačítka



Obrázek 13.7: Diagram aktivit pro obslužnou rutinu pravého tlačítka



Obrázek 13.8: Diagram aktivit algoritmu zápisu na LCD

13.5 Autodetekce osazených snímačů

Zařízení se může objevit celkem ve třech variantách osazení. Při návrhu bylo rozhodnuto, že pro zařízení bude vytvořeno pouze jedno programové vybavení. Proto bylo nutné, aby při zapnutí zařízení proběhla automatická detekce osazených integrovaných obvodů. Proto v prvotní inicializaci (po zapnutí zařízení) byl vytvořen algoritmus automatické detekce osazených integrovaných obvodů. Byla zvolena následující realizace. Ze všech tří integrovaných obvodů je vyčtena hodnota určeného registru a porovnána s očekávanou hodnotou. Pokud se vyčtená a očekávaná hodnota neshodují, je vyhodnoceno, že daný integrovaný obvod není na DPS osazen. Například integrovaný obvod pro měření teploty AT30TS74 obsahuje registry teplotních limitů, které nebyly v práci využity. Podle katalogu výrobce registr teplotních limitů po startu nabývá hodnoty v hexadecimální soustavě 0x4B00. Adresy příslušných registrů a očekávané hodnoty jsou vypsány v tabulce 13.3.

Tabulka 13.3: Registry a očekávané hodnoty při automatické detekci osazení

Integrovaný obvod	Adresa registru	Očekávaná hodnota
AT30TS74	0x02	0x4B00
KX126-1063	0x1A	0x86
MAX30003	0x1D	0x003B8100

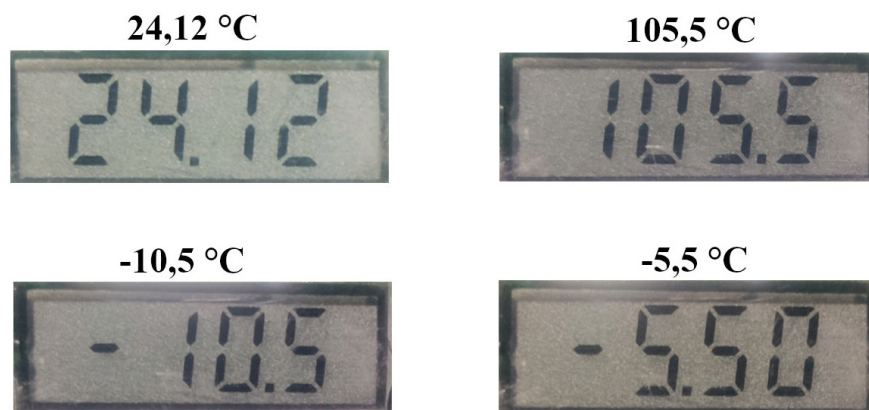
13.6 Měření teploty

Po implementaci základních funkcí bylo přistoupeno k návrhu měřících algoritmů. V první řadě byl vytvořen a implementován měřící algoritmus pro měření teploty. Dle požadavků je žádáno, aby teplota byla aktualizována každých deset sekund a její hodnota byla zobrazena na displeji (tedy v případě, že byla zvolena položka **tEMP** ve výběrovém menu). Na základě těchto stanovených požadavků byl navržen měřící algoritmus.

Tabulka 13.4: Výpis všech registrů teplotního snímače [34]

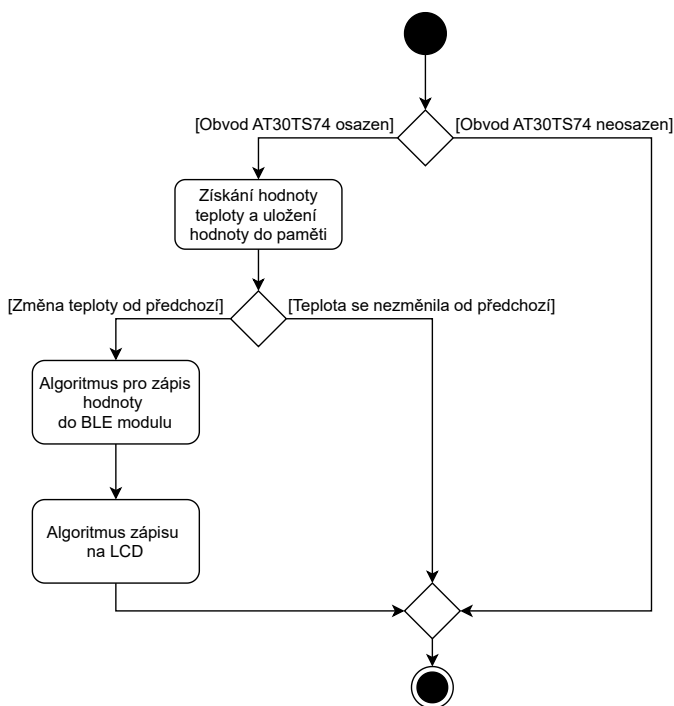
Název registru	Adresa	Zápis/Čtení	Velikost	Výchozí hodnota
Pointer registr	n/a	Zápis	8 bitů	0x00
Teplotní registr	0x00	Čtení	16 bitů	0x0000
Konfigurační registr	0x01	Zápis/Čtení	16 bitů	0x0000
Registr teplotních limitů (Spodní hranice)	0x02	Zápis/Čtení	16 bitů	0x4B00 (75 °C)
Registr teplotních limitů (Horní hranice)	0x03	Zápis/Čtení	16 bitů	0x5000 (80 °C)

Jak již bylo zmíněno pro měření teploty byl zvolen integrovaný obvod AT30TS74, uvedený v kapitole 12.3. Tento obvod obsahuje celkem pět registrů, které jsou uvedeny v tabulce 13.4. Pointer registr má velikost osmi bitů a je určen pouze pro zápis. Slouží k zápisu adresy jednoho ze čtyř registrů. Z registru, který má uloženou adresu v pointer registru, je pak možné číst. Při zápisu do registru je nutné pokaždé zadat adresu registru, do kterého bude zapisováno. Adresy jednotlivých registrů jsou uvedeny v tabulce 13.4. Teplotní registr je určený pouze pro čtení, a obsahuje hodnotu teploty v okolí integrovaného obvodu. Jedná se o šestnáctibitový registr, kdy maximální rozlišení snímače je dvanáct bitů. Toto rozlišení lze nastavit v konfiguračním registru. Tento registr obsahuje mnoho dalších nastavení, které jsou uvedeny v katalogu výrobce. Poslední dva registry slouží k nastavení teplotních limitů. [34]



Obrázek 13.9: Hodnoty teploty zobrazené na LCD

Pro obsluhu teplotního snímače AT30TS74 byla vytvořena knihovna, ve které jsou implementovány funkce pro inicializaci a sběr dat ze snímače, případně zápis hodnoty teploty na LCD. V inicializační funkci je provedena konfigurace pro rozlišení teploměru na jedenáct bitů, tedy zobrazovaná hodnota teploty je zobrazena se dvěma desetinnými místy. Další funkce, která je v programu pojmenována GetTemperature již obsahuje algoritmus sběru dat ze snímače teploty AT30TS74. Funkce neobsahuje vstupní parametr a návratová hodnota funkce je znaménková, šestnáctibitová celočíselná hodnota. Jedná se o hodnotu změřené teploty bez desetinné čárky. Například, pokud tedy změřená teplota byla 23,50 °C, návratová hodnota funkce je ve tvaru 2350. Tento tvar je vhodný pro zápis na LCD, kdy desetinná tečka je následně doplněna při zobrazení. K tomu slouží poslední funkce vytvořené knihovny. Vstupní parametr funkce je hodnota teploty uložené v šestnáctibitové celočíselné proměnné. Algoritmus zápisu hodnoty na displej je navrhnut na rozsah teplotního snímače. Jelikož je zobrazení hodnoty omezeno na čtyři sedmissegmenty, bylo nutné dle velikosti hodnoty teploty upravit formu zápisu. Pro kladné hodnoty teploty není zobrazeno znaménko, tedy je možné zobrazit hodnotu s přesností na setiny. Nicméně při překročení teploty nad 100 °C je na displeji zobrazena hodnota teploty s přesností na desetiny. Obdobně je to i u záporných hodnot teploty, kdy první znak na displeji je znaménko mínus a poté následují desítky, jednotky a za desetinnou tečkou desetiny. Pokud je ale záporná hodnota teploty větší než -10 °C, tak jsou na displeji zobrazeny jednotky a za desetinou tečkou následují desetiny a setiny. Příklad zobrazení různých hodnot teploty na displeji je možné vidět na obrázku 13.9.



Obrázek 13.10: Diagram aktivit algoritmu měření teploty

Proces měření a zápisu hodnoty teploty na LCD probíhá v obslužné rutině přerušení časovače PIT_SLOW. Blíže je tento časovač popsán v kapitole 13.1, kde je uveden diagram aktivit (obrázek 13.1), kdy blok algoritmu čtení teploty je dále rozveden diagramem aktivit na obrázku 13.10. Nejprve je ověřeno, zda je obvod AT30TS74 osazen. Následně je z obvodu vyčtena hodnota teploty pomocí implementované funkce a uložena do paměti. Pokud je získaná hodnota teploty odlišná od poslední naměřené hodnoty teploty, tak je hodnota teploty zapsána na displej a proveden algoritmus pro zápis hodnoty do BLE modulu (pouze v případě připojeného chytrého telefonu). Nicméně pokud se hodnota právě naměřené teploty nezměnila od předchozí, tak nedochází k žádné akci.

13.7 Detekce počtu kroků

Po návrhu měřicího algoritmu pro měření teploty bylo přistoupeno k návrhu a implementaci detekci počtu kroků. Pro detekci počtu kroků byl vybrán akcelerometr KX126-1063, který je blíže popsán v kapitole 12.4. Akcelerometr obsahuje integrovanou periférii krokoměru, které přísluší mj. dva osmibitové registry zvané PED_STP_L a PED_STP_H. Registry obsahují počet vzniklých kroků od posledního vyčtení registru PED_STP_H. Hodnota počtu kroků je tedy šestnáctibitová, přičemž registr PED_STP_L obsahuje nižších osm bitů a registr PED_STP_H naopak osm vyšších bitů.[35]

V prvé řadě bylo nutné navrhnout, jakým způsobem budou čtena data z registrů PED_STP_L a PED_STP_H. Jedna z možností byla cyklickým dotazováním, nicméně toto řešení není moc efektivní. Pokud by zařízení bylo v klidu, byly by z příslušných registrů akcelerometru vyčítány nulové hodnoty počtu kroků. Proto bylo přistoupeno k druhé možnosti, a to využití přerušení od akcelerometru. Jak již bylo zmíněno v kapitole 12.4 bylo využito přerušení INT1. Pro povolení, nastavení zdroje a další konfiguraci periférie přerušení jsou určeny registry INC1, INC5 a INC7. Zdroj přerušení byl nastaven na překročení definované hranice počtu vzniklých kroků. Tato hranice je nastavena registry PED_STPWM_L a PED_STPWM_H. V této realizaci byla hranice zvolena na dvanáct kroků. Tedy pokud počet kroků v registrech PED_STP_L a PED_STP_H bude roven nebo větší než dvanáct kroků, bude vygenerován přerušovací signál na přerušovacím pinu INT1. [35]

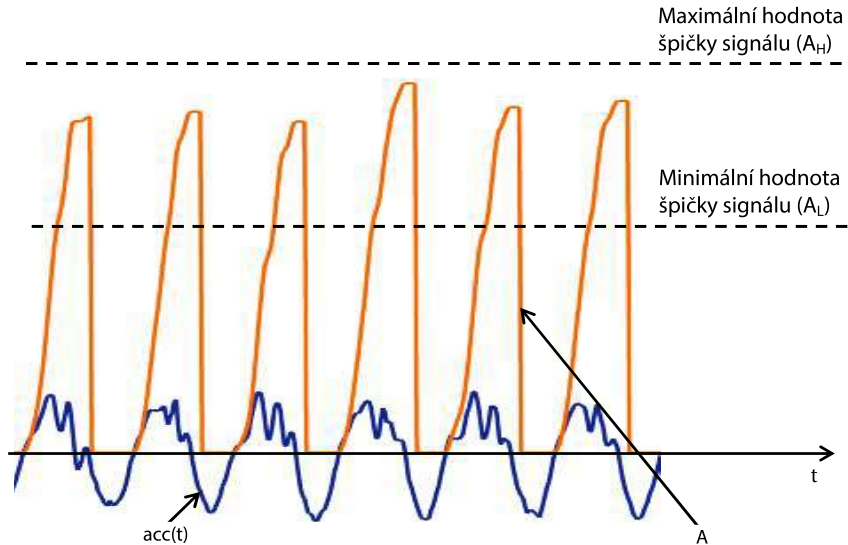
Opět byla vytvořena knihovna, která obsahuje funkce pro inicializaci akcelerometru pro vyčtení počtu kroků a zápisu kroků na displej. Pro integrovaný krokoměr je v akcelerometru obsaženo několik nastavitelných registrů, kterými lze parametrizovat chod periférie. Funkce detekce počtu kroků je založena na vzniklých změnách 3D akcelerometru při dopadu nohy na zem. Nejprve je vypočtena absolutní hodnota z tří rozměru akcelerometru:

$$acc(t) = \sqrt{(acc_x(t))^2 + (acc_y(t))^2 + (acc_z(t))^2}. \quad (13.4)$$

Pomocí integrace je vypočtena okamžitá plocha vzniklé špičky signálu:

$$A = \int acc(t) dt. \quad (13.5)$$

Obě matematické funkce 13.4 a 13.5 jsou uvedeny v manuálu výrobce *Getting Started with Pedometer* [40].

