



**ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE**

F3

**Fakulta elektrotechnická
Katedra Telekomunikace**

Bakalářská práce

Monitor teploty a vlhkosti vzduchu s výstupem na displej realizovaný pomocí přípravku DE10-Lite a jazyka VHDL

Michal Rieger
Elektronika a Komunikace

Květen 2024
Vedoucí práce: Ing. Pavel Lafata, Ph.D.

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Rieger** Jméno: **Michal** Osobní číslo: **507327**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra telekomunikační techniky**
Studijní program: **Elektronika a komunikace**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Monitor teploty a vlhkosti vzduchu s výstupem na displej realizovaný pomocí přípravku DE10-Lite a jazyka VHDL

Název bakalářské práce anglicky:

Air Temperature and Humidity Monitor with Display Output Using DE10-Lite Kit in VHDL

Pokyny pro vypracování:

Seznamte se s přípravkem DE10-Lite (FPGA MAX10) a jeho obsluhou pomocí jazyka VHDL. K přípravku připojte alespoň 2 dostupné digitální senzory teploty a vlhkosti vzduchu, např. DS18B20, DHT22, DHT11 aj. dostupné. Využijte rovněž ovládací prvky přípravku – přepínače, tlačítka, segmentový displej. Dále navrhnete a realizujete připojení znakového displeje k přípravku pro zobrazování naměřených hodnot, nebo využijte VGA výstup pro připojení externího monitoru. Vytvořte VHDL kódy a knihovny pro ovládání čidel a senzorů a vyčítání naměřených veličin. Využijte segmentový displej jako pomocný způsob zobrazení naměřených hodnot. Hlavní zobrazení realizujte pomocí samostatného znakového displeje nebo externího VGA monitoru. Využijte také přepínače a tlačítka na přípravku pro přepínání zobrazených hodnot a výběr čidla. K přípravku připojte jednoduchý piezo měnič a využijte jej jako jednoduchý alarm pro případ, že dojde k překročení hodnoty měřené veličiny mimo nastavený interval (např. teploty apod.). Výstupem projektu budou VHDL kódy a knihovny pro ovládání různých čidel pomocí přípravku DE10-Lite a zobrazení hodnoty na segmentovém displeji i samostatném znakovém displeji.

Seznam doporučené literatury:

- [1] Lafata, P. - Hampel, P. - Pravda, M.: Digitální technika. 1. vyd. Praha: Česká technika - nakladatelství ČVUT, 2011. 164 s. ISBN 978-80-01-04914-3.
- [2] Pinker, J. - Poupa, M.: Číslicové systémy a jazyk VHDL. Praha: BEN - technická literatura, 2006. 349 s. ISBN 80-7300-198-5.
- [3] Ashender, P., J.: The VHDL Cookbook [online]. Dostupné z: <https://tams-www.informatik.uni-hamburg.de/vhdl/doc/cookbook/VHDL-Cookbook.pdf>.
- [4] Terasic: DE10-Lite User Manual [online]. Dostupné z: <https://www.intel.com/content/dam/www/programmable/us/en/portal/dsn/42/doc-us-dsnbk-42-2912030810549-de10-lite-user-manual.pdf>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Pavel Lafata, Ph.D. katedra telekomunikační techniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **02.02.2024**

Termín odevzdání bakalářské práce: **24.05.2024**

Platnost zadání bakalářské práce: **21.09.2025**

Ing. Pavel Lafata, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování / Prohlášení

Rád bych poděkoval svému vedoucímu práce Ing. Pavlu Lafatovi, PhD. za vedení práce, pomoc při řešení problémů a zapůjčení hardveru potřebného k realizaci práce. Nakonec bych rád poděkoval své rodině a přátelům, kteří mě při studiu podporují.

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.

.....

Abstrakt / Abstract

Tato bakalářská práce se zabývá návrhem a realizací monitoru teploty a vlhkosti v jazyce VHDL s využitím přípravku DE10-Lite. První část práce se věnuje teoretickému rozboru klíčových technologií a komponent, jako jsou jazyk VHDL, FPGA technologie, sběrnice 1-Wire a Aosong 1-Wire, senzory DS18B20 a DHT11 a znakový LCD displej s řadičem HD44780. V praktické části jsou podrobně popsány vytvořené kódy v jazyce VHDL, které byly hlavním cílem této práce.

Výsledkem této práce je plně funkční systém, který monitoruje a zobrazuje aktuální hodnoty teploty a vlhkosti na znakovém LCD displeji a sedmisegmentových displejích přípravku DE10-Lite. Systém také umožňuje pomocí přepínačů a tlačítek na přípravku nastavovat mezní hodnoty teploty a vlhkosti. Pokud jsou tyto mezní hodnoty překročeny, je spuštěn zvukový alarm.

Klíčová slova: FPGA, VHDL, DS18B20, DHT11, LCD displej, 1-Wire, Terasic DE10-Lite

This bachelor's thesis focuses on the design and implementation of a temperature and humidity monitor using the VHDL language and the DE10-Lite board. The first part of the thesis is dedicated to the theoretical analysis of key technologies and components, such as the VHDL language, FPGA technology, 1-Wire and Aosong 1-Wire buses, DS18B20 and DHT11 sensors, and a character LCD display with an HD44780 controller. The practical part provides a detailed description of the VHDL code developed, which was the main goal of this work.

The result of this work is a fully functional system that monitors and displays the current values of temperature and humidity on a character LCD display and the seven-segment displays of the DE10-Lite board. The system also allows the user to set threshold values for temperature and humidity using the switches and buttons on the board. If these threshold values are exceeded, an audible alarm is triggered.

Keywords: FPGA, VHDL, DS18B20, DHT11, LCD display, 1-Wire, Terasic DE10-Lite

Obsah /

1 Úvod	1	A Blokové schéma výsledné práce	27
2 Teoretický rozbor	2	B Seznam příložených kódů v jazyce VHDL	28
2.1 Jazyk VHDL	2		
2.1.1 Entita	2		
2.1.2 Architektura	2		
2.2 Intel Quartus Prime	3		
2.3 FPGA	3		
2.4 Přípravek Terasic DE10-Lite	4		
2.5 Sběrnice 1-Wire	5		
2.5.1 Komunikace na sběrnici 1-Wire	5		
2.5.2 Komunikace na sběr- nici 1-Wire na úrovni signálů	6		
2.6 Senzor DS18B20	6		
2.6.1 Paměť senzoru DS18B20	7		
2.7 Sběrnice Aosong 1-Wire	8		
2.7.1 Komunikace na sběrnici Aosong 1-Wire	8		
2.8 Senzor DHT11	9		
2.9 Senzor deště	9		
2.10 Znakový LCD displej	10		
2.10.1 Řadič HD44780	10		
2.10.2 Paměti řadiče	10		
2.10.3 Komunikace s řadičem HD44780	11		
2.10.4 Příkazy pro komunika- ci s řadičem HD44780	12		
3 Realizace v jazyce VHDL	13		
3.1 Komponenty pro práci s hodinovým signálem	13		
3.1.1 Komponenta PLL	13		
3.1.2 Komponenta Divider	13		
3.2 Komponenta DHT11	14		
3.3 Komponenta DS18B20	15		
3.4 Komponenta LCD	16		
3.4.1 Komponenta LCD con- troller	18		
3.5 Komponenta BinarytoBCD	19		
3.5.1 Double dabble algoritmus	20		
3.6 Komponenta BCDto7segment	20		
3.7 Komponenta main	21		
3.8 Výsledná realizace	22		
4 Závěr	24		
Literatura	25		

Tabulky / Obrázky

2.1 Závislost doby převodu na rozlišení.....7	2.1 Struktura bloků FPGA3
2.2 Piny znakového LCD displeje.. 10	2.2 Vývojová deska DE10-Lite4
2.3 Příkazy pro komunikaci s řadičem HD44780 12	2.3 Časování inicializace na sběrnici 1-Wire.....5
	2.4 Časování zápisu a čtení bitů na sběrnici 1-Wire.....6
	2.5 Paměť SRAM senzoru DS18B20.....7
	2.6 Generátor CRC8
	2.7 Zahájení komunikace na sběrnici Aosong 1-Wire8
	2.8 Časování bitů při komunikaci na sběrnici Aosong 1-Wire9
	2.9 Adresy DDRAM HD44780..... 11
	2.10 Tabulka znaků řadiče HD44780..... 11
	3.1 Komponenta PLL 13
	3.2 Komponenta Divider..... 13
	3.3 Komponenta DHT11..... 14
	3.4 Diagram stavového automatu komponenty DHT11 15
	3.5 Komponenta DS18B20 15
	3.6 Diagram stavového automatu komponenty DS18B20 16
	3.7 Komponenta LCD..... 17
	3.8 Ukázka VHDL kódu pro detekci stisku tlačítka 17
	3.9 Ukázka VHDL kódu pro výběr dat..... 17
	3.10 Komponenta LCD controller .. 18
	3.11 Diagram stavového automatu komponenty LCD 19
	3.12 Komponenta BinaryToBCD ... 19
	3.13 Double dabble algoritmus ve VHDL 20
	3.14 Komponenta BCDto7segment . 20
	3.15 Dekodér BCD na 7segmentový displej ve VHDL 21
	3.16 Komponenta main 21
	3.17 Celkové zapojení práce a zobrazení naměřených hodnot na LCD displeji 23
	3.18 Zobrazení maximálních mezích hodnot na LCD displeji .. 23
	3.19 Zobrazení minimálních mezích hodnot na LCD displeji .. 23

Kapitola 1

Úvod

V posledních letech se technologie FPGA (Field Programmable Gate Array) stává stále důležitější součástí digitálních systémů díky své flexibilitě, vysokému výkonu a možnosti paralelního zpracování dat. Hlavní výhodou FPGA je, že umožňují snadný a rychlý vývoj systémů s konkrétními požadavky. Oproti mikrokontrolérům poskytují větší flexibilitu a ve srovnání s aplikačně specifickými integrovanými obvody mají nižší počáteční náklady a nabízejí možnost rekonfigurace, což vede ke zkrácení doby vývoje systému.

Tato práce je zaměřena na implementaci monitorovacího systému teploty a vlhkosti vzduchu s výstupem na displej, který je realizován pomocí přípravku DE10-Lite a programovacího jazyka VHDL (VHSIC Hardware Description Language, kde zkratka VHSIC představuje Very High Speed Integrated Circuits).

Jedním z cílů této práce je seznámit se s vývojovým přípravkem Terasic DE10-Lite a jeho obsluhou pomocí jazyka VHDL. Hlavním cílem je vytvořit knihovny pro ovládání připojených senzorů DHT11, DS18B20 a senzoru deště. A pro zobrazení hodnot na sedmisegmentových displejích a znakovém LCD displeji.

Kapitola 2

Teoretický rozbor

2.1 Jazyk VHDL

Jazyk VHDL slouží k popisu a simulaci digitálních integrovaných obvodů. Řadíme ho do skupiny popisovacích jazyků, do kterých patří dále například Verilog. Jazyk vznikl v rámci programu VHSIC armády USA na začátku 80. let 20. století. První varianta jazyku byla publikována v roce 1985 a od té doby bylo vydáno mnoho standardů, které nesou označení IEEE (Institute of Electrical and Electronics Engineers) [1–2].

