



Zadání bakalářské práce

Název:	Zařízení pro ovládání základních periférií připojených k FPGA obvodu
Student:	Michal Šebek
Vedoucí:	Ing. Pavel Kubalík, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Počítačové inženýrství
Katedra:	Katedra číslicového návrhu
Platnost zadání:	do konce letního semestru 2023/2024

Pokyny pro vypracování

Prozkoumejte existující řešení ovládání periférií z FPGA obvodu firmy XILINX. Zaměřte se zejména na tyto periférie: OLED displej, alfanumerický displej, tlačítka, přepínače, led diody. Navrhněte vlastní knihovnu zaměřenou na ovládání periférií zejména přes sběrnici I2C. Navrženou knihovnu zrealizujte a řádně otestujte. Pro demonstraci správné funkce periférií napište obslužnou aplikaci pro FPGA Artix-7. Aplikace bude obsahovat menu pro volbu testování periférií a nastavení obsahu registrů. Výsledné řešení řádně otestujte.

Bakalářská práce

**ZAŘÍZENÍ PRO
OVLÁDÁNÍ
ZÁKLADNÍCH PERIFERIÍ
PŘIPOJENÝCH K FPGA
OBVODU**

Michal Šebek

Fakulta informačních technologií
Katedra číslicového návrhu
Vedoucí: Ing. Pavel Kubalík, Ph.D.
11. května 2023

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2023 Michal Šebek. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení, je nezbytný souhlas autora.

Odkaz na tuto práci: Šebek Michal. *Zařízení pro ovládání základních periferií připojených k FPGA obvodu*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2023.

Obsah

Poděkování	vii
Prohlášení	viii
Abstrakt	ix
Seznam zkratk	x
Úvod	1
1 Cíle práce	3
2 Rešerše dostupných periférií	5
2.1 Základní periférie	5
2.2 OLED displej	5
2.3 Alfamerický displej	5
2.4 Další periférie	6
2.5 Převodník logických úrovní	6
2.6 Vývojová deska	6
3 Analýza ovládání periférií	7
3.1 Sběrnice Inter-Integrated Circuit	7
3.1.1 Průběh přenosu	7
3.1.2 Existující implementace pro FPGA	8
3.2 Řadiče OLED displejů	9
3.2.1 Řadič SSD1306	9
3.2.1.1 Příkazy	9
3.2.1.2 Komunikace přes I ² C	9
3.2.2 Řadič SH1106	10
3.3 Řadiče alfanumerických displejů	10
3.3.1 Řadič HD44780	10
3.3.1.1 Komunikace	11
3.3.1.2 Inicializace	12
3.3.2 Řadič ST7066U	12
3.4 Vstupně-výstupní expandér PCF8574	12
3.4.1 Komunikace	13
3.5 Hodiny reálného času DS3231	13
3.5.1 Registry	13
3.5.2 Komunikace	15
3.6 Senzor teploty a tlaku BMP280	16
3.6.1 Registry	16
3.6.2 Komunikace	16
3.6.3 Převod hodnot	19
3.7 Senzor teploty a vlhkosti SHT30	19

3.7.1	Příkazy a komunikace	19
3.8	Maticová klávesnice	21
3.8.1	Snímání stisknutých kláves	21
3.9	Rotační enkodér	22
3.10	Celočíselné dělení	23
3.11	Písma	23
3.12	Nástroje pro testování	24
4	Návrh knihovny a aplikace	25
4.1	Společné vlastnosti modulů	25
4.2	Modul pro řízení I ² C sběrnice	25
4.3	Modul pro ovládání OLED displeje	26
4.4	Modul pro ovládání alfanumerického displeje	27
4.5	Modul pro celočíselné dělení	27
4.6	Modul pro převod čísla do binárně kódovaných dekadických číslic	27
4.7	Modul pro ovládání senzoru BMP280	29
4.8	Modul pro ovládání senzoru SHT30	29
4.9	Modul pro ovládání hodin reálného času DS3231	30
4.10	Moduly pro zpracování vstupů	30
4.11	Modul pro čtení rotačního enkodéru	30
4.12	Modul pro čtení vstupu z maticové klávesnice	30
4.13	Aplikace pro vývojovou desku Basys 3	31
5	Testování	33
5.1	Použité součástky a moduly	33
5.2	Použité nástroje	33
5.3	Postup testování	34
	Závěr	37
	A Návrh obrazovek aplikace	39
	Obsah přiloženého archivu	47

Seznam obrázků

3.1	Schéma maticové klávesnice 4×4	21
3.2	Konečný automat pro detekci otáčení rotačního enkodéru	22
3.3	Algoritmus celočíselného dělení se zbytkem	23
4.1	Schéma modulu <code>divider_core</code> celočíselné děličky	28
4.2	Schéma zřetězení tří modulů <code>divider_core</code> celočíselné děličky pro převod do BCD	28
5.1	Vývojová deska Basys 3 se spuštěnou testovací aplikací	35
5.2	Vývojová deska Basys 3 se spuštěnou testovací aplikací (detail na I ² C periferie)	36
A.1	Návrh obrazovky menu	39
A.2	Návrh obrazovky aktivity pro testování senzoru BMP280	39
A.3	Návrh obrazovky aktivity pro testování senzoru SHT30	39
A.4	Návrh první obrazovky aktivity pro testování hodin reálného času DS3231	40
A.5	Návrh druhé obrazovky aktivity pro testování hodin reálného času DS3231	40
A.6	Návrh třetí obrazovky aktivity pro testování hodin reálného času DS3231	40
A.7	Návrh čtvrté obrazovky aktivity pro testování hodin reálného času DS3231	40
A.8	Návrh obrazovky aktivity pro testování klávesnice, tlačítek, přepínačů a rotačního enkodéru	40
A.9	Návrh obrazovky aktivity pro testování světelných diod	41
A.10	Návrh obrazovky aktivity pro testování alfanumerického displeje	41

Seznam tabulek

3.1	Důležité příkazy řadičů SSD1306 a SH1106	9
3.2	Důležité příkazy řadiče HD44780	11
3.3	Registry hodin reálného času DS3231	14
3.4	Význam bitů konfiguračního registru obvodu DS3231	14
3.5	Význam bitů stavového registru obvodu DS3231	15
3.6	Hodnoty nastavení počtu vzorků měření senzoru BMP280	17
3.7	Hodnoty nastavení režimu senzoru BMP280	17
3.8	Hodnoty nastavení doby mezi měřeními senzoru BMP280	17
3.9	Hodnoty nastavení koeficientu filtru senzoru BMP280	17
3.10	Příkaz pro zahájení jednorázového měření pomocí SHT30	20
3.11	Příkaz pro zahájení periodického měření pomocí SHT30	20
3.12	Význam jednotlivých bytů naměřených dat z SHT30	20

Seznam výpisů kódu

3.1	Kód pro výpočet teploty ze senzoru BMP280	18
3.2	Kód pro výpočet tlaku ze senzoru BMP280	18

Chtěl bych poděkovat především vedoucímu této práce, Ing. Pavlu Kubalíkovi, Ph.D., za jeho rady a připomínky.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užit. Tyto osoby jsou oprávněny Dílo užit jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 11. května 2023

Michal Šebek

Abstrakt

Tato bakalářská práce se zabývá ovládáním běžně dostupných periférií z FPGA obvodu společnosti Xilinx. Z periférií řízených přímo jsou vybrána tlačítka, přepínače, LED diody, rotační enkodér a maticová klávesnice. Prostřednictvím I²C sběrnice je ovládán OLED displej s řadičem SSD1306 nebo SH1106, alfanumerický LCD displej s řadičem HD44780 nebo kompatibilním připojený skrze vstupně-výstupní expandér PCF8574, senzor teploty a tlaku vzduchu BMP280, senzor teploty a vlhkosti vzduchu SHT30 a hodiny reálného času DS3231. Pro ovládání periférií je v této práci navržena knihovna v jazyce VHDL, kterou lze použít pro FPGA obvody společnosti Xilinx. V jazyce VHDL je vytvořena také aplikace pro otestování správné funkce knihovny a periférií. Aplikaci lze spustit na vývojové desce Basys 3 od společnosti Digilent, která obsahuje FPGA obvod Artix-7 společnosti Xilinx. Aplikace obsahuje menu umožňující výběr periferie, po jejím výběru je zobrazena obrazovka specifická pro danou periferii.

Klíčová slova knihovna, ovládání periférií, ověření funkčnosti, testovací aplikace, FPGA, Xilinx, OLED displej, LCD, I²C, VHDL

Abstract

This bachelor thesis focuses on the control of commonly available peripherals from a Xilinx FPGA circuit. Out of the peripherals controlled directly, buttons, switches, LEDs, rotary encoder and matrix keyboard are selected. Via the I²C bus, OLED displays with the SSD1306 or SH1106 controller, alphanumeric LCD displays with the HD44780 controller or compatible connected through the PCF8574 input/output expander, BMP280 air temperature and pressure sensor, SHT30 air temperature and humidity sensor, and DS3231 real-time clock are controlled. In this thesis, a VHDL library is designed to control the peripherals, which can be used for Xilinx FPGA circuits. A VHDL application is also developed to test the correct operation of the library and peripherals. The application can be run on a Basys 3 development board from Digilent, which contains an Artix-7 FPGA from Xilinx. The application includes a menu that allows selection of a peripheral, after which a peripheral-specific screen is displayed.

Keywords library, peripheral control, functional verification, test application, FPGA, Xilinx, OLED display, LCD, I²C, VHDL

Seznam zkratek

BCD	Binary-Coded Decimal
EEPROM	Electrically Erasable Programmable Read-Only Memory
FPGA	Field-Programmable Gate Array
GHDL	G Hardware Design Language
GHW	GHDL Waveform
I ² C	Inter-Integrated Circuit
JTAG	Joint Test Action Group
LCD	Liquid Crystal Display
LED	Light-Emitting Diode
LSB	Least Significant Byte
MSB	Most Significant Byte
OLED	Organic Light-Emitting Diode
RTC	Real-Time Clock
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus
VCD	Value Change Dump
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit

Úvod

Programovatelná hradlová pole (FPGA) umožňují jednoduchý a rychlý vývoj systémů se specifickými požadavky. Oproti mikrokontrolérům nabízí větší flexibilitu a oproti aplikačně specifickým integrovaným obvodům mají nižší počáteční náklady a nabízí rekonfigurovatelnost, a díky tomu i kratší dobu vývoje systému.

Součástí takových systémů jsou běžně různé periferie, například tlačítka, přepínače, LED diody a displeje, které slouží ke konfiguraci systému a jeho interakci s uživatelem, nebo senzory, které se používají k měření okolního prostředí. Dalšími periferiemi mohou být paměti a úložiště nebo jiné periferie, které jsou specifické pro danou aplikaci.

Některé periferie je triviální ovládat přímo z FPGA obvodu (např. tlačítka a LED diody), pro jiné je přímočařejší použít mikroprocesor (např. displej komunikující přes I²C sběrnici nebo senzor, u kterého je třeba k získání správné hodnoty v běžných jednotkách provést složitý převod). Procesor může být mimo FPGA obvod, alternativně lze FPGA obvod nakonfigurovat, aby byl procesor jeho součástí.

Pro některé aplikace by bylo vhodnější a jednodušší ovládat i složitější periferie z FPGA obvodu přímo, bez nutnosti používat procesor. U některých běžných periférií však pro tento způsob ovládání často není k dispozici žádné existující řešení. V této práci bude vyvinuta knihovna, která pro vybrané periferie tento způsob ovládání umožní.



Kapitola 1

Cíle práce

Hlavním cílem práce je navrhnout a realizovat knihovnu pro ovládání OLED displeje, alfanumerického displeje, tlačítek, přepínačů, LED diod a některých dalších periférií z FPGA obvodu firmy Xilinx. Funkčnost knihovny a periférií bude ověřena pomocí testovací aplikace, která bude obsahovat menu pro volbu periférií a pro každou periférii umožní její ovládání.

Prvním cílem je provést rešerši periférií a současného stavu řešení jejich ovládání z FPGA. Rešerše se bude věnovat jak displejům, tlačítkům, přepínačům a LED diodám, tak i některým dalším perifériím, které jsou na trhu dostupné. Z periférií pak budou vybrány ty, kterým se bude věnovat zbytek práce.

Po rešerši je cílem detailněji analyzovat způsob ovládání vybraných periférií. V případě periférií obsahujících vlastní řadič se analýza bude věnovat funkcionalitám řadiče a způsobu komunikace s ním. U periférií, které neobsahují vlastní řadič, se bude analýza věnovat jejich samotnému řízení.

Dalším cílem je na základě analýzy navrhnout a implementovat knihovnu, která bude umožňovat ovládání vybraných periférií z obvodu FPGA. Knihovna bude obsahovat moduly pro základní ovládání všech vybraných periférií.

Posledním cílem je s pomocí nově implementované knihovny navrhnout a implementovat testovací aplikaci, která bude obsahovat menu pro volbu periférií a pro každou periférii umožní její ovládání. Tato aplikace bude otestována na FPGA obvodu Artix-7 firmy Xilinx.

Rešerše dostupných periférií

Tato kapitola se zabývá rešerší a výběrem vhodných periférií.

2.1 Základní periférie

Základní součástky, jako jsou tlačítka, přepínače, LED diody, se prodávají v různých druzích a často bývají také součástí vývojových desek, viz dále.

Z existujících desek obsahujících některé tyto periférie je k dispozici deska obsahující tlačítkovou maticovou klávesnici o 4×4 tlačítkách, další 4 tlačítka a 8 LED diod, kterou lze koupit z e-shopu GM electronic [1].

Maticové klávesnice jsou k dispozici i samostatně v různých provedeních, zejména v provedení se 4×4 [2], [3] nebo 4×3 tlačítka [4].

Mezi základní periférie se dá zařadit i rotační enkodér. V nabídce e-shopu dratek.cz je např. modul s rotačním enkodérem KY-040 [5]. Pro rotační enkodér na desce Pmod ENC firmy Digilent existuje ukázkový projekt pro FPGA [6].

Pro komunikaci s těmito základními perifériemi přes sběrnici I²C je třeba použít I/O expandér podporující I²C rozhraní, v této práci však budou tyto periférie ovládány přímo.

2.2 OLED displej

Na trhu jsou k dispozici malé OLED displeje s rozlišením 128×64 nebo 128×32 bodů, které používají řadič SSD1306 nebo SH1106. Komunikace je u nich řešena buď přes rozhraní I²C, nebo SPI. Akceptují napájecí napětí 3,3 V. Příkladem může být displej s úhlopříčkou 0,96", rozlišením 128×64 bodů a I²C rozhraním s řadičem SSD1306, který je k dispozici na e-shopu dratek.cz [7], nebo 1,3" displej se stejným rozlišením a rozhraním, ale s řadičem SH1106, který je k dispozici tamtéž [8].

Firma Digilent nabízí OLED displej s rozlišením 128×32 bodů a řadičem SSD1306 s SPI rozhraním na modulu Pmod OLED [9]. Pro tento modul firma nabízí i ukázkou použití z FPGA [10]. Jedná se o jednoduchou ukázkou, která na displeji zobrazí text [10].

Pro komunikaci FPGA s těmito displeji přes I²C nebyla nalezena žádná existující ukázkou.

2.3 Alfnumerický displej

Dostupné jsou LCD displeje s řadičem HD44780 nebo kompatibilním (např. ST7066U). Tyto displeje jsou k dispozici v provedeních 16×2 nebo 20×4 znaků. Napájeny jsou napětím 5 V. [11], [12]

Ke komunikaci používají paralelní rozhraní blíže specifikované v katalogovém listu řadiče [13]. Pro tento typ komunikace existují knihovny, např. jednoduchá knihovna z OpenCores zaměřená na ovládání 16×2 displejů [14].

