VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

LABORATORNÍ ÚLOHY PRO MIKROKONTROLÉRY HCS 08

LABORATORY ASSIGNMENT FOR HCS 08 MICROCONTROLLERS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Jan Bilík**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Tomáš Macho, Ph.D.**
SUPERVISOR

BRNO 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**
Ústav automatizace a měřicí techniky

Student: Jan Bilík ***ID:*** 186032
Ročník: 3 ***Akademický rok:*** 2017/18

NÁZEV TÉMATU:

Laboratorní úlohy pro mikrokontroléry HCS 08

POKYNY PRO VYPRACOVÁNÍ:

1. Seznamte se s mikrokontroléry MC9S08LH64, vývojovou deskou TWR-S08LH64, sběrnicemi 1-Wire, IIC a SPI.
2. Navrhněte laboratorní úlohy demonstrující připojení snímače teploty po sběrnici 1-Wire k mikrokontroléru, kalendářového obvodu po sběrnici IIC a skupiny 8 LED diod (s využitím posuvného registru) přes sběrnici SPI.
3. Laboratorní úlohy implementujte včetně potřebných přizpůsobovacích obvodů.
4. Pro laboratorní úlohy vytvořte srozumitelné návody.
5. Ověřte funkčnost laboratorních úloh a vyhodnoťte dosažené výsledky.

DOPORUČENÁ LITERATURA:

MALÝ, Martin. Sběrnice 1-Wire. Vyvoj.hw [online]. 2014 [cit. 2017-08-07]. Dostupné z: http://vyvoj.hw.cz/navrh-obvodu/rozhrani/sbernice-1-wiretm.html

Termín zadání: 5. 2. 2018 ***Termín odevzdání:*** 21.5.2018

Vedoucí práce: Ing. Tomáš Macho, Ph.D.
Konzultant:

doc. Ing. Václav Jirsík, CSc.
předseda oborové rady

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta elektrotechniky a komunikačních technologií
Ústav automatizace a měřicí techniky

UPOZORNĚNÍ:

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

Abstrakt

Cílem bakalářské práce je seznámení se sériovými sběrnicemi I2C, SPI a 1-Wire a jejich implementace při realizaci komunikace mezi mikrokontrolérem NXP MC9S08LH64, teplotním čidlem, kalendářovým obvodem a posuvným registrem. Práce je také zaměřena na hardwarové připojení periferií k mikrokontroléru včetně přizpůsobovacích obvodů a softwarovou obsluhu včetně vysvětlení jednotlivých knihovních funkcí využitých při výměně dat mezi obvody a mikrokontrolérem.

Klíčová slova

HCS08, MC9S08LH64, IIC, I2C, SPI, 1-Wire, DS18B20, DS3231, SN74HC164, mikrokontrolér, sběrnice

Abstract

The goal of my bachelor thesis is getting acquainted with serial buses I2C, SPI and 1-Wire and their implementation in communication between microcontroller NXP MC9S08LH64, temperature sensor, real-time clock circuit and shift register. The thesis also focuses on hardware connection with necessary circuits and behaviour of software functions used during data exchange.

Keywords

HCS08, MC9S08LH64, IIC, I2C, SPI, 1-Wire, DS18B20, DS3231, SN74HC164, microcontroller, bus

Bibliografická citace:

BILÍK, J. *Laboratorní úlohy pro mikrokontroléry HCS 08.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 58 s. Vedoucí bakalářské práce Ing. Tomáš Macho, Ph.D..

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma Laboratorní úlohy pro mikrokontrolér HCS 08 jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brně dne: **10. května 2018** ……
podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Tomáši Machovi, Ph.D. za rady, poskytnutou volnost a odbornou pomoc při zpracování mé bakalářské práce.

V Brně dne: **10. května 2018** ……
podpis autora

Obsah

1. Úvod 10
2. Sériové sběrnice 11
2.1 Sběrnice 1-Wire 11
2.1.1 Napájení slave zařízení 11
2.1.2 Sběrnice 12
2.1.3 Přenos dat po sběrnici 1-Wire 12
2.2 Sběrnice IIC (I2C) 15
2.2.1 Obousměrná komunikace 15
2.2.2 Přenos logické úrovně 15
2.2.3 Přenos vysoké logické úrovně 16
2.2.4 Obecné operace na sběrnici I2C 16
2.2.5 Příkazy START a STOP 17
2.2.6 Formát přenášeného bytu 18
2.2.7 Potvrzení přenášeného bytu 18
2.2.8 Data 18
2.2.9 Přenos dat ze zařízení typu master do zařízení typu slave 19
2.2.10 Přenos dat ze zařízení typu slave do zařízení typu master 19
2.3 Sběrnice SPI 20
2.3.1 Přenos dat po sběrnici SPI 20
2.3.2 Signály MOSI a MISO 21
2.3.3 Výběr slave zařízení 21
2.3.4 Připojení zařízení slave 22
3. Mikrokontrolér MC9S08LH64 23
3.1 Periferie 23
3.2 Modul I2C 23
3.3 Modul SPI 24
3.4 Vývojová deska TWR-S08LH64 24
3.4.1 Vybavení desky 24
3.4.2 Napájení 25
4. Teplotní čidlo DS18B20 26
4.1 Vstupy a výstupy obvodu 26

5. Kalendářový obvod DS3231 27
5.1 Vstupy a výstupy obvodu 27
6. Posuvný registr SN74HC164 29
6.1 Vstupy a výstupy obvodu 29
7. Připojení periferií k vývojovému systému 30
7.1 Signály mikrokontroléru 30
7.2 Hardwarové zapojení periferií 33
8. Softwarová realizace 36
8.1 Knihovna 1-Wire 36
8.1.1 Inicializace komunikace 36
8.1.2 Časování komunikace 36
8.1.3 Funkce Reset 37
8.1.4 Zápis bitu 37
8.1.5 Čtení bitu 37
8.1.6 Zápis bytu 38
8.1.7 Čtení bytu 38
8.2 Knihovna I2C 39
8.2.1 Inicializace komunikace 40
8.2.2 Začátek komunikace 40
8.2.3 Opětovný začátek komunikace 41
8.2.4 Ukončení komunikace 41
8.2.5 Zápis bytu 42
8.2.6 Čtení bytu 43
8.3 Knihovna SPI 45
8.3.1 Inicializace knihovny 45
8.3.2 Čtení a zápis jednoho bytu 45
8.4 Knihovna DS18B20 47
8.4.1 Čtení teploty 47
8.4.2 Čtení a konverze teploty 49
8.4.3 Čtení jedinečné adresy čidla 50
8.4.4 Zadání úlohy k otestování knihovny 50
8.5 Knihovna DS3231 51

8.5.1 Nastavení bytu 51
8.5.2 Nastavení 2 a více bytů 51
8.5.3 Vyčítání bytu 52
8.5.4 Vyčítání 2 a více bytů 52
8.5.5 Zadání úlohy k otestování knihovny 53
8.6 Knihovna ShiftRegister 53
8.6.1 Nastavení bytu 53
8.6.2 Změna pinu 53
8.6.3 Nastavení pinu 54
8.6.4 Rotace vlevo a rotace vpravo 54
8.6.5 Bitová inverze 54
8.6.6 Zadání úlohy k otestování knihovny 54
9. Závěr 56
10. Seznam použité literatury 57
11. Seznam příloh na CD 58