Kód napsaný v jazyce VHDL musí být syntetizovatelný. To znamená, že překladač musí být schopný převést kód na zapojení logických členů a hradel. Dále je vždy při psaní VHDL kódu nutné uvažovat pro jaký obvod je program vytvářen. Ačkoliv by měl být jazyk univerzální, můžou se u programovatelných polí vyskytovat omezení, která jsou dána jejich hardwerovou realizací [1].

Při vytváření kódu v jazyce VHDL je možné rozdělit program do menších bloků, které se nazývají komponenty. Výsledný obvod pak můžeme složit z komponent, které navzájem propojíme signály. Funkci signálu lze přirovnat k funkci vodiče v číslicovém systému. Základní strukturu komponenty tvoří entita a architektura.

2.1.1 Entita

Entita je v jazyce VHDL označení pro výsledný navrhovaný obvod. Každá entita má název, deklaraci portů (vstupů a výstupů obvodu) a popis funkce entity. Základními typy portů jsou:

- IN – vstupní port
- OUT – výstupní port
- INOUT – vstupně/výstupní port
- BUFFER – výstupní port s možností čtení

2.1.2 Architektura

Druhou základní částí entity po deklaraci jejích portů je tzv. architektura. Ta obsahuje vlastní popis entity. Architektura může být popsána několika způsoby v závislosti na úrovni abstrakce, kterou chceme použít. Tyto úrovně abstrakce mohou být následující:

- Behaviorální popis – Jedná se o nejvyšší úroveň abstrakce. Obvod je popsán z hlediska funkce a jeho chování. Výhodou je rychlost zápisu kódu, jednoduchost a flexibilita. Nevýhodou je, že nad procesem syntézy a implementace máme pouze malou kontrolu. Nedokážeme tak ovlivnit složení a zapojení samotného obvodu.
- Dataflow (RTL-Register Transfer Logic) popis – RTL popis je o úroveň nižší stupeň abstrakce než Behaviorální popis. Výstupní funkce obvodu lze popsat pomocí Booleových rovnic. Výhoda je větší kontrola nad syntézou a implementací obvodu než při behaviorálním popisu. Nevýhodou však zůstává omezená kontrola nad detailní syntézou a implementací, protože známe pouze složení obvodu na úrovni bloků a logických hradel, nikoliv přesné zapojení.

- Strukturální popis – Jedná se o nejnižší úroveň abstrakce. Při popisu známe přesné zapojení obvodu a jeho složení z jednotlivých logických hradel. Výhodou je naprostá kontrola nad syntézou a implementací obvodu. Nevýhodou je nejvyšší náročnost na vytvoření kódu a také nejmenší flexibilita, kdy i při malé změně může být potřeba změnit podstatnou část kódu.

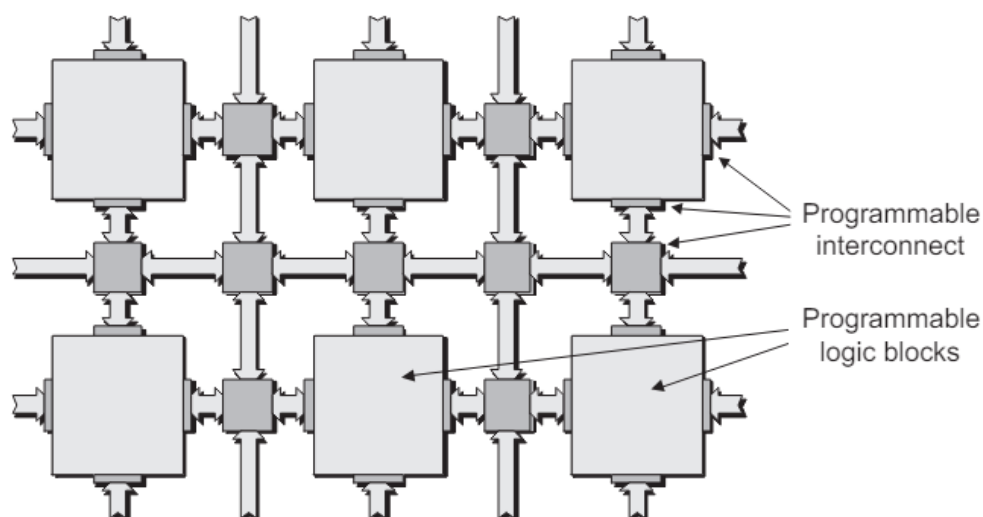
2.2 Intel Quartus Prime

Intel Quartus Prime je software pro navrhování programovatelných logických zařízení od společnosti Intel. Tento software obsahuje vše potřebné pro analýzu a syntézu HDL návrhů, což zahrnuje zadání návrhu, syntézu až po optimalizaci, ověření a simulaci [3]. Quartus Prime obsahuje implementaci VHDL a Verilog pro popis hardwaru.

Program je dostupný ve více verzích. V této práci je použita verze Intel Quartus Prime Lite, verze je bezplatná a vhodná pro studijní účely nebo pro menší vývojáře. Ve verzi Lite je možné kompilovat a programovat pouze omezené množství zařízení. V práci je použito zařízení z rodiny Intel MAX 10, která je ve verzi Lite podporována.

2.3 FPGA

FPGA neboli programovatelné hradlové pole je digitální obvod, který se skládá z logických bloků. Tyto bloky jsou nejčastěji uspořádány maticově a navzájem propojeny vodiči. Nejčastěji se logický blok skládá z vyhledávacích tabulek LUT (Look up table), klopných obvodů typu D, multiplexorů a sběrnice mezi nimi, tyto bloky můžeme označit jako univerzální. Dále FPGA běžně obsahují bloky I/O (vstupně/výstupní), bloky fázových závěsů PLL (Phased locked loop) a pro správu hodinového signálu, bloky paměti RAM (Random Access Memory) a násobičky. Propojení mezi jednotlivými logickými bloky závisí na programovatelných maticových přepínačích. Konstrukce FPGA je zobrazena na obrázku 2.1. Hradlová pole lze programovat pomocí popisovacích jazyků (VHDL, Verilog). [1–2]



Obrázek 2.1. Struktura bloků FPGA [2]

Hradlová pole zaujímají střední cestu mezi PLD (Programmable Logic Device) a ASIC (Application Specific Integrated Circuit), protože jejich funkčnost může být

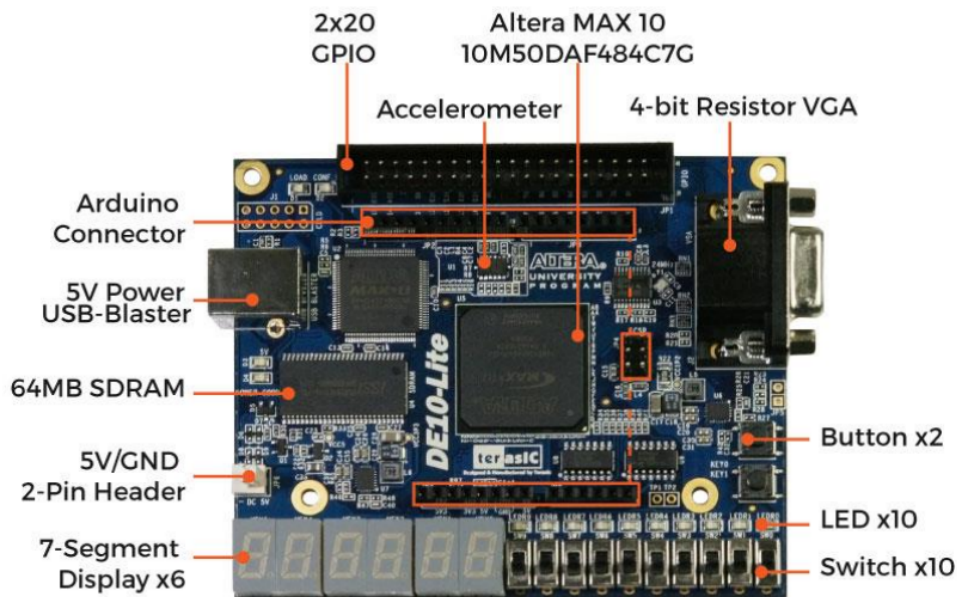
přizpůsobena v terénu podobně jako u PLD, ale mohou obsahovat miliony logických hradel a být použity k implementaci extrémně velkých a složitých funkcí, které dříve byly realizovatelné pouze pomocí ASIC [2].

Často se používají k návrhu prototypů ASIC nebo jako hardwarová platforma, na které se ověřuje fyzická implementace nových algoritmů. Dále může být výhodou paralelní zpracovávání dat. Oproti běžným procesorům, které zpracovávají sekvenčně jednotlivé instrukce, se v FPGA vytvoří reálná hardwarová spojení a můžou zpracovávat data co nejeefektivněji.

2.4 Přípravek Terasic DE10-Lite

Je to vývojová deska od firmy Terasic. Představuje robustní hardwarovou platformu postavenou kolem FPGA Altera MAX 10M50DAF484C7G. FPGA obsahuje 50 000 programovatelných logických prvků, 1 638 Kbit paměti M9k, 5 888 Kbit uživatelské flash paměti, 4 fázové závěsy, 2 integrované ADC a 144 násobiček. [4]

Kromě FPGA je deska vybavena 64 MB SDRAM (Synchronous Dynamic Random Access Memory), 40 GPIO (General Purpose Input/Output) konektory, rozšiřujícím konektorem Arduino UNO R3 s 6 analogovými vstupy a akcelerometrem. K desce je možné připojit externí displej přes VGA (Video Graphics Array) konektor. Dále ve spodní části desky nalezneme 10 LED, 10 přepínačů, 2 tlačítka a 6 7segmentových displejů. Pro naprogramování desky je zde zabudovaný programátor USB blaster. Na obrázku 2.2 lze vidět desku DE10-Lite s popisem jednotlivých částí.



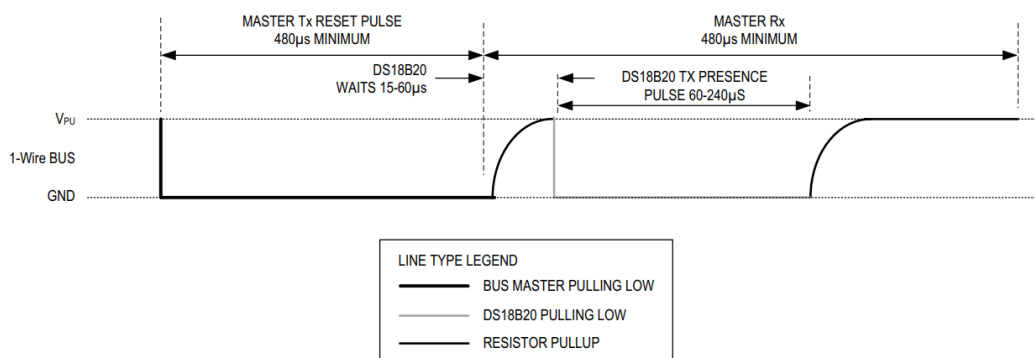
Obrázek 2.2. Vývojová deska DE10-Lite [4]

2.5 Sběrnice 1-Wire

1-Wire je sériová obousměrná half-duplex sběrnice, která byla navržena firmou Dallas Semiconductor. Její zapojení je možné provést pouze s jedním datovým vodičem a jednou společnou zemí. Jedná se o sběrnici s otevřeným kolektorem, což znamená že je připojena přes rezistor na napájecí napětí a její klidový stav je logická 1. Řídící obvod, který se nazývá master, iniciuje a řídí komunikaci s jedním nebo více podřízenými zařízeními. Tato zařízení se nazývají slave. Každé slave zařízení má unikátní 64bitovou adresu, která umožňuje zapojení více zařízení slave na stejné 1-Wire sběrnici. Nejmenších 8 bitů obsahuje adresu rodiny zařízení, následujících 48 bitů obsahuje unikátní sériovou adresu a 8 nejvýznamějších bitů obsahuje CRC (Cyclic Redundancy Check) [5–6].