Pro komunikaci přes I²C je třeba použít I/O expandér. V nabídce jsou i displeje s již přidaným I/O expandérem PCF8574 [15], [16]. Pro tento typ komunikace FPGA s řadičem displeje nebyla nalezena žádná existující ukázka či knihovna.

2.4 Další periferie

V nabídce je mnoho dalších periférií, výběr byl omezen pouze na ty, které disponují I²C rozhraním a které se prodávají osazené na malé desce plošných spojů umožňující jejich snadné připojení k nepájivému poli nebo vývojové desce bez nutnosti návrhu vlastní desky plošných spojů a nákupu dalších komponent, jako jsou rezistory nebo kondenzátory.

Z nabídky byly pro účely této práce vybrány následující periferie:

Senzor SHT30 Jedná se o senzor teploty a relativní vlhkosti vzduchu [17]. K napájení lze použít napětí 3,3 V [17]. Modul s tímto senzorem lze zakoupit např. z e-shopu LaskaKit.cz [18].

Senzor BMP280 Senzor teploty vzduchu a atmosférického tlaku [19]. Lze napájet napětím 3,3 V [19]. Modul s klonem tohoto senzoru lze zakoupit např. z e-shopu LaskaKit.cz [20].

Hodiny reálného času DS3231 Tento obvod umožňuje udržování aktuálního data a času [21]. K napájení lze použít napětí 3,3 V [21]. Modul s klonem tohoto obvodu lze zakoupit např. z e-shopu LaskaKit.cz [22].

Pro žádnou z těchto periférií nebyla nalezena existující ukázka ovládání z FPGA obvodu.

2.5 Převodník logických úrovní

Protože vstupně-výstupní piny FPGA obvodu Artix-7 pracující maximálně na 3,3V úrovni nejsou kompatibilní s výstupy zařízení komunikujících na 5V úrovni [23], je třeba použít převodník logických úrovní. Z převodníků vhodných pro použití s I²C sběrnici je v nabídce například dvoukanálový převodník, který navíc obsahuje i napěťový regulátor z 5 V na 3,3 V [24], nebo čtyřkanálový převodník, který regulátor neobsahuje [25].

2.6 Vývojová deska

Vývojové desky, sloužící k usnadnění vývoje a testování, také obsahují periferie. Zde je několik příkladů vývojových desek obsahujících FPGA obvod Artix-7 firmy Xilinx:

Cmod A7-35T Deska od firmy Digilent s 48 piny, vhodná mj. i pro připojení na nepájivé pole. Z těchto pinů jsou 2 piny analogové vstupy a 44 pinů je vstupně-výstupních, digitálních. Deska dále obsahuje Pmod konektor s 8 digitálními vstupně-výstupními piny, paměť SRAM o kapacitě 512 KiB, flash paměť o kapacitě 4 MiB, hodinový oscilátor s frekvencí 12 MHz a obvod umožňující programování (obsahuje USB-JTAG převodník) a komunikaci (obsahuje USB-UART převodník). [26]

Basys 3 Další deska od společnosti Digilent, vhodná pro výukové účely. Obsahuje 16 přepínačů, 16 LED diod, 5 tlačítek, 4ciferný 7segmentový displej, 4 Pmod konektory, každý obsahující 8 digitálních vstupně-výstupních pinů, u jednoho Pmod konektoru lze piny nakonfigurovat i jako analogové vstupy. Dále obsahuje hodinový oscilátor frekvencí 100 MHz, VGA port pro výstup na monitor, USB port pro klávesnici nebo myš a obvod umožňující programování (obsahuje USB-JTAG převodník) a komunikaci (obsahuje USB-UART převodník). [27]

Analýza ovládání periférií

Tato kapitola se zabývá podrobnější analýzou funkcionalit vybraných periférií a způsobu komunikace s nimi.

3.1 Sběrnice Inter-Integrated Circuit

Sběrnice *Inter-Integrated Circuit* (I²C) je sériová sběrnice umožňující komunikaci jednoho či více zařízení typu *master* (typicky mikrokontrolér; poskytuje hodinový signál a iniciuje přenosy) s jedním či více zařízeními typu *slave*¹ (typicky periférie) pomocí dvou vodičů – hodinového (SCL) a datového (SDA). [28]

Vodiče jsou připojeny ke kladnému napětí přes *pull-up* rezistory a výstupy všech zařízení pracují na *open-drain* nebo *open-collector* principu, tj. jejich výstupy jsou buď na nízké logické úrovni, nebo ve stavu vysoké impedance. Logická úroveň vodiče je tak vysoká právě tehdy, když na něj žádné zařízení nevysílá nízkou logickou úroveň. [28]

Každé zařízení typu *slave* má 7- nebo 10bitovou adresu, která jej na sběrnici jednoznačně identifikuje. [28]

3.1.1 Průběh přenosu

V tomto shrnutí se předpokládá, že sběrnice pracuje ve standardním (*Standard-mode*) nebo rychlém režimu (*Fast-mode*) a propojuje pouze jedno zařízení typu *master* s jedním nebo více zařízeními typu *slave* se 7bitovými adresami.

Ve výchozím stavu jsou výstupy všech zařízení ve stavu vysoké impedance, a tak je logická úroveň obou vodičů SCL a SDA díky *pull-up* rezistorům vysoká. [28]

Zahájení přenosu (*START condition*) Přenos je zahájen zařízením typu *master* sestupnou hranou na vodiči SDA při vysoké logické úrovni na vodiči SCL. [28]

Ukončení přenosu (*STOP condition*) Přenos je ukončen zařízením typu *master* vzestupnou hranou na vodiči SDA při vysoké logické úrovni na vodiči SCL. [28]

S výjimkou zahájení a ukončení přenosu není dovoleno měnit logickou úroveň na vodiči SDA, když je na vodiči SCL vysoká logická úroveň. [28]

V průběhu přenosu je hodinový signál SCL generován zařízením typu *master* s možným prodloužením nízké úrovně zařízením typu *slave* (jev *clock stretching*). [28]

¹V současné revizi specifikace byly termíny *master* a *slave* nahrazeny termíny *controller* a *target* [28]. Tato práce se však bude držet původní terminologie, kterou lze nalézt v předchozí revizi specifikace [29], tj. *master* a *slave*.

Vyslání jednoho bitu Po sestupné hraně na vodiči SCL vysílající zařízení nastaví požadovanou logickou úroveň na vodiči SDA; vyšle nízkou logickou úroveň při vysílání nuly nebo přepne svůj výstup do stavu vysoké impedance při vysílání jedničky. Vyslání bitu končí při další sestupné hraně na vodiči SCL, kdy může začít vysílání dalšího bitu. [28]

Vyslání celého bytu Podle předchozího bodu vysílající zařízení vyšle 8 bitů v pořadí od nejvýznamnějšího po nejméně významný. Následuje potvrzení přijetí. [28]

Potvrzení přijetí (*ACK*, *NACK*) Probíhá stejně jako vyslání jednoho jedničkového bitu vysílajícím zařízením s tím rozdílem, že současně může přijímající zařízení vyslat nulový bit jako potvrzení přijetí (*ACK*). Jestliže tak neučiní, přijetí není potvrzeno (*NACK*). [28]

Adresace zařízení Zařízení typu *master* vyšle jeden byte sestávající z adresy požadovaného zařízení typu *slave* v sedmi nejvýznamnějších bitech a bitu R/\bar{W} v nejméně významném bitu. Bit R/\bar{W} je nulový v režimu pro zápis a jedničkový v režimu pro čtení. Zařízení typu *slave* s odpovídající adresou potvrdí přijetí (*ACK*). [28]

Zřetěžením výše uvedených dílčích částí lze provést celý přenos:

Zápis Přenos ve směru *master–slave* probíhá následovně [28]:

1. Zařízení typu *master* zahájí přenos (*START condition*).
2. Zařízení typu *master* provede adresaci zařízení v režimu pro zápis a zařízení typu *slave* vyšle *ACK*.
3. Zařízení typu *master* po bytech vysílá data a zařízení typu *slave* potvrzuje přijetí každého bytu (*ACK*).
4. Zařízení typu *master* ukončí přenos (*STOP condition*) vysláním nízké úrovně na vodič SDA, následně vzestupné hrany na vodič SCL a nakonec vzestupné hrany na vodič SDA.

Čtení Přenos ve směru *slave–master* probíhá následovně [28]:

1. Zařízení typu *master* zahájí přenos (*START condition*).
2. Zařízení typu *master* provede adresaci zařízení v režimu pro čtení a zařízení typu *slave* vyšle *ACK*.
3. Zařízení typu *slave* po bytech vysílá data a zařízení typu *master* potvrzuje přijetí každého bytu (*ACK*) s výjimkou posledního bytu, po němž nastane *NACK*. To zabrání nechtěnému zásahu zařízení typu *slave* do ukončování/zahajování přenosu zařízením typu *master*.
4. Zařízení typu *master* ukončí přenos (*STOP condition*) vysláním nízké úrovně na vodič SDA, následně vzestupné hrany na vodič SCL a nakonec vzestupné hrany na vodič SDA.

V obou případech může *master* místo ukončení přenosu (*STOP condition*) a opětovného zahájení nového přenosu (*START condition*) rovnou zahájit nový přenos bez ukončení předchozího (*repeated START condition*, stejně jako *START condition*), a to vysláním vysoké úrovně na vodič SDA, následně vzestupné hrany na vodič SCL a nakonec sestupné hrany na vodič SDA. [28]

3.1.2 Existující implementace pro FPGA

Pro FPGA existují následující implementace I²C *master* zařízení:

I²C Controller Core od Richarda Herveilleho Implementace řadiče I²C sběrnice zařízení typu *master* v jazycích VHDL a Verilog. Podporuje i pokročilejší funkcionality, jako je podpora *multi-master* sběrnice (sběrnice s více zařízeními typu *master*) nebo *clock stretching*. [30]

I²C Master od Scotta Larsona Jednoduchá implementace řadiče I²C sběrnice zařízení typu *master* v jazyce VHDL. Podporuje *clock stretching*, ale už ne *multi-master*. [31]

3.2 Řadiče OLED displejů

V rešerši bylo zjištěno, že běžně dostupné OLED displeje obsahují řadič SSD1306 nebo SH1106. Tato sekce se zabývá analýzou funkcionalit a ovládání těchto řadičů.

3.2.1 Řadič SSD1306

Řadič SSD1306 podporuje OLED displeje s rozlišením až 128×64 bodů. Podporuje komunikaci přes rozhraní SPI, I²C a paralelní rozhraní blíže specifikované v katalogovém listě. [32]

Řadič obsahuje obrazovou paměť, která je rozdělena na 8 stránek na výšku (každá stránka je vysoká 8 řádků) a 128 sloupců na šířku. Adresuje se pomocí dvou ukazatelů, stránkového a sloupcového. [32]

V závislosti na adresačním režimu (stránkový, horizontální, vertikální) se po zápisu do obrazové paměti ukazatele automaticky posunou na další pozici. Výchozí adresační režim je stránkový, který po každém zápisu do obrazové paměti inkrementuje sloupcový ukazatel o 1 a stránkový ukazatel ponechá beze změny. [32]

3.2.1.1 Příkazy

Řadič nabízí ovládání pomocí příkazů různých kategorií, příkazy základní (např. zapnutí/vypnutí displeje, nastavení kontrastu), rolovací (*scrolling*), adresační (nastavení adresačního režimu, nastavení ukazatelů do obrazové paměti), příkazy na konfiguraci hardwaru (nastavení počtu řádků displeje a jejich pořadí, ...). [32]

Pro potřeby této práce jsou důležité příkazy uvedené v tabulce 3.1. U ostatních příkazů jsou výchozí hodnoty postačující.

■ **Tabulka 3.1** Důležité příkazy řadičů SSD1306 a SH1106 z [32], [33], [34]. Příkazy jsou uvedeny v šestnáctkové soustavě

Příkaz	Popis
A0/A1	Prohození pořadí SEG-pinů (sloupců): A0 = neprohozeno (výchozí), A1 = prohozeno (obraz převrácený horizontálně).
C0/C8	Prohození pořadí COM-pinů (řádků): C0 = neprohozeno (výchozí), C8 = prohozeno (obraz převrácený vertikálně).
8D, 10/14	Zapnutí nábojové pumpy (pouze pro SSD1306): 8D, 10 = vypnuta (výchozí), 8D, 14 = zapnuta.
AE/AF	Zapnutí displeje: AE = vypnut (výchozí), AF = zapnut.
00–0F	Nastavení spodních 4 bitů sloupcového ukazatele.
10–1F	Nastavení horních 4 bitů sloupcového ukazatele.
B0–B7	Nastavení stránkového ukazatele.

3.2.1.2 Komunikace přes I²C

Řadič SSD1306 v I²C komunikaci vystupuje v roli *slave* s adresou 0111100 nebo 0111101 v závislosti na logické úrovni na pinu D/C#. Podporuje frekvenci SCL až 400 kHz. [32]

Komunikace probíhá podle následujícího schématu [32]:

1. *Master* zahájí I²C přenos a adresuje řadič v režimu pro zápis.
2. *Master* vyšle řídicí byte:

Bit	Zn.	Popis
7	Co	viz níže
6	D/C#	0 = příkaz, 1 = data
5-0		0

Jestliže je hodnota Co rovna 1, řadič očekává, že bude následovat pouze jeden příkazový nebo datový byte, a před případným dalším příkazovým nebo datovým bytem je třeba opět zaslat řídicí byte. To umožňuje přenos příkazů i dat v rámci jednoho I²C přenosu. [32]

Pokud je hodnota Co rovna 0, jsou všechny byty až do konce přenosu příkazové nebo datové (v závislosti na hodnotě D/C#) a krok 3 se tak může opakovat vícekrát. [32]

3. *Master* vyšle příkazový/datový byte:

Podle hodnoty D/C# v řídicím bytu se jedná buď o příkaz (popř. část vícebytového příkazu), nebo o data. Příkaz ovlivňuje konfiguraci, data jsou zapsána do obrazové paměti na adresu danou sloupcovým a stránkovým ukazatelem. Po zápisu do obrazové paměti jsou automaticky aktualizovány ukazatele podle aktuálního adresačního režimu. [32]

Postup se může se opakovat od bodu 2 nebo 3 v závislosti na hodnotě Co.

4. *Master* ukončí I²C přenos.

Řadič podporuje i čtení obrazových dat z obrazové paměti [32], ale tento režim není v této práci použit.

3.2.2 Řadič SH1106

Řadič SH1106 se svými funkcionalitami velmi podobá řadiči SSD1306. Na rozdíl od řadiče SSD1306 však podporuje OLED displeje s rozlišením až 132×64 bodů a obrazová paměť tak má 132 sloupců [34]. Displeje široké 128 sloupců bývají připojeny na prostředních 128 sloupcových pinů (např. displej z [8]).

Adresační režim má pouze jeden, a to stránkový. Rolování není podporováno vůbec. Dále jsou drobné rozdíly v některých příkazech a jejich výchozích hodnotách, např. příkaz pro ovládání nábojové pumpy má odlišný kód a nábojová pumpa je ve výchozím stavu zapnutá. [34]

Díky podobnosti obou řadičů je možné použít příkazy z tabulky 3.1 a způsob ovládání předchozího řadiče vč. I²C adresy i pro řadič SH1106. [34]

3.3 Řadiče alfanumerických displejů

V rešerši bylo zjištěno použití řadičů Hitachi HD44780 a kompatibilních pro ovládání alfanumerických LCD displejů. Tato sekce se zabývá analýzou funkcionalit a ovládání těchto řadičů.