Seznam symbolů a zkratek

Zkratky:

BCD	…	Binary Coded Decimal (dvojkově reprezentované dekadické číslo)
CPU	…	Central processing unit (centrální procesorová jednotka)
CS	…	Chip Select
FET	…	Field Effect Transistor (unipolární tranzistor řízený polem)
IIC (I2C)	…	Inter-Integrated Circuit
IRQ	…	External Interrupt Request
KBI	…	Keyboard Interrupt
LCD	…	Liquid Crystal Display (displej z tekutých krystalů)
MISO	…	Master In Slave Out
MOSI	…	Master Out Slave In
PWM	…	Pulse Width Modulation (pulzní šířková modulace)
RAM	…	Random Access Memory (paměť s náhodným příštupem)
ROM	…	Read Only Memory (paměť určena pouze ke čtení)
RTC	…	Real Time Clock (kalendářový obvod)
SPI	…	Serial Peripheral Interface (sériové periferní rozhraní)
SS	…	Slave Select
TOD	…	Time Of Day (modul mikrokontroléru)

Symboly:

C	…	kapacita	[F]
f	…	frekvence	[Hz]
Q	…	elektrický náboj	[Ah]
R	…	odpor	[Ω]
T	…	teplota	[°C]
t	…	čas	[s]

Seznam obrázků

OBR. 4-1 ČASOVÉ SLOTY SBĚRNICE 1-WIRE (PŘEVZATO Z [3]) 14
OBR. 5-1 STÁHNUTÍ SBĚRNICE K LOG.0 POMOCÍ TRANZISTORU FET (PŘEVZATO Z [5]) ... 16
OBR. 5-2 PŘÍKAZ START A STOP (PŘEVZATO Z [5]) 17
OBR. 6-1 SÉRIOVÉ ŘETĚZENÍ SLAVE ZAŘÍZENÍ NA SBĚRNICI SPI (UPRAVENO PODLE [8]) 22
OBR. 7-1 PŘIPOJENÍ SIGNÁLŮ NA PINY TEPLOTNÍHO ČIDLA (PŘEVZATO Z [3]) 26
OBR. 8-1 PŘIPOJENÍ SIGNÁLŮ NA PINY KALENDÁŘOVÉHO OBVODU (PŘEVZATO Z [9]) ... 28
OBR. 9-1 VSTUPY A VÝSTUPY POSUVNÉHO REGISTRU (PŘEVZATO Z [10]) 29
OBR. 10-1: SCHÉMA PŘIPOJENÍ PERIFERIÍ K VÝVOJOVÉMU SYSTÉMU 30
OBR. 10-2 ZÁKLADNÍ VYVEDENÍ SIGNÁLŮ VÝVOJOVÉ DESKY 31
OBR. 10-3 ZÁKLADNÍ VYVEDENÍ SIGNÁLŮ MODULU TOWER 31
OBR. 10-4 PÁJIVÉ POLE S EXTERNÍMI OBVODY 33
OBR. 10-5 ROZMÍSTĚNÍ SOUČÁSTEK A VYVEDENÍ SIGNÁLŮ NA PÁJIVÉM POLI 34
OBR. 11-1 VÝVOJOVÝ DIAGRAM - ODESÍLÁNÍ BYTU 1-WIRE 38
OBR. 11-2 VÝVOJOVÝ DIAGRAM - PŘIJÍMÁNÍ BYTU 1-WIRE 39
OBR. 11-3 VÝVOJOVÝ DIAGRAM - ZÁPIS BYTU I2C 43
OBR. 11-4 VÝVOJOVÝ DIAGRAM - ČTENÍ BYTU I2C 44
OBR. 11-5 VÝVOJOVÝ DIAGRAM - ČTENÍ A ZÁPIS BYTU SPI 47
OBR. 11-6 VÝVOJOVÝ DIAGRAM - VYČTENÍ TEPLOTY Z DS18B20 49

Seznam tabulek

TAB. 10-1 PŘIPOJENÍ SIGNÁLŮ 32

1. ÚVOD

Fakulta elektrotechniky a komunikačních technologií VUT v Brně ve své výuce nabízí předmět Mikrokontroléry, který bude brzy měnit své jméno na Embedded systémy. Náplní tohoto předmětu je seznámení se s mikrokontrolérem MC9S08LH64 a práce s vývojovou deskou TWR-S08LH64.

Vývojová deska obsahuje velké množství periferií, ale neobsahuje obvody, které by umožňovaly využívat funkce sběrnic I2C, SPI a 1-Wire.

Přiložená bakalářská práce je zaměřena na připojení periferií a umožňuje seznámení se a využívání sériových sběrnic při komunikaci s kalendářovým obvodem, teplotním čidlem a posuvným registrem.

Pro demonstraci funkce sběrnic je nutno zvolit vhodné obvody, které komunikaci umožňují. Pro sběrnici SPI jsem zvolil posuvný registr SN74HC164, ke kterému je připojena osmice vysoce svítivých LED diod. Dále jsem zvolil kalendářový obvod DS3231, který komunikuje po sběrnici I2C. Speciálním případem sériové sběrnice je 1-Wire. Tato asynchronní sběrnice se využívá při připojení jednoduchých zařízení, která pro komunikaci využívají menší množství dat. Jedním z těchto zařízení je teplotní čidlo firmy Maxim Integrated DS18B20, které v této práci využívám.

Svou práci jsem rozdělil do dvou částí. První část věnuji teoretickému seznámení s mikrokontrolérem, použitými externími obvody, funkcemi jednotlivých sběrnic, jejich signály a způsoby zapojení. Druhá část se zabývá hardwarovým zapojením přípravku a softwarovou realizací. Součástí práce jsou i návody pro jednotlivé laboratorní úlohy.

2. SÉRIOVÉ SBĚRNICE

Sériové sběrnice jsou takové sběrnice, u kterých je průběh přenosu dat veden sekvenčně po jednotlivých bitech, na rozdíl od paralelních sběrnic, které přenáší několik bitů zároveň.

Mezi sériové sběrnice patří také 1-Wire, I2C a SPI. Teoretický popis těchto tří sběrnic se nachází v následující kapitole.

2.1 Sběrnice 1-Wire

Sběrnice 1-Wire je navržená pro komunikaci s jednoduchými senzory a zařízeními s jednodrátovým rozhraním. Tento systém komunikace se používá s pomalejšími zařízeními, které mají nižší výkon. Existují dva druhy komunikace se zařízeními. Komunikace standardní rychlostí a overdrive rychlostí. U standardní rychlosti komunikace lze dosáhnout rychlosti dat 16,3 kbit/s, zatímco u overdrive módu komunikace lze rychlost přibližně zdesetinásobit.

1-Wire sběrnice umožňuje obousměrný tok dat, který se nesmí konat ve stejném čase. Pomocí sběrnice lze zařízení slave i napájet. Druhou možností je připojení samostatného napájecího vodiče.