2.5.1 Komunikace na sběrnici 1-Wire

Každá komunikace je zahájena resetovacím pulzem, který stáhne sběrnici do logické 0 po dobu nejméně 480 μs , následně sběrnici uvolní a odpor vrátí sběrnici do logické 1. Pokud je na sběrnici zařízení slave, reaguje na vzestupnou hranu přítomnostním pulzem a po prodlevě 15–60 μs stáhne sběrnici do logické 0 na dobu 60–240 μs . Průběh signálu je vidět na obr. 2.3.



Obrázek 2.3. Časování inicializace na sběrnici 1-Wire [6]

Poté co master detekuje přítomnostní pulz, může poslat zařízení slave ROM (Read-Only Memory) příkaz. ROM příkazy bývají podobné pro všechna zařízení, která komunikují po 1-Wire sběrnici.

Když Master úspěšně odešle ROM příkaz zařízení slave, se kterým chce komunikovat, může následně poslat funkční příkaz. Funkční příkazy se mohou pro každé zařízení lišit. V této práci jsou dále uvedeny příkazy pro senzor DS18B20. Všechny příkazy pro komunikaci se senzorem DS18B20 jsou uvedeny zde [6].

- Read ROM [33h] – Master přečte 64bitovou adresu slave zařízení. Tento příkaz může být použit pouze, když je jedno zařízení na sběrnici, jinak dojde ke kolizi dat.
- Match ROM [55h] – Po odeslání tohoto příkazu, jenž je následován 64bitovou adresou, odpoví pouze zařízení slave, které má shodnou 64bitovou adresu.
- Skip ROM [CCh] – Tímto příkazem může master adresovat všechna zařízení na sběrnici, aniž by musel posílat adresu zařízení. Také se dá tento příkaz použít, pokud je na sběrnici pouze jedno slave zařízení, aniž bychom museli pracovat s jeho adresou.
- Convert T [44h] – Po tomto příkazu se provede převod teploty. Když je převod dokončen, dojde k uložení teplotních dat do 2bytového registru v paměti scratchpad. Pokud je senzor napájen externím zdrojem, může master po příkazu číst sběrnici,

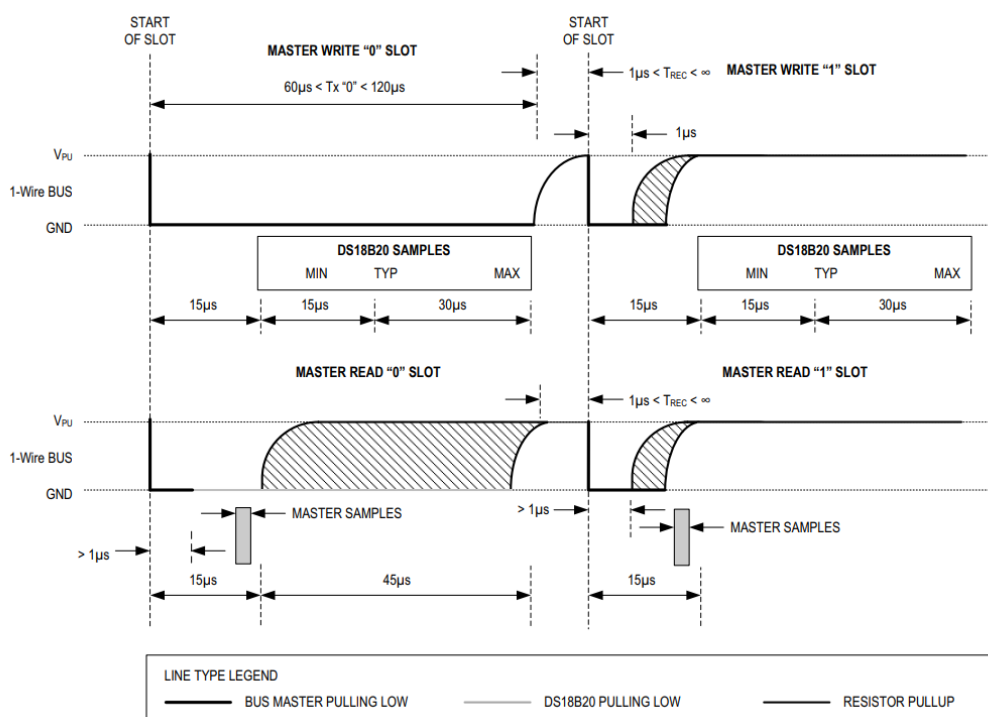
když je převod u konce na sběrnici bude logická 1. Pokud je senzor napájen parazitně ze sběrnice, musí master po dobu konverze nastavit sběrnici do logické 1. Minimální doba převodu závisí na nastaveném rozlišení senzoru, závislost je uvedena v tabulce 2.1.

- Read Scratchpad [BEh] – Master přečte data z paměti scratchpad. Data budou poslána od nejnižšího bitu po nejvyšší bit. Master může přenos dat kdykoliv ukončit, pokud je potřeba pouze část dat ze scratchpad.

2.5.2 Komunikace na sběrnici 1-Wire na úrovni signálů

Master zapisuje příkazy nebo čte data ze slave po jednotlivých bitech. Bity čte nebo zapisuje během tzv. časových slotů. Každý časový slot musí trvat minimálně $60 \mu\text{s}$ a maximálně $120 \mu\text{s}$. Mezi dvěma časovými sloty musí být minimální doba zotavení $1 \mu\text{s}$. Rozlišujeme 4 druhy časových slotů a jejich průběhy jsou znázorněny na obrázku 2.4.

- Zápis 1 – Master nastaví sběrnici do logické 0 a nejdříve po $15 \mu\text{s}$ ji uvolní.
- Zápis 0 – Master nastaví na sběrnici logickou 0 po dobu nejméně $60 \mu\text{s}$.
- Čtení 1 – Master nastaví na sběrnici logickou 0 po dobu kratší než $15 \mu\text{s}$, potom slave uvolní sběrnici a ta se vrátí do klidového stavu, čímž odešle logickou 1.
- Čtení 0 – Master nastaví na sběrnici logickou 0 po dobu kratší než $15 \mu\text{s}$, potom slave podrží sběrnici v logické 0, čímž pošle logickou 0.



Obrázek 2.4. Časování zápisu a čtení bitů na sběrnici 1-Wire [6]

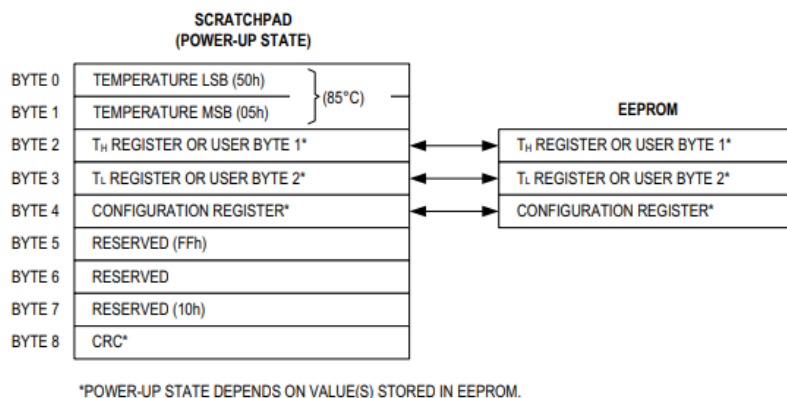
2.6 Senzor DS18B20

DS18B20 je digitální senzor pro měření teploty v rozsahu $-55 \text{ }^\circ\text{C}$ až $+125 \text{ }^\circ\text{C}$. Senzor dosahuje přesnosti $0,5 \text{ }^\circ\text{C}$ v rozsahu teplot $-10 \text{ }^\circ\text{C}$ až $+85 \text{ }^\circ\text{C}$. Rozlišení senzoru je programovatelné od 9 do 12 bitů. Senzor komunikuje přes sběrnici 1-Wire. Obsahuje

64bitovou paměť ROM, v které je uložena unikátní adresa, díky níž je možné adresovat senzor na sběrnici. Napájení senzoru je možné z externího zdroje s hodnotou napětí 3 V až 5,5 V, případně přímo z datové sběrnice tzv. parazitní napájení, čímž eliminujeme potřebu externího zdroje [6]. Senzor má tři piny, pin VDD sloužící pro připojení napájení. Pin DQ slouží pro komunikaci po sběrnici a pin GND pro připojení k elektrické zemi.

2.6.1 Paměť senzoru DS18B20

Senzor obsahuje 9bajtovou paměť typu SRAM (Static Random Access Memory), jejíž označení je scratchpad, a energeticky nezávislou paměť EEPROM (Electrically Erasable Programmable Read-Only Memory), jejichž rozdělení je ukázáno na obrázku 2.5. Bajty 0 a 1 paměti scratchpad obsahují informace o teplotě. Nejvyšších 5 bitů v bajtu 1 určuje znaménko, pokud jsou rovny logické 0, teplota je kladná a pokud jsou rovny logické 1, tak je teplota záporná. Tři nejnižší bity bajtu 1 a 4 nejvyšší bity bajtu 0 jsou určeny pro celou hodnotu teploty. Čtyři nejnižší bity bajtu 0 jsou určeny pro desetinnou část teploty. Další jsou bajty 2 a 3, které mohou sloužit k nastavení krajních hodnot teploty, jejichž překročení je možné detekovat pomocí příkazů. Bajt 4 je konfigurační registr pro nastavení rozlišení měřené teploty. Bity 7 a 4–0 bajtu 4 jsou pevně nastaveny a nedají se měnit, ale volbou bitů 6 a 5 můžeme měnit rozlišení senzoru. S větším rozlišením narůstá potřebný čas pro převod teploty. Dobu převodu a rozlišení při všech kombinacích bitů 5–6 můžeme vidět v tabulce 2.1, kde bit 5 je označen R0 a bit 6 jako R1.



Obrázek 2.5. Rozdělení paměti SRAM [6]

Tabulka 2.1. Konfigurace bitů rozlišení teploty [6]

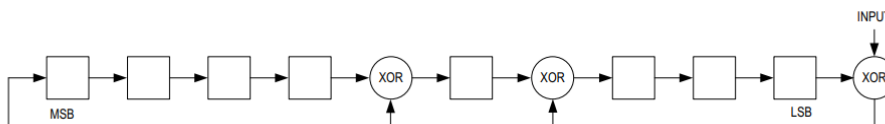
R1	R0	rozlišení (v bitech)	maximální doba převodu (ms)
0	0	9	93,75
0	1	10	187,5
1	0	11	375
1	1	12	750

Bajty 5 až 7 nemohou být přepsány a jsou určeny pro interní potřeby senzoru. Osmý bajt obsahuje CRC (Cyclic Redundancy Check) pro bajty 0 až 7. CRC slouží pro ověření správnosti dat, která získal master při čtení. Pokud se CRC vypočítaný z přečtených dat shoduje s CRC přečteným se senzoru, tak se data přenesla v pořádku.