3.3.1 Řadič HD44780

Řadič HD44780 podporuje alfanumerické LCD displeje do velikosti 8×2 nebo 16×1 znaků, s rozšířením je schopný ovládat i větší displeje. Velikost každého znaku je 5×8 nebo 5×11 bodů (v katalogovém listě je tento režim označen 5×10 bodů). Obsahuje datovou paměť pro 80 znaků (2 řádky po 40 znacích), data z ní jsou použity jako adresa pro generátor znaků. [13]

Ke komunikaci používá paralelní rozhraní sestávající z vodičů RS, R/\bar{W} , E a 8 datových vodičů DB0–DB7, resp. 4 datových vodičů DB4–DB7. [13]

Podporuje různé příkazy (instrukce), nejdůležitější pro tuto práci jsou uvedené v tabulce 3.2.

■ **Tabulka 3.2** Důležité příkazy řadiče HD44780 [13]. Hodnoty DB jsou uvedeny hexadecimálně

RS	R/\bar{W}	DB [7:0]	Název	Popis	Doba trvání
0	0	01	Clear Display	Vymaže datovou paměť a přesune kurzor na začátek.	1,52 ms
0	0	04–07	Entry Mode Set	DB1 udává směr pohybu kurzoru (0 pro směr doleva, 1 pro směr doprava) a DB0 povoluje posuv při zápisu do datové paměti (0 pro vypnutí posuvu, 1 pro zapnutí posuvu).	37 μ s
0	0	08–0F	Display On/Off Control	DB2 zapíná (1) / vypíná (0) displej, DB1 zapíná (1) / vypíná (0) kurzor (horizontální čára pod znakem), DB0 zapíná (1) / vypíná (0) blikání kurzoru (přes celý znak).	37 μ s
0	0	20–3F	Function Set	DB4 nastavuje šířku paralelního rozhraní (0 pro 4bitové, 1 pro 8bitové), DB3 nastavuje režim na jednořádkový (0) nebo dvouřádkový (1), DB3 nastavuje velikost znaků (0 pro 5×8, 1 pro 5×11).	37 μ s
0	0	80–FF	Set DDRAM Address	Nastaví ukazatel do datové paměti na DB [6:0].	37 μ s
1	0	00–FF	Write data to RAM	Zapíše data do datové nebo znakové paměti a inkrementuje/dekrementuje ukazatel.	37 μ s

3.3.1.1 Komunikace

V klidovém stavu je logická úroveň vodiče E nízká. Postup zaslání příkazu závisí na zvolené šířce rozhraní. [13]

8bitové rozhraní Pro data se používá všech 8 datových vodičů DB0–DB7 [13]. Příkaz se pošle následovně [13]:

1. Jsou nastaveny požadované hodnoty na vodičích RS, R/\bar{W} a datových vodičích DB0–DB7.
2. Na vodič E je vyslán pulz (tj. vzestupná hrana následovaná sestupnou hranou).

4bitové rozhraní Pro data se používají pouze 4 datové vodiče DB4–DB7 [13]. Data příkazu jsou rozděleny na dvě poloviny a posílají se zvlášť [13]. Celý příkaz se pošle následovně [13]:

1. Jsou nastaveny požadované hodnoty na vodičích RS, R/\bar{W} . Na datové vodiče DB4–DB7 je nastavena horní polovina dat příkazu, tj. čtyři nejvýznamnější bity (DB4–DB7 z tabulky).

2. Na vodič E je vyslán pulz (tj. vzestupná hrana následovaná sestupnou hranou).
3. Na datové vodiče DB4–DB7 je nastavena spodní polovina dat příkazu, tj. čtyři nejméně významné bity (DB0–DB3 z tabulky).
4. Na vodič E je vyslán pulz (tj. vzestupná hrana následovaná sestupnou hranou).

Hodnoty na RS a R/\bar{W} jsou stabilní po dobu t_{AS} před vzestupnou hranou na E po celou šířku pulzu na E a po dobu t_{AH} po sestupné hraně na E. Hodnoty na datových vodičích jsou stabilní po dobu t_{DSW} před sestupnou hranou na E a po dobu t_H po sestupné hraně na E. Šířka pulzu je PW_{EH} . Minimální hodnoty všech těchto časových intervalů se pohybují v rozmezí do 1 μ s. Minimální doba mezi dvěma náběžnými hranami na E je do 2 μ s. [13]

3.3.1.2 Inicializace

První část (inicializace) probíhá v 8bitovém režimu následovně [13]:

1. Po zapnutí napájení je nutno 15–40 ms počkat.
2. V 8bitovém režimu je řadiči poslán příkaz *Function Set* s nastavením rozhraní na 8bitové. Na hodnotě čtyř nejméně významných datových bitů nezáleží.
3. Je nutno počkat alespoň 4,1 ms.
4. Opět je v 8bitovém režimu poslán příkaz *Function Set* s nastavením rozhraní na 8bitové; na hodnotě čtyř nejméně významných datových bitů nezáleží.
5. Je nutno počkat alespoň 100 μ s.
6. Naposledy je v 8bitovém režimu poslán příkaz *Function Set* s nastavením rozhraní na 8bitové, kde na hodnotě čtyř nejméně významných datových bitů nezáleží.
7. Pokud je žádané komunikaci s řadičem displeje přepnout do 4bitového režimu, je v 8bitovém režimu poslán příkaz *Function Set* s nastavením rozhraní na 4bitové, kde na hodnotě čtyř nejméně významných datových bitů nezáleží.

Druhá část (konfigurace) již probíhá v žádaném režimu a skládá se z následujících částí [13]:

1. Příkaz *Function Set* se všemi parametry.
2. Příkaz *Display On/Off Control* se všemi parametry.
3. Příkaz *Display Clear*.
4. Příkaz *Entry Mode Set* se všemi parametry.

3.3.2 Řadič ST7066U

Řadič ST7066U se od řadiče HD44780 svými funkcionalitami a ovládáním nijak významně neliší. Příkazy včetně doby trvání i způsob komunikace jsou totožné. U některých časových intervalů se liší jejich minimální délky, stále se však pohybují v rozmezí do 1 μ s. Mírně se liší také inicializační sekvence. [35]

Pro rozšíření na větší displeje se používá pomocný řadič ST7065C. [35]

3.4 Vstupně-výstupní expandér PCF8574

Vstupně-výstupní expandér PCF8574 umožňuje připojit k I²C sběrnici periférie, které nemají vlastní I²C rozhraní. Nabízí 8 vstupně-výstupních pinů. [36]

3.4.1 Komunikace

Expandér podporuje čtení i zápis. V I²C komunikaci vystupuje v roli *slave* s adresou z rozsahu 0100000 až 0100111 (poslední tři bity adresy se volí pomocí logické úrovně na určených pinech expandéru). Podporuje frekvenci SCL až 100 kHz. [36]

Zápis Při zápisu komunikace probíhá následovně [36]:

1. *Master* zahájí přenos (*START condition*) a adresuje expandér v režimu pro zápis.
2. Expandér vyšle *ACK*.
3. *Master* pošle 8 bitů dat.
4. Expandér potvrdí přijetí dat (*ACK*) a v tento moment všechny bity dat současně promítne na stav vstupně-výstupních pinů. Pokud byla zapsána hodnota 0, je úroveň na pinu stažena na nízkou logickou úroveň. V opačném případě je úroveň na pinu krátce (po dobu mezi vzestupnou a sestupnou hranou na SCL) vytažena na vysokou logickou úroveň, následně je tažena na vysokou úroveň pouze slabým vnitřním rezistorem a pin tak může být použit jako vstup.
Master může postup opakovat od bodu 1 (*repeated START*) nebo od bodu 3 (tj. poslat další data).
5. Nakonec *master* ukončí přenos (*STOP condition*).

Čtení Při čtení komunikace probíhá následovně [36]:

1. *Master* zahájí přenos (*START condition*) a adresuje expandér v režimu pro čtení.
2. Expandér vyšle *ACK*. V tomto okamžiku přečte hodnoty úrovní na vstupně-výstupních pinech.
3. Expandér pošle přečtené hodnoty zařízení *master*.
4. *Master* buď potvrdí přijetí dat (*ACK*), kdy expandér přečte další hodnoty a pokračuje se bodem 3, nebo přijetí nepotvrdí (*NACK*) a ukončí přenos (*STOP condition*).

3.5 Hodiny reálného času DS3231

Hodiny reálného času DS3231 nabízejí přesné udržování času v sekundách, minutách, hodinách (ve 12- nebo 24hodinovém formátu), dnech v týdnu, dnech v měsíci (bere v úvahu rozdílný počet dnů v různých měsících i přestupné roky), měsících a rocích (v rozsahu 0–99). Pro případ výpadku hlavního napájení nabízí možnost připojení záložní baterie. [21]

3.5.1 Registry

Seznam registrů je uveden v tabulce 3.3. Všechny časové údaje jsou ve formátu *binary-coded decimal* (BCD) [21].

U hodinových registrů (0x02, 0x09, 0x0C) je možné pomocí 2. nejvýznamnějšího bitu zvolit formát času jako 24hodinový (pokud je hodnota bitu rovna 0) nebo 12hodinový (pokud je hodnota bitu rovna 1). V případě 12hodinového formátu slouží 3. nejvýznamnější bit k označení *AM* (hodnota bitu je 0) nebo *PM* (hodnota bitu je 1). [21]

Druhý nejvýznamnější bit v registrech dnů u budíků (0x0A, 0x0D) označuje, zda se jedná o den v týdnu (hodnota bitu je 1) nebo o den v měsíci (hodnota bitu je 0). [21]

Nejvýznamnější bit každého registru času budíku (0x07–0x0D) značí, zda se údaj z daného registru má pro aktivaci budíku shodovat s odpovídající hodnotou aktuálního času (hodnota bitu je 0) či nikoliv (hodnota bitu je 1). Pro správnou funkci musí adresy registrů v rámci jednoho

■ **Tabulka 3.3** Registry hodin reálného času DS3231 [21]

Adresa	Popis
0x00	Sekunda
0x01	Minuta
0x02	Hodina
0x03	Den v týdnu
0x04	Den v měsíci
0x05	Měsíc
0x06	Rok
0x07	Budík 1: sekunda
0x08	Budík 1: minuta
0x09	Budík 1: hodina
0x0A	Budík 1: den
0x0B	Budík 2: minuta
0x0C	Budík 2: hodina
0x0D	Budík 2: den
0x0E	Konfigurační registr
0x0F	Stavový registr
0x10	Kompenzační registr
0x11	Teplota (MSB)
0x12	Teplota (LSB)

■ **Tabulka 3.4** Význam bitů konfiguračního registru (0x0E) obvodu DS3231 [21]

Bit	Název	Popis															
7	\overline{EOSC}	Pokud je tento bit nastaven na 1, oscilátor, podle něhož se udržuje čas, není aktivní při napájení z baterie.															
6	BBSQW	Pokud jsou bity BBSQW a INTCN nastaveny na 0, pin \overline{INT}/SQW se při napájení z baterie namísto obdélníkového výstupu přepne do stavu vysoké impedance.															
5	CONV	Zapsáním hodnoty 1 je vynucena teplotní konverze.															
4, 3	RS2, RS1	Nastavují frekvenci výstupu obdélníkového průběhu na pinu \overline{INT}/SQW podle následující tabulky:															
		<table border="1"> <thead> <tr> <th>RS2</th> <th>RS1</th> <th>Frekvence</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1 Hz</td></tr> <tr><td>0</td><td>1</td><td>1024 Hz</td></tr> <tr><td>1</td><td>0</td><td>4096 Hz</td></tr> <tr><td>1</td><td>1</td><td>8192 Hz</td></tr> </tbody> </table>	RS2	RS1	Frekvence	0	0	1 Hz	0	1	1024 Hz	1	0	4096 Hz	1	1	8192 Hz
RS2	RS1	Frekvence															
0	0	1 Hz															
0	1	1024 Hz															
1	0	4096 Hz															
1	1	8192 Hz															
2	INTCN	Určuje funkci pinu \overline{INT}/SQW . Při nastavení na 1 se pin chová jako výstup přerušování, při nastavení na 0 se pin chová jako výstup obdélníkového průběhu.															
1	A2IE	Pokud jsou bity A2IE a INTCN nastaveny na 1, a pokud je příznak A2IF z následujícího registru nastaven na 1, pak je pin \overline{INT}/SQW přepnut na nízkou logickou úroveň k indikaci přerušování.															
0	A1IE	Význam tohoto bitu je analogický k významu bitu A2IE, pouze s budíkem č. 1 namísto budíku č. 2.															

■ **Tabulka 3.5** Význam bitů stavového registru (0x0F) obvodu DS3231 [21]

Bit	Název	Popis
7	OSF	Tento bit je nastaven na 1, pokud oscilátor je nebo byl zastavený. Lze vynulovat zápisem 0.
3	EN32KHZ	Pokud je tento bit nastaven na 1, je zapnut obdélníkový výstup o frekvenci 32,768 kHz na pinu 32kHz. V opačném případě je pin přepnut do stavu vysoké impedance.
2	BSY	Tento bit je nastaven na 1, pokud probíhá teplotní konverze.
1	A2F	Tento bit je nastaven na 1, pokud je aktivní budík č. 2 a byla splněna podmínka pro aktivaci budíku. Lze vynulovat zápisem 0.
0	A1F	Význam tohoto bitu je analogický k významu bitu A2F, pouze s budíkem č. 1 namísto budíku č. 2.

budíku, které mají tento bit nastaven na 0, být sousední a začínat na indexu prvního registru pro daný budík (0x07 nebo 0x0B). [21]

Konfigurační registr nabízí ke konfiguraci celkem 8 bitů [21], jejichž význam je uveden v tabulce 3.4. Stavový registr obsahuje jeden konfigurační bit a 4 stavové bity [21]. Jejich význam je uveden v tabulce 3.5.

3.5.2 Komunikace

Hodiny DS3231 vystupují při I²C komunikaci v roli *slave* s adresou 1101000. Podporují frekvenci SCL až 400 kHz. [21]

Čtení Proces čtení z jednoho či více po sobě jdoucích registrů probíhá následovně [21]:

1. *Master* zahájí komunikaci a adresuje hodiny v režimu pro zápis.
2. *Master* pošle adresu registru, ze kterého chce číst, tím se nastaví index registru.
3. *Master* znovuzahájí komunikaci a adresuje hodiny v režimu pro čtení.
4. *Master* čte hodnoty registrů. Po každém bytu hodiny automaticky inkrementují index čteného registru. Po přečtení posledního registru (0x12) se index vrátí na začátek (0x00).
5. *Master* po přečtení posledního čteného registru nepotvrdí přijetí a ukončí komunikaci.

Při zahájení komunikace nebo přetočení indexu na první pozici (0x00) jsou hodnoty registrů udržujících čas zkopírovány do vedlejších registrů, odkud jsou čteny. To zabrání čtení nekonzistentních dat v momentě inkrementace času. [21]

Kroky 1 a 2 nejsou povinné, pokud se již ukazatel registru nachází na požadované adrese. [21]

Zápis Proces zápisu do jednoho či více po sobě jdoucích registrů probíhá následovně [21]:

1. *Master* zahájí komunikaci a adresuje hodiny v režimu pro zápis.
2. *Master* pošle adresu registru, do kterého chce zapsat, tím se nastaví index registru.
3. *Master* zapisuje hodnoty do registrů. Po každém bytu hodiny automaticky inkrementují index zapisovaného registru. Po zápisu do posledního registru (0x12) se index vrátí na začátek (0x00).
4. *Master* po zápisu do posledního zapisovaného registru ukončí komunikaci.