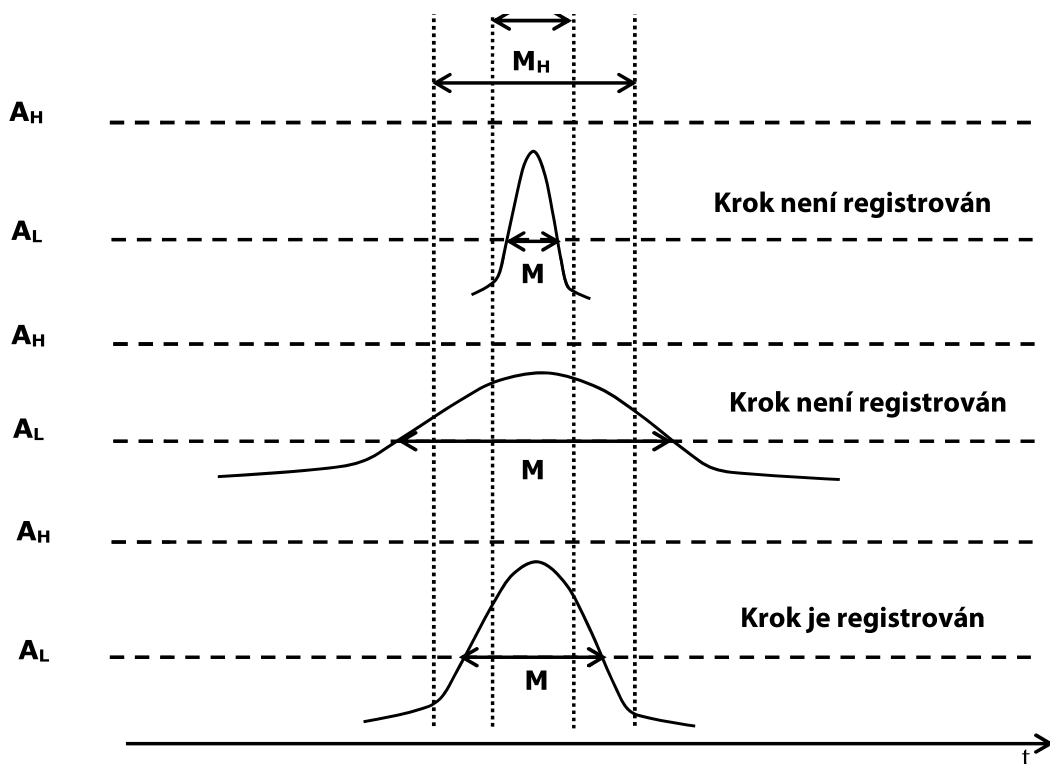


Obrázek 13.11: Průběh absolutní hodnoty tří rozměrů akcelerometru a okamžitá hodnota špičky signálu [40]

Grafické zobrazení těchto funkcí je možné vidět na obrázku 13.11. Pomocí parametrů A_L a A_H je definováno rozmezí, ve kterém je korektně registrován krok. Pokud tedy vrchol nedosahuje minimální hodnoty A_L nebo naopak přesahuje hodnotu A_H , není daná špička signálu registrována jako korektní krok. To znamená, že registry PED_STP_L a PED_STP_H nejsou aktualizovány. Tyto úrovně jsou nastavitelné v příslušných registrech akcelerometru. Názvy těchto registrů jsou uvedeny v tabulce 13.5. [40]

Jeden z dalších nastavitelných parametrů je časový interval špičky signálu. Opět tento interval (značený M) musí nabývat hodnot v předem daných mezích určených parametry M_L a M_M . Pokud hodnota intervalu vybočuje z předepsaných mezí, tak opět detekovaná špička signálu není registrována jako korektní krok. [40]

Dalším podstatným parametrem je časový interval mezi dvěma po sobě vzniklými kroky. Opět je možné definovat mezní parametry (T_P a T_M) a vymezit tím, v jakém rozmezí se může hodnota intervalu (T) mezi dvěma kroky nacházet. Pokud je hodnota intervalu mezi dvěma kroky příliš krátká, či naopak příliš dlouhá, vzniklá špička signálu není detekována jako korektní krok. S tímto parametrem úzce souvisí parametr STP_TH , který určuje prahovou hodnotu, od které

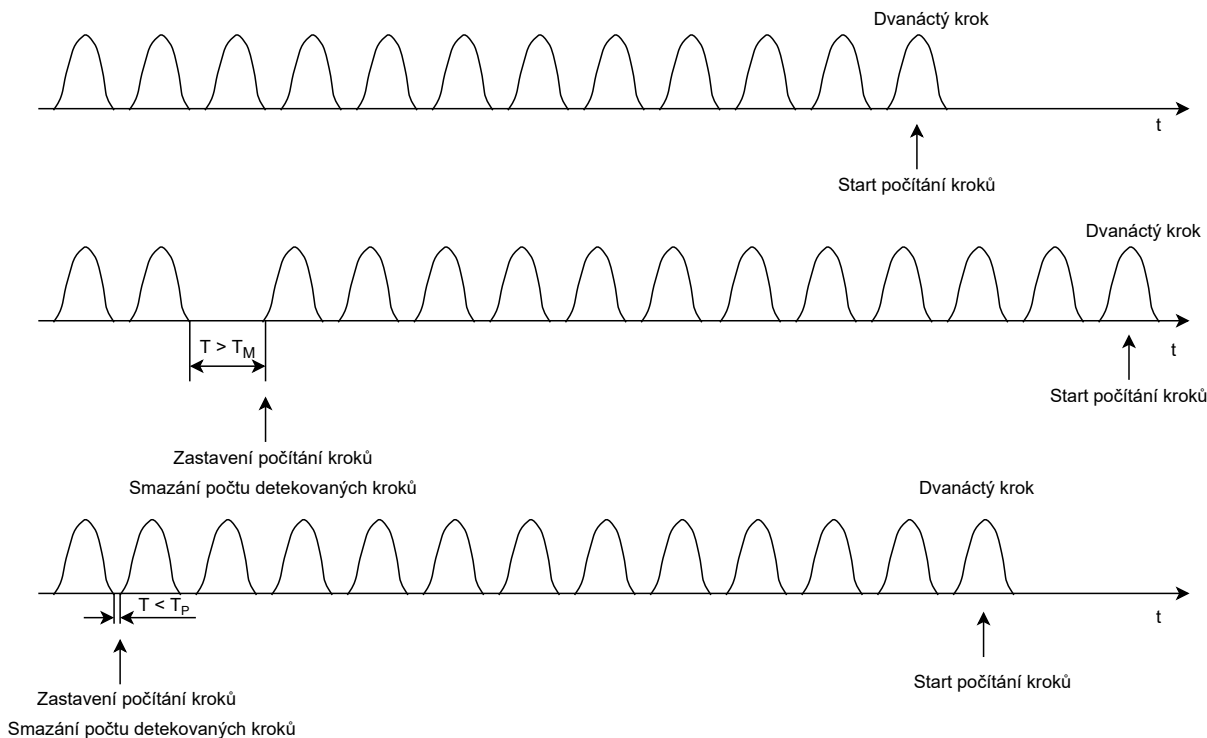


Obrázek 13.12: Časový interval špičky signálu [40]

jsou kroky registrovány. Pro lepší demonstraci je uveden příklad na obrázku 13.13. Zde je parametr STP_TH nastaven na dvanáct kroků (stejné nastavení je v této realizaci). Pokud tedy nastane v sérii za sebou dvanáct kroků (akcelerometr byl předtím v klidu), kdy každý z nich vyhovuje nastaveným parametrům, tak od dvanácti kroků začne počítání každého následujícího kroku. Vzniklé kroky jsou samozřejmě přičítány k hodnotě registrů PED_STP_L a PED_STP_H . Nicméně pokud počet vzniklých kroků v řadě za sebou (opět každý krok vyhovuje určeným parametrům) nepřekročí určenou hranici dvanáct kroků, vzniklé kroky jsou smazány a nejsou přičteny k hodnotě registrů PED_STP_L a PED_STP_H . [40]

Tabulka 13.5: Soupis nastavitelných parametrů a jejich příslušných registrů

Označení	Název registru	Popis parametru
A_L	PED_CNTL5	minimální úroveň špičky signálu pro detekci kroku
A_H	$PED_CNTL4[0:3]$	maximální úroveň špičky signálu pro detekci kroku
M_L	PED_CNTL7	minimální časový interval špičky signálu pro detekci kroku
M_M	PED_CNTL6	maximální časový interval špičky signálu pro detekci kroku
T_P	PED_CNTL10	minimální časový interval mezi dvěma kroky
T_M	PED_CNTL9	maximální časový interval mezi dvěma kroky
STP_TH	$PED_CNTL1[4:6]$	prahová hodnota počtu pro spuštění detekce kroků



Obrázek 13.13: Prahová hodnota STP_TH [40]

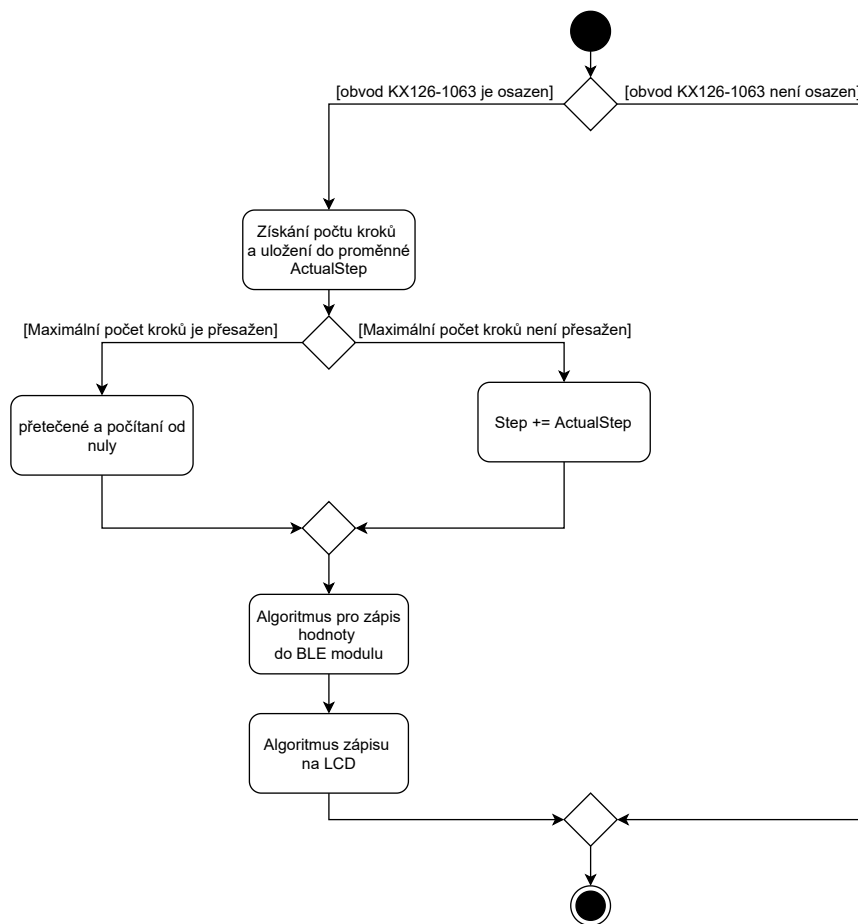
Pro integrovaný krokoměr lze nastavit tzv. ODR. Pomocí tohoto parametru je možné nastavit s jakou frekvencí budou data vzorkována. Pro tuto realizaci bylo zvoleno ODR na 100 Hz.

Pro nastavení výše zmíněných parametrů byla vytvořena inicializační funkce, ve které se provádí zápis do příslušných registrů. Při zápisu parametrů do registru integrovaného obvodu je nutné, aby byl akcelerometr v režimu stand-by a po nadefinování jednotlivých parametrů jej opět spustit. Po implementaci inicializační funkce byla implementována funkce, pomocí které je vyčten obsah registrů PED_STP_H a PED_STP_L. Tato funkce byla pojmenována GetStep.

13.7.1 Algoritmus čtení počtu kroků

Aktivita pro získávání počtu kroků, která je uveden v diagramu aktivit na obrázku 13.1 je podrobněji popsána diagramem na obrázku 13.14. Jedná se o algoritmus obslužné rutiny přerušení (IRQ_PTC7), která je generována akcelerometrem. V diagramu jsou použity dvě proměnné zvané ActualStep a Step. ActualStep je lokální proměnná, do které je ukládán aktuální počet vzniklých kroků vyčtený z akcelerometru při generování přerušení. Proměnná Step je globální proměnná, ve které je uložen celkový počet detekovaných kroků. V obslužné rutině přerušení je z příslušných registrů akcelerometru vyčtena hodnota aktuálního počtu detekovaných kroků a uložena do lokální proměnné ActualStep. Poté je ověřeno, zda přičtením právě získaného počtu kroků (uložené

v proměnné ActualStep) k celkovému počtu kroků (uložené v proměnné Step) nedojde k překročení maximální hodnoty, která činí 999 999 kroků. Pokud dojde k překročení této maximální hodnoty, tak počet kroků je automaticky nulován. Tedy pokud je v proměnné uložen počet korek odpovídající hodnotě 999 995 kroků a z akcelerometru je vyčten počet nově detekovaných kroků odpovídající hodnotě 12 kroků, tak v proměnné step je uložena hodnota 8 kroků. Pokud ale nedojde k překročení maximální hodnoty zobrazené displejem, aktuální počet kroků je přičten k celkovému počtu kroků. Následně probíhá zápis do BLE modulu (v případě, že je připojen chytrý telefon) a pokud je aktivní režim zobrazování počtu kroků, tak je hodnota počtu kroků aktualizována na displeji.