Jestliže se neshodují, čtení musí proběhnout znovu. Ekvivalentní polynomiální funkce pro CRC je dána vztahem

$$CRC = X^8 + X^5 + X^4 + 1. \quad (1)$$

Master může k výpočtu CRC z přečtených dat použít polynomiální generátor. Generátor se skládá z posuvného registru a hradel XOR, jeho blokové schéma je zobrazeno na obrázku 2.6.



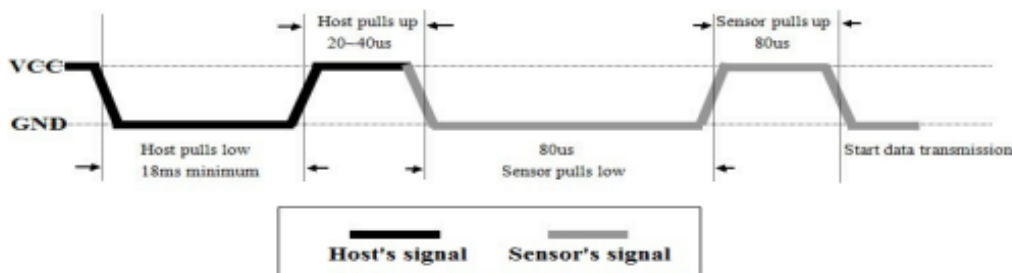
Obrázek 2.6. Generátor CRC [6]

2.7 Sběrnice Aosong 1-Wire

Jedná se o sériovou obousměrnou half-duplex sběrnici. Sběrnice byla navržena firmou Aosong Electronics Co., Ltd., liší se od 1-Wire sběrnice od společnosti Dallas Semiconductor. Rozdíly jsou v časování pulzů při komunikaci a zařízení nemají určenou unikátní adresu, tak že může být na sběrnici připojeno pouze jedno zařízení. Taktéž se jedná o sběrnici s otevřeným kolektorem, a proto je klidový stav sběrnice logická 1.

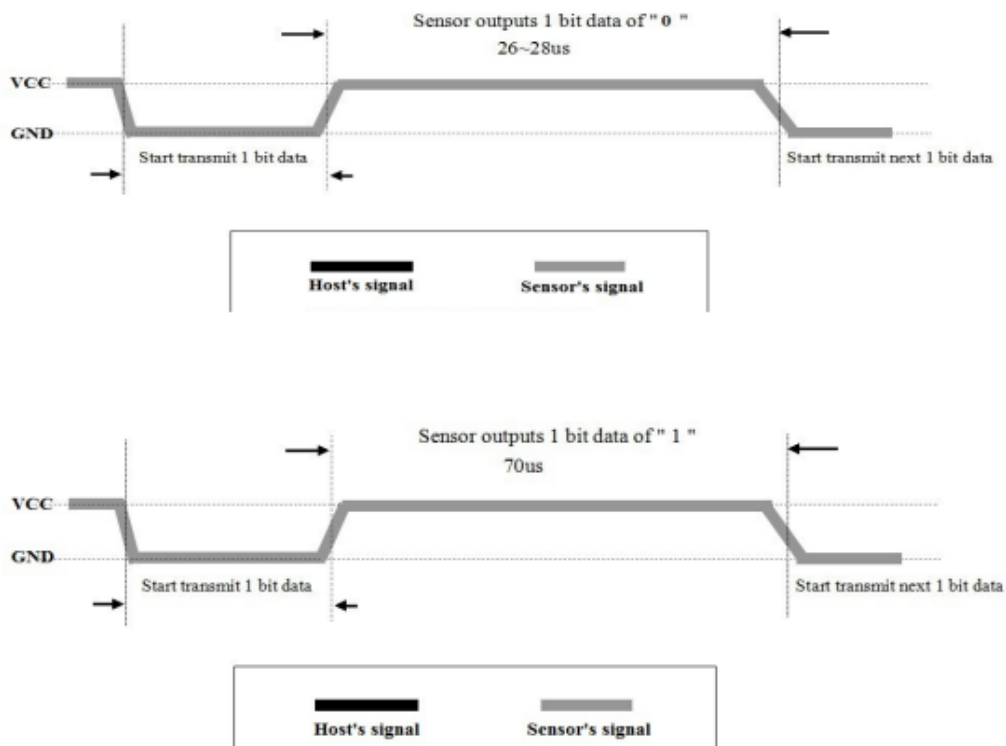
2.7.1 Komunikace na sběrnici Aosong 1-Wire

Komunikace je zahájena, když master stáhne sběrnici do logické 0 po dobu nejméně 18 ms. Poté master uvolní sběrnici na 20–40 μs a čeká na odpověď slave zařízení. Pokud slave detekuje signál, stáhne sběrnici do logické 0 po dobu 80 μs jako odpověď. Poté slave uvolní sběrnici na 80 μs před přenosem dat. Průběh začátku komunikace je na obrázku 2.7.



Obrázek 2.7. Zahájení komunikace na sběrnici Aosong 1-Wire [7]

Po inicializaci je zahájen přenos bitů. Každý přenos bitu začíná stáhnutím sběrnice na logickou 0 po dobu 50 μs . Následující doba, kdy je sběrnice v logické 1, určuje zda je hodnota bitu logická 0 nebo 1. Doba odpovídající logické 0 je 26–28 μs a doba odpovídající logické 1 je 70 μs . Průběh signálů je ukázán na obrázku 2.8.



Obrázek 2.8. Časování bitů při komunikaci na sběrnici Aosong 1-Wire [7]

2.8 Senzor DHT11

DHT11 je digitální senzor pro měření teploty v rozsahu 0 až 50 °C s přesností 2 °C a relativní vlhkosti v rozsahu 20 až 90 % s přesností 5 %. Napájecí napětí by mělo být 3,3 až 5,5 V. Komunikace se senzorem probíhá přes sběrnici Aosong 1-Wire [7]. Senzor má čtyři piny, pin VDD je připojen k napájení. Pin DATA slouží ke komunikaci po sběrnici. Pin GND se připojuje k zemi. Poslední pin NC nemá žádnou funkci a nezapojuje se.

Při komunikaci po sběrnici pošle senzor 40 bitů, které reprezentují hodnoty relativní vlhkosti a teploty. Prvních 8 bitů reprezentuje celočíselnou hodnotu relativní vlhkosti. Následujících 8 bitů je určeno pro desetinnou část relativní vlhkosti, ale měly by být vždy nulové. Další 8 bitů reprezentuje celočíselnou hodnotu teploty. Následujících 8 bitů je určeno pro desetinnou část teploty, ale měly by být vždy nulové. Posledních 8 bitů je kontrolní součet předchozích 32 bitů.

2.9 Senzor deště

Tento senzor se používá k detekci srážek a lze jím detekovat déšť i sníh. Skládá se ze samotného senzoru a modulu. Modul se dále skládá z komparátoru LM393 a potenciometru [8]. Pomocí potenciometru je možné nastavit citlivost senzoru. Senzor je založen na hřebenové konstrukci, na povrchu jsou 2 vodivé cesty oddělené malou mezerou. Kapka při dopadu na senzor vodivě propojí oba hřebeny, a tím sníží odpor. Snížení odporu je poté vyhodnoceno komparátorem na modulu. Díky této konstrukci je možné detekovat i malou kapku.

Modul má 4 piny, kde 2 jsou pro připojení napájecího napětí a elektrické země. Zbývající 2 piny jsou digitální výstup D0 a analogový výstup A0. Na digitálním výstupu

je ve výchozím stavu logická 1, pokud sensor detekuje srážky, na výstupu bude logická 0. Na analogovém výstupu se mění napětí od výchozích 5 V až po 0 V.

2.10 Znakový LCD displej

Jedná se o znakový LCD (liquid crystal display) displej, který zobrazuje 4 řádky po 20 znacích. Komunikace s displejem probíhá paralelně, což znamená, že je posíláno více bitů současně. Displej je ovládán pomocí řadiče HD44780 a celkově má 16 pinů. Názvy všech pinů a jejich funkce jsou uvedeny v tabulce 2.2. Na pin VO je dobré připojit potenciometr, aby bylo možné nastavovat kontrast displeje.

Tabulka 2.2. Piny LCD displeje a jejich funkce.

Pin	Funkce
VSS	Elektrická zem
VDD	Napájecí napětí +5 V
VO	Změna kontrastu
RS	Výběr mezi registrem pro data nebo instrukce
R/W	Změna mezi módem pro zápis a čtení
E	Povolení čtení/zápisu dat
D0-D7	Digitální piny pro čtení a zápis dat
A	Anoda podsvícení displeje
K	Katoda podsvícení displeje

2.10.1 Řadič HD44780

Řadič HD44780 byl vyvinut firmou Hitachi. Především díky jeho nízké ceně, spolehlivosti a kompatibilitě s mnoha alfanumerickými displeji se stal jedním z nejpopulárnějších řadičů alfanumerických displejů. Řadič může být ovládán 4bitovým nebo 8bitovým mikroprocesorem.

2.10.2 Paměti řadiče

- DDRAM (Display Data RAM) – V paměti jsou uloženy znaky, které se následně zobrazí na displeji. Do paměti je možné uložit až 80 znaků, kde každý má velikost 8 bitů. Adresy v DDRAM jsou rozděleny na 2 řádky. Pokud není všechna DDRAM využita, zbylá paměť se využije jako běžná data RAM. Na obrázku 2.9 můžeme vidět vztah mezi pozicemi na displeji a adresami z DDRAM při zobrazení 4 řádků po 20 znacích.
- CGROM (Character Generator ROM) – Paměť uchovává předdefinované znaky. Může generovat až 208 znaků o velikosti 5x8 nebo 32 znaků o velikosti 5x10 znaků. Ukázka znaků je na obrázku 2.10.
- CGRAM (Character Generator RAM) – V této paměti si může uživatel nadefinovat osm vlastních znaků o velikosti 5x8 nebo čtyři znaky o velikosti 5x10.