Na sběrnici smí být připojen jeden nebo více slave zařízení. Je zde ovšem jen jeden master, který řídí přenos informací po sběrnici. Master inicializuje všechny přenosy na datové lince. Přenos dat je možný pouze mezi masterem a řízenými zařízeními, nikoli mezi zařízeními slave navzájem.

Pro komunikaci není třeba používat samostatný synchronizační signál. Každý slave je synchronizován na sestupnou hranu signálu přítomného na sběrnici.

Při přenosu jednoho bytu se jako první přenáší nejméně významný bit (LSB).

2.1.1 Napájení slave zařízení

Slave připojený ke sběrnici 1-Wire může být napájen dvěma způsoby. Ke vnějšímu napájení dochází v případě připojení napájecího pinu slave zařízení k napájecímu napětí. Tato topologie se využívá pokud slave vyžaduje více energie pro svou práci.

Parazitní napájení lze použít pro slave zařízení, které nevyžadují při činnosti velké množství energie. V tomto případě je slave připojen ke sběrnici, ze které ukládá energii v interním kapacitoru. Ten zásobuje zařízení energií ve chvílích, kdy je na sběrnici vystavená logická 0.

2.1.2 Sběrnice

Každý pin, na který je sběrnice připojena, musí být open-drain. 1-Wire protokol umožňuje přenos dat mezi dvěma zařízeními i ve stavu, kdy jsou ostatní podřízená zařízení ve stavu nečinnosti. Ke sběrnici se připojuje pull-up rezistor spojující sběrnici a napájecí napětí. To má za následek, že pokud jedno zařízení stahuje sběrnici k nule, objeví se tato hodnota na sběrnici. Velikost rezistoru se určuje na základě následujících kritérií:

Je-li zařízení napájeno externě, lze použít pull-up odpor s hodnotou okolo 10 kΩ nebo více.

Je-li zařízení napájeno externě, lze použít pull-up odpor s hodnotou menší než 1 kΩ v případě, kdy používáme overdrive mód a je tedy potřeba rychlých změn na sběrnici.

Je-li zařízení napájeno přes sběrnici, musí se použít pull-up odpor, který zvládá napájení všech slave zařízení i v situacích, kdy tato zařízení vyžadují více energie v důsledku náročných operací.

Zařízení pracující na bázi open-drain smí sběrnici připojovat pouze k potenciálu země. Pro vystavení nízké úrovně, za kterou se považuje napětí nižší než 0,3 násobek napájecího napětí, slouží tranzistor FET, který po aktivaci připojí sběrnici k zemi a tím sníží její napětí. Po vypnutí tranzistoru se pomocí pull-up odporu napětí na sběrnici opět zvýší.

Pro přenos vysoké logické úrovně, za kterou se považuje napětí vyšší než 0,7 násobek napájecího napětí, postačuje, pokud zařízení pouze vypne tranzistor FET. Odpojením tranzistoru dojde k rozpojení sběrnice a země. Pull-up odpor zvýší napětí na sběrnici na vysokou logickou úroveň.

2.1.3 Přenos dat po sběrnici 1-Wire

Komunikace se zařízeními pracujícími na sběrnici 1-Wire začíná inicializační sekvencí, která obsahuje Reset puls vyvolaný masterem a Presence puls vyvolaný slave zařízením. Zařízení odpovídá na Reset puls vystavením Presence pulsu. Ten slouží masterovi k indikaci přítomnosti slave zařízení na sběrnici, zároveň puls informuje mastera, že je slave připraven k činnosti.

Inicializační sekvenci zahajuje master Reset pulsem, kdy je na sběrnici vystavena logická 0 po dobu minimálně 480 μs. Po Reset pulsu master uvolní sběrnici a naslouchá. Jakmile je sběrnice masterem uvolněna, vytáhne pull-up odpor sběrnici do logické 1. Přítomný slave detekuje vzestupnou hranu, po které čeká 15 až 60 μs. Poté vystaví na sběrnici Presence puls v podobě logické 0 po dobu 60 až 240 μs.

Pro přenos dat po sběrnici 1-Wire se využívají časové sloty. Master vystavuje data na sběrnici v průběhu zapisovacích časových slotů a vyčítá data v průběhu čtecích

časových slotů. V průběhu jednoho časového slotu je po sběrnici přenesen právě jeden bit. Oba typy časových slotů jsou iniciovány masterem, který stahuje sběrnici dolů.

Zápis 1 do časového slotu slouží k zápisu logické 1 a zápis 0 pro zápis logické 0 do slave zařízení připojeného ke sběrnici. V obou případech musí mít tento slot minimální délku 60 μs a před dalším časovým slotem prodlevu minimálně 1 μs pro zotavení.

Generování časového slotu pro zápis 1 zahajuje master, který stáhne sběrnici k nule a drží ji staženou po dobu maximálně 15 μs. Poté uvolní sběrnici, pull-up rezistor vytáhne sběrnici na vysokou úroveň a počká do konce časového okna. Generování slotu pro zápis 0 probíhá obdobným způsobem, ale master neuvolňuje sběrnici, ale drží ji stáhnutou po celou dobu časového slotu.

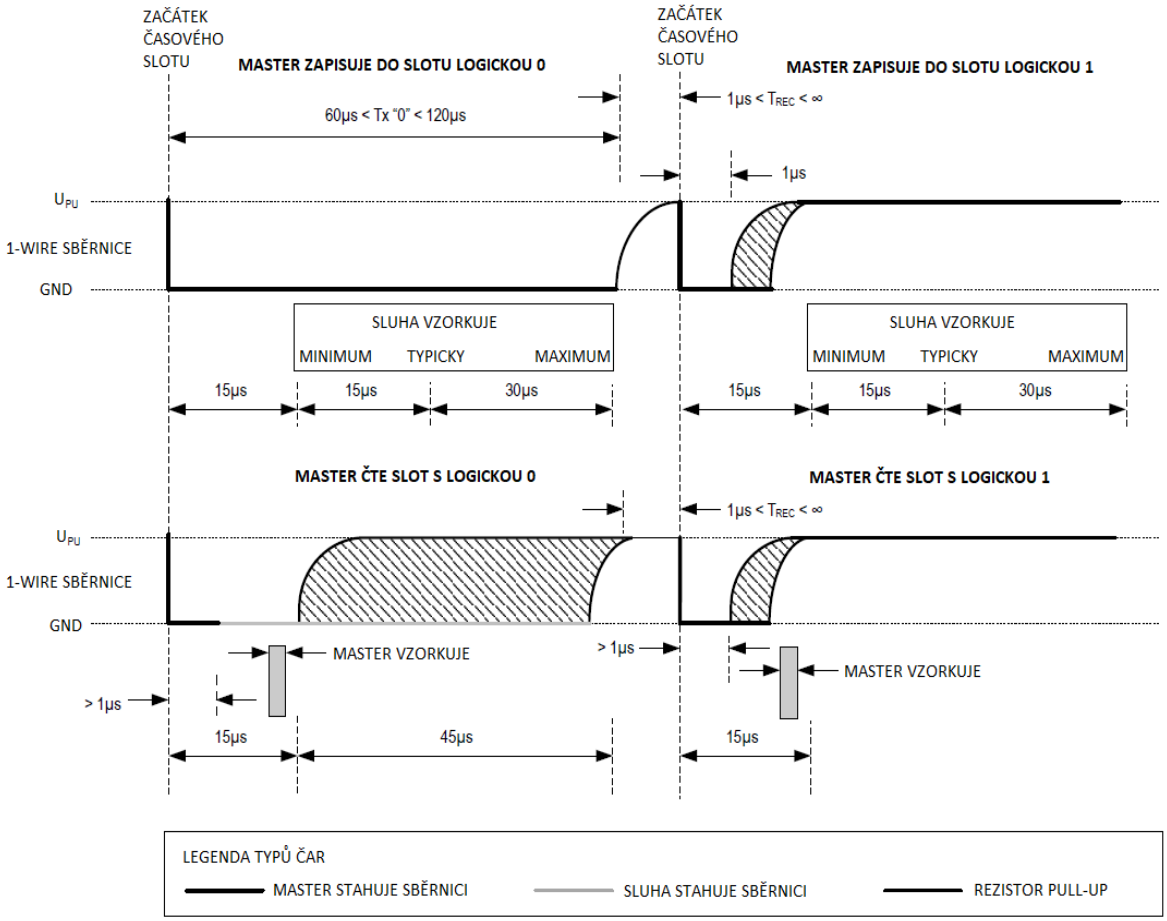
Podřízené zařízení pracující na sběrnici detekuje sestupnou hranu. Po 15 μs od sestupné hrany vzorkuje stav sběrnice. Vzorkování může probíhat v jakémkoliv čase od 15 do 60 μs od sestupné hrany. Pokud má v době vzorkování sběrnice vysokou úroveň, zapíše zařízení do paměti logickou 1, pokud nízkou úroveň, zapíše do paměti logickou 0.