Obrázek 13.14: Diagram aktivit algoritmu čtení počtu kroků

13.7.2 Nulování počtu kroků

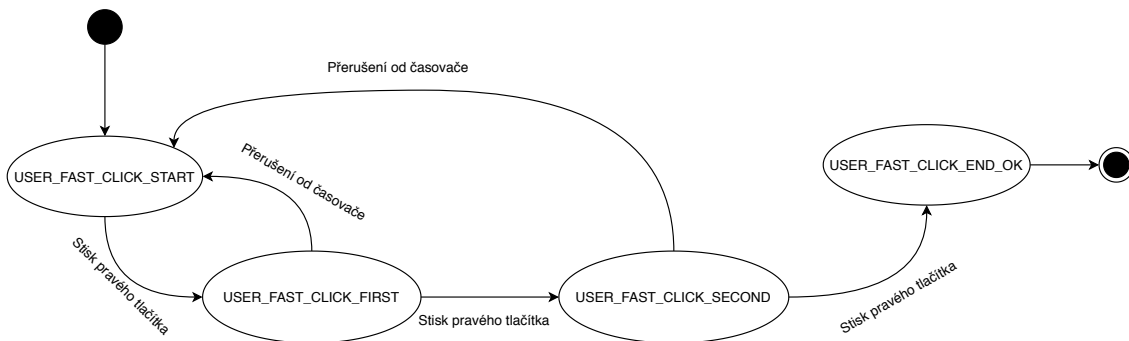
Hodnota počtu kroků je při chůzi stále inkrementována. Bylo tedy nutné navrhnout, kdy a jak budou kroky nulovány. Byly definovány tři možnosti:

- počet kroků dosáhne maximální hodnoty,

- uživatel provede akci pro smazání počtu kroků,
- dojde k vypnutí zařízení, případně jeho restartování.

Nulování počtu kroků při dosažení maximální hodnoty počtu kroků bylo popsáno v kapitole 13.7.1.

V druhé možnosti může uživatel provést nulování počtu kroků. To je možné třemi rychlými stisky pravého tlačítka ve chvíli, kdy je na displeji zobrazen počet kroků. Pro tuto realizaci byla využita periférie časovače implementována v mikrokontroléru. Jedná se o časovač PIT_FAST, zmíněný v kapitole 13.1. Tento časovač je nastaven na časovou periodu 500 ms. V tomto časovém rozmezí musí dojít celkem k třem stiskům pravého tlačítka, aby došlo k nulování počtu kroků. Tato situace je zobrazena ve stavovém diagramu na obrázku 13.15. Jednotlivé názvy stavů korespondují s názvy ve zdrojovém kódu. Při prvním kliknutí pravého tlačítka (na LCD je zobrazen počet kroků) je nastaven bit ResetStep proměnné PitFastProcessStart. Zároveň je spuštěný zmíněný časovač. Pokud je generováno přerušení časovače dříve, než dojde k přechodu do stavu USER_FAST_CLICK_END_OK, tak je spuštěna obslužná rutina časovače. Zde je nastaven stav USER_FAST_CLICK_START a vynulován bit ResetStep proměnné PitFastProcessStart. Pokud ale uživatel stiskne pravé tlačítko třikrát v časovém intervalu 500 ms, zařízení vstoupí do stavu USER_FAST_CLICK_END_OK. V tomto stavu je opět vynulován bit ResetStep proměnné PitFastProcessStart a provedeno nulování počtu kroků.



Obrázek 13.15: Stavový diagram pro smazání počtu kroků uživatelem

13.8 Měření tepové frekvence

Jako poslední z návrhů byl proveden návrh měřicího algoritmu pro měření tepové frekvence. K tomu byl využit integrovaný obvod MAX30003, ve kterém je implementována detekce QRS komplexu zvaná Pan–Tompkins. Stejně jako v předešlých případech byla vytvořena knihovna, ve které jsou obsaženy funkce pro inicializaci integrovaného obvodu MAX30003 a další podpůrné funkce pro práci s obvodem MAX30003. V inicializační funkci je implementováno nastavení příslušných registrů

pro měření tepové frekvence. Dále je inicializováno přerušení (označován INTB), kterému byl zvolen zdroj přerušení od R to R intervalu. Přerušení je generováno při detekci nového R vrcholu. [36]

Po vytvoření inicializační funkce bylo přistoupeno k návrhu měřicího algoritmu. Dle požadavku je měření spuštěno uživatelem pomocí pravého tlačítka, při zvolené položce **BMP** cyklického menu. Měření trvá maximálně patnáct sekund nebo definovaný počet sérií měření. Pokud do určeného času, nebo počtu sérií měření není tepová frekvence změřena správně, je na displeji zobrazena chybová hláška Err3.

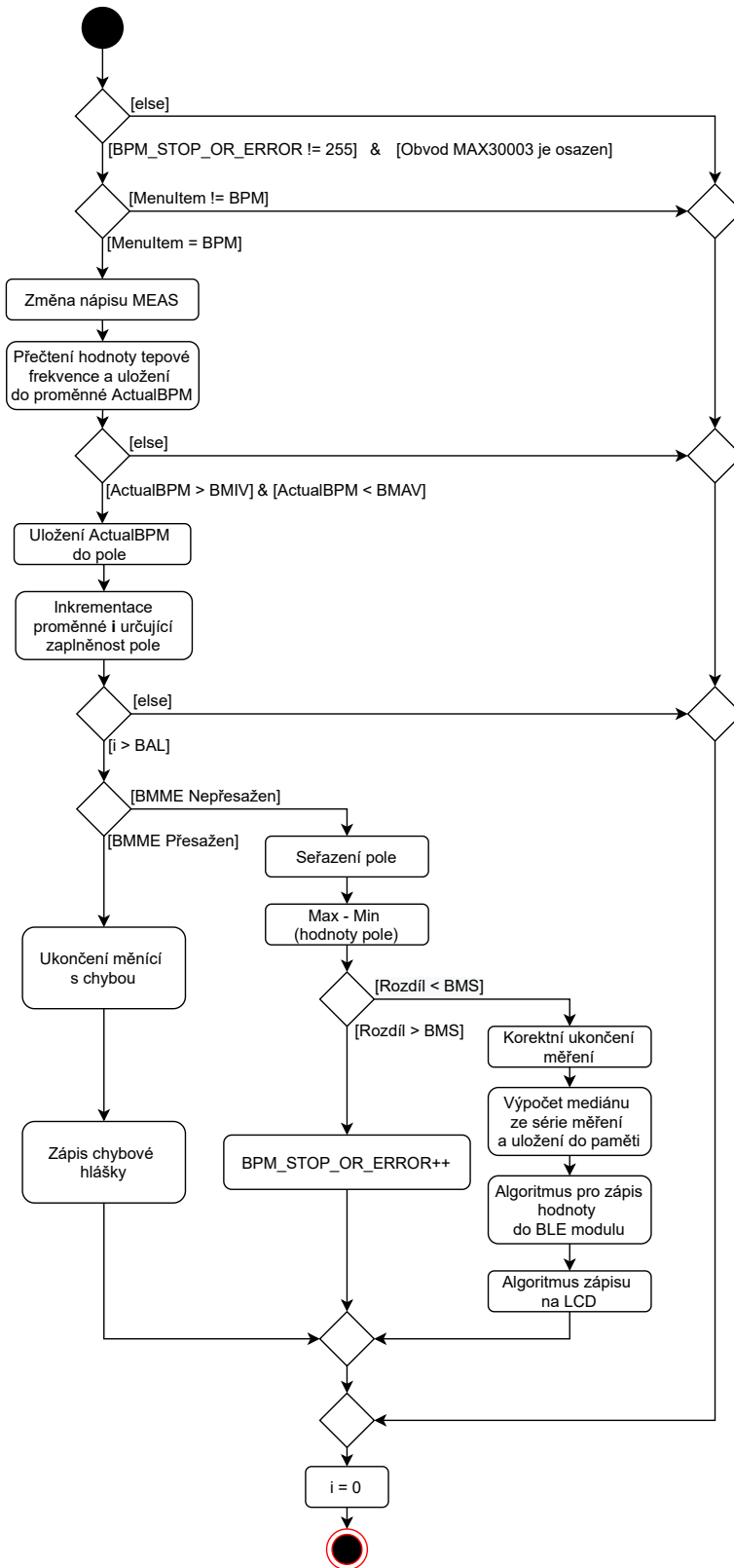
13.8.1 Algoritmus měření tepové frekvence

Nejprve byly vytvořeny konstanty, pomocí kterých je inicializován proces měření. Jedná se o následující konstanty, které definují:

- BPM_ARRAY_LENGTH (BAL): počet měření tepové frekvence pro jednu sérii měření (určuje velikost pole, ve kterém jsou obsaženy hodnoty tepové frekvence pro jednu sérii měření),
- BPM_MIN_VALUE (BMIV): minimální možnou hodnotu tepové frekvence,
- BPM_MAX_VALUE (BMAV): maximální možnou hodnotu tepové frekvence,
- BPM_MAX_SPAN (BMS): maximální rozptyl mezi minimální a maximální naměřenou hodnotou v jedné sérii měření,
- BPM_MAX_MEAS_ERROR (BMME): maximální počet sérií měření.

Dále je pak podstatná proměnná pojmenována BPM_STOP_OR_ERROR. Pomocí této proměnné je signalizováno, zda bylo spuštěno měření či nikoli. Proměnná je inkrementována po každé neúspěšné sérii měření. Pokud tento počet přesáhne hodnotu určenou konstantou BPM_MAX_MEAS_ERROR, měření je ukončeno a na displeji je zobrazena chybová hláška **Err3**. Jako neúspěšný pokus je brána série měření (jedna série měření obsahuje počet naměřených hodnot tepové frekvence definovaných konstantou BPM_ARRAY_LENGTH), kdy rozptyl mezi maximální a minimální hodnotou změřené tepové frekvence je větší, než jak je definováno pomocí konstanty BPM_MAX_SPAN. To může být způsobeno tím, že uživatel při měření není v klidu. Pokud ale je změřená tepová frekvence menší, než hodnota určená konstantou BPM_MIN_VALUE (nebo naopak větší než hodnota konstanty BPM_MAX_VALUE), hodnota není zahrnuta do série měření. Je provedeno opětovné měření. Pokud naměřené hodnoty srdečního tepu vyhovují všem zmíněným parametrům, je z naměřených hodnot vypočten medián jako výsledná hodnota srdečního tepu.

Celý měřicí algoritmus (vyvolán přerušením INTB) je zobrazen diagramem aktivit na obrázku 13.16. V první řadě je vyčtena tepová frekvence z registru zvaného RTOR. Hodnota časové délky mezi dvěma QRS komplexy má velikost čtrnácti bitů a je uložena jako počet intervalů délky 7,8125 ms (vychází z inicializace). Pro získání R to R intervalu v milisekundách, musí být vyčtená



Obrázek 13.16: Diagram aktivit algoritmu měření tepové frekvence

hodnota z registru vynásobena s hodnotou délky jednoho intervalu (7,8125 ms). Vypočtená hodnota ($RtoR_{ms}$) je dále přepočtena na údery za minutu (BPM):

$$BPM = \frac{60}{\frac{RtoR_{ms}}{1000}}. \quad (13.6)$$

Zmíněný výpočet je uveden v katalogu výrobce [36].

Vypočtená hodnota BPM z rovnice 13.6 je vrácena jako návratová hodnota vytvořené funkce GetBPM. Pomocí funkce je vyčten počet intervalů z registru RTOR a proveden přepočet na údery za minutu. Měření tepové frekvence nesmí být delší než patnáct sekund. Proto, když uživatel spouští tlačítkem měření tepové frekvence, je spuštěna periferie časovače PIT_FAST a nastaven bit StopMeasBPM v proměnné PitFastProcessStart popsané v kapitole 13.1. Jelikož tento časovač je nastaven na 500 ms, je zde implementováno počítadlo průchodu. Pokud dojde k třiceti průchodům, je měřící sekvence ukončena s chybou a bit StopMeasBPM je vynulován. [36]

13.9 Identifikační číslo zařízení

Zařízení bude vyráběno po více kusech. Pro každé zařízení bylo tedy nutné vytvořit unikátní identifikační číslo, pomocí kterého lze v mobilní aplikaci dané zařízení jednoduše vyhledat. Dále bylo požadováno, aby bylo toto číslo možné zobrazit na displej zařízení. Tedy počet míst identifikačního čísla je omezen pouze na čtyři místa. Na základě tohoto ustanovení bylo rozhodnuto, že číslo bude zobrazováno v hexadecimální soustavě. Maximálně může být tedy vyrobeno 65 535 kusů. Přičemž reálný počet vyrobených kusů se bude pohybovat ve stovkách, což nepřesahuje maximální počet možných identifikačních čísel. Hodnota identifikačního čísla 0000 určuje, že zařízení nebylo přiřazeno identifikační číslo.

Po návrhu struktury identifikačního čísla bylo přistoupeno k jeho implementaci. V první řadě byla vytvořena v linker skriptu nová výstupní sekce s názvem serial o velikosti 2 bajty umístěná ve flash paměti. Tato sekce začíná na adrese 0x410 a končí na adrese 0x411. V kódu byla vytvořena šestnáctibitová proměnná, která byla přiřazena k sekci .serial.