Při zápisu do registru udržujícího sekundy hodiny resetují čítač generující signál pro inkrementaci času, a tak má *master* celou sekundu na zápis do všech registrů. Tím se předejde zápisu nesprávného času způsobeným inkrementací času v průběhu zápisu do dalších registrů. [21]

3.6 Senzor teploty a tlaku BMP280

Senzor BMP280 dokáže měřit teplotu vzduchu s přesností 1 °C a atmosférický tlak s přesností 1,7 hPa v rozsahu 300 hPa až 1100 hPa. Podporuje komunikaci pomocí I²C a SPI rozhraní. [19]

3.6.1 Registry

Zde jsou shrnuty nejdůležitější registry.

Kalibrační data Jsou uložena v registrech 0x88 až 0x9F, které jsou jen pro čtení, a slouží ke korekci měřených hodnot. Sestávají z dvanácti 16bitových hodnot, `dig_T1` až `dig_T3` a `dig_P1` až `dig_P9`. Všechny hodnoty jsou uloženy v *little-endian* formátu. Hodnoty `dig_T1` a `dig_P1` jsou kladná 16bitová čísla, všechny ostatní hodnoty jsou celá 16bitová čísla ve dvojkovém doplňku. [19]

Reset Zápisem hodnoty 0xB6 na adresu 0xE0 dojde k resetu BMP280. Po resetu se senzor přepne do režimu *sleep*. [19]

Konfigurační registry Slouží k nastavení režimu a konfiguračních hodnot senzoru. Jsou uloženy na adresách 0xF4 (`ctrl_meas`) a 0xF5 (`config`). [19]

V registru `config` je možné nastavit, kolik vzorků se má sejmout při měření teploty (`osrs_t`, 3 nejvýznamnější bity) a tlaku (`osrs_t`, následující 3 bity) a režim senzoru (`mode`, 2 nejméně významné bity). Možné hodnoty jsou v tabulkách 3.6 a 3.7. [19]

Registr `ctrl_meas` obsahuje nastavení doby mezi měřeními v režimu *normal* (`t_sb`, 3 nejvýznamnější bity), nastavení koeficientu filtru (`filter`, následující 3 bity; více info v katalogovém listě) a nastavení režimu rozhraní SPI (v této práci nevyužito). Registry mohou nabývat hodnot z tabulek 3.8 a 3.9. [19]

Naměřené hodnoty Naměřené hodnoty teploty a tlaku jsou kladná 20bitová čísla uložená v registrech na adresách 0xF7 až 0xF9 (tlak) a 0xFA až 0xFC (teplota). 20bitová hodnota je rozšířena na 24 bitů připojením čtyř nulových bitů na konec čísla a takto je do trojice registrů uložena v *big-endian* formátu. [19]

3.6.2 Komunikace

Senzor BMP280 v I²C komunikaci vystupuje v roli *slave* s adresou 1110110 nebo 1110111 v závislosti na logické úrovni na pinu `SDO`. Podporuje frekvenci `SCL` až 3,4 MHz. [19]

Čtení Proces čtení z jednoho či více po sobě jdoucích registrů probíhá následovně [19]:

1. *Master* zahájí komunikaci a adresuje senzor v režimu pro zápis.
2. *Master* pošle adresu prvního registru, ze kterého chce číst.
3. *Master* znovuzahájí komunikaci a adresuje senzor v režimu pro čtení.
4. *Master* čte hodnoty registrů. Po každém bytu senzor automaticky inkrementuje index čteného registru.
5. *Master* po přečtení posledního čteného registru nepotvrdí přijetí a ukončí komunikaci.

Pokud v průběhu čtení dojde k aktualizaci naměřených hodnot, jsou nové hodnoty uloženy do vedlejších, nepřístupných registrů, odkud jsou do hlavních registrů propsány až po dokončení čtení. To zabrání čtení nekonzistentních dat v momentě aktualizace hodnot. [19]

■ **Tabulka 3.6** Hodnoty nastavení počtu vzorků měření (*osrs_t* a *osrs_p*) senzoru BMP280 [19]

Hodnota	Počet vzorků
000	0 (na výstupu konstanta)
001	1
010	2
011	4
100	8
101–111	16

■ **Tabulka 3.7** Hodnoty nastavení režimu (*mode*) senzoru BMP280 [19]

Hodnota	Režim	Popis
00	<i>sleep</i>	Senzor neprovádí měření.
01, 10	<i>forced</i>	Senzor provede jedno měření a vrátí se do stavu <i>sleep</i> .
11	<i>normal</i>	Senzor provádí měření periodicky.

■ **Tabulka 3.8** Hodnoty nastavení doby mezi měřeními v režimu *normal* (*t_sb*) senzoru BMP280 [19]

Hodnota	Doba mezi měřeními
000	0,5 ms
001	62,5 ms
010	125 ms
011	250 ms
100	500 ms
101	1000 ms
110	2000 ms
111	4000 ms

■ **Tabulka 3.9** Hodnoty nastavení koeficientu filtru (*filter*) senzoru BMP280 [19]

Hodnota	Filtr
000	vypnutý
001	2
010	4
011	8
100	16

■ **Výpis kódu 3.1** Kód pro výpočet teploty ze senzoru BMP280 [19]

```
// Returns temperature in DegC, resolution is 0.01 DegC.
// Output value of "5123" equals 51.23 DegC.
BMP280_S32_t t_fine; // t_fine carries fine temperature as global value
BMP280_S32_t bmp280_compensate_T_int32(BMP280_S32_t adc_T)
{
    BMP280_S32_t var1, var2, T;
    var1 = (((adc_T>>3) - ((BMP280_S32_t)dig_T1<<1))) *
            ((BMP280_S32_t)dig_T2)) >> 11;
    var2 = (((((adc_T>>4) - ((BMP280_S32_t)dig_T1)) * ((adc_T>>4) -
            ((BMP280_S32_t)dig_T1))) >> 12) * ((BMP280_S32_t)dig_T3)) >> 14;
    t_fine = var1 + var2;
    T = (t_fine * 5 + 128) >> 8;
    return T;
}
```

■ **Výpis kódu 3.2** Kód pro výpočet tlaku ze senzoru BMP280. Používá hodnotu `t_fine` vypočtenou předchozí funkcí [19]

```
// Returns pressure in Pa as unsigned 32 bit integer.
// Output value of "96386" equals 96386 Pa = 963.86 hPa
BMP280_U32_t bmp280_compensate_P_int32(BMP280_S32_t adc_P)
{
    BMP280_S32_t var1, var2;
    BMP280_U32_t p;
    var1 = (((BMP280_S32_t)t_fine)>>1) - (BMP280_S32_t)64000;
    var2 = (((var1>>2) * (var1>>2)) >> 11 ) * ((BMP280_S32_t)dig_P6);
    var2 = var2 + ((var1*((BMP280_S32_t)dig_P5)<<1);
    var2 = (var2>>2)+(((BMP280_S32_t)dig_P4)<<16);
    var1 = (((dig_P3 * ((var1>>2) * (var1>>2)) >> 13 )) >> 3) +
            (((BMP280_S32_t)dig_P2) * var1)>>1))>>18;
    var1 = (((32768+var1))*((BMP280_S32_t)dig_P1)>>15);
    if (var1 == 0)
    {
        return 0; // avoid exception caused by division by zero
    }
    p = (((BMP280_U32_t)(((BMP280_S32_t)1048576)-adc_P)-(var2>>12))*3125;
    if (p < 0x80000000)
    {
        p = (p << 1) / ((BMP280_U32_t)var1);
    }
    else
    {
        p = (p / (BMP280_U32_t)var1) * 2;
    }
    var1 = (((BMP280_S32_t)dig_P9) * ((BMP280_S32_t)(((p>>3) * (p>>3))>>13)))
            >>12;
    var2 = (((BMP280_S32_t)(p>>2)) * ((BMP280_S32_t)dig_P8))>>13;
    p = (BMP280_U32_t)((BMP280_S32_t)p + ((var1 + var2 + dig_P7) >> 4));
    return p;
}
```

Zápis Proces zápisu do jednoho či více registrů probíhá následovně [19]:

1. *Master* zahájí komunikaci a adresuje senzor v režimu pro zápis.
2. *Master* pošle adresu registru, do kterého chce zapsat.
3. *Master* zapíše hodnotu registru.
4. Postup se buď opakuje pro další registry od kroku 2 posláním další adresy registru, nebo pokračuje krokem 5.
5. *Master* ukončí komunikaci.

3.6.3 Převod hodnot

K převodu hodnot naměřených senzorem BMP280 na hodnoty v jednotkách °C a Pa je nutné použít algoritmy uvedené v katalogovém listu BMP280, které k výpočtu využívají i kalibrační data. Algoritmy jsou zveřejněné v jazyce C ve třech verzích: s využitím aritmetiky s plovoucí řádovou čárkou, s využitím 64bitové celočíselné aritmetiky nebo s využitím 32bitové celočíselné aritmetiky s omezenou přesností. Tato práce bude vycházet z verze s 32bitovou aritmetikou, které je uvedena ve výpisech kódu 3.1 a 3.2. [19]

3.7 Senzor teploty a vlhkosti SHT30

Senzor SHT30 měří teplotu vzduchu a relativní vlhkost vzduchu. Ke komunikaci používá pouze rozhraní I²C. [17]

3.7.1 Příkazy a komunikace

Senzor SHT30 v I²C komunikaci vystupuje v roli *slave* s adresou 1000100 nebo 1000101 v závislosti na logické úrovni na pinu ADDR. Podporuje frekvenci SCL až 1 MHz. [17]

Jednorázové měření Pro jednorázové měření lze nastavit opakovatelnost měření a zda se má provádět *clock stretching*. [17]

Průběh komunikace pro jednorázové měření je následující [17]:

1. *Master* zahájí komunikaci a adresuje senzor v režimu pro zápis.
2. *Master* odešle oba byty příkazu podle tabulky 3.10. Senzor začne provádět měření.
3. *Master* může ukončit komunikaci.
4. *Master* zahájí komunikaci a adresuje senzor v režimu pro čtení.
5. Jestliže ještě nebylo dokončeno měření, pak v závislosti na nastavení *clock stretching*:
 - vypnuto** Senzor na adresaci odpoví *NACK*. Postup se opakuje od bodu 4.
 - zapnuto** Senzor odpoví *ACK* a stáhne vodič SCL na nízkou logickou úroveň. Po dokončení měření senzor vodič SCL opět uvolní. Postup pokračuje bodem 6.
6. *Master* přečte až 6 bytů dat. Jejich význam je popsán v tabulce 3.12.
7. *Master* ukončí komunikaci.

Periodické měření Pro periodické měření lze nastavit opakovatelnost a frekvenci měření. [17]

Pro měření v tomto režimu je nejprve nutné režim periodického měření zapnout [17]:

1. *Master* zahájí komunikaci a adresuje senzor v režimu pro zápis.
2. *Master* odešle oba byty příkazu podle tabulky 3.11. Senzor začne provádět měření.
3. *Master* ukončí komunikaci.

■ **Tabulka 3.10** Příkaz pro zahájení jednorázového měření pomocí SHT30 [17]

Opakovatelnost	<i>Clock stretching</i>	Příkaz
vysoká	vypnuto	0x24, 0x00
střední	vypnuto	0x24, 0x0B
nízká	vypnuto	0x24, 0x16
vysoká	zapnuto	0x2C, 0x06
střední	zapnuto	0x2C, 0x0D
nízká	zapnuto	0x2C, 0x10

■ **Tabulka 3.11** Příkaz pro zahájení periodického měření pomocí SHT30 [17]

Opakovatelnost	MPS ^a	Příkaz
vysoká	0.5	0x20, 0x32
střední	0.5	0x20, 0x24
nízká	0.5	0x20, 0x2F
vysoká	1	0x21, 0x30
střední	1	0x21, 0x26
nízká	1	0x21, 0x2D
vysoká	2	0x22, 0x36
střední	2	0x22, 0x20
nízká	2	0x22, 0x2B
vysoká	4	0x23, 0x34
střední	4	0x23, 0x22
nízká	4	0x23, 0x29
vysoká	10	0x27, 0x37
střední	10	0x27, 0x21
nízká	10	0x27, 0x2A

^a počet měření za sekundu

■ **Tabulka 3.12** Význam jednotlivých bytů naměřených dat z SHT30 [17]

Pořadí	Význam
1.	Hodnota z teplotního čidla (MSB)
2.	Hodnota z teplotního čidla (LSB)
3.	Kontrolní součet hodnoty z teplotního čidla
4.	Hodnota z čidla vlhkosti (MSB)
5.	Hodnota z čidla vlhkosti (LSB)
6.	Kontrolní součet hodnoty z čidla vlhkosti

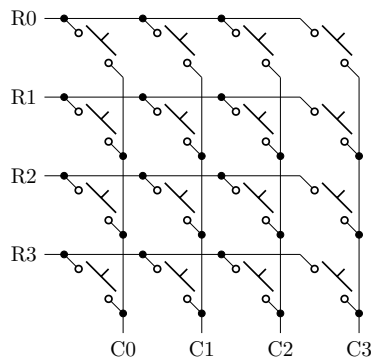
Následně lze přečíst naměřené hodnoty [17]:

1. *Master* zahájí komunikaci a adresuje senzor v režimu pro zápis.
2. *Master* odešle oba byty příkazu pro čtení hodnot (0xE0, 0x00).
3. *Master* zahájí komunikaci a adresuje senzor v režimu pro čtení.
4. Jestliže od posledního přečtení hodnoty ještě nebylo dokončeno další měření, senzor na adresaci odpoví *NACK* a postup lze opakovat od bodu 3 nebo lze komunikaci ukončit. V opačném případě postup pokračuje bodem 5.
5. *Master* přečte až 6 bytů dat. Jejich význam je popsán v tabulce 3.12.
6. *Master* ukončí komunikaci.

3.8 Maticová klávesnice

Maticová klávesnice o m řádcích a n sloupcích má celkem $m + n$ vodičů, jeden pro každý řádek a jeden pro každý sloupec, viz obr. 3.1. Ve výchozím stavu nejsou žádné z těchto vodičů propojeny. Stisk klávesy propojí řádkový vodič odpovídající řádku stisknuté klávesy se sloupcovým vodičem odpovídajícím sloupci stisknuté klávesy. [37]

Nevýhodou maticových klávesnic je, že při stisku více kláves nemusí jít jednoznačně rozpoznat, které klávesy jsou stisknuty (tzv. *ghosting*, příp. *masking*). Lze mu zamezit použitím klávesnice, která má s tlačítky sériově zapojené diody. [38], [39]



■ **Obrázek 3.1** Schéma maticové klávesnice 4×4. Inspirováno [40]

3.8.1 Snímání stisknutých kláves

Jednoduchým řešením čtení stavu kláves je následující postup [37]:

- Ke každému vodiči (nebo alespoň ke sloupcovým vodičům) je přiřazen *pull-up* rezistor.
- Řádkové vodiče jsou připojeny k výstupům řadiče a sloupcové vodiče ke vstupům řadiče.
- Řadič skenuje klávesnici řádek po řádku:
 1. Na aktuální řádek je vyslána nízká logická úroveň a všechny ostatní výstupy jsou přepnuty do stavu vysoké impedance. Pokud by byly přepnuty na nízkou logickou úroveň, byly by chybně snímány i ostatní řádky. Pokud by byly uvedeny na vysokou logickou úroveň, došlo by ke zkratu při stisku více kláves ve stejném sloupci.