Pozice na Displeji

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Adresy DDRAM	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13
	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53
	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27
	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	66	67

Obrázek 2.9. Adresy DDRAM HD44780 [9]

Lower 4 Bits	Upper 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
xxxx0000	CG RAM (1)			0	a	P	`	P					-	9	3	a	p	
xxxx0001	(2)			!	1	A	Q	a	q				.	7	7	4	ä	q
xxxx0010	(3)			"	2	B	R	b	r				「	イ	ウ	×	β	θ
xxxx0011	(4)			#	3	C	S	c	s				」	ウ	7	E	ε	ω
xxxx0100	(5)			\$	4	D	T	d	t				√	イ	ト	ト	μ	Ω
xxxx0101	(6)			%	5	E	U	e	u				.	オ	ナ	1	ε	Ü
xxxx0110	(7)			&	6	F	V	f	v				ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)			'	7	G	W	g	w				ア	キ	ヌ	ヲ	g	π
xxxx1000	(1)			<	8	H	X	h	x				ィ	ウ	ネ	リ	」	×
xxxx1001	(2)			>	9	I	Y	i	y				ゝ	ク	ル	ル	」	y
xxxx1010	(3)			*	:	J	Z	j	z				エ	コ	ン	ル	j	〒
xxxx1011	(4)			+	:	K	L	k	l				オ	サ	ヒ	ロ	*	〒
xxxx1100	(5)			,	<	L	¥	l	l				オ	シ	フ	フ	φ	〒
xxxx1101	(6)			-	=	M	J	m	}				ユ	ズ	ハ	ン	ε	÷
xxxx1110	(7)			.	>	N	^	n	+				ヨ	セ	ホ	」	ñ	
xxxx1111	(8)			/	?	O	_	o	+				ウ	リ	マ	」	ö	■

Obrázek 2.10. Tabulka znaků řadiče HD44780 [9]

2.10.3 Komunikace s řadičem HD44780

Komunikace s řadičem probíhá pomocí instrukcí a je možná v 8bitovém režimu nebo ve 4bitovém režimu. V 8bitovém režimu, který je využit v této práci, je zapojeno všech 8 datových vodičů DB0–DB7. Data jsou odeslána po přivedení pulzu na pin E. Při 4bitovém režimu jsou připojeny pouze datové vodiče DB4–DB7. Instrukce je potom rozdělena na 2 části, kdy první se odešlou 4 nejvýznamější bity instrukce a potom 4

nejnižší bity instrukce. Obě části musí být odeslány samostaným pulzem přivedeným na E.[9]

Inicializaci provede automaticky vnitřní resetovací obvod, pokud jsou splněny napájecí podmínky pro resetovací obvod. Když nejsou splněny podmínky, musí být inicializace provedena pomocí instrukcí [9]. Po připojení k napájení je nutné počkat nejméně 15 ms, následně je poslána instrukce Function set, kde na hodnotě 4 nejméně významných bitů instrukce nezáleží. Dále je nutné počkat nejméně 4,1 ms a poslat znovu instrukci Function set. Naposledy je nutné počkat nejméně 100 μ s a poslat instrukci Function set.

Poté následuje konfigurace, která je stejná ať už byla inicializace provedena přes instrukce nebo byla provedena automaticky resetovacím obvodem. První se pošle příkaz Function set, čímž se nastaví 8bitová nebo 4bitová komunikace, počet řádek a font znaků. Dále příkaz Display on, kterým je displej zapnut. Display clear příkaz nastaví všechny znaky na mezeru a posledním příkazem Entry mode set je nastavena inkrementace a posun displeje.

Po inicializaci je možné začít zapisovat data do DDRAM. Data jsou odesílána pomocí instrukce Write data to DDRAM a jsou následně zobrazena na displeji.

2.10.4 Příkazy pro komunikaci s řadičem HD44780

V této podkapitole jsou uvedeny pouze příkazy potřebné pro tuto práci všechny příkazy jsou uvedeny zde [9]. V tabulce 2.3 jsou uvedeny bity, které je nutné nastavit pro poslání příkazu řadiči a doba provedení příkazu.

Tabulka 2.3. Příkazy pro komunikaci s řadičem HD44780 [9]

Název	RS	R/W	DB7-DB0	Doba provedení
Clear Display	0	0	0 0 0 0 0 0 0 1	1,52 ms
Entry Mode Set	0	0	0 0 0 0 0 1 I/D S	37 μ s
Display on/off control	0	0	0 0 0 0 1 D C B	37 μ s
Function set	0	0	0 0 1 DL N F - -	37 μ s
Set DDRAM address	0	0	1 A A A A A A A	37 μ s
Write data to DDRAM	1	0	Write data	37 μ s

- Clear Display – Vymaže celý displej a poté nastaví DDRAM adresu 0 do ukazatele na paměť.
- Entry Mode Set – I/D: Zvyšuje (I/D = 1) nebo snižuje (I/D = 0) hodnotu adresy DDRAM, když proběhne zápis nebo čtení znaku z DDRAM. S: Posune celý displej doprava (I/D = 0) nebo doleva (I/D = 1), když je S nastaven na 1. Pokud je S nastaven do 0, displej není posunut.
- Display On/Off Control – Displej se zapne pokud je D nastaveno do 1 a vypne pokud je D nastaveno do 0. Kurzor je zapnut pokud je C nastaveno do 1 a vypnut pokud je nastaveno do 0. Pokud je B nastaveno do 1, tak číslice, pod kterou je kurzor, začne blikat.
- Function Set – DL: Nastavení jestli chceme komunikovat pomocí 8bitového rozhraní (DL = 1) nebo 4bitového rozhraní (DL = 0). N: Nastavení počtu zobrazovaných řádků. F: Nastavení fontu znaků.
- Set DDRAM Address – Nastaví DDRAM binární adresu AAAAAAA do ukazatele.
- Write data to DDRAM – zapíše data do DDRAM, hodnoty bitů DB0-DB7 zapisujeme podle toho jaký znak chceme zobrazit z CGROM.

Kapitola 3

Realizace v jazyce VHDL

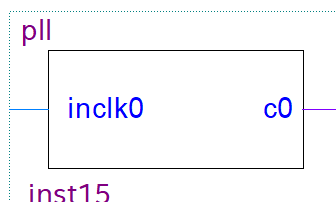
VHDL kód byl realizován v programu Quartus Prime Lite od firmy Intel. Tento program byl vybrán hlavně kvůli tomu, že obsahuje podporu pro vývojovou desku DE10-Lite. Tím je zaručeno, že syntetizovaný kód bude s deskou kompatibilní. VHDL kód je rozdělen do několika bloků (komponent), které jsou následně propojeny. Blokové schéma celého zapojení je uvedeno v příloze A.

3.1 Komponenty pro práci s hodinovým signálem

Přípravek Terasic DE-10Lite dodává dvojici hodinových signálů o kmitočtech 50 MHz a 10 MHz, ale pro různé části obvodu mohou být vhodnější jiné kmitočty. Proto jsou zde vytvořeny 2 děličky.

3.1.1 Komponenta PLL

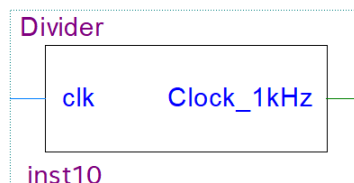
V komponentě PLL (Obrázek 3.1) je využit předdefinovaný blok fázového závěsu (PLL) v programu Quartus, který tuto konverzi mezi kmitočty realizuje. Na vstup bloku je přiveden hodinový signál o kmitočtu 50 MHz a výstupní signál má kmitočet 1 MHz, který je například vhodný pro časování komunikace se senzory.



Obrázek 3.1. Komponenta PLL

3.1.2 Komponenta Divider

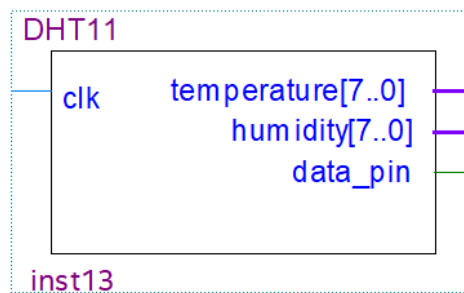
Na vstup komponenty Divider (Obrázek 3.2) je přiveden hodinový signál o kmitočtu 1 MHz a výstupní signál má kmitočet 1 kHz. Dělička je realizována pomocí čítače, který po stanoveném počtu náběžných hran vstupního signálu vždy neguje výstupní signál, čímž mění kmitočet, ale je zachována střída 1:1. Takový výstupní signál je například přiveden na vstup reproduktoru.



Obrázek 3.2. Komponenta Divider

3.2 Komponenta DHT11

Komponenta DHT11 (Obrázek 3.3) slouží k řízení komunikace se senzorem DHT11. Na vstupní port `clk` je přiveden hodinový signál o kmitočtu 1 MHz pro časování komunikace se senzorem. Dále je zde vstupně-výstupní port `data_pin` pro komunikaci se senzorem a dva výstupní porty `temperature` a `humidity`, pomocí kterých jsou hodnoty teploty a vlhkosti předávány k dalšímu zpracování.



Obrázek 3.3. Komponenta DHT11

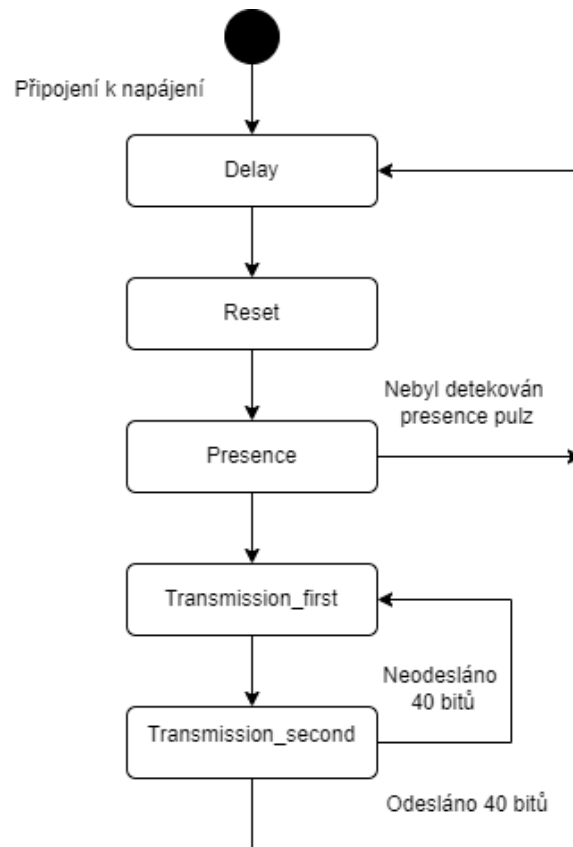
Komponenta je vytvořena jako stavový automat se stavy `Start`, `Response`, `Transmission_first`, `Transmission_second` a `Delay`. Stavový diagram tohoto automatu je na obrázku 3.4 a dodržuje časování, které je podrobněji rozebráno v kapitole 2.7.1.

Po připojení k napájení je doporučeno počkat alespoň 1 sekundu pro přečkání nestabilního stavu, než bude zahájena komunikace se senzorem. Zároveň je doporučeno mezi dvěma čteními dat ze senzoru počkat nejméně 2 sekundy. Proto výchozí stav je `Delay`, kde je nastaveno zpoždění 2 sekundy, po jejichž uplynutí následuje stav `Reset`.

Ve stavu `Reset` dojde k zahájení komunikace. Sběrnice je zde stažena na logickou 0 po dobu 18 ms. Po uplynutí této doby je sběrnice uvolněna a automat přejde do stavu `Presence`.

Ve stavu `Presence` probíhá kontrola, zda senzor stáhl sběrnici na logickou 0 a následně ji uvolnil. Pokud senzor stáhl sběrnici do logické 0, automat se přepne do stavu `Transmission_first`. Pokud senzor nestáhl sběrnici automat se přepne do stavu `Delay` a po 2 sekundách zkusí zahájit komunikaci znovu.

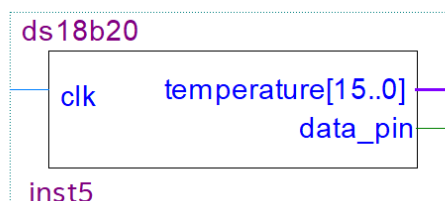
V posledních dvou stavech `Transmission_first` a `Transmission_second` probíhá přenos bitů ze senzoru. Ve stavu `Transmission_first` je zahájen odpočet délky pulzu, když je sběrnice v logické 1. Poté se přejde do dalšího stavu `Transmission_second` a čeká se než je sběrnice stažena do logické 0. Podle délky pulzu v logické 1 se určí zda je hodnota bitu logická 0 nebo 1 a je uložen do 40bitového vektoru. Následně proběhne kontrola zda už bylo odesláno všech 40 bitů. Pokud ne přejde automat zpět do stavu `Transmission_first` a proces se opakuje. Když je přeneseno všech 40 bitů, proběhne kontrolní součet přenesených bitů. Poslední bajt se musí rovnat součtu prvních 4 bajtů. Pokud se součet rovná data se uloží do vektorů `humidity` a `temperature`.