Obrázek 13.17: Grafické rozhraní pro editaci sériového čísla

Nyní tedy na adresách 0x410 a 0x411 je obsaženo identifikační číslo. Pro editaci identifikačního čísla byl vytvořen skript v pythonu 3.8.5. Pomocí tohoto skriptu je možné načíst binární soubor a vyčíst aktuální hodnotu sériového čísla. Dále je možné toto číslo aktualizovat, tedy zvětšit

jeho hodnotu o jedna, případně zmenšit o jedna. Při aktualizování je již nové identifikační číslo zapsáno do binárního souboru. Pro tuto aplikaci bylo vytvořeno jednoduché grafické rozhraní zobrazené na obrázku 13.17.

13.10 Komunikace mezi BLE modulem a mikrokontrolérem

Po návrhu a implementaci algoritmu sběru dat z příslušných obvodů a jejich následné úpravě bylo přistoupeno k návrhu a realizaci přenosu naměřených dat pomocí bezdrátové komunikace BLE. K tomu byl využit již zmíněný integrovaný obvod BLE112 a pro komunikaci s mikrokontrolérem bylo využito komunikační rozhraní UART. Komunikace mezi mikrokontrolérem a BLE modulem je obousměrná. Z mikrokontroléru se do BLE modulu přenáší naměřená data získaná ze snímačů, verze osazení a HexID. Naopak z BLE modulu je přenášena informace o připojení chytrého telefonu a MAC adresa BLE modulu.

Tabulka 13.6: Datový rámeček zprávy pro zaslání hodnoty do BLE modulu

Bajt	0	1 ... N	N+1	N+2
Popis	Typ zprávy	Data	Kontrolní součet	Ukončovací znak

Tabulka 13.7: Datový rámeček zprávy pro zaslání dat z BLE modulu

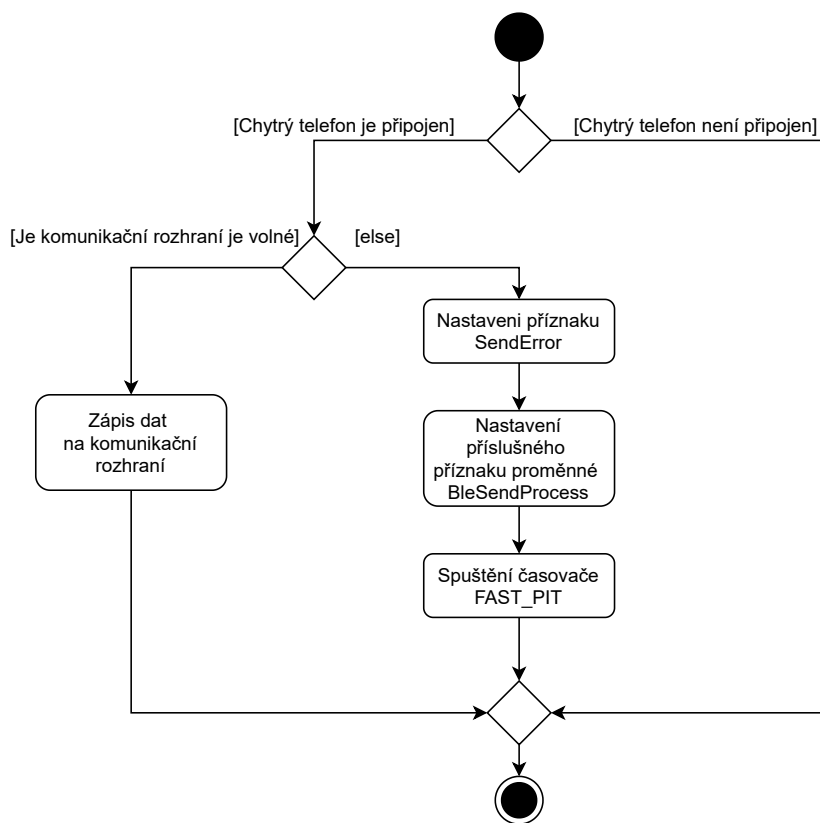
Bajt	0	(1 ... N)	1 (N+1)
Popis	Typ zprávy	(Data)	Ukončovací znak

Tabulka 13.8: Typ zprávy

Hodnota v hex	Popis
0x7E	Teplota
0x2A	Počet kroků
0x2F	Srdeční tep
0x26	hexID zařízení
0x80	Mobilní zařízení je odpojeno
0x81	Mobilní zařízení je připojeno
0x2D	MAC adresa zařízení

V prvé řadě bylo tedy nutné navrhnout komunikační protokol mezi mikrokontrolérem a BLE modulem. Bylo rozhodnuto, že přenášená data budou kódována v ASCII kódování. Datový rámeček zprávy odesílané z mikrokontroléru do BLE modulu je zobrazen v tabulce 13.6. První bajt datového rámečku určuje typ zasílané zprávy, kterým je definován začátek řetězce. Za tímto znakem následují data. Délka jednotlivých dat je variabilní a je závislá na typu přenášené zprávy. Následně po datech je zaslán kontrolní součet a celý řetězec je ukončen ukončovacím znakem, jehož hexadecimální

hodnota je 0x03. Kontrolní součet je prováděn operací XOR mezi jednotlivými bajty dat (tedy mezi 1 a N bajtem). Celkem jsou do BLE modulu zasílány čtyři zprávy. Jedná se o hodnotu změřené teploty, tepové frekvence, počet změřených kroků a hexID zařízení s verzí osazení. Informace o verzi osazení je přenášena společně s HexID. V prvním bajtu (podle mojí analogie bajt 1, tedy hned za typem zprávy) je obsažena verze osazení a v následujících čtyřech bajtech je definováno HexID zařízení. V tabulce 13.8 jsou zobrazeny hexadecimální hodnoty označující daný typ zprávy. V tabulce jsou také označeny typy zprávy, které jsou využity pro zasílání zpráv z BLE modulu do mikrokontroléru. Dále pak v tabulce 13.7 je uveden datový rámec zprávy zasílané z BLE modulu do mikrokontroléru. Jako první znak je uveden typ zprávy. Dále pak část data je volitelná. Využívá se pouze pro příkaz přenosu MAC adresy BLE modulu. MAC adresa je posílána bez kontrolního součtu, jelikož je předem známa její délka. Pokud tedy přijatá MAC adresa z BLE modulu nemá požadovanou délku, v paměti mikrokontroléru je MAC adresa uložena s nulovými hodnotami. Tím je signalizováno špatné přečtení MAC adresy. MAC adresa je potřebná pouze v testovací stanici. Pokud je tedy pomocí testovací stanice přečtena MAC adresa, která obsahuje pouze nuly, musí dojít k restartování BLE modulu. Restartování BLE modulu může být provedeno testovací stanicí zadáním příkazu pro restartování BLE modulu (viz kapitola 13.12). Pro ukončení rámce je využita hodnota v hexadecimální soustavě 0x03.



Obrázek 13.18: Diagram aktivit algoritmu měření zápisu hodnoty do BLE modulu

Jednotlivé změřené hodnoty (teplota, tepová frekvence, počet kroků) jsou posílány nezávisle na sobě. Bylo tedy nutné zamezit situaci, kdy je prováděn přenos dat přes komunikační sběrnici, a zároveň vznikne požadavek na posílání nových dat. Tato data mohou být poslána poté, co bude komunikační linka volná. Tato situace byla ošetřena následovně. V první řadě byla vytvořena nová proměnná s názvem `BleSendProcess`. V proměnné jsou obsaženy bity signalizující měřené hodnoty, které musí být znovu poslány. Tato proměnná je zobrazena v tabulce 13.9 s popisem jednotlivých bitů. Pokud má být změřená hodnota zapsána do BLE modulu je první ověřeno, jestli je komunikační linka volná. Pokud ano, může proběhnout zápis. Nicméně pokud ne, změřená hodnota není poslána, ale pouze uložena v paměti mikrokontroléru. Dále je povolena periferie časovače `PIT_FAST`, je nastaven bit `SendError` v proměnné `PitFastProcessStart` a je nastaven příslušný bit pro znovu poslání v proměnné `BleSendProcess`. Proměnná `PitFastProcessStart` i časovač `PIT_FAST` jsou popsány v kapitole 13.1. Poté, co je vyvolána obslužná rutina časovače, je opět zkontrolována dostupnost komunikační linky. A následuje stejný scénář. Pokud je linka volná, data mohou být poslána a je nulován příslušný bit. Nicméně pokud komunikační linka není volná, hodnota není zapsána. Poslání hodnoty je tedy opět odloženo na další přerušení časovače. Bit `SendError` může být nulován pouze tehdy, pokud není aktivní ani jeden z bitů v proměnné `BleSendProcess`.

Dále bylo nutné ošetřit reakci na vzniklou chybu v komunikaci signalizovanou signálem CTS. Při tvorbě programového vybavení bylo zjištěno následující chování. Pokud BLE modul není schopen správně přijímat data, je tato skutečnost signalizována změnou úrovně CTS signálu na log. 1. Nicméně již byl chybný bajt odeslán a přečten v BLE modulu. Proto pokud nastane situace, že signál CTS je aktivní, ukončí se posílání hodnoty a je nastaven bit `SendError` v proměnné `PitFastProcessStart` a příslušný bit v proměnné `BleSendProcess`. Poté je spuštěn časovač `PIT_FAST`. Nyní se již postupuje stejně jako kdyby hodnota nebyla poslána z důvodu obsazenosti komunikačního rozhraní.

Tabulka 13.9: Přehled obsahu jednotlivých bitů proměnné `BleSendProcess`

Bit	7..4	3	2	1	0
Název	X	SEND_ID	SEND_BPM	SEND_STEP	SEND_TEMP

13.11 Programové vybavení BLE112

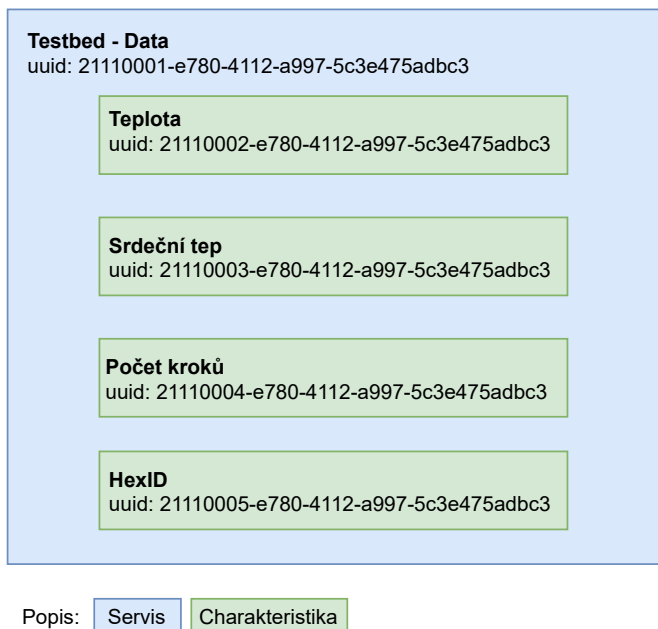
Po návrhu komunikace mezi mikrokontrolérem a BLE modulem bylo přistoupeno k návrhu a posléze implementaci programového vybavení BLE modulu. V programovém vybavení byly navrženy čtyři události. Jedná se o události, které jsou vyvolány při připojení nebo odpojení chytrého telefonu, zapnutí modulu, detekci příchozích dat na UART periferii.

V událost, která je vyvolána při zapnutí BLE modulu je provedena inicializace modulu. Zde je povolen advertisement mód a Undirected-Connectable mód. Poté je povoleno párování a je povolena

UART periferie (s nastavením watermarku) a pin RST je nastaven jako výstupní. V poslední části události je přečtena MAC adresa BLE modulu a následně poslána pomocí UART rozhraní do mikrokontroléru. Datový rámec zasílání zprávy z BLE modulu je zobrazen v tabulce 13.7.

Po návrhu startovací události byl vytvořen algoritmus pro událost, pomocí které jsou detekovány příchozí data na UART periférii. Pokud jsou detekována příchozí data, je tato událost vyvolána. V obslužné rutině události byl implementován algoritmus pro zpracování příchozích dat a jejich následný zápis do příslušných charakteristik GATT, které byly vytvořeny. V první řadě jsou vyčtena příchozí data na periférii UART. Následně je detekován typ zprávy podle prvního bajtu a proveden kontrolní součet. Pokud vypočtený kontrolní součet je roven příchozímu součtu a zároveň je detekován konec příchozího řetězce, příchozí data jsou zapsána do příslušných charakteristik GATT.

V posledních dvou událostech (pomocí kterých je detekováno připojení, nebo naopak odpojení chytrého telefonu od BLE modulu) je obsaženo pouze odeslání informace o připojení přes komunikační sběrnici. Pokud se chytrý telefon připojí k BLE modulu, je poslána hodnota v hexadecimální soustavě 0x80. V opačné případě, tedy při odpojení chytrého telefonu od BLE modulu, je poslána hodnota v hexadecimální soustavě 0x81. Hodnoty jsou posílány v datovém rámci zobrazeném v tabulce 13.7.



Obrázek 13.19: Struktura vytvořeného servisu a jeho charakteristiky

Dále byly pro tuto realizaci vytvořeny vlastní servisy a charakteristiky. Pro jedinečné identifikační číslo byl použit generátor 128 bitového UUID. Servis a příslušné charakteristiky s jejich UUID jsou zobrazeny na obrázku 13.19.