2. Řadič přečte logické úrovně na sloupcových vodičích. Jestliže je logická úroveň na vodiči vysoká, znamená to, že klávesa na pozici odpovídající sloupcovému a aktuálnímu řádkovému vodiči není stisknuta. V opačném případě tato klávesa stisknuta je, příp. se může jednat o falešný obraz z důvodu jevu *ghosting* při stisku více kláves.
3. Postup se opakuje s dalším řádkem.

Jev *ghosting* lze buď vůbec neřešit, pak mohou být chybně zaznamenány stisky kláves, které stisknuty nebyly, nebo lze rozpoznat problémové situace a zpracovat je jako chybu nebo je ignorovat. Alternativně lze ignorovat jakýkoli stisk více kláves, což je vhodné v případě, kdy se očekává stisk pouze jedné klávesy.

3.9 Rotační enkodér

Enkodér na modulu KY-040 je mechanický inkrementální rotační enkodér s 20 zarážkami (stabilními pozicemi) [5]. Typ použitého enkodéru není uveden, ale z pouzder podobných enkodérů vychází 5 vývodů, z nichž 3 jsou pro samotný enkodér (A, B a C) a zbylé 2 jsou spínány vestavěným tlačítkem, které lze stisknout zatlačením hřídele enkodéru [41], [42].

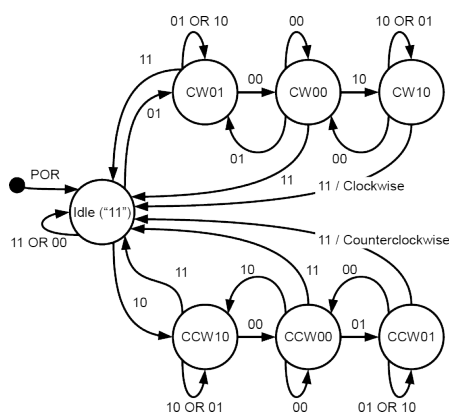
V klidovém stavu ve stabilní pozici jsou vodiče A, B a C rozpojeny [42]. Při otáčení enkodéru se stavy mění následovně:

Otáčení doprava Při otáčení vpravo mezi dvěma stabilními pozicemi proběhne následující sekvence událostí [42]:

1. Vodiče A a C jsou propojeny.
2. Vodiče B a C jsou propojeny.
3. Vodiče A a C jsou rozpojeny.
4. Vodiče B a C jsou rozpojeny.

Otáčení doleva Při otáčení vlevo mezi dvěma stabilními pozicemi pak postupně proběhnou následující události [42]:

1. Vodiče B a C jsou propojeny.
2. Vodiče A a C jsou propojeny.
3. Vodiče B a C jsou rozpojeny.
4. Vodiče A a C jsou rozpojeny.



■ **Obrázek 3.2** Konečný automat pro detekci otáčení rotačního enkodéru [43]

Po propojení vodiče C se zemí a připojení *pull-up* rezistorů k vodičům A a B, jak je doporučeno v [41] a jak tomu je i na modulu KY-040, lze na vodičích A a B pozorovat logické úrovně a podle nich detekovat rotaci enkodéru a její směr. V případě, že je vodič A, resp. B uvnitř enkodéru propojen s vodičem C, je na vodiči A, resp. B nízká logická úroveň. V opačném případě je na daném vodiči vysoká logická úroveň. Tyto hodnoty pak může použít řadič k detekci otáčení hřídele enkodéru.

Pro detekci otáčení existuje návrh konečného automatu na obr. 3.2. Má celkem 7 stavů a je odolný vůči záměnitelnosti za předpokladu, že k nim v každý okamžik dochází nejvýše na jednom z vodičů A a B. [43]

3.10 Celočíselné dělení

Pro výpočet některých hodnot bude zapotřebí provést celočíselné dělení. Na rozdíl od operátorů jazyka VHDL „+“, „-“ a „*“ pro sčítání, odčítání a násobení, které jsou při syntéze pomocí nástroje Xilinx Vivado plně podporovány, dělení pomocí VHDL operátoru „/“ je oficiálně podporováno pouze, pokud je dělitel mocninou dvou nebo jsou oba operandy konstantní [44]. Pro dělení je tak nutné použít jiné řešení, lze se inspirovat např. algoritmem písemného dělení se zbytkem.

Algoritmus dělení k -bitového nezáporného dělece A (binárně a_{k-1} až a_0) l -bitovým kladným dělitelem D (binárně d_{l-1} až d_0) s k -bitovým podílem Q (binárně q_{k-1} až q_0) a l -bitovým zbytkem R (binárně r_{l-1} až r_0) je vyobrazen na obr. 3.3. Algoritmus lze přímočaře implementovat pro FPGA (podrobněji v kapitole Návrh), taková možnost je i přes možné nedostatky pro tuto práci postačující.

Input: A, D

Output: Q, R

```

R ← 0;
for i ← k - 1 downto 0 do
    rl-1, ..., r1 ← rl-2, ..., r0;
    r0 ← ai;
    if R ≥ D then
        R ← R - D;
        qi ← 1;
    else
        qi ← 0;
    end
end
end
```

■ **Obrázek 3.3** Algoritmus celočíselného dělení se zbytkem [45]

3.11 Písma

Pro generování textu v obrazových datech pro OLED displej je vhodné použít nějaký font. Aby nebylo potřeba v této práci navrhovat vlastní, lze použít některý z již existujících:

Spleen Bitmapový font dostupný v několika sadách velikostí znaků, vč. 5×8 bodů. Všechny znaky v jedné sadě mají stejný rozměr. [46]

Font8x8 Bitmapový font s velikostí všech znaků 8×8 bodů. [47]

3.12 Nástroje pro testování

V průběhu vývoje knihovny a aplikace v jazyce VHDL lze pro účely testování použít následující nástroje:

GHDL Tento nástroj provádí analýzu zdrojových kódů v jazyce VHDL a kompiluje je do spustitelného souboru, který provádí simulaci návrhu. Volitelně může při simulaci vygenerovat soubor ve formátu GHW nebo VCD se záznamem průběhu simulovaných signálů v čase. [48]

GTKWave Slouží ke grafickému zobrazení průběhu signálů v čase. Podporuje několik formátů včetně GHW a VCD. [49]

Xilinx Vivado Tato sada nástrojů společnosti Xilinx je určena pro vývoj a testování návrhů pro její FPGA obvody. Nabízí mnoho dílčích nástrojů pokrývajících celý vývojový proces. Tyto nástroje jsou přístupné společně v jedné aplikaci. [50]

Návrh knihovny a aplikace

V této kapitole je popsán návrh jednotlivých bloků knihovny a testovací aplikace.

4.1 Společné vlastnosti modulů

Moduly jsou vyvíjeny v jazyce VHDL. Je použita konvence pojmenování vstupů a výstupů modulů, kde pojmenování vstupních signálů začíná předponou `i_` a výstupních signálů předponou `o_`. Vstupně-výstupní signály jsou pak pojmenovány s předponou `io_`. Není-li uvedeno jinak, je aktivní logická úroveň všech řídicích a stavových signálů vysoká, a výchozí tedy nízká.

Většina modulů obsahuje následující vstupy a výstupy:

`i_clk` Hodinový signál. Modul mění svůj stav při náběžné hraně tohoto signálu.

`i_reset` Synchronní *active-high* reset. Slouží k uvedení modulu do počátečního stavu.

`o_done` Signál signalizující dokončení požadovaného příkazu. Vstupy modulů není doporučeno měnit před signalizací dokončení pomocí tohoto signálu. Signál je aktivní po dobu jednoho hodinového cyklu. Po signalizaci dokončení příkazu modul ještě následující hodinový cyklus ignoruje vstupní signály.

Moduly ovládající periferie přes I²C sběrnici navíc obsahují řídicí a datové signály pro ovládání modulu `i2c_master` popsané dále a generický parametr `g_I2C_ADDRESS` na specifikaci adresy periferie.

Některé moduly mají generický parametr `g_CLK_PERIOD_NS`, který používají k určení konstant čítačů k měření času pro čekání.

4.2 Modul pro řízení I²C sběrnice

Modul `i2c_master` využívá modul `i2c_master_byte_ctrl` v jazyce VHDL z implementace *I2C Controller Core* od Richarda Herveilleho. Umožňuje přizpůsobení pomocí generického parametru `g_CLOCK_DIVIDER`, který slouží k nastavení frekvence hodin I²C sběrnice.

Pro připojení na fyzické piny FPGA obsahuje vstupně-výstupní signály `io_scl_pad` (pro hodinový signál SCL) a `io_sda_pad` (pro datový signál SDA).

Ovládání z ostatních modulů na FPGA je realizováno pomocí následujících signálů:

`i_start` Příkaz na zahájení přenosu (*START* nebo *repeated START*). Lze zkombinovat s příkazy `i_read` a `i_write`, pak je před danou operací (znovu)zahájen přenos.

i_stop Příkaz na ukončení přenosu (*STOP*). Lze zkombinovat s příkazy **i_read** a **i_write**, pak je po dané operaci ukončen přenos.

i_read Příkaz na přečtení jednoho bytu dat (**o_data**) a vyslání potvrzovacího signálu (**i_nack**).

i_write Příkaz na zápis jednoho bytu dat (**i_data**) a přečtení potvrzovacího signálu (**o_nack**).

o_done Příznak dokončení příkazu.

i_data Data (1 byte) k vyslání na sběrnici.

o_nack Potvrzovací signál přečtený ze sběrnice (0 = *ACK*, 1 = *NACK*).

o_data Data (1 byte) přečtená ze sběrnice.

i_nack Potvrzovací signál k vyslání na sběrnici (0 = *ACK*, 1 = *NACK*).

4.3 Modul pro ovládání OLED displeje

Modul `oled_display` dokáže ovládat OLED displej s řadičem SSD1306 nebo SH1106. Umožňuje přizpůsobení pomocí čtyř generických parametrů:

g_I2C_ADDRESS 7bitová I²C adresa ovládaného řadiče displeje.

g_CONTROLLER_TYPE Typ řadiče displeje. Může nabývat hodnot `SSD1306` a `SH1106`.

g_FLIP_HORIZONTAL Ovlivňuje, zda se při inicializaci má odeslat příkaz k horizontálnímu převrácení displeje.

g_FLIP_VERTICAL Ovlivňuje, zda se při inicializaci má odeslat příkaz k vertikálnímu převrácení displeje.

Modul podporuje následující 3 příkazy:

Inicializace Provede příkazy nutné k inicializaci a zapnutí displeje. Spouští se signálem **i_init**.

Nastavení pozice Nastaví stránkový a sloupcový ukazatel do obrazové paměti řadiče. Spouští se signálem **i_set_position** a parametry jsou zadány signály **i_page** a **i_column**.

Zápis obrazových dat Zapiše data do obrazové paměti řadiče displeje. Spouští se signálem **i_write_data** a data jsou zadána 8bitovým signálem **i_data**. Umožňuje zápis několika bytů při jednom přenosu. S posledním zapisovaným bytem je nutné aktivovat signál **i_last_byte**.

Pro usnadnění vykreslování na displej z aplikace byly vytvořeny následující moduly a balíčky:

Modul `oled_draw` Usnadňuje překreslení celého displeje. Po zahájení příkazu **i_redraw** pomocí výstupů **o_page** a **o_col** adresuje nějaký modul aplikace, který vrací data k vykreslení, a pomocí řídicích signálů **o_oled_set_position**, **o_oled_write** a **o_oled_last** řídí modul `oled_display` pro vykreslení dat na displej.

Balíček `display_utils_pkg` Obsahuje funkce pro usnadnění tvorby obvodu pro generování obrazových dat. Obsahuje funkce na vykreslování textu (pomocí fontů `Spleen` a `font8x8`), tlačítek a záhlaví obrazovky aplikace vč. volitelného tlačítka *zpět*.

4.4 Modul pro ovládání alfanumerického displeje

Modul `character_lcd` dokáže ovládat alfanumerický LCD displej s řadičem HD44780 nebo kompatibilním. Z generických parametrů má pouze `g_CLK_PERIOD_NS` a `g_I2C_ADDRESS`.

Podporuje následující příkazy:

Inicializace Provede příkazy nutné k inicializaci a zapnutí displeje. Spouští se signálem `i_init`.

Vymazání displeje Vymaže obsah displeje. Spouští se signálem `i_clear`.

Nastavení pozice Nastaví ukazatel do datové paměti řadiče displeje. Spouští se pomocí signálu `i_set_position` a parametry jsou zadány signály `i_row` (řádek) a `i_col` (sloupec).

Zápis znaku Zapiše data do datové paměti řadiče displeje. Spouští se signálem `i_send` a data jsou zadána 8bitovým signálem `i_data`. Umožňuje zápis několika bytů při jednom přenosu. S posledním zapisovaným bytem je nutné aktivovat signál `i_send_last`.

Modul `character_lcd` je rozdělen na dvě části:

Modul `character_lcd_controller` V reakci na žádané příkazy modulu generuje dílčí příkazy (hodnoty `RS`, `R/W` a `DB0–DB7` pojmenované jako `o_rs`, `o_rw` a `o_data`), které předává spolu s řídicími signály následujícímu modulu. Obstarává i potřebná čekání na vyřízení příkazu řadičem displeje. Krátká čekání jsou vynechána díky malé použité rychlosti sběrnice I²C.

Modul `character_lcd_i2c_sender` Příkazy transformuje na dílčí části spolu se signálem `E` řadiče (v této knihovně pojmenovaným `w_en`) podle schématu 4bitové komunikace. Části jsou následně pomocí I²C posílány vstupně-výstupnímu expandéru PCF8574, který je promítnut na pinech řadiče displeje.

4.5 Modul pro celočíselné dělení

Celočíselná dělička `divider` umožňuje dělení dvou celých čísel, nezáporného dělence a kladného dělitele. Je realizována za pomoci následujících dílčích modulů:

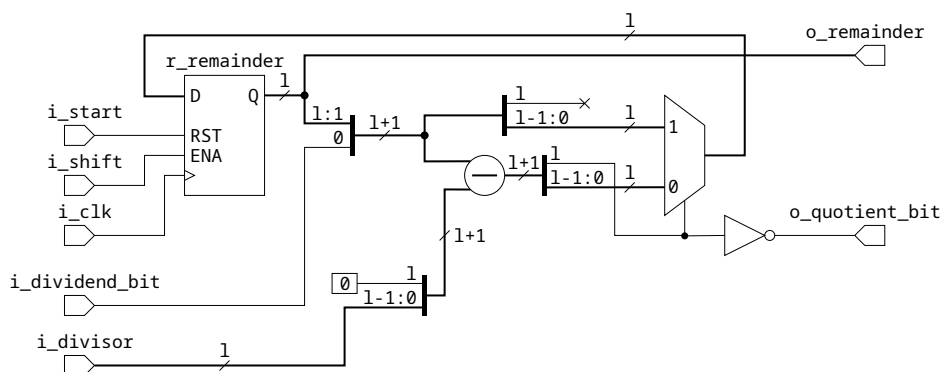
`divider_core` Modul, který provádí hlavní tělo algoritmu dělení – na vstupu má bit dělence, na výstupu bit podílu a udržuje zbytek po dělení. Schéma tohoto modulu je na obrázku 4.1.

`divider_datapath` Obsahuje posuvný registr, do kterého je na počátku dělení načten dělenec, který je následně po bitech předáván modulu `divider_core`. Do tohoto posuvného registru je zároveň po bitech načítán výsledný podíl.