Obrázek 3.4. Diagram stavového automatu komponenty DHT11

3.3 Komponenta DS18B20

Tato komponenta (Obrázek 3.5) slouží k řízení komunikace se senzorem DS18B20. Na vstupní port `clk` je přiveden hodinový signál o kmitočtu 1 MHz pro časování komunikace se senzorem. Dále je zde vstupně-výstupní port `data_pin` pro komunikace se senzorem a výstupní port `temperature`, který předává data získaná od senzoru dále ke zpracování.



Obrázek 3.5. Komponenta DS18B20

Komponenta je vytvořena jako stavový automat se stavy `Reset`, `Presence`, `Commands`, `Write_data`, `Read_data` a `Conversion_delay`. Stavový diagram automatu je ukázán na obrázku 3.6 a časování komunikace je podrobněji rozebráno v kapitole 2.5.1.

Po připojení k napájení se nachází stavový automat ve stavu `Reset`, ve kterém je zahájena komunikace. Ve stavu je sběrnice stažena na logickou 0 po dobu $500 \mu\text{s}$. Následně je sběrnice uvolněna a stavový automat přejde do stavu `Presence`.

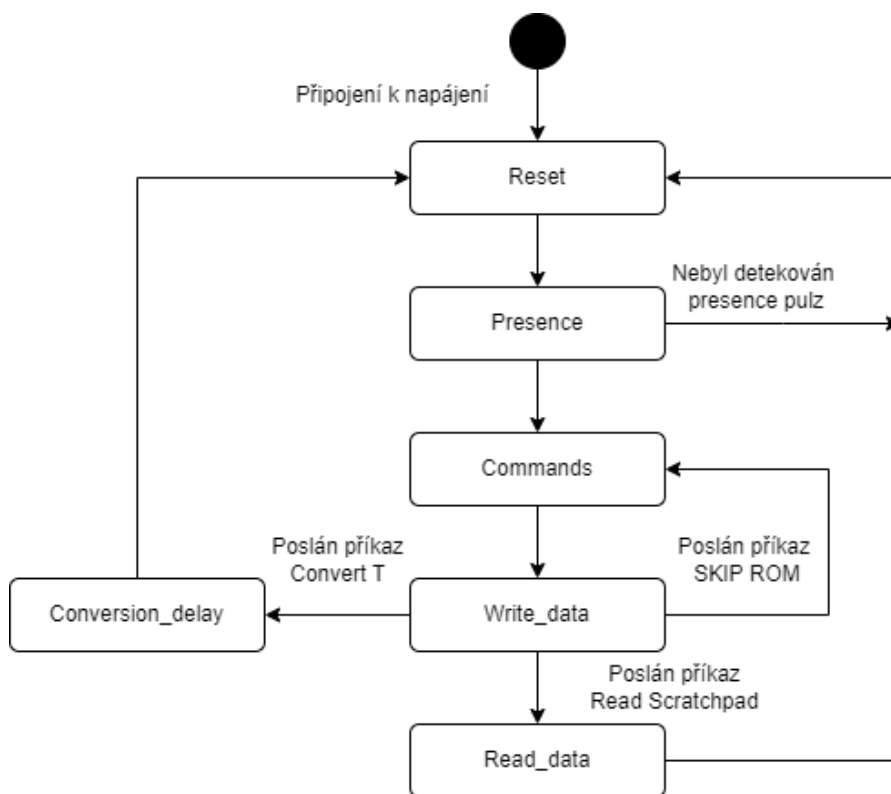
Ve stavu `Presence` probíhá kontrola jestli je sběrnice stažena do logické 0. Pokud byla detekována logická 0, senzor odpověděl prezenčním pulzem a komunikace může po 500

μs pokračovat přechodem do stavu Commands, čímž je inicializace dokončena. Jestliže nebyla detekována logická 0, stavový automat přejde do stavu Reset a zkusí zahájit komunikaci znovu.

Stav commands obsahuje binární hodnoty příkazů Skip ROM, Convert T a Read Scratchpad, které jsou potřebné pro komunikaci s jedním senzorem DS18B20 na sběrnici. První se odešle příkaz Skip ROM. Všechny příkazy jsou odesílány ve stavu Write_data, kde je nastaveno správné časování pro odesílání jednotlivých bitů. Druhý příkaz je odeslán Convert T. Po odeslání tohoto příkazu se stavový automat přepne do stavu Conversion_delay, ve kterém zůstane dokud není dokončen převod teploty. Převod je ukončen, když senzor uvolní sběrnici a je detekována logická 1.

Poté stavový automat přejde znovu do stavu Reset a Presence, čímž proběhne znovu inicializace. Ve stavu Commands opět odešle příkaz Skip ROM a následně pošle příkaz Read Scratchpad. Po tomto příkazu se stavový automat přepne do stavu Read_data.

V posledním stavu Read_data probíhá přenos dat ze senzoru. Podle délky pulzu je vždy určeno zda je hodnota bitu logická 1 nebo logická 0. Celkově by senzor měl poslat 9 bytů, ale přenos lze ukončit dříve, pokud nejsou všechna data ze scratchpad potřeba. V této práci po přenosu 16 bitů dojde k přechodu automatu do stavu Reset a celá komunikace začne od začátku.



Obrázek 3.6. Diagram stavového automatu komponenty DS18B20

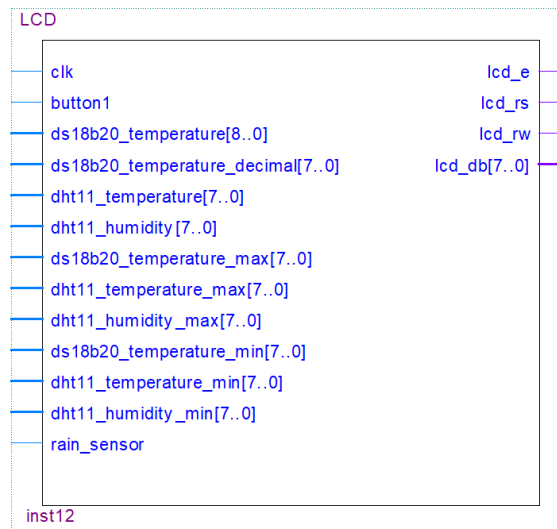
3.4 Komponenta LCD

V komponentě LCD (Obrázek 3.7) probíhá výběr dat, která budou zobrazena na LCD, a řízení komponenty LCD controller. Komponenta má vstupní port clk, na který je přiveden hodinový signál o kmitočtu 50 MHz. Na další vstupní port button1 je přiváděn signál z tlačítka. Na zbylé vstupní porty jsou přiváděna data ze senzorů a nastavené

maximální a minimální mezní hodnoty teploty a vlhkosti v kódu BCD (Binary Coded Decimal). Výstupní porty lcd-e, lcd-rs, lcd-rw a lcd-db slouží ke komunikaci s řadičem HD44780, komunikace probíhá v komponentě LCD controller.

Výběr dat, která jsou zobrazena na displeji, je řízen hodnotou signálu counter, která se mění při stisku tlačítka. Pokud je detekována sestupná hrana signálu z tlačítka (u přípravku DE-10Lite je výstupní signál z tlačítka invertován), k signálu counter se přičte jedna. Podle hodnoty signálu counter je určeno jaká data jsou zobrazena na displeji.

Na obrázku 3.8 je možné vidět zjednodušenou část kódu, který je použit pro detekci stisku tlačítka a na obrázku 3.9 je ukázka zjednodušeného kódu pro výběr dat zobrazovaných na displeji.



Obrázek 3.7. Komponenta LCD

```
process (button1) is    -- proces pro detekci stisku tlačítka
begin
  if button1 = '0' and button1'event then    --detekce sestupne hrany
    counter <= counter + 1;
    if counter > 1 then
      counter <= 0;
    end if;
  end if;
end process;
```

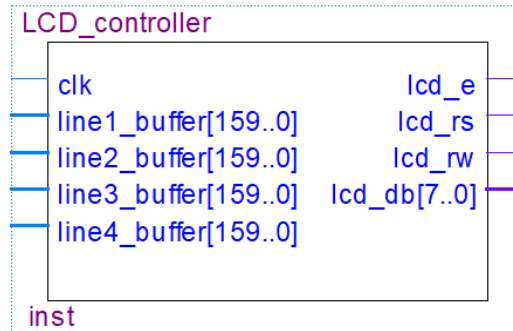
Obrázek 3.8. Ukázka VHDL kódu pro detekci stisku tlačítka

```
process(clk)    -- proces pro vyber dat zobrazovanych na displej
begin
  if counter = 1 then
    line1(95 downto 88) <= X"6D";    --m
    line1(87 downto 80) <= X"61";    --a
    line1(79 downto 72) <= X"78";    --x
  elsif counter = 2 then
    line1(95 downto 88) <= X"6D";    --m
    line1(87 downto 80) <= X"69";    --i
    line1(79 downto 72) <= X"6E";    --n
  end if;
end process;
```

Obrázek 3.9. Ukázka VHDL kódu pro výběr dat

3.4.1 Komponenta LCD controller

Komponenta slouží ke komunikaci s řadičem LCD displeje. Na vstupní port clk je přiveden hodinový signál o kmitočtu 50 MHz. Dále jsou zde vstupní porty line1-buffer, line2-buffer, line3-buffer a line4-buffer, na které jsou přiváděny binární hodnoty znaků, které mají být zobrazeny. Výstupní porty lcd-e, lcd-rs, lcd-rw a lcd-db slouží ke komunikaci s řadičem displeje. Ukázka komponenty je na obrázku 3.10.



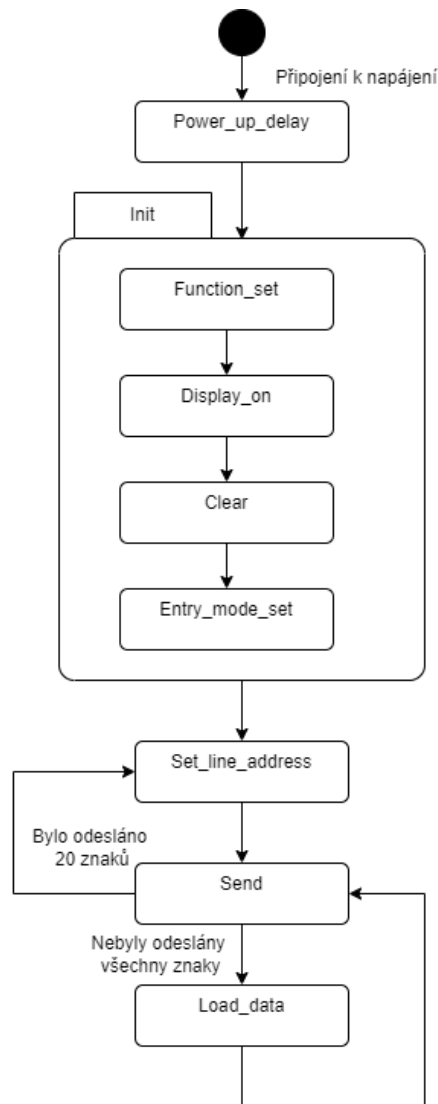
Obrázek 3.10. Komponenta LCD controller

Komponenta je vytvořena jako stavový automat se stavy `Power_up_delay`, `Init`, `Set_line_address`, `Send`, `Load_data`. Uvnitř stavu `Init` je vytvořen další stavový automat, který bude popsán dále. Celý stavový automat je vyobrazen na obrázku 3.11.