13.12 Komunikační protokol pro testovací stanici

Zařízení bude ve výrobní lince testováno v testovací stanici. Dle požadavků tvůrců testovací stanice byla navržena a implementována komunikace mezi zařízením a testovací stanicí. Bylo rozhodnuto, že pro realizaci komunikace bude využito komunikační rozhraní UART. Dále byl navržen komunikační protokol, který byl následně implementován v mikrokontroléru. Bylo navrženo textově orientovaný komunikační protokol. Komunikace je koncipována tak, že ze strany testovací stanice je zadán příkaz, a následně je ze strany zařízení poslána odpověď. Byla tedy vytvořena struktura příkazů a odpovědí. Dále byly definovány jednotlivé příkazy podle požadavků tvůrce testovací sekvence.

Tabulka 13.10: Struktura příkazu bez parametru

Bajt	0	1	2
Popis	Počáteční znak	Příkaz	Ukončovací znak

Tabulka 13.11: Soupis příkazů bez parametru

Znak	Hodnota v Hex	Popis
!	0x21	Zahájení komunikace s testovací stanicí
"	0x22	Ukončení komunikace s testovací stanicí
#	0x23	Detekce zdroje napájení
%	0x25	Spuštění rutiny pro sledování vibrací
&	0x26	Změření aktuální teploty
'	0x27	Vyžádání hodnoty poslední změřené tepové frekvence
*	0x2A	Funkce levého tlačítka
+	0x2B	Funkce pravého tlačítka
-	0x2D	Vyžádání MAC adresy BLE modulu BLE112
.	0x2E	Akce pro restartování BLE modulu BLE112

Struktura řetězce příkazů může mít dva tvary. Jedním z tvarů je příkaz bez parametru, který je zobrazen v tabulce 13.10. Skládá se ze tří bajtů, kdy první bajt je počáteční znak, který nesmí být použitý nikde dále v řetězci. K tomu byl vybrán znak označující začátek textu. Hexadecimální hodnota toho znaku je v hexadecimální soustavě 0x02. Po počátečním znaku následuje bajt určující příkaz, který má být proveden. Soupis všech příkazů bez parametru je zobrazen v tabulce 13.11. Na konec řetězce je umístěn ukončovací znak, pro který byl opět vybrán znak určující konec textu. Hexadecimální hodnota znaku je v hexadecimální soustavě 0x03. Druhým tvarem je příkaz s parametrem. V příkazu jsou obsaženy všechny náležitosti, které jsou u příkazu bez parametru. Navíc je zde možné zadat parametr příkazu, který je ukončen znakem '\0'. Struktura příkazu je zobrazena v tabulce 13.12. Soupis všech příkazů s parametrem je uvedena v tabulce 13.13. Obdobná struktura řetězce je využita i pro odpověď (tabulka 13.14), která je zaslána do testovací stanice jako reakce na zadaný příkaz. Začátek řetězce je označen počátečním znakem. Poté následuje bajt určující pří-

kaz, který byl zadán testovací stanicí. Za tímto bajtem následuje posloupnost bajtů, ve kterých je obsažena odpověď na zadaný příkaz, která je ukončena znakem '\0'. Na konec řetězce je umístěn ukončovací znak. Soupis všech odpovědí je uveden v tabulce 13.15.

Tabulka 13.12: Struktura příkazu s parametrem

bajt	0	1	2	...	N	N+1
Popis	Počáteční znak	Příkaz	Parametr			Ukončovací znak

Tabulka 13.13: Soupis příkazů s parametrem

Znak	Hodnota v Hex	Popis	Příkaz
)	0x29	Ovládání LED	"BF"- Vypnout modrou LED
			"BT"- Zapnout modrou LED
			"GF"- Vypnout zelenou LED
			"GT"- Zapnout zelenou LED
			"WF"- Vypnout bílou LED
			"WT"- Zapnout bílou LED
/	0x2F	Režim spotřeby BLE112	"L"-Nízká spotřeba
			"H"- Vysoká spotřeba

Následně bylo rozhodnuto, že komunikace musí být zahájena příkazem, jehož hexadecimální hodnota je 0x21. Pokud není tento příkaz zadán, není možné využívat další příkazy. Odpověď na tento příkaz je hodnota HexID a verze osazení. Pro ukončení komunikace je nutné komunikaci ukončit daným příkazem (0x22). Díky této skutečnosti je v programovém vybavení zařízení jasně oddělen normální běh zařízení a běh zařízení v testovací stanici. Dále bylo nutné zajistit nedefinované stavy a příslušnou reakci na tyto stavy. Jedním z nedefinovaných stavů je zadání špatného příkazu. Pokud tak nastane, je vyvolána odpověď, která má stejnou strukturu jako je uvedena v tabulce 13.14. Jako příkaz je uveden příkaz, který je odpovídáno. V datovém rámci je zapsána hodnota Err1. Jedná se o chybovou hlášku signalizující zadání špatného příkazu. Dalším nedefinovaným stavem je zadání špatné hodnoty parametru v příkazu s parametrem. Odpověď na tento příkaz je obdobná jako v prvním případě. Nicméně v datovém rámci je obsažena chybová hláška Err2. Dále je možné, že je oslovený integrovaný obvod, který ale není v dané verzi osazen. V takovém případě je v odpovědi vrácena chybová hláška Err3.

V komunikačním protokolu není implementován příkaz pro vyčtení hodnoty počtu kroků. V testovací stanici není možné simulovat pohyb potřebný pro detekci kroků, který byl definován v kapitole 13.7. Proto pro otestování funkčnosti akcelerometru je testovací lůžko rozvibrováno. Pro detekci těchto vibrací je určen příkaz, kde hexadecimální hodnota je 0x25.

Tabulka 13.14: Struktura odpovědi

Bajt	0	1	2	...	N	N+1
Popis	Počáteční znak	Příkaz	Odpověď			Ukončovací znak

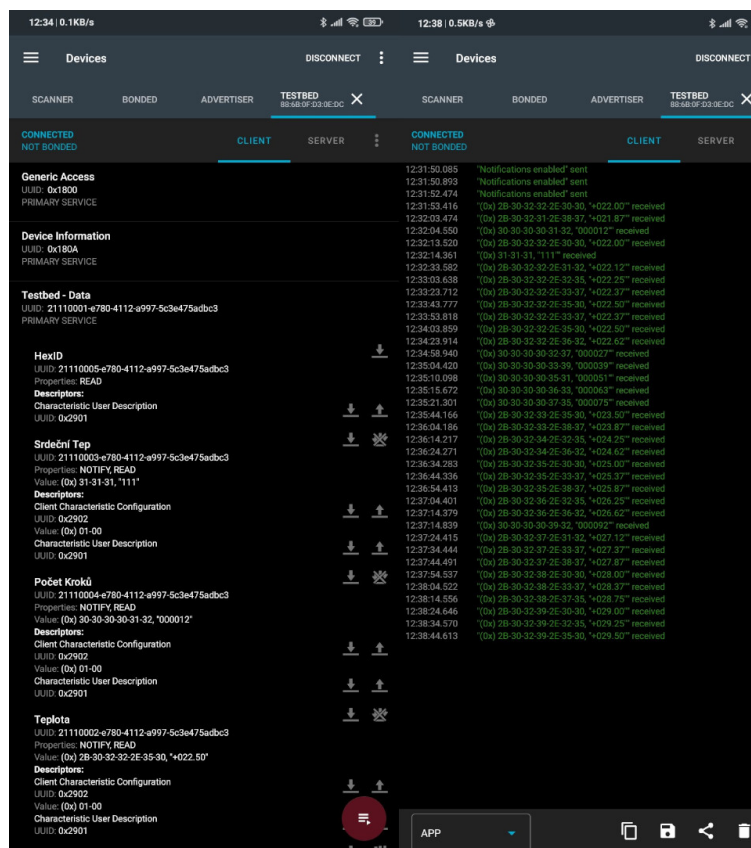
Tabulka 13.15: Soupis odpovědí

Odpověď na příkaz	Hodnota příkazu v Hex	Data
!	0x21	-5 bytové číslo např. 7A4F8 - Nejvyšší bajt určuje verzi osazení může nabývat hodnot: - "7"- plná verze osazení - "4"- pouze krokoměr - "1"- pouze teploměr - Zbylé 4 byty obsahují HexID
"	0x22	- "GOOD"- Příkaz proběhl zprávně
#	0x23	- "BAT"- Napájení z baterie - "USB"- napájení z USB
%	0x25	- "TRUE"- Vibrace byli detekovány - "FALSE"- Vibrace nebyli detekovány do 5 s - "Err3"- Integrovaný obvod není osazen
&	0x26	- Teplota ve formátu "+25.75"nebo "-10.75" - Teplota je uvedena ve °C - "Err3"- Integrovaný obvod není osazen
'	0x27	- Hodnota tepové frekvence ve formátu "89" - "MEAS"- právě probíhá měření tepové frekvence - "Err3"- Integrovaný obvod není osazen
)	0x29	- "GOOD"- vše je v pořádku
*	0x2A	- "GOOD"- vše je v pořádku
+	0x2B	- "GOOD"- vše je v pořádku
,	0x2C	- "GOOD"- vše je v pořádku
-	0x2D	- MAC adresa ve formátu "CC:CC:CC:FE:0A:B4"
.	0x2E	- "GOOD"- vše je ve pořádku
/	0x2F	- "GOOD"- vše je v pořádku

Kapitola 14

Testování navrženého produktu

Po zhotovení návrhu DPS a implementování programového vybavení bylo přistoupeno k testování funkčnosti produktu. Byla provedena experimentální měření při turistice, kdy byly zaznamenávány přijaté hodnoty teploty a počtu kroků chytrým telefonem. Tato data byla následně kontrolována, zda nedošlo k chybě, ať už při přenosu dat nebo při procesu měření. Tímto způsobem byla otestována

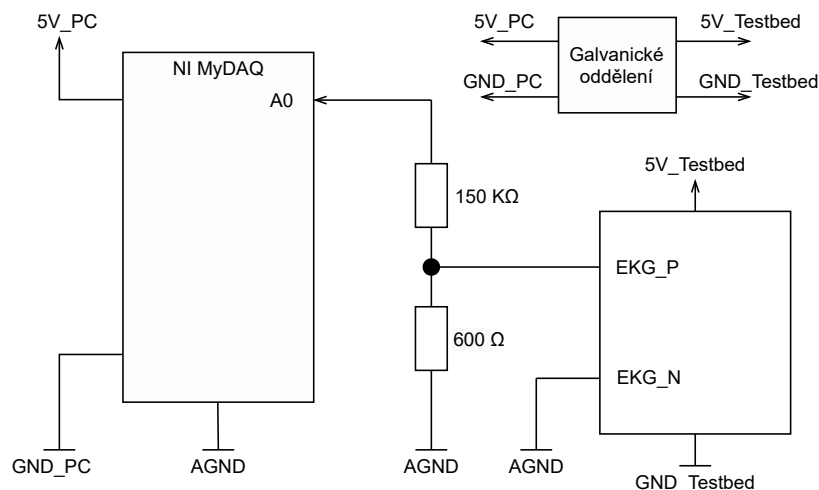


Obrázek 14.1: Snímek z aplikace nRF Connect for Mobile

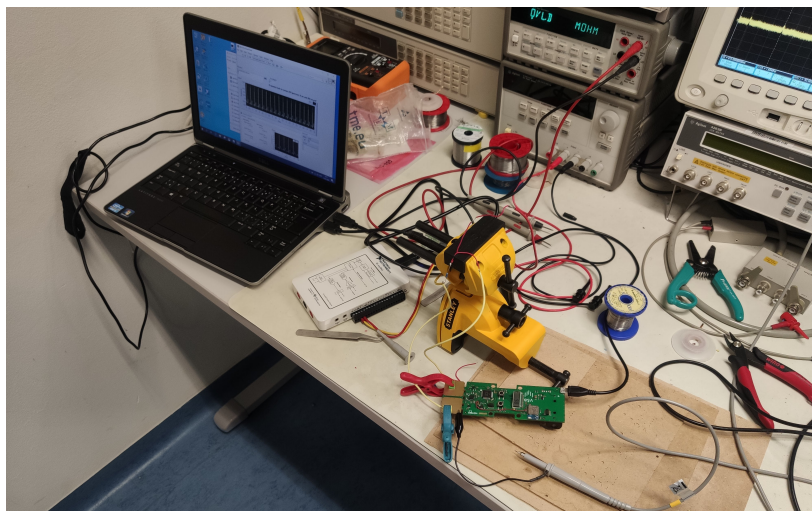
funkčnost navržených měřících algoritmů a správnost přenosu dat pomocí bezdrátové komunikace pro teploměr a krokoměr. Oba snímače jsou kalibrovány výrobcem při výrobě. Pro testování přenosu dat mezi chytrým telefonem a zařízením byla využita aplikace nRF Connect for Mobile dostupná na Google Play. Testování probíhalo na verzích 4.24.2 a 4.24.3 této aplikace. Pro testování nebyla využita vyvíjená aplikace určená pro toto zařízení, jelikož v době testování byla v rané fázi vývoje. Pomocí aplikace nRF Connect for Mobile je možné si zobrazit jednotlivé servery a jejich charakteristiky. V aplikaci je možno vidět historii příchozích dat, které je možno exportovat do textového souboru. Prostředí aplikace je zobrazeno na obrázku 14.1. Posléze bylo přistoupeno k testování měřícího algoritmu pro měření tepové frekvence a navrženého komunikačního protokolu s testovací stanicí. Zmíněná testování jsou popsána v následujících kapitolách.