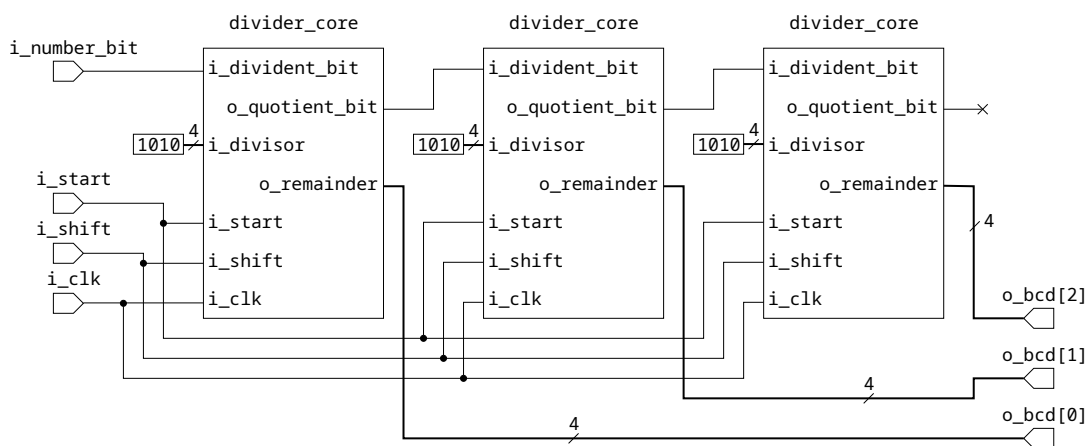
`divider_controller` Řadič, který vysílá řídicí signály do ostatních částí.

4.6 Modul pro převod čísla do binárně kódovaných dekadických číslic

Modifikací modulu `divider` je vytvořen modul `bcd_converter`, který převádí číslo na sekvenci binárně kódovaných dekadických číslic (BCD). Modul je velice podobný modulu `divider`, liší se přejmenováním označení `dividend` na `number` a tím, že blok `divider_core` je nahrazen jejich sekvencí, jedním pro každou výstupní číslici, viz obr. 4.2. Také je dělitel pevně nastaven na 10. Na výstupu modulu je pole všech dílčích podílů `o_bcd`, což je výsledek převodu čísla do BCD. Díky sudosti dělitele lze modul implementovat tak, aby výstup `o_quotient_bit` modulu `divider_core` přímo nezávisel na jeho vstupu `i_dividend_bit`.



■ **Obrázek 4.1** Schéma modulu `divider_core` celočíselné děličky. Vlevo je registr udržující aktuální zbytek po dělení, vpravo je pomocí rozdílu hodnot určena hodnota bitu podílu a nová hodnota zbytku po dělení.



■ **Obrázek 4.2** Schéma zřetězení tří modulů `divider_core` celočíselné děličky pro převod do BCD

4.7 Modul pro ovládání senzoru BMP280

Modul `bmp280` dokáže ovládat digitální tlakový a teplotní senzor BMP280. Z generických parametrů má `g_CLK_PERIOD_NS` a `g_I2C_ADDRESS`.

Modul podporuje následující příkazy:

Inicializace Přečte kalibrační data do paměti FPGA obvodu pro další použití. Spouští se signálem `i_init`.

Nastavení konfigurace Nastaví konfiguraci senzoru. Spouští se signálem `i_set_config` a parametry jsou zadány složeným signálem `i_config`, který obsahuje položky `t_sb`, `filter`, `osrs_t`, `osrs_p` a `mode`.

Čtení teploty a tlaku Přečte hodnoty ze senzoru a převede je na správné jednotky vč. kompenzace pomocí kalibračních dat. Spouští se signálem `i_update`. Výstupem jsou signály `o_temp` (teplota v setinách °C ve dvojkovém doplňku) a `o_press` (tlak v jednotkách Pa).

Modul `bmp280` je složen z několika dílčích modulů:

Modul `bmp280_communication` Tento modul obsluhuje komunikaci se senzorem a podporuje příkazy `set_config` (nastavení konfigurace senzoru), `read_calib` (přečtení kalibračních dat ze senzoru) a `read_adc` (přečtení naměřených hodnot teploty a tlaku ze senzoru bez jejich konverze).

Modul `bmp280_compensation` Převádí naměřené hodnoty do správných jednotek a provádí kompenzaci podle kalibračních dat. Postupuje podle 32bitové verze algoritmu. Pro dělení využívá modul `divider`.

Modul `bmp280_controller` Přeposílá příkazy z `bmp280` do `bmp280_communication` a v případě příkazu `i_update` po přečtení dat ze senzoru navíc zahájí jejich převod.

4.8 Modul pro ovládání senzoru SHT30

Modul `sht30` ovládá digitální teplotní a vlhkostní senzor SHT30. Má jediný generický parametr, `g_CLK_PERIOD_NS`.

Modul podporuje následující příkazy:

Nastavení konfigurace Nastaví konfiguraci senzoru pro periodické měření. Spouští se signálem `i_set_config` a parametry jsou zadány složeným signálem `i_config`, který obsahuje položky `repeatability` a `mps`.

Čtení teploty a vlhkosti Přečte hodnoty ze senzoru a převede je na správné jednotky. Spouští se signálem `i_read`. Výstupem jsou signály `o_temperature` (teplota v setinách °C ve dvojkovém doplňku) a `o_humidity` (relativní vlhkost v setinách %).

Modul `sht30` je složen z následujících dílčích modulů:

Modul `sht30_communication` Tento modul obsluhuje komunikaci se senzorem a provádění příkazů bez konverze hodnot.

Modul `sht30_conversion` Převádí naměřené hodnoty do správných jednotek. Pro dělení využívá modul `divider`.

Modul `sht30_controller` Přeposílá příkazy z `sht30` do `sht30_communication` a v případě příkazu `i_read` po přečtení dat ze senzoru navíc zahájí jejich převod.

4.9 Modul pro ovládání hodin reálného času DS3231

Modul *rtc* ovládá hodiny reálného času DS3231. Registry hodin jsou rozděleny na čtyři části: aktuální čas, alarm 1, alarm 2 a konfigurační/stavové. Pro každou z těchto částí existuje příkaz pro její přečtení i zápis.

Modul je rozdělen do následujících částí:

rtc_controller Hlavní modul, který reaguje na příkazy a řídí zbylé moduly.

i2c_communicator Modul, který zařizuje čtení a zápis do registrů hodin přes rozhraní I²C na úrovni bytů. Na vstupu přijímá počáteční a koncovou adresu registru, příkaz čtení/zápis a případná data k zápisu. Na výstupu má adresu aktuálního bytu a v případě čtení i přečtená data a signál, že byte byl přečten. Modul je složen ze dvou dílčích modulů, **i2c_reader** pro čtení a **i2c_writer** pro zápis.

rtc_data Modul, který uchovává přečtená data registrů a poskytuje rozhraní pro přečtená data i data k zápisu ve formě obalujících struktur pro každou ze čtyř částí.

4.10 Moduly pro zpracování vstupů

Pro použití vstupů z tlačítek nebo jiných periférií a jejich zpracování jsou v knihovně navrženy následující moduly:

synchronizer Synchronizuje asynchronní vstupy do hodinové domény pomocí dvou sériově zapojených klopných obvodů typu D.

debouncer Odstraňuje záškuby vstupů. Svůj výstup změní na hodnotu vstupu pouze, pokud je tato hodnota stabilní po dobu danou generickým parametrem `g_DEBOUNCE_TIME_MS`.

edge_detector Detekuje hrany na vstupu. Typ detekované hrany lze nastavit pomocí generického parametru `g_EDGE_TYPE` na vzestupnou hranu, sestupnou hranu, nebo libovolnou z nich. Na výstupu se v okamžik detekce hrany objeví pulz o délce jednoho hodinového cyklu.

4.11 Modul pro čtení rotačního enkodéru

Modul **rotary_encoder** umožňuje detekci otočení hřídele rotačního enkodéru doleva nebo doprava. Při detekci otočení doleva vyšle pulz na výstup `o_dec` a při detekci otočení doprava jej vyšle na výstup `o_inc`. Výstupy jsou aktivní po dobu jednoho hodinového cyklu. Oproti návrhům z [6] a [43] je přidán chybový stav, do kterého automat přechází při neočekávaném vstupu (tj. když se úrovně na signálech A a B změní současně) a který je zároveň počátečním stavem. Z chybového stavu automat přejde do výchozího stavu po návratu enkodéru do výchozí polohy.

4.12 Modul pro čtení vstupu z maticové klávesnice

Modul **keypad** umožňuje čtení vstupu z maticové klávesnice. Je rozdělen na dvě části:

keypad_core Zajišťuje čtení vstupu z klávesnice pomocí algoritmu popsaného v předchozí kapitole. Na výstupu má pole signálů udávajících informaci o stavu kláves (1 pokud klávesa je stisknuta, 0 pokud klávesa stisknuta není), pro každou klávesu obsahuje jeden signál.

keypad_getkey Z výstupu předchozího modulu určuje index právě stisknuté klávesy. Pokud není stisknuta žádná klávesa, vrací hodnotu `-1`. Pokud je stisknuta právě jedna klávesa, vrátí její index převodem z kódu 1 z *n*. V případě, že je stisknuto více kláves současně, je výstup nezměněn.

4.13 Aplikace pro vývojovou desku Basys 3

Aplikace je rozdělena do několika aktivit. Jako první je spuštěno menu, které umožňuje spuštění všech dalších aktivit. Vstup z rotačního enkodéru s tlačítkem je posílán pouze aktuálně spuštěné aktivitě, která má zároveň exkluzivní přístup k OLED displeji v podobě řídicích a datových signálů pro moduly `oled_display` a `oled_draw` a příp. i k samotné I²C sběrnici skrze řídicí a datové signály pro modul `i2c_master`. K ostatním periferiím přístup být řízen nemusí, protože mohou být ovládány pouze z jedné aktivity k tomu určené.

Každá aktivita je složena z modulu, který může obsahovat instance dalších modulů pro ovládání periferií a řadič, který slouží jako prostředník pro řízení většiny komunikace aktivity s řadiči na stejné nebo vyšší úrovni, dále z generátoru obrazových dat pro OLED displej a z modulu obsahujícího vlastní logiku aktivity.

Vlastní logika aktivity zpracovává vstupy z rotačního enkodéru s tlačítkem a případně i z dalších periferií nebo řadičů a reaguje na ně změnou hodnot vnitřních registrů nebo vyvoláním jiné akce, např. vysláním příkazu periférii nebo návratu do menu. Také žádá o překreslení displeje.

Příkazy z modulu s vlastní logikou nejsou zasílány přímo odpovídajícím řadičům, ale pro zachování exkluzivního přístupu jsou předány řadiči aktivity. Tím je zajištěn exkluzivní přístup k těmto řadičům, zejm. s ohledem na řadič I²C sběrnice, se kterým může komunikovat jak modul ovládající OLED displej na úrovni aplikace, tak i modul ovládající jinou I²C periférii na úrovni aktivity.

Pro data vykreslovaná na displej je v každé aktivitě použit vlastní modul generující obrazová data z vnitřních registrů aktivity s pomocí funkcí z balíčku `display_utils_pkg`. Návrh obrazovek všech aktivit je v příloze A.

Celá aplikace pro vývojovou desku Basys 3 se skládá z následujících částí:

Modul `application_basys` Hlavní modul aplikace. Obsahuje instance modulů pro čtení rotačního enkodéru s tlačítkem, pro řízení I²C sběrnice a pro ovládání OLED displeje. Obsahuje také instance všech aktivit aplikace a řadič `application_basys_controller` pro počáteční spuštění aplikace, což zahrnuje inicializaci OLED displeje a spuštění aktivity menu. Pro přepínání přístupu k periferiím modul používá výstup `o_active_activity` modulu aktivity menu, který udává aktuálně spuštěnou aktivitu.

Aktivita Menu Jedná se o speciální aktivitu, která neobsahuje tlačítko *zpět* a slouží pro výběr další aktivity a řízení přístupu. Při spuštění jiné aktivity přejde aktivita Menu do pozastaveného stavu a řízení je předáno právě spouštěné aktivitě. Při návratu z aktivity znovu aktivita Menu převezme řízení.

Aktivita LCD Nabízí možnost inicializovat displej, vymazat obsah jeho datové paměti, nastavit pozici kurzoru a zapsat libovolný znak nebo sekvenci „Hello, World!“ do datové paměti alfanumerického displeje.

Aktivita LEDs Nabízí možnost rozsvítit nebo zhasnout jednotlivé svítivé diody na vývojové desce.

Aktivita Buttons Zobrazuje stav všech tlačítek a přepínačů na vývojové desce, právě stisknutou klávesu na maticové klávesnici a čítač inkrementovaný a dekrementovaný pomocí rotačního enkodéru. Nakonec ukazuje čítač inkrementovaný s každou změnou některé z těchto hodnot, což je využito pro sledování správného potlačení zákmitů.

Aktivita BMP280 Umožňuje nastavit konfiguraci senzoru teploty a tlaku BMP280 a přečíst naměřené hodnoty. Hodnota teploty je zobrazena v setinách °C a hodnota tlaku pak v jednotkách Pa.

Aktivita SHT30 Umožňuje nastavit konfiguraci senzoru teploty a vlhkosti SHT30 pro periodické měření a přečíst naměřené hodnoty. Teplota je zobrazena v setinách °C a vlhkost v setinách %.

Aktivita DS3231 Tato aktivita je z důvodu omezeného prostoru na OLED displeji rozdělena na čtyři obrazovky. Každá obrazovka má podobnou strukturu jako celá jiná aktivita, instance modulu pro komunikaci s RTC obvodem je však společná. První obrazovka umožňuje nastavení a zobrazení aktuálního data a času, druhá obrazovka umožňuje totéž s budíkem č. 1 a třetí obrazovka s budíkem č. 2. Čtvrtá obrazovka slouží k zobrazení a nastavení konfigurace a příznaků.

V této kapitole je popsáno, jakým způsobem byla testována knihovna a aplikace vyvíjená v této práci.

5.1 Použité součástky a moduly

Pro testování knihovny a aplikace na fyzickém přípravku byla použita vývojová deska Basys 3, která byla pro účely této práce vypůjčena od katedry, a následující součástky:

OLED displej Pro testování byly použity dva OLED displeje s rozlišením 128×64 bodů. Jeden z nich, koupený z [7], obsahoval řadič SSD1306, druhý, zakoupený z [8], pak SH1106.

Alfanumerický displej Pro testování byl použit modul dodaný vedoucím práce, který kromě displeje obsahoval i vstupně-výstupní expandér PCF8574. Modul s displejem vypadá totožně s modulem dostupným z [15].

Převodník logických úrovní Pro převod logických úrovní I²C sběrnice mezi alfanumerickým displejem pracujícím při napětí 5 V a vývojovou deskou akceptující napětí 3,3 V byl použit čtyřkanálový převodník z [25], z něhož byly použity pouze dva kanály. Dvoukanálový převodník s přidaným napěťovým regulátorem není pro toto použití vhodný.

Rotační enkodér Byl použit modul s enkodérem KY-040 z [5]. Modul obsahoval *pull-up* rezistory u vývodů A a B enkodéru. U vývodu tlačítka *pull-up* rezistor chyběl, a tak bylo nutné aktivovat interní *pull-up* FPGA obvodu.

Maticová klávesnice Byla použita membránová maticová 4×4 klávesnice z [2].

Další moduly Byly použity další moduly ovládané přes I²C sběrnici, a to modul s hodinami reálného času DS3231 z [22], modul se senzorem BMP280 z [20] a modul se senzorem SHT30 z [18].

5.2 Použité nástroje

V průběhu vývoje knihovny a aplikace byl pro simulaci používán nástroj GHDL, který byl díky své rychlosti často používán před spuštěním nástroje Xilinx Vivado. Některé chyby dokázal odchytnout už při analýze zdrojového kódu nebo na začátku simulace návrhu.