Výchozí stav stavového automatu je `Power_up_delay`, kde je nastaveno doporučené minimální zpoždění po připojení k napájení 15 ms. Po 15 ms přejde stavový automat do stavu `Init`.

Stav `Init` slouží k inicializaci displeje a je vytvořen jako další stavový automat se stavy `Function_set`, `Display_on`, `Clear` a `Entry_mode_set`. V každém z těchto stavů je poslán příkaz řadiči a nastaveno zpoždění, které je nutné pro provedení příkazu. Příkazy mají stejné jméno jako stavy a jsou popsány v kapitole 2.10.4 a tabulce 2.3. Po odeslání instrukce `Entry_mode_set` stavový automat přejde do stavu `Set_line_address`.

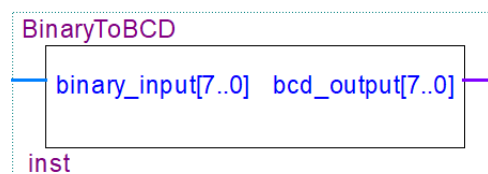
Ve stavu `Set_line_address` je na port `lcd_db` nastavena adresa prvního znaku prvního řádku (Obrázek 2.9). Následně dojde k přechodu do stavu `Send`, kde je příkaz odeslán. Po odeslání příkazu přejde automat do stavu `Load_data`, ve kterém je nastaveno prvních 8 bitů (první znak) z portu `line1_buffer` na port `lcd_db`. Následně automat přejde do stavu `Send` a odešle příkaz pro zobrazení znaku na displeji. Potom proběhne kontrola, zda byl poslední znak vypsaný na displej poslední znak první řádky. Pokud tomu tak není automat přejde do stavu `Load_data` a načte další znak a zobrazí ho na displej. Když je načten poslední znak první řádky automat přejde do stavu `Set_line_address`, kde je nastavena adresa prvního znaku druhé řádky a proces zobrazování znaků na displej se opakuje pro druhou řádku. Po zobrazení všech řádek na displeji, začne opět výpis první řádky a proces se opakuje.



Obrázek 3.11. Diagram stavového automatu komponenty LCD

3.5 Komponenta BinaryToBCD

V komponentě je realizován převod z binárních dat na data BCD. BCD kód je binární reprezentace desítkových čísel, kde je každá desítková číslice (0-9) reprezentována čtyř-bitovým binárním kódem. Data jsou převáděna, aby se po další úpravě mohli zobrazovat na 7segmentovém displeji a LCD. Ukázka komponenty je na obrázku 3.12.



Obrázek 3.12. Komponenta BinaryToBCD

3.5.1 Double dabble algoritmus

Tento algoritmus se běžně používá k převodu binárního čísla na BCD kód. Binární číslo je posunuto doleva pro každý jeho bit. Bity se posouvají z nejvýznamějšího bitu binárního čísla do nejméně významného bitu BCD. Po každém posunu se zkontrolují všechny BCD číslice, pokud je hodnota libovolné číslice větší nebo rovna 5, přičte se hodnota 3 k této číslici. Tento posun se provádí dokud nejsou přesunuty všechny bity z binárního čísla do BCD. Ukázka double dabble algoritmu v jazyce VHDL je zobrazena na obrázku 3.13.

```
entity BinaryToBCD is -- definice entity
  port ( binary_input : in  STD_LOGIC_VECTOR(7 downto 0);
        bcd_output   : out STD_LOGIC_VECTOR(7 downto 0));
end BinaryToBCD;

architecture Behavioral of BinaryToBCD is
begin
  process (binary_input)
    variable temp : STD_LOGIC_VECTOR(7 downto 0);
  begin
    temp := (others => '0'); -- vynulování vektoru

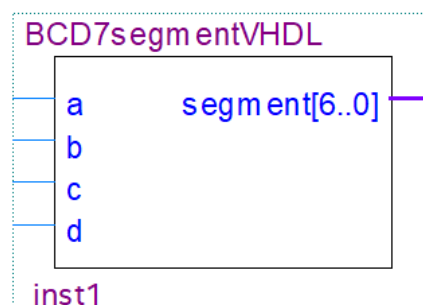
    for i in 7 downto 0 loop
      if temp(3 downto 0) >= "0101" then -- kontrola první číslice
        temp(3 downto 0) := temp(3 downto 0) + "0011"; -- přičtení 3 k první číslici
      end if;
      if temp(7 downto 4) >= "0101" then -- kontrola druhé číslice
        temp(7 downto 4) := temp(7 downto 4) + "0011"; -- přičtení 3 k druhé číslici
      end if;
      temp(7 downto 1) := temp(6 downto 0); -- posun vlevo
      temp(0) := binary_input(i);
    end loop;

    bcd_output <= temp;
  end process;
end Behavioral;
```

Obrázek 3.13. Double dabble algoritmus v jazyce VHDL

3.6 Komponenta BCDto7segment

Tato komponenta plní funkci dekodéru BCD na kód 7segmentového displeje. Na vstup bloku jsou přivedeny čtyři signály pojmenovány a,b,c,d, které odpovídají řádkovým místům kódu BCD. Výstupní signály jsou pojmenovány segment0 až segment6, které odpovídají řádkovým místům kódu 7segmentového displeje. Hodnoty výstupních signálů jsou rovny výsledkům logických operací, které je možné vidět na obrázku 3.15. Ukázka komponenty je na obrázku 3.14.



Obrázek 3.14. Komponenta BCDto7segment

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity BCD7segmentVHDL is --define entity
port (a,b,c,d : in std_logic;
      segment : out std_logic_vector(6 downto 0));
end BCD7segmentVHDL;