14.1 Měření tepové frekvence generovaného EKG signálu

Pro ověření funkčnosti měřícího algoritmu měření tepové frekvence byl využit program vytvořený v Labview 2013 s biomedicínskou sadou nástrojů (LabVIEW Biomedical Toolkit 2013). Program byl vytvořen tvůrci testovací stanice a obsahoval generování EKG signálu na výstupní analogový pin karty NI myDAQ. Na obrázku 14.2 je zobrazeno schéma propojení NI karty s EKG ploškami zařízení. V programu bylo možné měnit hodnotu tepové frekvence generovaného EKG signálu. Bylo provedeno několik měření v různých hodnotách tepové frekvence. Fotografie fyzického propojení je zobrazena na obrázku 14.3.



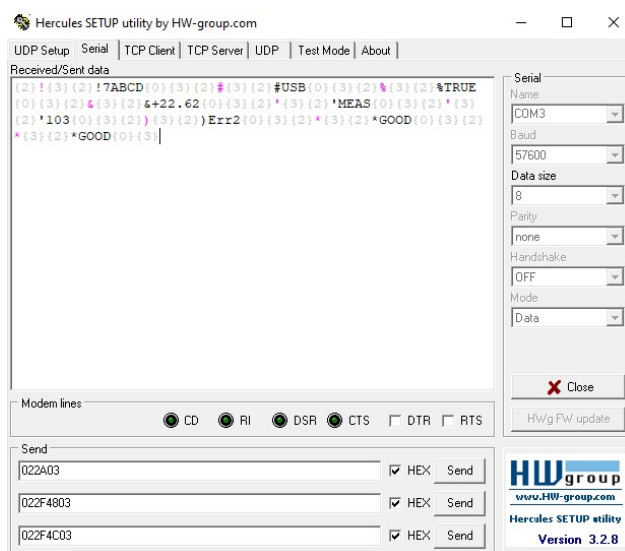
Obrázek 14.2: Schéma propojení měřící karty a zařízení Testbed



Obrázek 14.3: Fotografie pracoviště pro měření generovaného signálu EKG

14.2 Testování komunikačního protokolu pro testovací stanici

Navržený komunikační protokol pro komunikaci mezi zařízením a testovací stanicí byl testován pomocí USB převodníku komunikačního rozhraní UART označovaný UM232R FTDI 2005. Byl využíván program Hercules ve verzi 3.2.8. Záznam z testování je uveden na obrázku 14.4. Příkazy byly zadávány v hexadecimálním tvaru. Postupně byly provedeny veškeré příkazy a zkontrolovány jejich odpovědi. Správnost navržení komunikačního protokolu a funkce příkazů byla otestována v navržené testovací stanici. Testování probíhalo za kontroly tvůrců testovací stanice a byly provedeny všechny testovací rutiny.



Obrázek 14.4: Snímek z programu Hercules při testování komunikačního protokolu

Kapitola 15

Závěr

Cílem mé práce byl návrh a konstrukce testovacího produktu pro výrobní linku s prvky Průmyslu 4.0, která je umístěna v budově CPIT TL3 porubského areálu VŠB. V první části jsem se zaměřil na teorii týkající se principů měření teploty, počtu kroků a tepové frekvence. Následně jsem se zaobíral architekturou a programováním mikrokontroléru s jádrem ARM Cortex-M0+, kde jsem popsal základní vlastnosti jádra, architekturu MCU Von Neumann, řetězené zpracování instrukcí a instrukční sadu. Dále jsem se obeznámil s jednotlivými komunikačními rozhraními, která jsou využívána ve zvolených snímačích, popř. v komunikační modul. Prostudoval jsem segmentové LCD s multiplexovaným řízením a technologii Bluetooth Low Energy. V neposlední řadě jsem se seznámil s funkcí softwarových vývojových nástrojů, které jsem využil při tvorbě návrhu testovacího produktu.

Po nastudování teorie jsem přistoupil k návrhu elektrického zapojení a motivu plošného spoje. Nejprve jsem provedl průzkum trhu za účelem výběru komponent. Při výběru komponent jsem kladl největší důraz na nízkou energetickou náročnost komponenty. Proto byl například zvolen segmentový LCD displej, který navíc vyhovoval svými rozměry. Dále pak zvolený akcelerometr obsahuje integrovanou periferii krokoměru, díky které byl usnadněn návrh programového vybavení a tím pádem snížena celková spotřeba zařízení. Nebylo nutné implementovat algoritmus pro detekci kroku. Z podobného důvodu jsem vybral integrovaný obvod MAX30003, který je využit pro měření tepové frekvence. V integrovaném obvodu MAX30003 je implementován detekční algoritmus QRS komplexu zvaný Pan-Tompkins. Tato skutečnost má opět kladný vliv na celkovou spotřebu elektrické energie. Po provedení průzkumu trhu a výběru jednotlivých komponent jsem přistoupil k návrhu elektrického zapojení a motivu plošného spoje. Pro tento návrh jsem využil EDA program Altium Designer ve verzi 14.3. V první řadě jsem vytvořil knihovnu vybraných komponent a následně schéma elektrického zapojení. Poté jsem přistoupil k tvorbě motivu plošného spoje dle zadaných požadavků na rozměr zařízení. Podle složitosti zapojení jsem rozhodl, že DPS bude obsahovat čtyři vrstvy motivu. Jako povrchovou úpravu jsem zvolil galvanické pozlacení, u kterého nehrozí opotřebení. Nicméně tento typ povrchové úpravy je dražší než úprava H.A.S.L., proto tato úprava byla použita pouze pro verzi s EKG ploškami. U ostatních typů osazení jsem dal přednost levnější vari-

antě H.A.S.L. Pro výrobu DPS byla vybrána firma MARPOS, s.r.o., které jsem můj návrh předal. Po zhotovení návrhu DPS a jeho výrobě jsem přistoupil k návrhu programového vybavení. Návrh jsem koncipoval jako jeden program pro více verzí osazení zařízení. Program je sice složitější, ale výhodou tohoto řešení je možnost využití jednoho programu pro všechny verze osazení. Nejprve jsem navrhl algoritmus časovačů, které jsem nazval PIT_SLOW a PIT_FAST. Následně jsem navrhl algoritmus pro detekci zdroje napájení, ke kterému byla využita integrovaná periferie komparátoru v mikrokontroléru. Dále jsem provedl návrh ovládání zařízení. Pro ovládání zařízení je možné využít dva tlačítka a pro podání informace uživateli sedmsegmentový LCD displej. Nicméně bylo zde několik omezení, ať už v oblasti zobrazování textu nebo počtu tlačítek. Proto jsem navrhl, že zařízení bude obsahovat cyklické menu, které usnadnilo přepínání mezi zobrazováním jednotlivých měřených hodnot. Po návrhu ovládání zařízení jsem přistoupil k návrhu měřících algoritmů zvolených snímačů. Navržené měřící algoritmy jsou v souladu s požadavky zadavatele. Jako první jsem provedl návrh algoritmu měření teploty. Následně jsem přistoupil k návrhu algoritmu měření počtu vykonaných kroků, kde jsem využil periferii integrovaného krokoměru ve zvoleném akcelerometru, která výrazně zjednodušila návrh algoritmu měření počtu kroků. Obdobně jsem využil implementovaný algoritmus detekce QRS komplexu ve zvoleném obvodu MAX30003. Čtení a následné zpracování jednotlivých na měřeních hodnot jsem implementoval v příslušných obslužných rutinách přerušení. Následně po návrhu měřících algoritmů jsem navrhl komunikační protokol (mezi mikrokontrolérem a BLE modulem) a programové vybavení BLE modulu. Vygenerované UUID jednotlivých servisů s jejich popisem jsem předal tvůrci mobilní aplikace. V neposlední řadě jsem provedl návrh komunikačního protokolu mezi mikrokontrolérem a testovací stanicí. Při návrhu jsem bral v potaz požadavky tvůrců testovací stanice. Pro implementaci návrhu jsem využil softwarové podpory pracovníka katedry pro zvolený typ mikrokontroléru. Na základě této podpory jsem implementovat mé řešení. Celkem byly vyrobeny dvě prototypové verze, na základě kterých byly doladěny veškeré požadavky ze strany návrhářů výrobní linky a návrhářů krabičky pro zařízení.

Po implementování návrhu jsem provedl několik testů, pomocí kterých byla ověřena funkčnost zařízení. Zařízení jsem testoval při běžné turistice, kdy naměřená data byla přijímána pomocí aplikace nRF Connect for Mobile, a posléze kontrolována. Pro zařízení je vyvinuta aplikace, která ale v období testování zařízení byla stále ve vývoji, proto pro testování nebyla použita. Na základě zmíněného testování jsem ověřil funkčnost měřícího algoritmu pro měření teploty a počtu kroků, kdy přesnost měření je dána kalibrací výrobce při výrobě. Pro testování měření tepové frekvence jsem využil program dodaný tvůrci testovací stanice, pomocí kterého bylo možné simulovat EKG signál na výstupní analogové piny karty NI myDAQ. Takto generovaný signál byl přiveden na EKG plošky mnou navrženého zařízení a měřen generovaný signál. Tímto testem byla ověřena funkčnost navrženého algoritmu měření tepové frekvence. Zařízení bylo dále otestováno v navržené testovací stanici při asistenci tvůrců testovací stanice. Byly provedeny všechny definované testy, jejichž výsledky potvrdily funkčnost zařízení. Pro testovací stanici jsem dále pak vytvořil celkem tři úmyslně poškozená zařízení, která mají reprezentovat chybné výrobky ve výrobní lince. Jedná se pouze o softwarová

poškození, která zamezují komunikaci s jednotlivými snímači.

Jak již bylo zmíněno mnou vytvořené zařízení je součástí výrobní linky s prvky Průmyslu 4.0. V lince koluje několik typů těchto zařízení s různou verzí osazení. Celkem bylo vyrobeno 90 kusů zařízení (30 kusů od každé verze osazení). V lince dochází s těmito zařízeními k několika operacím. Jedná se například o testování funkčnosti produktu v testovací stanici, montáž a demontáž zařízení do krabičky, případně do lego domku apod.

Mnou navržené a zkonstruované zařízení je možné v budoucnu dále inovovat. Například po návržení a zkonstruování zařízení bylo oznámeno dodavatelem akcelerometru, že daný typ není vhodné používat pro nové návrhy. Postupem času dojde k zastavení produkce. Na základě této informace jsem kontaktoval výrobce, který mi tuto informaci potvrdil a doporučil mi možnou alternativu. Jedná se o akcelerometr KX132-1211, který je vyráběn ve stejném pouzdře a se stejným uspořádáním pinů. Nicméně doporučený akcelerometr neobsahuje periferii krokoměru, proto jako další možná alternativa je modul STP201M. V modulu je implementován algoritmus krokoměru, napájecí napětí je v rozmezí 2,3 V až 3,6 V a obsahuje komunikační rozhraní.

Zadání diplomové práce bylo naplněno ve všech bodech a řešení bylo konzultováno s vedoucím práce.

Literatura

- [1] KREIDL, Marcel. *Měření teploty: senzory a měřící obvody*. Praha: BEN - technická literatura, 2005. Senzory neelektrických veličin. ISBN 80-730-0145-4.
- [2] VOJÁČEK, Antonín. Přehled principů el. měření teploty - 1. díl. *Automatizace.hw.cz: rady a poslední novinky z oboru* [online]. Praha: automatizace.hw.cz, ©1997 - 2014, 24. Červen 2014 [cit. 2021-01-19]. Dostupné z: <https://automatizace.hw.cz/prehled-principu-el-mereni-teploty-1-dil>
- [3] VOJÁČEK, Antonín. Principy akcelerometrů - 1. díl - Piezoelektrické. *Automatizace.hw.cz: rady a poslední novinky z oboru* [online]. Praha: automatizace.hw.cz, ©1997 - 2014, 14. Leden 2007 [cit. 2020-12-06]. Dostupné z: <https://automatizace.hw.cz/clanek/2007011401>
- [4] VOJÁČEK, Antonín. Jak pracují nové 3D MEMS akcelerometry Freescale? *Vyvoj.hw.cz: profesionální elektronika* [online]. Praha: vyvoj.hw.cz, ©1997 - 2014, 30. Duben 2007 [cit. 2020-12-06]. Dostupné z: <https://vyvoj.hw.cz//soucastky/jak-pracuji-nove-3d-mems-akcelerometry-freescale.html>
- [5] HAMPTON, John R. *EKG stručně, jasně, přehledně*. 7. vydání. Praha: Grada, 2013. ISBN 978-80-247-4246-5.
- [6] TOMPKINS, Willis J. a Jiapu PAN. A Real-Time QRS Detection Algorithm. *IEEE Transactions on Biomedical Engineering*. 1985, **32**(3), 230-236.
- [7] DUBEY, Harishchandra. Illustration of QRS detection using Pan-Tompkins algorithm. In: *ResearchGate* [online]. ResearchGate, © 2008-2021, 24.6.2017 [cit. 2021-02-18]. Dostupné z: https://www.researchgate.net/figure/Illustration-of-QRS-detection-using-Pan-Tompkins-algorithm-a-Raw-ECG-data-b-ECG_fig14_311274279
- [8] TIŠNOVSKÝ, Pavel. Architektura mikrořadičů s jádry ARM Cortex-M0 a ARM Cortex-M0+. *ROOT.CZ* [online]. ROOT.CZ, ©1998 – 2021, 1. 10. 2015 [cit. 2020-10-02]. Dostupné z: <https://www.root.cz/clanky/architektura-mikroradicu-s-jadry-arm-cortex-m0-a-arm-cortex-m0/>