Pro zobrazování průběhu signálů ze simulace vytvořené nástrojem GHDL byl používán nástroj GTKWave. Uživatelská přívětivost tohoto nástroje byla horší než u nástroje Xilinx Vivado, ale pro odhalení některých chyb byl tento nástroj postačující.

Balík nástrojů Xilinx Vivado byl v této práci používán nejen pro simulaci, kdy jeho spuštění trvalo déle než u předchozích nástrojů, ale zato nabízel přívětivější prostředí a ovládání, ale zejména pro syntézu a implementaci návrhu, následného generování konfiguračního souboru pro FPGA (souboru *bitstream*) a jeho nahrání na FPGA obvod na přípravku Basys 3, což umožňovalo testování knihovny a aplikace na fyzickém přípravku.

Nástroj Vivado byl využíván ve verzi 2022.2, ale z důvodu zachování kompatibility navržené knihovny a aplikace s verzí nástroje 2018.3 byla občas využita i tato starší verze. Protože starší verze nepodporovala syntézu některých konstruktů, které v novější verzi fungovaly správně, bylo nutné návrh knihovny a aplikace místy upravit.

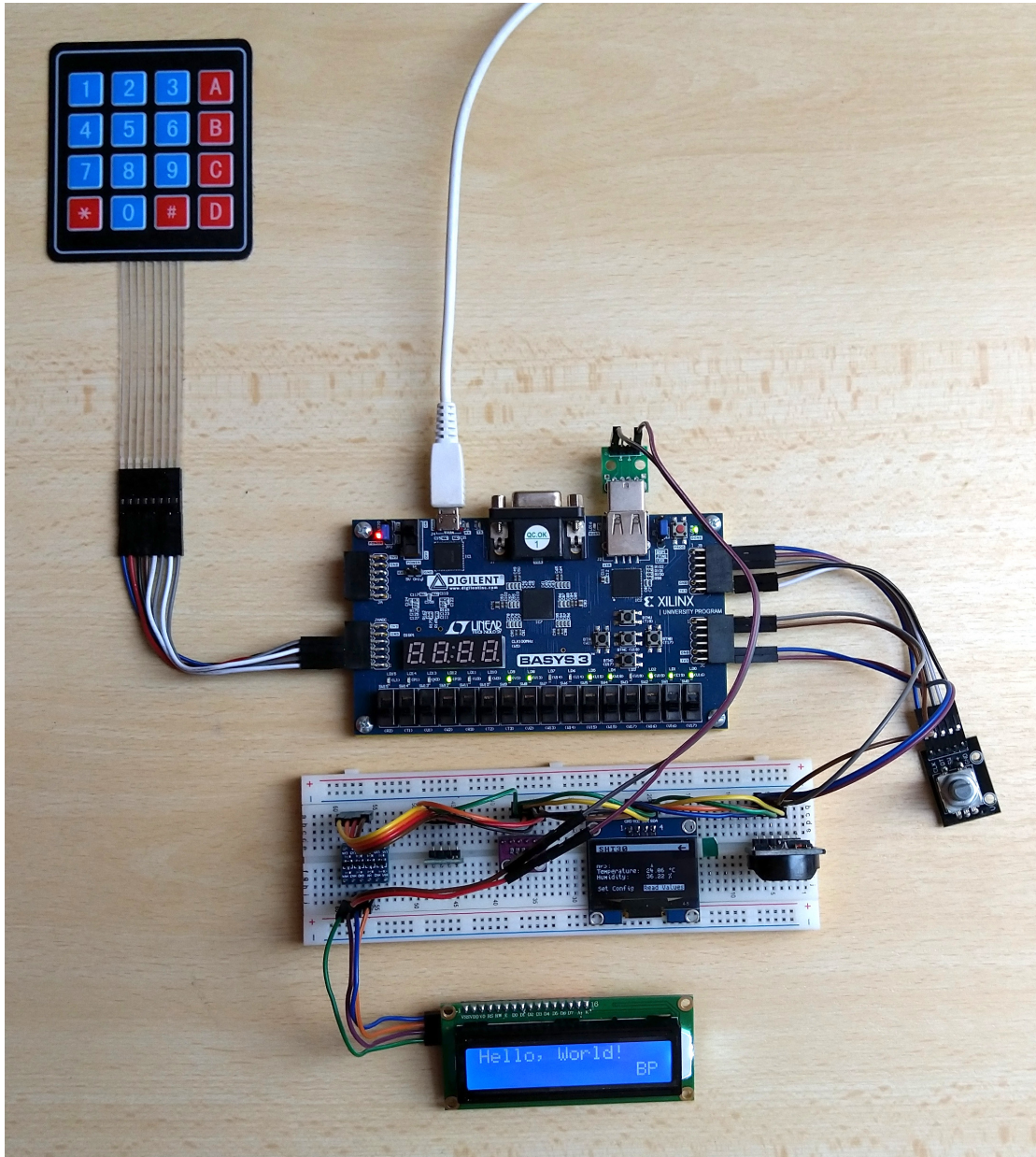
5.3 Postup testování

Pro některé moduly knihovny, zejm. výpočetní, byly vytvořeny testy, které při simulaci kontrolovaly jejich správnou funkčnost. Ostatní moduly byly při vývoji také průběžně simulovány, avšak nebyla automaticky kontrolována jejich správnost. Často se jednalo o jednoduchou simulaci, kde byl generován pouze hodinový signál a signál reset, dokázala však odchytnout některé základní chyby.

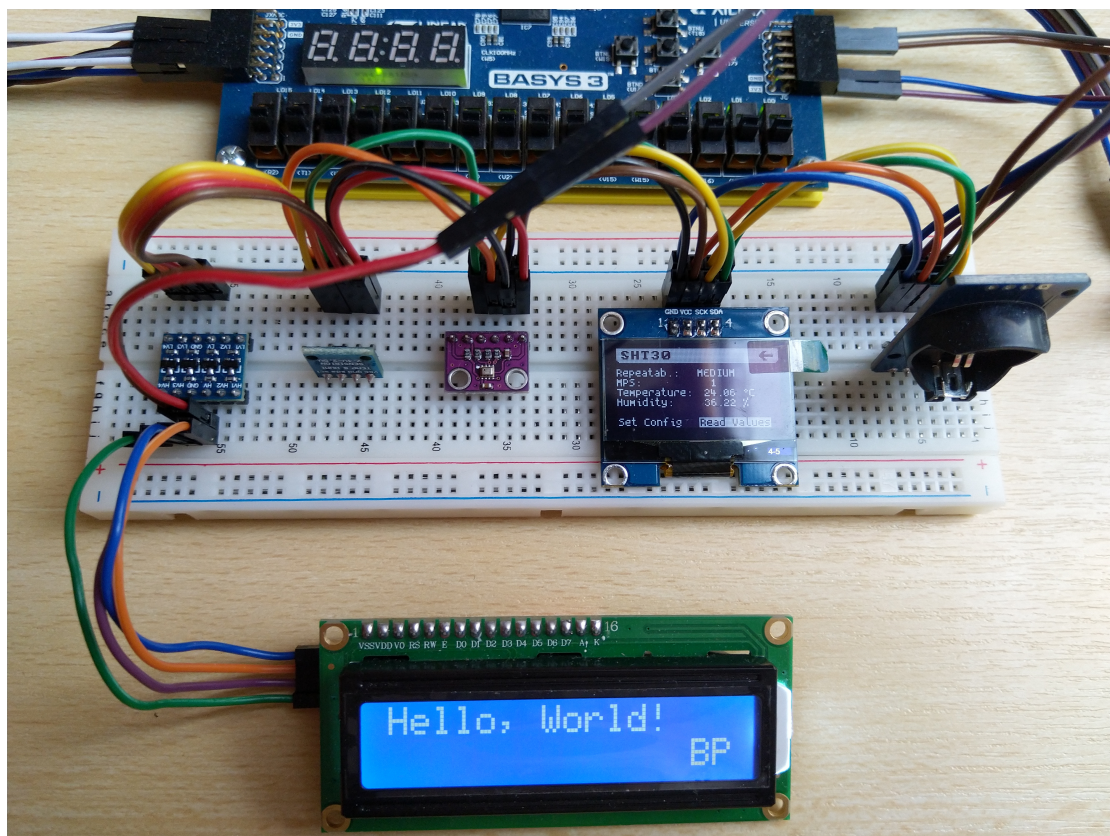
Pro větší bloky knihovny, např. moduly pro ovládání jednotlivých periférií, byly při vývoji vytvořeny i jednoduché aplikace pro jejich testování na vývojové desce Basys 3 s využitím tlačítek, přepínačů, LED diod a 7segmentového displeje obsažených na této desce. Testované periferie byly připojeny k pinům Pmod portů vývojové desky.

Také hlavní aplikace byla testována na přípravku Basys 3. Všechny periferie komunikující přes sběrnici I²C byly připojeny k pravému dolnímu Pmod portu na společnou sběrnici se společným napájením pomocí 4 vodičů (napájení 3,3 V, zem, datový a hodinový signál I²C sběrnice). LCD displej vyžadující napětí 5 V byl ke sběrnici připojen přes převodník logických úrovní a k napájení byl použit USB port desky Basys 3. Rotační enkodér byl připojen k pravému hornímu Pmod portu pomocí 5 vodičů (napájení pro *pull-up* rezistory, zem, dva signály z enkodéru a signál z tlačítka). Maticová klávesnice byla připojena k levému spodnímu Pmod portu pomocí 8 vodičů (4 řádkových a 4 sloupcových).

Na přípravku pak byla testována funkčnost aplikace a periférií volením položek v menu a nastavováním hodnot či prováděním jiných akcí na obrazovkách navržených k ovládání jednotlivých periférií.



■ **Obrázek 5.1** Vývojová deska Basys 3 s připojenými periferiemi a spuštěnou testovací aplikací. Deska je připojena k počítači, ze kterého byla naprogramována a ze kterého je zároveň napájena. Přimo na desce jsou tlačítka, přepínače a LED diody. Stav LED diod byl v aplikaci nastaven před pořízením fotografie. Vlevo je k desce připojena maticová klávesnice, vpravo pak rotační enkodér a pomocí čtyř vodičů i I²C periferie na nepájivém poli dole. USB port vývojové desky je použit pro napájení alfanumerického displeje



■ **Obrázek 5.2** Detail na I²C periférie připojené k vývojové desce Basys 3 se spuštěnou testovací aplikací. Na nepájivém poli je zleva převodník logických úrovní pro LCD displej pod nepájivým polem, senzor SHT30, senzor BMP280, OLED displej a hodiny reálného času DS3231. V aplikaci je v tomto okamžiku spuštěna aktivita pro ovládání senzoru SHT30. Stav alfanumerického displeje byl v aplikaci nastaven před pořízením fotografie

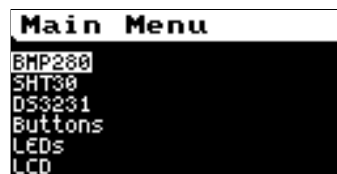
Závěr

Cílem práce bylo navrhnout a implementovat knihovnu pro FPGA firmy Xilinx pro ovládání základních periférií z FPGA obvodu firmy Xilinx se zaměřením na OLED displej, alfanumerický displej, tlačítka, přepínače a LED diody. Funkčnost knihovny a periférií měla být ověřena pomocí testovací aplikace pro FPGA obvod Artix-7 firmy Xilinx, která měla obsahovat menu pro volbu periférií a pro každou periférii měla umožnit její ovládání.

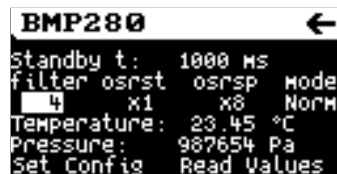
V práci byla navržena knihovna, která byla následně implementována v jazyce VHDL. Tato knihovna umožňuje ovládat některé běžně dostupné malé OLED displeje a alfanumerické LCD displeje, obsahuje pomocné moduly i pro tlačítka a přepínače. Pro LED diody připojené k vstupně-výstupnímu expandéru existuje podpora v podobě řadiče I²C sběrnice, pro LED diody připojené přímo není žádné knihovny třeba. Knihovna navíc podporuje i další periférie, senzor teploty a tlaku BMP280, senzor teploty a vlhkosti SHT30, hodiny reálného času DS3231 a rotační enkodér.

Dále byla navržena a implementována testovací aplikace, která umožňuje ovládat všechny periférie, které jsou podporovány knihovnou. Testovací aplikaci je možné vyzkoušet na vývojové desce Basys 3 firmy Digilent, která obsahuje FPGA obvod Artix-7 firmy Xilinx.

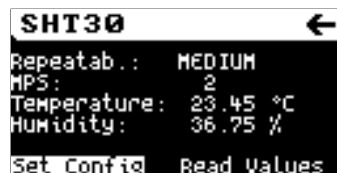
Návrh obrazovek aplikace



■ **Obrázek A.1** Návrh obrazovky menu. Obsahuje možnosti pro přechod na jednotlivé aktivity. Přepínat mezi možnostmi lze pomocí rotačního enkodéru a výběr lze potvrdit stiskem tlačítka na rotačním enkodéru. Aktuálně je zvýrazněna první možnost, BMP280



■ **Obrázek A.2** Návrh obrazovky aktivity pro testování senzoru BMP280. Umožňuje změnit hodnoty konfigurace senzoru a zobrazení naměřených hodnot. Aktuálně je zvýrazněna možnost volby koeficientu filtrování



■ **Obrázek A.3** Návrh obrazovky aktivity pro testování senzoru SHT30. Umožňuje změnit hodnoty konfigurace senzoru a zobrazení naměřených hodnot. Aktuálně je zvýrazněna možnost zapsat konfiguraci do senzoru

```

RTC Time ←
                Alarm 1 >
Time format: 12-hour
Time:         12:34:56 PM
Day:         4
Date (YMD):  1 23/12/31
Read Time   Write Time

```

■ **Obrázek A.4** Návrh první obrazovky aktivity pro testování hodin reálného času DS3231. Tato obrazovka umožňuje zobrazení a nastavení data a času. Aktuálně je zvýrazněna možnost změny hodnoty hodin

```

RTC Alarm 1 ←
< Time           Alarm 2 >
Time format: 12-hour
Time:         08:52:06 AM
Day:         DoM   13
Mask:        Minute
Read Alarm   Write Alarm

```

■ **Obrázek A.5** Návrh druhé obrazovky aktivity pro testování hodin reálného času DS3231. Tato obrazovka umožňuje zobrazení a nastavení času aktivace alarmu 1. Aktuálně je zvýrazněna možnost pro návrat na první obrazovku této aktivity

```

RTC Alarm 2 ←
< Alarm 1 Ctrl/status >
Time format: 24-hour
Time:         13:37
Day:         DoW   6
Mask:        Day
Read Alarm   Write Alarm

```

■ **Obrázek A.6** Návrh třetí obrazovky aktivity pro testování hodin reálného času DS3231. Tato obrazovka umožňuje zobrazení a nastavení času aktivace alarmu 2. Aktuálně je zvýrazněna možnost pro návrat do hlavního menu

```

RTC Ctrl/St ←
< Alarm 2
EOSC=1 BBSQW=0 CONV=0
PS=11 EN32kHz=1 INTCN=1
A1IE=0 A2IE=0 OSF=1
A1F=1  A2F=0  BSY=0
Read C/S   Write C/S