Slave je schopen vysílat data na sběrnici pouze tehdy, pokud master inicializuje čtecí sloty. Jako všechny časové sloty, i tyto musejí mít minimální délku 60 μs, a mezi jednotlivými sloty se musí nacházet mezera minimálně 1 μs.

Čtecí slot inicializuje master stáhnutím sběrnice 1-Wire dolů po dobu minimálně 1 μs. Po tomto úkonu master uvolní sběrnici a naslouchá. V návaznosti na stažení sběrnice začne zařízení slave přenášet log.1 nebo log.0. Slave přenáší logickou 1 uvolněním sběrnice, kdy vlivem pull-up odporu dojde ke zvýšení úrovně. Pokud je přenášena logická 0, slave uvolní sběrnici až na konci čtecího slotu.

Od slave zařízení jsou výstupní data platná po dobu 15 μs od sestupné hrany zahajující čtecí slot. Z toho důvodu musí master opustit sběrnici a číst její stav do 15 μs od začátku čtecího slotu.[1]

[1] Převzato z [3]



Obr. 2-1 Časové sloty sběrnice 1-Wire (převzato z [3])

2.2 Sběrnice IIC (I2C)

Sběrnice I2C je velmi populární synchronní sériová sběrnice typu multimaster sloužící ke komunikaci mezi řídicím zařízením (master) a řízeným zařízením (slave), popřípadě mezi více zařízeními typu master a několika slave zařízeními. Původně byla navržena firmou Philips pro spotřební elektroniku, postupem času byla rozšířena do embedded systémů, například pro připojení kalendářových obvodů. Sběrnice je tvořena pouze dvěma signálovými vodiči: vodičem SCL obsahujícím hodinový signál a vodičem SDA obsahujícím přenášená data. V porovnání s ostatními sběrnicemi je malé množství vodičů jedna z největších výhod komunikace po této sběrnici.

2.2.1 Obousměrná komunikace

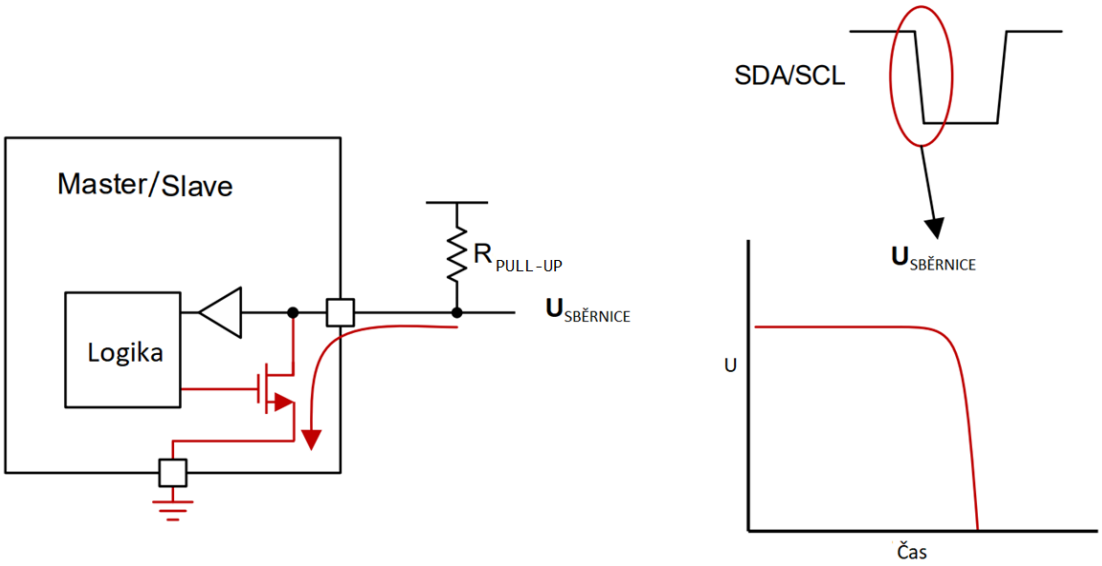
Na sběrnici I2C je pro komunikaci použit open-drain výstup. Tento výstup umožňuje uvolnění datového vodiče, nebo přitáhnutí vodiče k napětí. Ve většině případů se používá potenciál země.

Při uvolnění je datový vodič vytažen pomocí pull-up rezistoru k napájecímu napětí. V případě, kdy je vodič uvolněn masterem i zařízením slave, pull-up rezistor vytáhne signál k napájecímu napětí. To znamená, že žádnému zařízení připojenému k datovému vodiči není umožněno násilně vnutit vysokou logickou úroveň. Díky tomuto řešení se předchází problému spojenému s připojením napájecího napětí k zemi, který by mohl nastat v případě, že by jedno zařízení vnucovalo na vodič vysokou logickou úroveň a jiné zařízení stahovalo vodič k zemi. Podle pravidel komunikace na I2C sběrnici je od mastera vyžadováno pozdržet se komunikace v době, kdy je na datovém vodiči nízká úroveň značící komunikaci jiného zařízení.