- [9] *Cortex-M0+* [online]. Cambridge, UK: arm developer, ©1995-2021 [cit. 2021-10-04]. Dostupné z: <https://developer.arm.com/ip-products/processors/cortex-m/cortex-m0-plus>
- [10] TIŠNOVSKÝ, Pavel. Jak pracuje počítač? *ROOT.CZ* [online]. ROOT.CZ, ©1998 – 2021, 27. 2. 2008 [cit. 2021-01-28]. Dostupné z: <https://www.root.cz/clanky/jak-pracuje-pocitac/#k02>
- [11] ARM Cortex®-M0+ Pipeline. In: *Microchip Developer Help* [online]. Arizona: Microchip Technology, © 2021 [cit. 2021-03-14]. Dostupné z: <https://microchipdeveloper.com/32arm:m0-pipeline>
- [12] PINKER, Jiří. *Mikroprocesory a mikropočítače*. Praha: BEN - technická literatura, 2004. ISBN 80-730-0110-1.
- [13] VÁŇA, Vladimír. *ARM pro začátečníky*. Praha: BEN - technická literatura, 2009. ISBN 978-80-7300-246-6.
- [14] TIŠNOVSKÝ, Pavel. Sběrnice v domácích a osobních počítačích. *ROOT.CZ* [online]. ROOT.CZ, ©1998 – 2021, 2. 10. 2008 [cit. 2020-10-12]. Dostupné z: <https://www.root.cz/clanky/sbernice-v-domacich-a-osobnich-pocitacich/>
- [15] PETERKA, Jiří. Synchronní, asynchronní a arytmičtý přenos. *CHIPweek* [online]. 1996, prosinec 1996, **1996**(50), 24 [cit. 2020-10-23]. Dostupné z: <https://www.earchiv.cz/a96/a650k150.php3>
- [16] TIŠNOVSKÝ, Pavel. Komunikace po sériové sběrnici I2C. *ROOT.CZ* [online]. ROOT.CZ, ©1998 – 2021, 8. 1. 2009 [cit. 2020-10-12]. Dostupné z: <https://www.root.cz/clanky/komunikace-po-seriove-sbernici-isup2supc/>
- [17] OLEJÁR, Martin. Stručný popis sběrnice I2C a její praktické využití k připojení externí eeprom 24LC256 k mikrokontroléru PIC16F877. *vyvoj.hw.cz* [online]. Praha: vyvoj.hw.cz, ©1997 - 2014, 20. Květen 2000 [cit. 2020-10-12]. Dostupné z: <https://vyvoj.hw.cz/navrh-obvodu/strucny-popis-sbernice-i2c-a-jeji-prakticke-vyuziti-k-pripojeni-externi-eeeprom-24lc256>
- [18] TIŠNOVSKÝ, Pavel. Externí sériové sběrnice SPI a I²C. *ROOT.CZ* [online]. ROOT.CZ, ©1998 – 2021, 30. 12. 2008 [cit. 2020-10-16]. Dostupné z: <https://www.root.cz/clanky/externi-seriove-sbernice-spi-a-i2c/>
- [19] DUNCOMBE, Mark. Is UART Flow Control (RTS/CTS) Necessary for Proper Operation in Wireless Modules? *Laidr Technologies: Wireless Connectivity Blog* [online]. ©2006-2015, 27 August, 2015 [cit. 2020-10-23]. Dostupné z: <http://www.summitdata.com/blog/uart-flow-control-rtscts-necessary-proper-operation-wireless-modules/>

- [20] *Static Drive LCD Technology* [online]. Pacific Display Devices, ©1996 - 2012 [cit. 2020-10-25]. Dostupné z: http://www.pacificdisplay.com/lcd_static_drive.htm
- [21] *Multiplex Drive and Bias of LCD Technology* [online]. Pacific Display Devices, ©1996 - 2012 [cit. 2020-10-25]. Dostupné z: https://www.pacificdisplay.com/lcd_multiplex_drive.htm
- [22] NILSSON, Rolf. Bluetooth Low Energy není jen nová verze standardu Bluetooth. *Automa: Časopis pro automatizační techniku* [online]. 2013, prosinec 2013, **2013**(12), 40 - 43 [cit. 2020-10-03]. Dostupné z: https://www.automa.cz/Aton/FileRepository/pdf_articles/10907.pdf
- [23] GUPTA, Naresh Chand. *Inside bluetooth low energy*. Boston: Artech House: Artech House mobile communications series, ©2013. ISBN 978-1-60807-579-9.
- [24] TOWNSEND, Kevin. Introduction to Bluetooth Low Energy. *Adafruit Learning System* [online]. New York, ©2005 - 2021, 20 Mar, 2014 [cit. 2020-11-02]. Dostupné z: <https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gap>
- [25] *How GAP and GATT Work* [online]. Minneapolis: Punch Through, 24 June, 2013n. 1. [cit. 2020-11-03]. Dostupné z: <https://punchthrough.com/how-gap-and-gatt-work/>
- [26] TOŠOVSKÝ, Petr. Úvod do Altium Designeru I. *Vyvoj.hw.cz: profesionální elektronika* [online]. Praha: vyvoj.hw.cz, ©1997 - 2014, 27. Březen 2010 [cit. 2020-11-05]. Dostupné z: <https://vyvoj.hw.cz/navrh-obvodu/software/uvod-do-altium-designeru-i.html>
- [27] TOŠOVSKÝ, Petr. Úvod do Altium Designeru IV. - knihovny. *Vyvoj.hw.cz: profesionální elektronika* [online]. Praha: vyvoj.hw.cz, ©1997 - 2014, 1. Leden 2011 [cit. 2020-11-02]. Dostupné z: <https://vyvoj.hw.cz/teorie-a-praxe/software/uvod-do-altium-designeru-iv-knihovny.html>
- [28] TOŠOVSKÝ, Petr. Úvod do Altium Designeru II. – tvorba PCB. *Vyvoj.hw.cz: profesionální elektronika* [online]. Praha: vyvoj.hw.cz, ©1997 - 2014, 26. Duben 2010 [cit. 2021-02-04]. Dostupné z: <https://vyvoj.hw.cz/navrh-obvodu/software/uvod-do-altium-designeru-ii-tvorba-pcb.html>
- [29] *A powerful eclipse-based C/C++ integrated development tool for your STM32 projects* [online]. STMicroelectronics, ©2021 [cit. 2021-01-27]. Dostupné z: <https://www.st.com/en/development-tools/truестudio.html>
- [30] *About the Eclipse Foundation* [online]. Ottawa: The Eclipse Foundation [cit. 2021-01-27]. Dostupné z: <https://www.eclipse.org/org/>

- [31] OBERLI, Nicolas. SWD – ARM’S ALTERNATIVE TO JTAG. *Kudelski Security Research: The Latest News from Research at Kudelski Security* [online]. 16 May 2019 [cit. 2021-01-27]. Dostupné z: <https://research.kudelskisecurity.com/2019/05/16/swd-arms-alternative-to-jtag/>
- [32] Micrel, Inc. [online katalogový list]. *MIC5501/2/3/4*. ©2015 [cit. 2020-06-23]. Dostupné z: <https://www.lumex.com/spec/LCD-S401M16KR.pdf>
- [33] Lumex, Inc. [online katalogový list]. *LCD-S401M16KR*. ©2011 [cit. 2020-06-23]. Dostupné z: <http://ww1.microchip.com/downloads/en/devicedoc/mic550x.pdf>
- [34] Atmel Corporation [online katalogový list]. *AT30TS74*. ©2017 [cit. 2020-05-31]. Dostupné z: http://ww1.microchip.com/downloads/en/DeviceDoc/Web_AT30TS74_0217.pdf
- [35] KIONIX, Inc. [online katalogový list]. *KX126-1063*. ©2019 [cit. 2020-05-24]. Dostupné z: <https://d10bqar0tuhard.cloudfront.net/en/datasheet/KX126-1063-Specifications-Rev-3.0.pdf>
- [36] Maxim Integrated [online katalogový list]. *MAX30003*. ©2019 [cit. 2020-06-24]. Dostupné z: <https://datasheets.maximintegrated.com/en/ds/MAX30003.pdf>
- [37] Bluegiga Technologies [online katalogový list]. *BLE112*. ©2020 [cit. 2020-06-23]. Dostupné z: <https://www.silabs.com/documents/public/data-sheets/BLE112-DataSheet.pdf>
- [38] Freescale Semiconductor, Inc. [online katalogový list]. *Kinetis KL33 Microcontroller*. ©2015 [cit. 2020-06-23]. Dostupné z: <https://www.nxp.com/docs/en/data-sheet/KL33P64M48SF6.pdf>
- [39] Freescale Semiconductor, Inc. [online katalogový list]. *KL33 Sub-Family Reference Manual*. ©2016 [cit. 2020-07-01]. Dostupné z: <https://www.nxp.com/docs/en/reference-manual/KL33P64M48SF6RM.pdf>
- [40] KIONIX, Inc. [online katalogový list]. *Getting Started with Pedometer*. ©2019 [cit. 2020-07-10]. Dostupné z: <https://d10bqar0tuhard.cloudfront.net/en/document/AN073-Getting-Started-with-Pedometer.pdf>

Přílohy

A Schéma zapojení (4 strany)

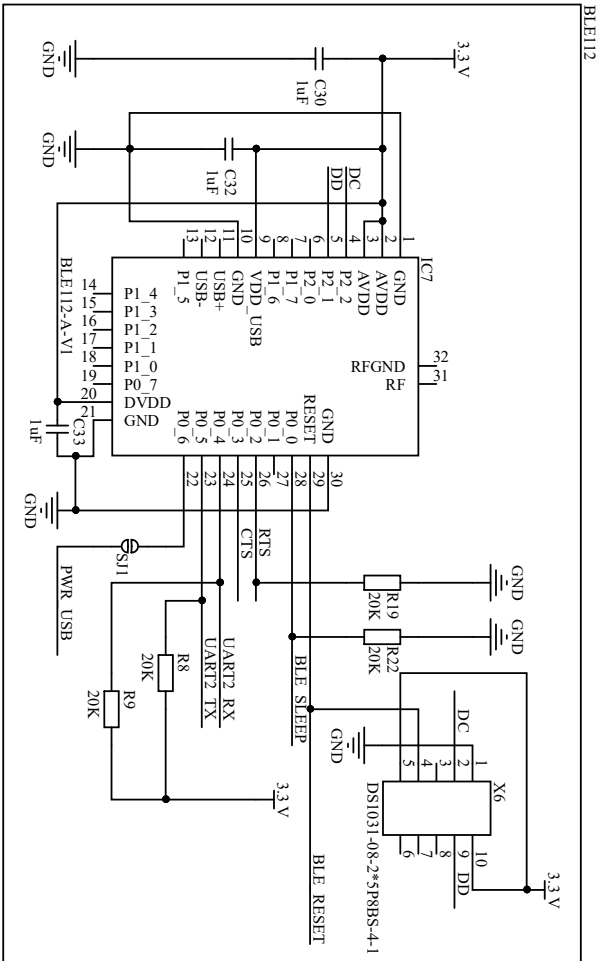
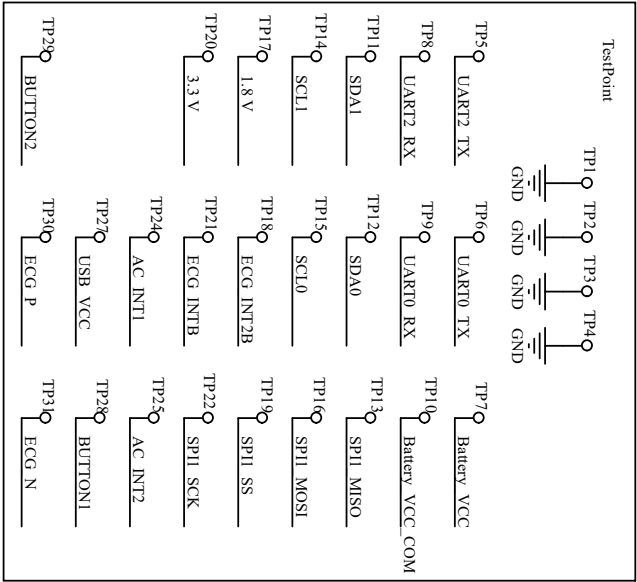
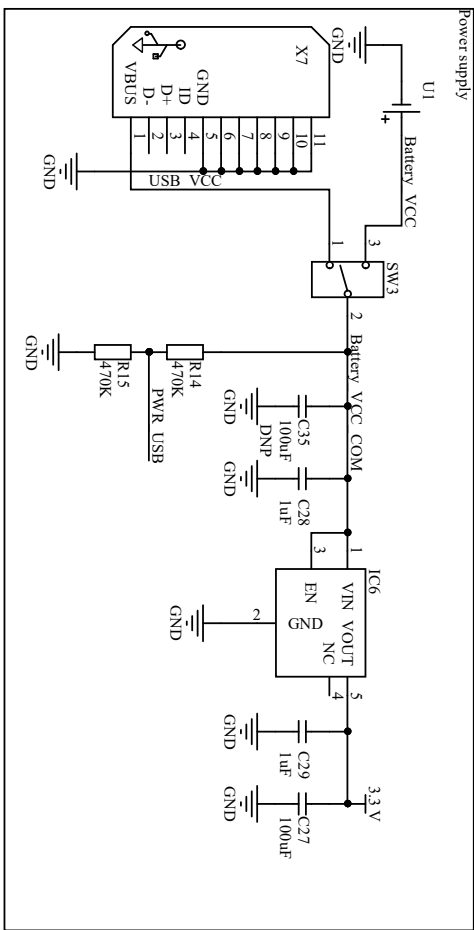
Součástí DP jsou elektronické přílohy.