```

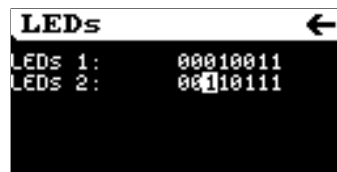
■ **Obrázek A.7** Návrh čtvrté obrazovky aktivity pro testování hodin reálného času DS3231. Tato obrazovka umožňuje načíst nebo nastavit/vynulovat různá nastavení a příznaky. Aktuálně je zvýrazněna možnost pro vynulování příznaku aktivace alarmu 1

```

Buttons, etc. ←
Keypad:      7
Rotary:      21
Buttons:     01001
Switches 1:  10110001
Switches 2:  00100100
Changes:     0003

```

■ **Obrázek A.8** Návrh obrazovky aktivity pro testování klávesnice, tlačítek, přepínačů a rotačního enkodéru. Umožňuje zobrazení právě stisknuté klávesy na maticové klávesnici, stavu jednotlivých tlačítek a přepínačů, hodnoty registru ovládaného rotačním enkodérem a celkový počet změn. Vybrat lze pouze tlačítko pro návrat do hlavního menu



■ **Obrázek A.9** Návrh obrazovky aktivity pro testování světelných diod. Umožňuje zapnout nebo vypnout jednotlivé světelné diody. Aktuálně je zvýrazněna možnost pro přepnutí stavu jedenácté diody



■ **Obrázek A.10** Návrh obrazovky aktivity pro testování alfanumerického displeje. Umožňuje inicializaci a vymazání displeje, nastavení pozice kurzoru a zaslání jednoho znaku nebo řetězce „Hello, World!“. Aktuálně je zvýrazněna možnost pro nastavení pozice kurzoru

Bibliografie

1. *Modul tlačítkové klávesnice 4x4 + 1x4, LED* [online]. GM electronic, spol. s r.o., © 2023. [cit. 2023-04-28]. Dostupné z: <https://www.gme.cz/v/1508390/modul-tlacitkove-klavesnice-4x4-1x4-led>.
2. *Membránová klávesnice 4x4 pro jednodeskové počítače* [online]. ECLIPSE s.r.o., © 2023. [cit. 2023-04-28]. Dostupné z: <https://dratek.cz/arduino/824-eses-membranova-klavesnice-4x4-pro-jednodeskove-pocitace.html>.
3. *4x4 Maticová tlačítková klávesnice Plastová* [online]. LaskaKit s.r.o., © 2023. [cit. 2023-04-28]. Dostupné z: <https://www.laskakit.cz/arduino-4x4--maticova-tlacitkova-klavesnice-plastova/>.
4. *Membránová klávesnice 4x3 pro jednodeskové počítače* [online]. ECLIPSE s.r.o., © 2023. [cit. 2023-04-28]. Dostupné z: <https://dratek.cz/arduino/7530-membranova-klavesnice-4x3-pro-jednodeskove-pocitace.html>.
5. *Rotační enkodér* [online]. ECLIPSE s.r.o., © 2023. [cit. 2023-04-28]. Dostupné z: <https://dratek.cz/arduino/837-rotacni-encoder.html>.
6. DIGILENT, INC. *PmodENC™ Demo Project: Nexys 3 VHDL Example - ISE 14.2* [software]. © 2011. [cit. 2023-05-07]. Dostupné z: https://digilent.com/reference/_media/reference/pmod/pmodenc/pmodenc_ise_demo_14-2.zip.
7. *IIC I2C OLED displej 0,96"- Bílý, 128 x 64* [online]. ECLIPSE s.r.o., © 2023. [cit. 2023-04-28]. Dostupné z: <https://dratek.cz/arduino/1569-iic-i2c-oled-display-0-96-128x64-bily.html>.
8. *IIC I2C OLED displej 1,3"- Bílý, 128 x 64* [online]. ECLIPSE s.r.o., © 2023. [cit. 2023-04-28]. Dostupné z: <https://dratek.cz/arduino/3181-iic-i2c-oled-1-3-displej-128x64-bily.html>.
9. *Pmod OLED* [online]. Digilent. [cit. 2023-04-28]. Dostupné z: <https://digilent.com/reference/pmod/pmodoled/start>.
10. DIGILENT, INC. *PmodOLED Interface Reference Project: Nexys 4 VHDL Example - Vivado 15.1* [software]. © 2011. [cit. 2023-04-28]. Dostupné z: https://digilent.com/reference/_media/reference/pmod/pmodoled/nexys4oleddemo.zip.
11. *LCD displej - Modrý, 16x2 znaků* [online]. ECLIPSE s.r.o., © 2023. [cit. 2023-04-29]. Dostupné z: <https://dratek.cz/arduino/836-display-modry-16x2-znaku.html>.
12. *LCD2004 Displej HD44780 - Modrý, 20 x 4 znaků* [online]. ECLIPSE s.r.o., © 2023. [cit. 2023-04-29]. Dostupné z: <https://dratek.cz/arduino/986-display-modry-20x4-znaku-hd44780.html>.

13. HITACHI, LTD. *HD44780U (LCD-II) (Dot Matrix Liquid Crystal Display Controller/Driver)* [online]. 1999. Rev. 0.0 [cit. 2023-04-04]. Dostupné z: <https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>.
14. DRESCHER, Daniel. *16x2 LCD controller* [software]. OpenCores, 2012. [cit. 2023-04-29]. Dostupné z: https://opencores.org/projects/16x2_lcd_controller.
15. *IIC I2C displej LCD 1602, 16 x 2 LCD znaků - Modrý modul* [online]. ECLIPSERA s.r.o., © 2023. [cit. 2023-04-29]. Dostupné z: <https://dratek.cz/arduino/1570-iic-i2c-display-lcd-1602-16x2-znaku-lcd-modul-modry.html>.
16. *I2C 20x4 displej pro jednodeskové počítače - Žluté podsvícení* [online]. ECLIPSERA s.r.o., © 2023. [cit. 2023-04-29]. Dostupné z: <https://dratek.cz/arduino/1421-eses-i2c-20x4-display-pro-jednodeskove-pocitace.html>.
17. SENSIRION. *Datasheet SHT3x-DIS: Humidity and Temperature Sensor* [online]. 2022. Ver. 7 [cit. 2022-04-15]. Dostupné z: https://sensirion.com/media/documents/213E6A3B/63A5A569/Datasheet_SHT3x_DIS.pdf.
18. *Senzor teploty a vlhkosti vzduchu SHT30* [online]. LaskaKit s.r.o., © 2023. [cit. 2023-05-01]. Dostupné z: <https://www.laskakit.cz/senzor-teploty-a-vlhkosti-vzduchu-sht30/>.
19. BOSCH SENSORTEC GMBH. *BMP280: Digital Pressure Sensor; BMP280 – Data sheet* [online]. 2021. Rev. 1.26 [cit. 2023-04-13]. Dostupné z: <https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmp280-ds001.pdf>.
20. *Senzor barometrického tlaku a teploty BMP280, klon* [online]. LaskaKit s.r.o., © 2023. [cit. 2023-05-01]. Dostupné z: <https://www.laskakit.cz/arduino-senzor-barometrickeho-tlaku-a-teploty-bmp280/>.
21. MAXIM INTEGRATED PRODUCTS, INC. *DS3231: Extremely Accurate I²C-Integrated RTC/TCXO/Crystal* [online]. 2015. Rev. 10 [cit. 2023-04-09]. Dostupné z: <https://www.analog.com/media/en/technical-documentation/data-sheets/ds3231.pdf>.
22. *RTC Hodiny reálného času DS3231 AT24C32* [online]. LaskaKit s.r.o., © 2023. [cit. 2023-05-01]. Dostupné z: <https://www.laskakit.cz/arduino-rtc-hodiny-realneho-casu-ds3231-at24c32/>.
23. XILINX, INC. *Artix-7 FPGAs Data Sheet: DC and AC Switching Characteristics* [online]. 2022. Ver. 1.27 [cit. 2023-04-27]. Dostupné z: https://docs.xilinx.com/v/u/en-US/ds181_Artix_7_Data_Sheet.
24. *Převodník úrovní I2C 5V na 3V* [online]. ECLIPSERA s.r.o., © 2023. [cit. 2023-04-29]. Dostupné z: <https://dratek.cz/arduino/999-prevodnik-urovni-i2c-5v-na-3v.html>.
25. *IIC I2C 5V na 3.3V Obousměrný převodník logické úrovně* [online]. ECLIPSERA s.r.o., © 2023. [cit. 2023-04-29]. Dostupné z: <https://dratek.cz/arduino/1481-iic-i2c-5v-na-3.3v-obousmerny-prevodnik-logicke-urovne.html>.
26. DIGILENT, INC. *Cmod A7 Reference Manual* [online]. 2019. [cit. 2023-04-29]. Dostupné z: https://digilent.com/reference/_media/reference/programmable-logic/cmod-a7/cmod_a7_rm.pdf.
27. DIGILENT, INC. *Basys 3™ FPGA Board Reference Manual* [online]. 2019. [cit. 2023-04-29]. Dostupné z: https://digilent.com/reference/_media/reference/programmable-logic/basys-3/basys3_rm.pdf.
28. NXP B.V. *I²C-bus specification and user manual* [online]. 2017. Rev. 7.0 [cit. 2023-03-23]. Dostupné z: https://web.archive.org/web/20211223011758id_/http://www.nxp.com/docs/en/user-guide/UM10204.pdf.
29. NXP SEMICONDUCTORS N.V. *I²C-bus specification and user manual* [online]. 2014. Rev. 6 [cit. 2023-03-23]. Dostupné z: https://web.archive.org/web/20170701210959id_/http://www.nxp.com/docs/en/user-guide/UM10204.pdf.

30. HERVEILLE, Richard. *I2C controller core* [software]. OpenCores, © 2000, 2001. [cit. 2023-04-29]. Dostupné z: <https://opencores.org/projects/i2c>.
31. LARSON, Scott. *I2C Master (VHDL)* [software]. Digi-Key TechForum, 2015. Ver. 2.2 [cit. 2023-04-29]. Dostupné z: <https://forum.digikey.com/t/i2c-master-vhdl/12797>.
32. SOLOMON SYSTECH LIMITED. *SSD1306: 128 x 64 Dot Matrix OLED/PLED Segment/Common Driver with Controller; Advance Information* [online]. 2008. Rev. 1.3 [cit. 2023-03-22]. Dostupné z: <https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf>. Nachází se na začátku PDF souboru (str. 1–59).
33. SOLOMON SYSTECH LIMITED. *SSD1306: 128 x 64 Dot Matrix OLED/PLED Segment/Common Driver with Controller; Application Note* [online]. 2009. Rev. 0.4 [cit. 2023-03-22]. Dostupné z: <https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf>. Nachází se na konci PDF souboru (str. 60–65).
34. SINO WEALTH ELECTRONIC LTD. *SH1106: 132 X 64 Dot Matrix OLED/PLED Segment/Common Driver with Controller* [online]. 2013. Rev. 2.3 [cit. 2023-03-22]. Dostupné z: <https://www.pololu.com/file/0J1813/SH1106.pdf>.
35. SITRONIX TECHNOLOGY CORPORATION. *ST7066U: Dot Matrix LCD Controller/Driver* [online]. 2015. Ver. 2.5 [cit. 2023-04-04]. Dostupné z: <https://focuslcds.com/content/ST7066Uv25.pdf>.
36. NXP B.V. *PCF8574; PCF8574A: Remote 8-bit I/O expander for I²C-bus with interrupt; Product data sheet* [online]. 2013. Rev. 5 [cit. 2023-04-05]. Dostupné z: https://www.nxp.com/docs/en/data-sheet/PCF8574_PCF8574A.pdf.
37. *Interface 4x3 & 4x4 Membrane Keypad with Arduino* [online]. LastMinuteEngineers.com, © 2023. [cit. 2023-05-08]. Dostupné z: <https://lastminuteengineers.com/arduino-keypad-tutorial/>.
38. DRIBIN, Dave. *Keyboard Matrix Help* [online]. dribin.org, 2000-06-24. Ver. 1.1 [cit. 2023-05-04]. Dostupné z: https://www.dribin.org/dave/keyboard/one_html/.
39. LAZARIDIS, Giorgos. *How a Key Matrix Work* [online]. PCBheaven, 2010-06-27. [cit. 2023-05-04]. Dostupné z: https://pcbheaven.com/wikipages/How_Key_Matrices_Works/.
40. COMPONENTS101. *4x4 Keypad Internal Structure* [online]. Components101, 2018-03-22. [cit. 2023-05-06]. Dostupné z: <https://components101.com/sites/default/files/inline-images/4x4-Keypad-Internal-Structure.png>.
41. ALPS ALPINE CO., LTD. *PDF Catalog: Encoders: EC11* [online]. 2021. [cit. 2023-05-04]. Dostupné z: <https://tech.alpsalpine.com/assets/products/catalog/ec11.en.pdf>.
42. BOURNS, INC. *PEC11L Series - 11 mm Low Profile Encoder* [online]. 2021. [cit. 2023-05-04]. Dostupné z: <https://www.bourns.com/docs/product-datasheets/pec11l.pdf>.
43. TERMINALJACK505. *One Pulse per Detent (4 Phase per Click) Bounce Tolerant Quadrature Encoder Finite State Machine* [online]. EEVblog, 2014-08-30. [cit. 2023-05-07]. Dostupné z: <https://www.eevblog.com/forum/projects/easy-rotary-encoder-implementation-good-easy-to-implement-code/?action=dlattach;attach=107185;image>.
44. ADVANCED MICRO DEVICES, INC. *Vivado Design Suite User Guide: Synthesis* [online]. 2022. Ver. 2022.2 [cit. 2023-05-03]. Dostupné z: <https://docs.xilinx.com/r/en-US/ug901-vivado-synthesis/VHDL-Constructs-Support-Status>. Sekce „VHDL Support“ → „VHDL Constructs Support Status“.
45. PŘISPĚVATELÉ WIKIPEDIE. *Division algorithm* [online]. Wikimedia Foundation, Inc., 2023-03-12. [cit. 2023-05-03]. Dostupné z: [https://en.wikipedia.org/wiki/Division_algorithm#Integer_division_\(unsigned\)_with_remainder](https://en.wikipedia.org/wiki/Division_algorithm#Integer_division_(unsigned)_with_remainder). Sekce „Long division“ → „Integer division (unsigned) with remainder“.

46. CAMBUS, Frederic. *Spleen: Monospaced bitmap fonts* [software]. 2023. [cit. 2023-05-07]. Dostupné z: <https://github.com/fcambus/spleen>.
47. HEPPER, Daniel. *font8x8: 8x8 monochrome bitmap fonts for rendering* [software]. 2019. [cit. 2023-05-07]. Dostupné z: <https://github.com/dhepper/font8x8>.
48. GINGOLD, Tristan et al. *GHDL: About* [online]. GitHub, Inc., 2022-03-01. [cit. 2023-05-01]. Dostupné z: <https://ghdl.github.io/ghdl/about.html>.
49. BYBELL, Anthony. *Welcome to GTKWave* [online]. Slashdot Media, 2023-04-02. [cit. 2023-05-01]. Dostupné z: <https://gtkwave.sourceforge.net/>.
50. *Vivado ML* [online]. Advanced Micro Devices, Inc., © 2023. [cit. 2023-05-01]. Dostupné z: <https://www.xilinx.com/products/design-tools/vivado/vivado-ml.html>.

Obsah přiloženého archivu

	readme.txt.....	stručný popis obsahu archivu
	bit.....	adresář se spustitelnou formou implementace aplikace
	src.....	zdrojové kódy
	impl.....	zdrojové kódy knihovny a aplikace
	thesis.....	zdrojová forma textu práce ve formátu L ^A T _E X
	text.....	text práce
	thesis.pdf.....	text práce ve formátu PDF