2.2.2 Přenos logické úrovně

Obvodové zapojení pracující na bázi open-drain umožňuje vodič pouze připojovat k potenciálu země pro přenos nízké logické úrovně, nebo uvolnit vodič a umožnit pull-up rezistoru zvýšit napětí pro přenos vysoké logické úrovně. Nízkou úrovní se v případě protokolu I2C rozumí napětí sběrnice v rozmezí -0,5 V až 0,3 násobek napájecího napětí VDD. Jako vysoká logická úroveň je pak považováno napětí sběrnice dosahující alespoň velikosti 0,7 násobku napájecího napětí VDD.

Na obrázku je nakreslena situace, kdy logika vyžaduje přenos nízké úrovně. Z toho důvodu je aktivován pull-down FET, který zařídí připojení sběrnice k potenciálu země, což způsobí snížení napětí na datovém vodiči.



Obr. 2-2 Stáhnutí sběrnice k log.0 pomocí tranzistoru FET (převzato z [5])

2.2.3 Přenos vysoké logické úrovně

V případě, že master nebo slave vyžadují přenos vysoké logické úrovně, stačí jim pouze uvolnit datový vodič. Uvolnění vodiče se provede pomocí vypnutí pull-down FET. Vypnutím pull-down tranzistoru dojde k odpojení vodiče od potenciálu země. Nyní je datový vodič připojen pouze k napájecímu napětí přes pull-up rezistor. Pull-up rezistor zvýší napětí na datovém vodiči, a to se dá interpretovat jako vysoká logická úroveň.[2]

2.2.4 Obecné operace na sběrnici I2C

Na obousměrném I2C rozhraní dochází ke komunikaci mezi mikrokontrolérem, který zde vystupuje jako master a ostatními slave zařízeními. Tomuto zařízení není dovoleno komunikovat po sběrnici, pokud nebyl osloven masterem. Oslovení probíhá pomocí adresy zařízení. Každé slave zařízení má svou specifickou nezaměnitelnou adresu. Tato adresa umožňuje masterovi zvolit jen jedno zařízení, se kterým hodlá komunikovat.

Mnoho slave zařízení vyžaduje konfiguraci při zapnutí zařízení. Konfiguraci provádí master, který přistoupí k interním registrům řízeného zařízení zápisem požadované hodnoty. Přístup k interním registrům se provádí pomocí unikátních registrových adres. Zařízení vystupující jako slave může mít jeden nebo více registrů, do kterých jsou data zapsána nebo čtena. Přenos dat může být zahájen pouze v případě

[2] Převzato z [4]

nečinnosti sběrnice. Nečinnost nastává v případě, že oba vodiče tvořící rozhraní mají vysokou logickou úroveň po příkazu STOP nebo doposud neproběhl příkaz START.

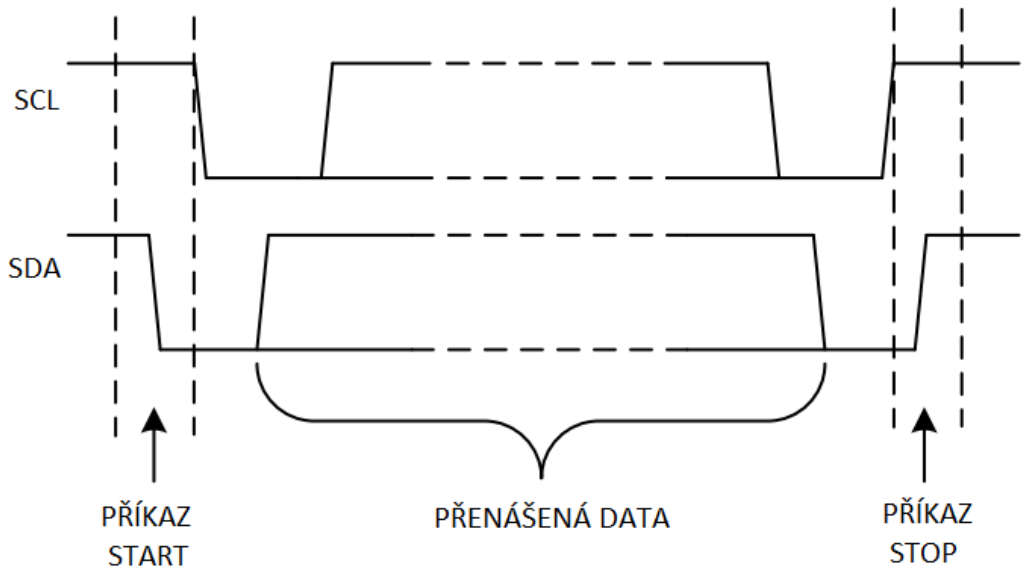
Rozhraní I2C je tvořeno dvěma vodiči. Sériovými hodinami, serial clock (SCL) a sériovými daty, serial data (SDA). Oba vodiče musí být připojeny k napájecímu napětí přes pull-up rezistor. Velikost pull-up rezistoru je nutno zvolit s ohledem na kapacitu komunikační linky, která se odvíjí od délky vodičů.

Obecný postup při komunikaci s podřízeným zařízením za účelem odeslání dat vyžaduje přenos příkazu START a adresaci přijímače zařízení, se kterým master hodlá komunikovat. Po adresaci master vysílá data, která daný adresát přijme a zpracuje. Po odeslání dat master ukončí komunikaci odesláním STOP příkazu.

Pro přijmutí dat od slave zařízení se postupuje obdobným způsobem. Master zahajuje konverzaci vysláním START příkazu, poté adresuje vysílač slave zařízení, od kterého vyžaduje data. Master následně přijímá data. Po přijmutí všech požadovaných dat ukončí master přenos pomocí příkazu STOP

2.2.5 Příkazy START a STOP

Pro inicializaci komunikace na sběrnici I2C slouží příkaz START, naopak pro ukončení komunikace slouží příkaz STOP. Tyto příkazy jsou generovány masterem. Příkaz START, sloužící pro začátek komunikace, je definován jako sestupná hrana na datovém vodiči SDA ve chvíli, kdy má hodinový vodič SCL vysokou úroveň. Náběžná hrana na datovém vodiči SDA v okamžiku vysoké úrovně SCL značí příkaz STOP. Na obrázku je zobrazen průběh příkazů START a STOP.



Obr. 2-3 Příkaz START a STOP (převzato z [5])

2.2.6 Formát přenášeného bytu

Přenos bitů probíhá v rytmu hodinových pulsů na vodiči SCL. Informace o délce jednoho bytu je rozdělena do jednotlivých bitů. S každým tikem hodin je odeslán jeden bit informace. Jako byte může vystupovat adresa registru nebo adresa zařízení a zapsaná nebo vyčtená data zařízení slave.

V průběhu přenosu jednoho bytu je prvním přeneseným bitem nejvýznamnější bit (MSB) a posledním nejméně významný bit (LSB). Master může vysílat jakékoliv byty v čase mezi START a STOP příkazem. Podmínkou je stabilita dat na vodiči SDA v průběhu vysoké logické úrovně na vodiči SCL. Pokud dojde během této úrovně na vodiči SCL ke změně dat na SDA, je změna interpretována jako řídicí příkaz, mezi které patří příkazy START a STOP.