Adresářová struktura elektronických příloh:

- B_Zdrojové_soubory_pro_DPS - Všechny soubory (pro Altium Designer v14.3) *.SchDoc, *PcbDoc a také projektový soubor Testbed_Project.PrjPcb (Příloha v IS EDISON)
- C_Program_pro_mikrokontroler - Projekt v souboru zip (Příloha v IS EDISON)
- D_Program_pro_BLE_modul - Všechny zdrojové soubory *.bgs, *.xml a projektový soubor TestbedUART.bgproj (Příloha v IS EDISON)
- E_Skript_pro_modifikaci_HexID - Skript v Pythonu 3.8.5 (Příloha v IS EDISON)

Příloha A

Schéma zapojení



Title: Testbed		Author: Daniel Kejzar	
Size: A4	Number: 1	Revision:	
Date: 21.03.2021	Time: 19:00:26	Schematic: 1.2	Board: 1.2

1

2

3

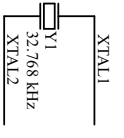
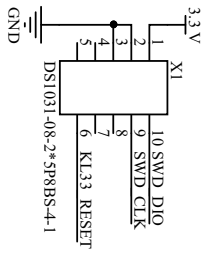
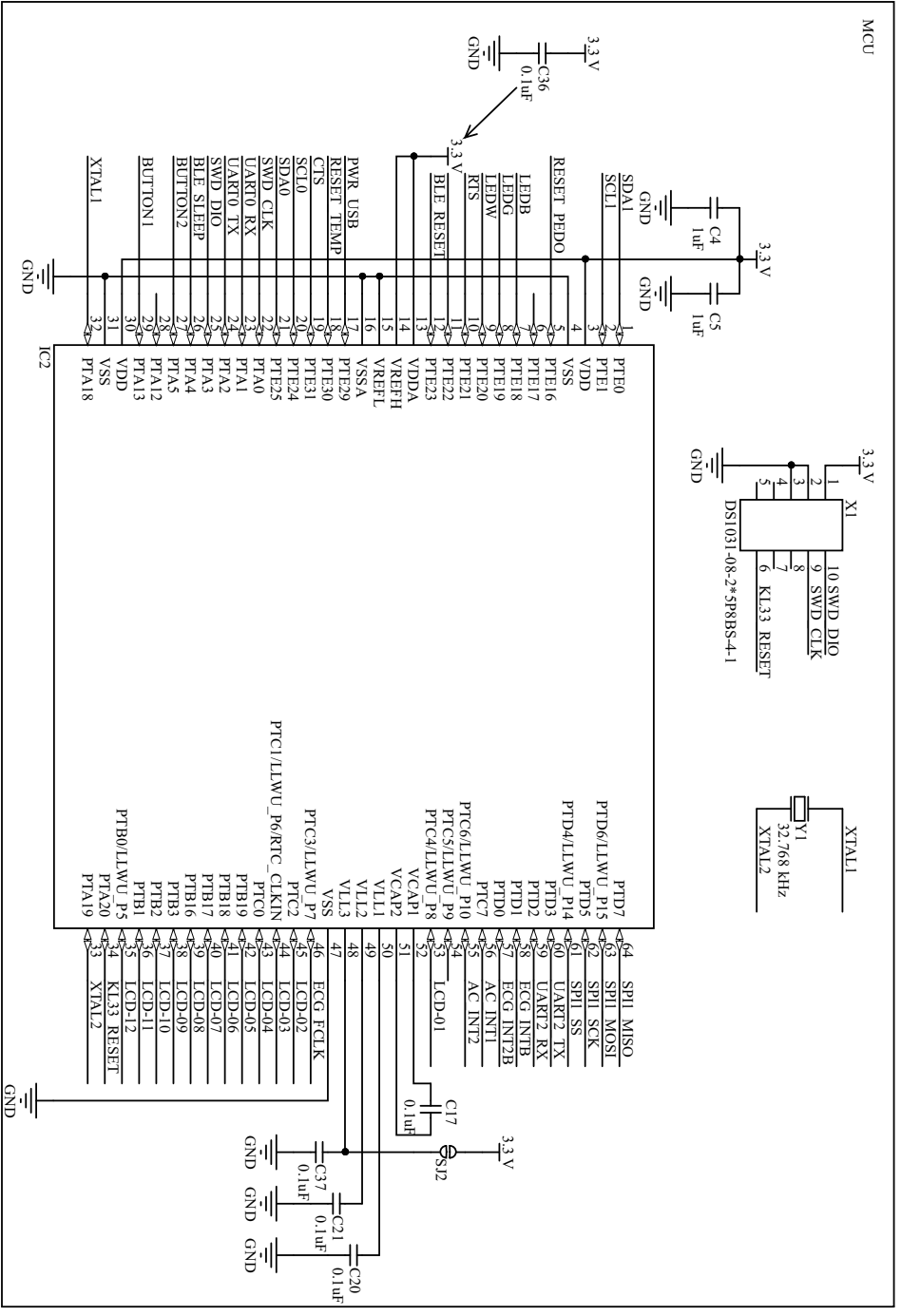
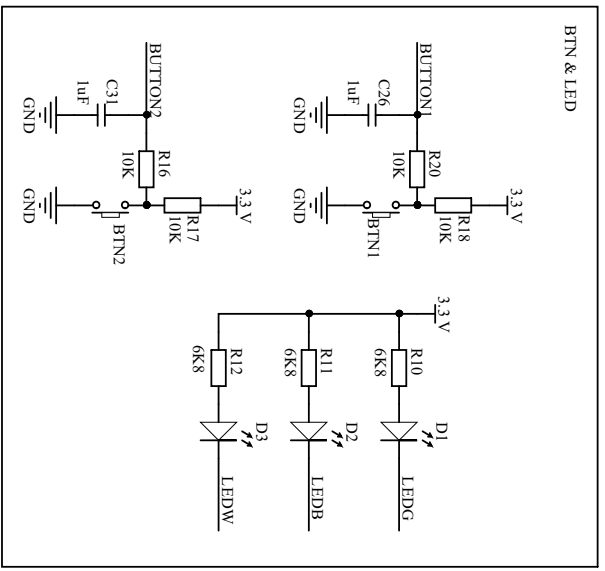
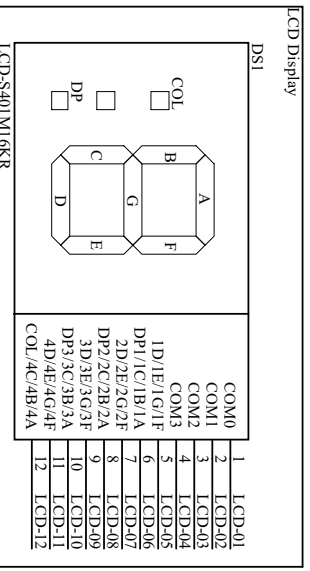
4

1

2

3

4



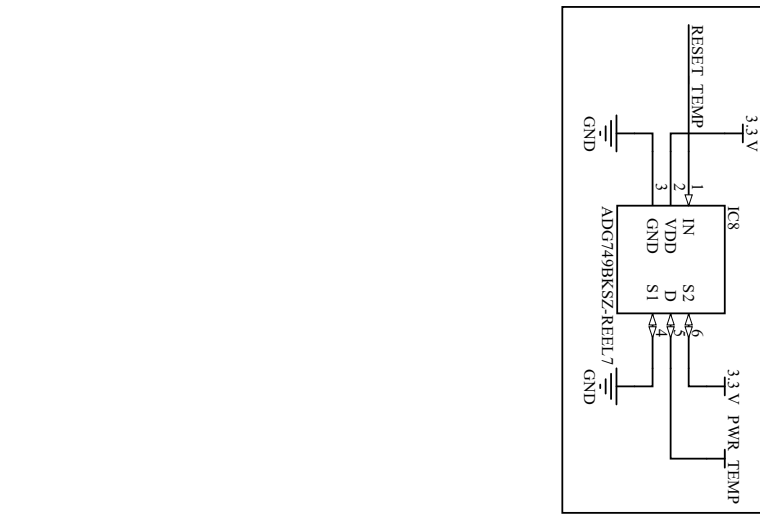
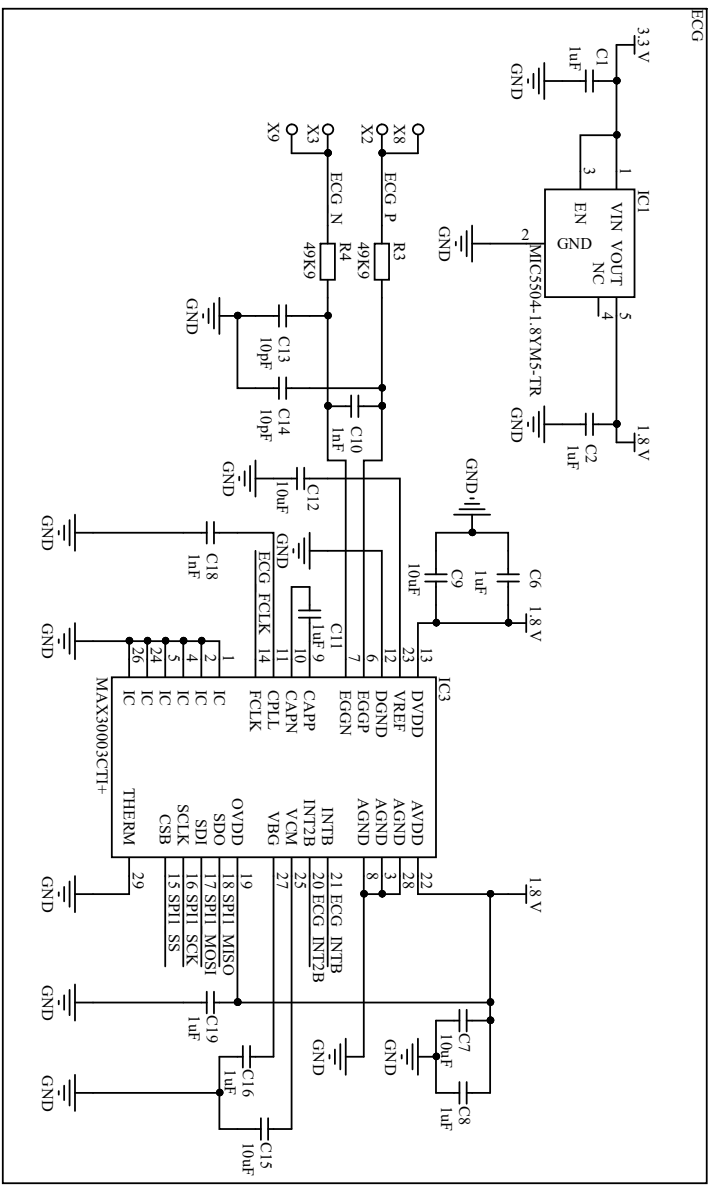
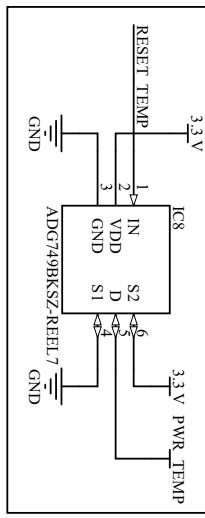
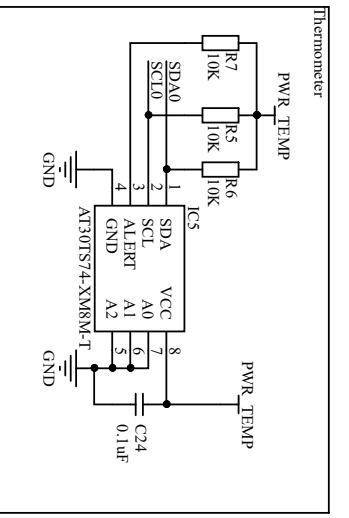
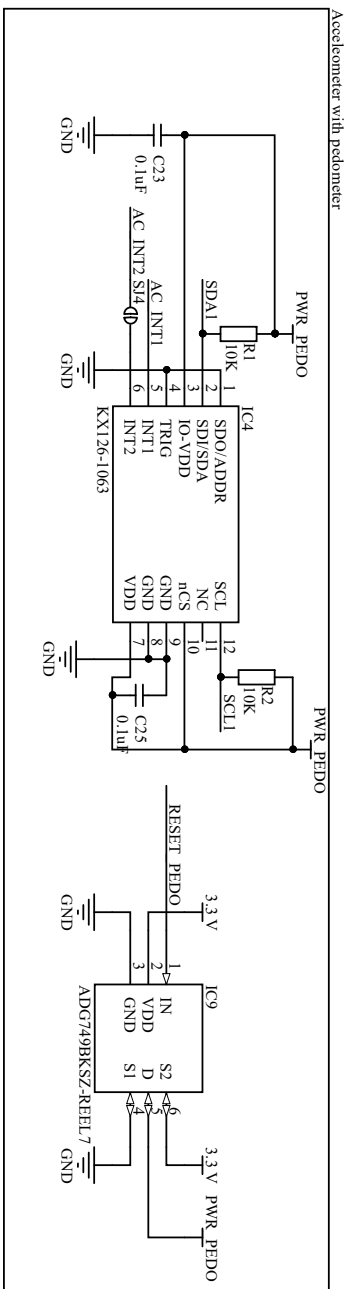
Title: Testbed		Author: Daniel Kajzar	
Size: A4	Number: 2	Revision:	
Date: 21.03.2021	Time: 19:00:26	Schematic: 1.2	Board: 1.2
		DEPARTMENT OF GOVERNMENTICS AND BIOLOGICAL ENGINEERING	

1

2

3

4



Title: Testbed		Author: Daniel Kojzar
Size: A4	Number: 3	Revision:
Date: 21.03.2021	Time: 19:00:26	Schematic: 1.2
Board: 1.2		