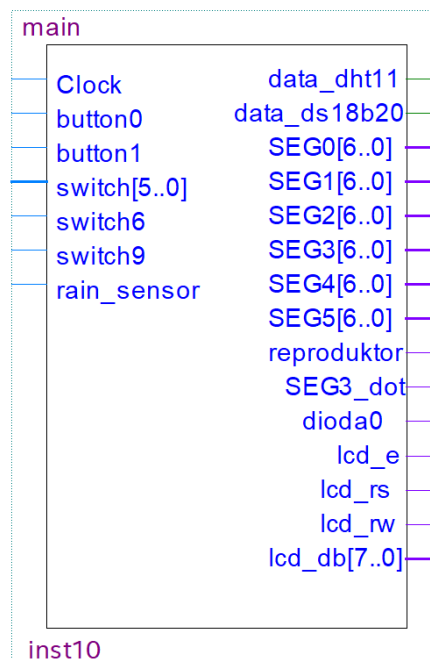
architecture RTL of BCD7segmentVHDL is
begin
segment(0) <= (not a and not b and c) or (a and not b and not c and not d);
segment(1) <= (a and not b and c) or (not a and b and c);
segment(2) <= (not a and b and not c);
segment(3) <= (not a and not b and c) or (a and b and c) or (a and not b and not c and not d);
segment(4) <= a or (not b and c);
segment(5) <= (b and not c) or (a and b) or (a and not c and not d);
segment(6) <= (not b and not c and not d) or (a and b and c);
end RTL;

```

Obrázek 3.15. Dekodér BCD na 7segmentový displej ve VHDL

3.7 Komponenta main

V komponentě main (Obrázek 3.16) dochází k propojení všech ostatních komponent, které byly uvedeny. Kromě propojení komponent je zde realizován výběr dat, která jsou zobrazována na 7segmentových displejích, nastavování maximálních a minimálních mezních hodnot pro teplotu a vlhkost a kontrola překročení těchto mezí.



Obrázek 3.16. Komponenta main

Výběr hodnot, které jsou zobrazeny, na 7segmentových displejích probíhá pomocí přepínačů, které jsou připojeny k vstupním portům switch a switch6.

Maximální a minimální meze mají přednastavené hodnoty. Hodnoty lze měnit pomocí tlačítka, které je přivedeno na vstupní port button0. Pokud je detekována náběžná hrana signálu, dojde k přičtení nebo odečtení hodnoty od dané meze podle polohy přepínače SW9 na přípravku. Ukázka detekce stiku tlačítka v jazyce VHDL je na obrázku 3.8.

Poslední část, kterou obsahuje komponenta main, je kontrola překročení mezí. Kontrola probíhá porovnáváním binárních dat ze sensorů s binárními hodnotami maximálních a minimálních mezí. Pokud je některá hodnota ze sensorů mimo meze, je vždy na

půl sekundy přiveden hodinový signál o kmitočtu 1 kHz na vstup reproduktoru. Poté je na reproduktor na půl sekundy nastavena logická 0, čímž je vytvořen přerušovaný zvukový alarm. Alarm je přerušovaný, aby se dal odlišit od alarmu, který slouží pro detekci deště. Tento alarm je spuštěn, pokud dojde k detekci logické 0 na výstupu senzoru deště. Po detekci deště je na reproduktor přiveden stejný hodinový signál o kmitočtu 1 kHz, který není přerušovaný.

3.8 Výsledná realizace

Přípravek získává data o teplotě a vlhkosti ze senzoru DHT11 a data o teplotě ze senzoru DS18B20. Data jsou poté převedena z binárního kódu na BCD kód. Data v BCD kódu jsou zobrazována na znakový LCD displej. Zároveň jsou data převáděna z BCD kódu na kód pro 7segmentový displej a zobrazována na displejích přípravku. Pro výběr dat, která se zobrazí na 7segmentových displejích, je využito šest přepínačů SW0–SW6. Zde jsou uvedeny jaká data se zobrazí při přepnutí přepínačů do hodnoty logická 1:

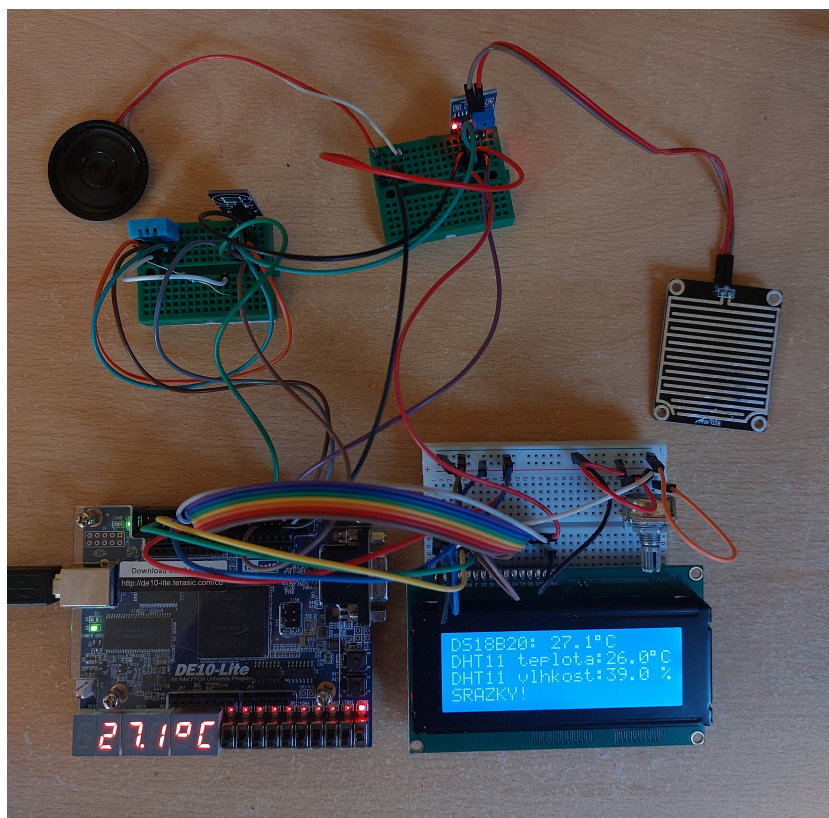
- SW0 – Teplota ze senzoru DHT11
- SW1 – Relativní vlhkost ze senzoru DHT11
- SW2 – Teplota ze senzoru DS18B20
- SW3 – Nastavení maximální hodnoty teploty pro senzor DHT11
- SW4 – Nastavení maximální hodnoty relativní vlhkosti pro senzor DHT11
- SW5 – Nastavení maximální hodnoty teploty pro senzor DS18B20
- SW6 – Pokud je přepínač SW6 ve stavu logická 1 společně s jedním z přepínačů SW3–SW5, je místo maximální hodnoty nastavována minimální hodnota pro daný senzor.

Při přepnutí dvou a více přepínačů SW0–SW5 se zobrazí na displejích samé nuly až na displej HEX5, který slouží pro zobrazení mínus při záporné teplotě ze senzoru DS18B20.

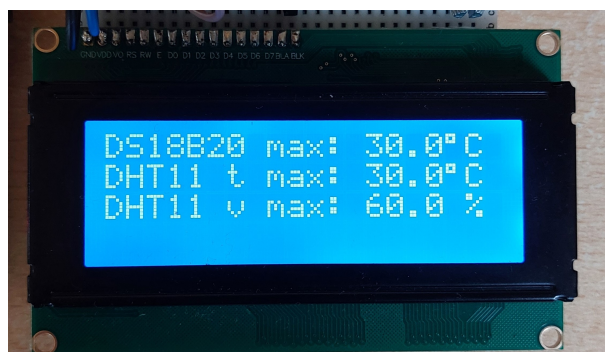
Pokud je vybrána nějaká mezní hodnota, lze její hodnotu měnit pomocí stisku tlačítka KEY0. Když dojde k stisku tlačítka, je k mezní hodnotě přičtena nebo odečtena 1, podle stavu přepínače SW9.

Pomocí tlačítka KEY1 je možné měnit zobrazovaná data na LCD displeji. Při základním zobrazení, které lze vidět na obrázku 3.17, jsou zobrazeny hodnoty teploty a vlhkosti ze senzorů. Po jednom stisku tlačítka jsou zobrazovány hodnoty maximálních mezí, což lze vidět na obrázku 3.19. Po druhém stisku tlačítka jsou vidět hodnoty minimálních mezí, což lze vidět na obrázku 3.18. Po dalším stisku se sekvence opakuje.

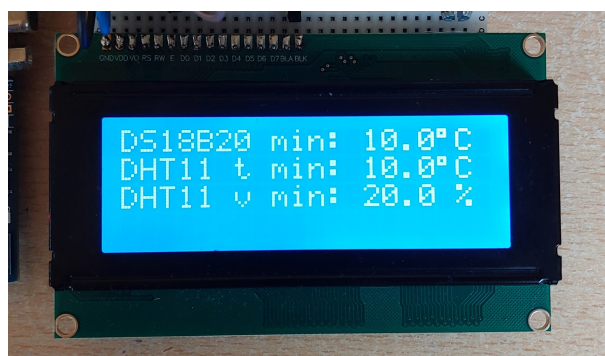
Při překročení mezí je spuštěn přerušovaný zvukový alarm na reproduktoru. Pokud jsou detekovány srážky na dešťovém senzoru, na LCD displeji se objeví varování a spustí se nepřerušovaný zvukový alarm.



Obrázek 3.17. Celkové zapojení práce a zobrazení naměřených hodnot na LCD displeji



Obrázek 3.18. Zobrazení maximálních mezních hodnot na LCD displeji



Obrázek 3.19. Zobrazení minimálních mezních hodnot na LCD displeji

Kapitola 4

Závěr

Zadáním této práce bylo seznámit se s přípravkem Terasic DE10-Lite a jazykem VHDL. Hlavním cílem bylo vytvořit VHDL kódy pro ovládání různých senzorů a zobrazení hodnot na segmentových displejích a znakovém LCD displeji. Tato práce dosáhla svého cíle vytvořením funkčního systému pro monitorování teploty a vlhkosti.

V teoretické části byly představeny základy funkce FPGA, jazyk VHDL a přípravek Terasic DE10-Lite. Dále byly uvedeny základní informace o senzorech DHT11 a DS18B20 a sběrnících Aosong 1-Wire a 1-Wire, které tyto senzory používají pro komunikaci. Popis zahrnoval také znakový LCD displej a řadič HD44780.

V praktické části byly podrobně popsány realizované VHDL kódy, které jsou rozděleny do jednotlivých komponent. Tyto komponenty umožňují řídit komunikaci se senzorem teploty DS18B20, senzorem teploty a vlhkosti DHT11 a dešťovým senzorem. Naměřené hodnoty jsou zpracovány a zobrazeny na segmentových displejích a znakovém LCD displeji. Zobrazení dat lze měnit pomocí tlačítek a přepínačů na přípravku. Dále lze nastavovat dolní a horní mezní hodnoty pro teplotu a vlhkost. Pokud naměřené hodnoty překročí tyto mezní hodnoty, uživatel je upozorněn alarmem v podobě reproduktoru.

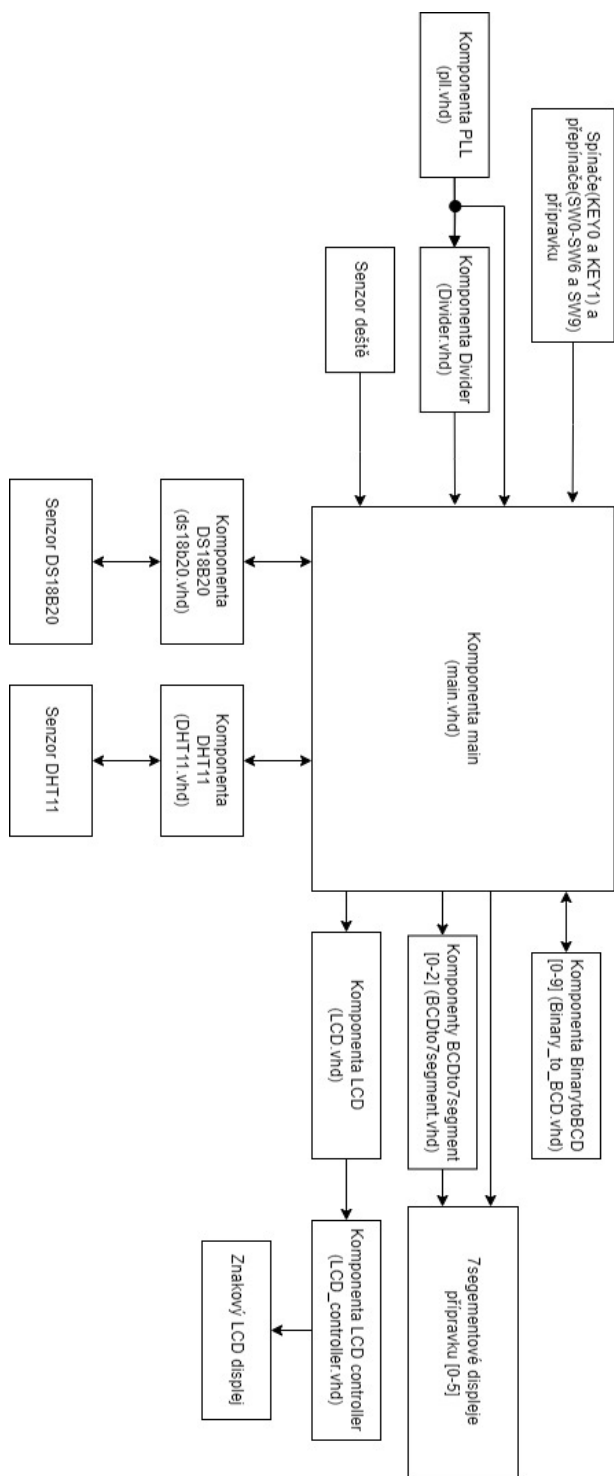
V budoucnu by se dala práce rozšířit o senzory pro měření dalších veličin nebo o komunikaci přes sériovou sběrnici s počítačem. Na počítači by mohla být vytvořena aplikace v programovacím jazyce Python nebo JavaScript, která by dále zpracovávala a zobrazovala naměřené hodnoty. Toto rozšíření by zvýšilo funkčnost a možnosti využití systému.

Literatura

- [1] LAFATA, Pavel, Petr HAMPL a Michal PRAVDA. *Digitální technika*. V Praze: ČVUT, 2011. ISBN 978-80-01-04914-3.
- [2] MAXFIELD, Clive. *The design warrior's guide to FPGAs: devices, tools and flows*. Amsterdam: Elsevier, 2004. ISBN 0-7506-7604-3.
- [3] MOUSER ELECTRONICS, Inc. *Altera Quartus® Prime Design Software* [online]. [cit.2024-5-15]. Dostupné na <https://cz.mouser.com/new/altera/intel-quartus/>.
- [4] *Terasic: DE10-Lite User Manual* [online]. Terasic, 2016. [cit.2023-12-14]. Dostupné na https://ftp.intel.com/Public/Pub/fpgaup/pub/Intel_Material/Boards/DE10-Lite/DE10_Lite_User_Manual.pdf.
- [5] LINKE, Bernhard. *Overview of 1-Wire Technology and Its Use* [online]. 2008. [cit.2023-12-14]. Dostupné na <https://www.analog.com/en/technical-articles/guide-to-1wire-communication.html>.
- [6] *DS18B20-Programmable Resolution 1-Wire Digital Thermometer* [online]. 2002. [cit.2023-12-12]. Dostupné na <https://www.analog.com/media/en/technical-documentation/data-sheets/ds18b20.pdf>.
- [7] AOSONG ELECTRONICS Co., Ltd. *Digital relative humidity & temperature sensor DHT11* [online]. [cit.2023-12-14]. Dostupné na https://www.electronicoscaldas.com/datasheet/DHT11_Aosong.pdf.
- [8] LASKAKIT S.R.O., © 2024. *Dešťový senzor* [online]. [cit.2024-5-18]. Dostupné na <https://www.laskakit.cz/destovy-senzor/>.
- [9] HITACHI, Ltd. *HD44780U (LCD-II) (Dot Matrix Liquid Crystal Display Controller/Driver)* [online]. 1999. [cit.2024-5-6]. Dostupné na <https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>.

Příloha A

Blokové schéma výsledné práce



Příloha B

Seznam příložených kódů v jazyce VHDL

- BCDto7segment.vhd
- Binary_to_BCD.vhd
- DHT11.vhd
- Divider.vhd
- ds18b20.vhd
- LCD.vhd
- LCD_controller.vhd
- main.vhd
- pll.vhd