2.2.7 Potvrzení přenášeného bytu

Poslední bit přenášené informace (LSB) je následován jedním bitem ACK. Tento bit umožňuje přijímači komunikovat s vysílačem a potvrdit správné přijetí celého bytu informace. Až po vyslání bitu ACK může být vyslán další byte informace. Před vysláním bitu ACK musí být vysílačem vodič SDA uvolněn.

Pro vyslání ACK bitu musí přijímač stáhnout vodič SDA do nízké logické úrovně v čase, kdy je na hodinovém vodiči SCL nízká logická úroveň. Během vysoké logické úrovně na SCL zůstává na vodiči SDA stabilní logická úroveň.

Pokud během deváté periody hodinového cyklu zůstane na datovém vodiči SCL vysoká logická úroveň, považuje se byte za nepotvrzený, tzv. bit NACK. Nepotvrzení přenášeného bytu může nastat v případě, kdy přijímač není schopen přijímat ani vysílat, jelikož je zaneprázdněn jinou činností. Dalším případem, kdy je vystaven bit NACK, je obdržení nesrozumitelných informací, popřípadě situace, kdy přijímač není schopen přijmout více dat. Bit NACK může být také vyslán, pokud je přijímačem master. V tomto případě bit značí dokončení čtení.

2.2.8 Data

Data jsou přijímána nebo vysílána slave zařízením nebo masterem. O směru dat rozhoduje master pomocí zápisu do registrů. Tyto registry jsou umístěny v paměti slave. Každý registr má svou specifickou adresu, na kterou se při zápisu odkazuje master. Zapsáním specifické hodnoty do určitého registru dává master pokyny slave zařízení.

Ne všechna podřízená zařízení mají více registrů. Některá jednoduchá zařízení obsahují pouze jeden registr, do kterého lze zapisovat zasláním dat bezprostředně po

adresaci zařízení. Tato zařízení mají pouze svou adresu a neobsahují adresy registrů. Příkladem zařízení, které neobsahuje adresy registrů, může být 8-bitový switch řízený pomocí I2C příkazů. Zařízení využívá hodnotu jednoho bitu k sepnutí nebo rozepnutí kanálu. Master do řízeného zařízení zapisuje jen jeden byte ihned po jeho adresaci. Přeskakuje tak adresu registru.

2.2.9 Přenos dat ze zařízení typu master do zařízení typu slave

Na začátku procesu je třeba podřízenému zařízení oznámit, že bude probíhat zápis. Pro začátek přenosu musí master vystavit na sběrnici příkaz START následovaný adresou zařízení, se kterým master hodlá komunikovat. Oznámení je posláno pomocí nejméně významného bitu v adrese slave zařízení. Proces zapisování signalizuje natavení tohoto bitu na hodnotu log.0.

Slave odpovídá bitem ACK. Po odpovědi master zašle adresu registru, do kterého si přeje zapisovat, to je následováno dalším potvrzovacím bitem ACK. Nyní master může zasílat data, které budou do slave zařízení zapisována. Po zaslání všech potřebných dat, v mnoha případech se jedná jen o jeden byte, ukončí master přenos příkazem STOP.

2.2.10 Přenos dat ze zařízení typu slave do zařízení typu master

Čtení probíhá velmi podobným způsobem jako zápis. Na rozdíl od zápisu se zde ale objevují další kroky. Při požadavku na vyčítání dat ze slave zařízení musí master uvést, ze kterého registru si přeje vyčíst data.

Master zahajuje komunikaci stejným způsobem jako při zápisu, tj. po vystavení příkazu START následuje adresa slave zařízení ukončená bitem s hodnotou log. 0, který signalizuje zápis. Po vystavení bitu ACK řízeným zařízením následuje vyslání adresy registru. Jakmile je vystaven bit ACK, provede master znovu příkaz START následovaný adresou slave. Adresa je zakončena bitem s hodnotou log.1, který signalizuje vyčítání dat. Slave opět reaguje vystavením bitu ACK. Master nyní musí uvolnit vodič SDA, ale na vodič SCL stále vysílat hodinový signál, kterým udává takt pro vysílání dat ze slave zařízení.

Uvolněním vodiče SDA dostává slave možnost přenášet data v rytmu hodinových pulsů. Za každým přeneseným bytem vystavuje master na sběrnici bit ACK, kterým vyjadřuje připravenost k přijetí dalších dat. Po přijmutí očekávaného počtu bytů vyšle master bit NACK způsobující zastavení komunikace ze strany slave zařízení. Slave uvolňuje sběrnici a master ukončuje komunikaci příkazem STOP.[3]

[3] Převzato z [5]

2.3 Sběrnice SPI

SPI je běžný komunikační protokol používaný v různých zařízeních. Prostřednictvím tohoto protokolu komunikují s mikrokontrolérem například SD karty, bezdrátové moduly pracující na frekvenci 2,4 GHz, kalendářové obvody, paměti s konfiguračními daty nebo další mikrokontroléry.

V jednom streamu může být zasláno velké množství bitů bez nutnosti zastavení. Například u sběrnice I2C jsou data zasílána v paketech, které limitují množství přenášených bitů. Zároveň je třeba používat příkazy START a STOP na začátku a konci paketu, což vede k přerušení toku dat.

Mezi zařízeními komunikujícími přes protokol SPI existuje vztah master-slave. Jako master obvykle vystupuje mikrokontrolér, který řídí komunikaci, zatímco jako slave vystupuje senzor, display nebo paměťový čip. Nejjednodušší konfigurace při komunikaci pomocí protokolu SPI představuje systém obsahující pouze jednoho mastera a jedno slave zařízení, ovšem obecně může master komunikovat s více než jedním zařízením typu slave.

Sběrnice SPI využívá 4 signály. Signál MOSI (Master Out, Slave In) představuje vodič, na který master vystavuje data určená pro slave zařízení. Na vodič MISO (Master In, Slave Out) naopak vystavuje data slave a master je přijímá. Hodiny, sloužící pro synchronizaci, se objevují na vodiči s názvem SCLK. Signál SS (Slave Select) nebo CS (Chip Select) slouží pro výběr slave zařízení, se kterým je vedena komunikace.

Mezi klady sběrnice SPI se řadí především rychlost přenosu dat, která může dosahovat téměř dvojnásobné rychlosti oproti sběrnici I2C při stejné frekvenci hodinového signálu. Využitím signálů CS/SS odpadá povinnost složitého adresování slave a také použití příkazů START a STOP. Samostatné signály MOSI a MISO umožňují vysílat a přijímat data ve stejném čase, což opět zrychluje komunikaci.

Mezi nevýhody patří větší počet vodičů než u sběrnic I2C a 1-Wire. Nepřítomnost potvrzovacího bitu, jakým je u protokolu I2C ACK bit vyslaný přijímačem, neumožňuje potvrzení správného přijetí dat.[4]

2.3.1 Přenos dat po sběrnici SPI

Komunikaci na sběrnici SPI řídí master, který také zahajuje komunikaci. Na zařízení, se kterým hodlá master komunikovat, je připojena nízká úroveň signálu CS/SS. Následně dochází k připojení hodinového signálu na vodič SCLK. Master začne vysílat data na lince MOSI rychlostí jeden bit za jeden hodinový cyklus. Slave vzorkuje stav sběrnice v čase náběžné nebo sestupné hrany hodinového signálu a ukládá

[4] Převzato z [6]

jednotlivé hodnoty. Volba hrany hodinového signálu, při které dochází ke vzorkování, je nastavitelná. Pokud je potřeba přijmout data vyslaná ze slave zařízení, ponechává master na lince SCLK připojený hodinový signál. Data jsou pak vysílána a přijímána stejným způsobem.[5]

Zařízení typu master i zařízení typu slave obsahují 8-bitový posuvný registr, který slouží k přenosu dat.

Hodinový signál na vodiči SCLK slouží k synchronizaci výstupu dat a vzorkování. V rytmu hodin jsou postupně vysouvána data z posuvného registru zařízení master a ukládána do posuvného registru zařízení slave. Zároveň jsou vysouvána i data z posuvného registru zařízení slave a jsou ukládána do posuvného registru zařízení master. Z obou posuvných registrů je vysunut vždy jeden bit s každým hodinovým cyklem, což znamená, že rychlost přenosu dat závisí na frekvenci hodinového signálu. Komunikace po sběrnici SPI je vždy iniciována masterem, jelikož ten konfiguruje a generuje hodinový signál.

Jakýkoliv komunikační protokol, který sdílí hodinový signál, je nazýván synchronním. Z toho důvodu je SPI synchronní komunikační protokol.

Použitý hodinový signál může být modifikovaný pomocí změny polarity a fáze hodin. Tyto vlastnosti určují, kdy bude na sběrnici vystaven bit a v jakém okamžiku bude vzorkován. Softwarově lze určit, zda budou bity vystaveny a vzorkovány na sestupné hraně, nebo na vzestupné hraně hodinového signálu.[6]

2.3.2 Signály MOSI a MISO

Prostřednictvím signálu MOSI zasílá master data, které přijímá slave. Tato data jsou zasílána bit po bitu. Signál MOSI je připojen k pinu slave zařízení s názvem MOSI. Obvykle master zasílá jako první nejvýznamnější bit (MSB).

Slave zasílá data masterovi pomocí signálu MISO. U takto zasílaných dat se povětšinou jako první zasílá nejvíce významný bit (MSB).

2.3.3 Výběr slave zařízení

Master si může vybrat, se kterým slave zařízením hodlá komunikovat. Výběr zařízení probíhá prostřednictvím signálu CS/SS. V okamžiku, kdy nedochází k přenosu, má signál CS/SS vysokou logickou úroveň, kterou se rozumí napětí vyšší než 0,7 násobek napájecího napětí. Pro slave je rozhodující nízká úroveň signálu CS/SS, která oznamuje, že s ním master hodlá komunikovat. Jako nízká logická úroveň je považováno menší napětí než 0,3 násobek napájecího napětí.

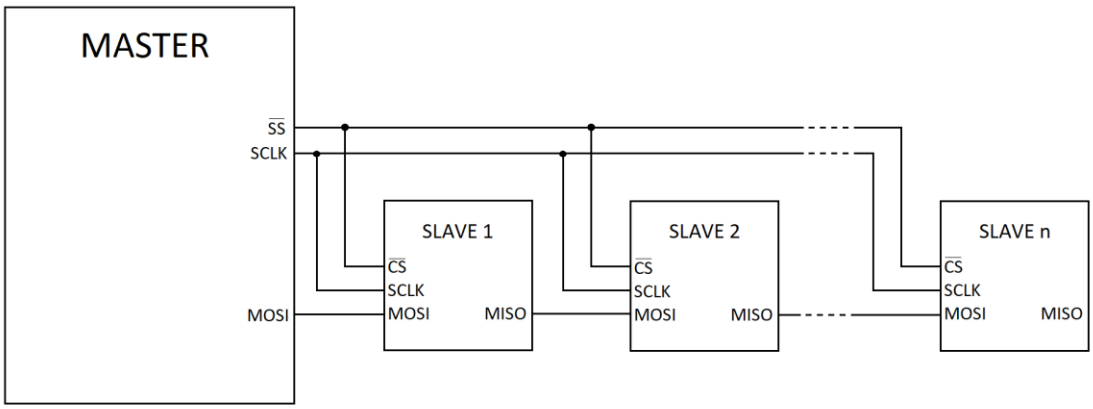
[5] Převzato z [8]
[6] Převzato z [7]

Pokud je komunikační protokol SPI použit při komunikaci mastera pouze s jedním zařízením typu slave, připojuje se pin slave zařízení CS/SS k zemi, tudíž zůstává stále aktivní. Master může obsahovat větší množství CS/SS pinů, to umožňuje připojit více slave zařízení paralelně. Pokud master umožňuje připojení pouze jednoho signálu CS/SS, větší množství slave přístrojů může být připojeno pomocí sériového řetězení.[7]

2.3.4 Připojení zařízení slave

SPI sběrnice umožňuje komunikaci jednoho mastera s jedním slave zařízením, ale také většího počtu slave. Existují zde dvě možnosti, jak připojit podřízená zařízení k masterovi. Pokud má master více pinů signálu slave select, lze podřízená zařízení připojit paralelním způsobem.

Pokud je dostupný pouze jeden pin slave select, budou podřízená zařízení připojena sériovým řetězením. V takovém případě se připojuje signál MOSI na stejnojmenný vstup prvního slave. Na vstup MOSI druhého slave se ovšem připojuje signál MISO z prvního slave a stejným způsobem se připojují další podřízená zařízení. Data vyslaná masterem poté přetékají z jednoho slave do druhého a zastaví se až u posledního. Pro dosažení určeného slave zařízení je tedy potřeba vyslat dostatečné množství dat. Tento typ zapojení se obvykle používá v situacích, kdy jsou slave pouze výstupní zařízení, jako například posuvné registry ovládající LED diody, u kterých není třeba přijímat žádné údaje zpět. U tohoto typu zapojení se ponechává vodič MISO odpojen.



Obr. 2-4 Sériové řetězení slave zařízení na sběrnici SPI (upraveno podle [8])

[7] Převzato z [4]

3. MIKROKONTROLÉR MC9S08LH64

MC9S08LH64 je 8-bitový mikrokontrolér firmy NXP patřící do rodiny HCS08, který dokáže pracovat s frekvencí CPU do 40 MHz při napětí 2,1V až 3,6V. Při frekvenci CPU do 20 MHz lze použít nižší napájecí napětí 1,8V až 2,1V. Mikrokontrolér podporuje 32 zdrojů přerušení. Díky několika úsporným režimům je možno dosáhnout velmi nízké spotřeby energie. Výhodou tohoto modelu je také příznivý poměr ceny a výkonu.

Na čipu je umístěna 4 KB RAM a 64 KB FLASH paměť. Zabudované bezpečnostní obvody nedovolují neoprávněný přístup k RAM a FLASH paměti. Hodinový signál mohou poskytovat vnější keramické rezonátory nebo vnitřní rezonátor, který díky možnosti programového dolaďování dosahuje odchylky jen 2% v závislosti na teplotě a napájecím napětí. Sběrnice pracují na poloviční frekvenci než jádro.

3.1 Periferie

Periferie jsou obecně zařízení, která se připojují k mikrokontroléru za účelem rozšíření jeho možností. Mikrokontrolér obsahuje následující periferie:

- Ovladač LCD
- Osmikanálový AD převodník s rozlišením 16 bitů
- Modul PWM
- Analogový komparátor
- TOD modul (hodiny reálného času)
- Moduly pro komunikaci po sběrnicích SCI, SPI a I2C

3.2 Modul I2C

Modul I2C slouží ke komunikaci mezi řadou zařízení, kdy maximální délka sběrnice a maximální počet připojených zařízení je limitován pouze maximální kapacitou sběrnice, která může být maximálně 400 pF. Signál SDA je defaultně přiveden na pin PTB4 a signál SCL na pin PTB5. Stejně jako v případě přesměrování signálů u modulu SPI, také u modulu I2C lze signály přesměrovat pomocí nastavení bitu IICPS v registru SOPT2. Po přesměrování se signál SDA připojí na pin PTA2 a signál SCL na pin PTA3. Hodinový signál lze na sběrnici připojovat a odpojovat pomocí nastavování bitu I2C v registru SCGC1. Po resetu je hodinový signál do modulu připojen.

3.3 Modul SPI

Mikrokontrolér má již z výroby implementovaný modul pro komunikaci přes rozhraní SPI. Komunikace probíhá pomocí signálů MISO, MOSI, SPSCK a SS. Tyto signály mohou být softwarově přesměrovány na jiné piny. Výchozí pozice signálu MISO je pin PTA2, signálu MOSI pin PTA3, signálu SPSCK pin PTA1 a signálu SS pin PTA0. Přesměrování pinů probíhá pomocí nastavení bitu SPIPS v registru SOPT2. Defaultně je nastavena log. 0, po nastavení na log.1 se signál MISO připojí na pin PTB4, MOSI na PTB5, SPSCK na PTB6 a SS na PTB7. Sběrnicové hodiny u SPI modulu, které se vyskytují na signálu SPSCK, mohou být vypínány a zapínány pomocí bitu SPI v registru SCGC2. Po resetu jsou bity nastaveny, což způsobí přivedení hodinového signálu do modulu. Pokud se modul nepoužívá, může být přívod signálu přerušen za účelem ušetření energie.[8]

3.4 Vývojová deska TWR-S08LH64

Pro mikrokontrolér MC9S08LH64 existují různé vývojové systémy, jedním z nich je vývojová deska TWE-S08LH64. Deska je navržena pro připojení do rozhraní NXP Tower System a umožňuje vývojářům odlaďování aplikačního software. Všechny signály generované mikrokontrolérem jsou dostupné na okrajových konektorech desky.

3.4.1 Vybavení desky

Mikrokontrolér se v praxi využívá k různorodým aplikacím. Z důvodu možnosti odladění softwaru sloužícího pro rozdílné aplikace je deska osazena různými periferiemi. Jednou z nich je LCD display GD-5306P, který je přímo připojen k mikrokontroléru. Další periferií je tříosý akcelerometr MMA7361L s nastavitelnou citlivostí. Ten slouží k odladění aplikací zaznamenávajících pohyb. Dále vývojová deska obsahuje piezo-bzučák, který slouží k odladění slyšitelných aplikací. Tento bzučák má rezonanční kmitočet 2400 Hz. Pro simulaci analogového vstupu je na okraji vývojové desky přítomen 5 kΩ potenciometr. Pro zaznamenání světelného záření je dále přítomen i senzor světla v podobě fototranzistoru. Proud tímto senzorem je dán intenzitou okolního světla. Na vývojové desce se také nacházejí 4 uživatelská tlačítka, ke kterým je připojen 22 pF kondenzátor pro minimalizaci zákmitů a které obsahují uživatelsky odpojitelný pull-up odpor. Dále zde lze nalézt 4 uživatelské LED diody zelené barvy, které jsou aktivní v nízké logické úrovni.

[8] Převzato z [1]

3.4.2 Napájení

Při použití Tower systému může být deska napájena přímo pomocí něj. Pokud napájení z Tower systému není použito, může být deska napájena přes rozhraní USB. Během napájení pomocí USB sběrnice je nutno dodržet omezení proudu z hostitelského USB, aby nedošlo k poškození desky, hostitelského počítače nebo celého Tower systému. Pro bateriové aplikace se na spodní straně desky nachází držák baterie o kapacitě 190 mAh.[9]

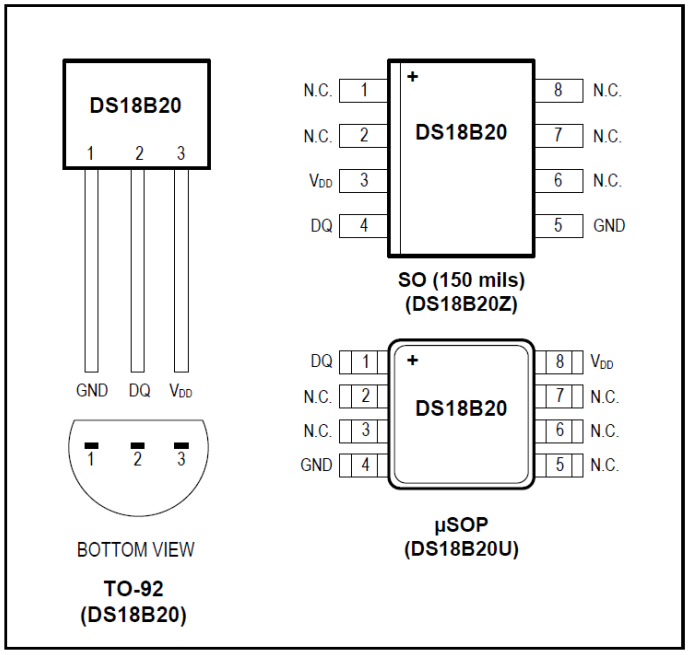
[9] Převzato z [2]

4. TEPLOTNÍ ČIDLO DS18B20

Snímač teploty DS18B20 vyráběný firmou Maxim, dříve Dallas, je oblíbený především pro svou přijatelnou cenu, teplotní rozsah -55 až +125°C a přesnost ±0,5 °C v rozsahu -10 až +85°C. Přesnost rozlišení lze nastavit od 9 do 12 bitů. Komunikace s mikrokontrolérem probíhá přes rozhraní 1-Wire, které k přenosu dat využívá pouze jeden vodič. Čidlo teploty lze napájet parazitně pomocí sběrnice 1-Wire nebo samotným napájecím vodičem. Každé čidlo teploty má své unikátní 64-bitové sériové číslo, což umožňuje připojení více čidel na jednu sběrnici. Toto číslo je uloženo v ROM paměti čidla.[10]

4.1 Vstupy a výstupy obvodu

Ve své práci využívám teplotní čidlo s pouzdrem TO-92. Napájení je uskutečňováno pomocí pinu VDD a není tak využito možnosti parazitního napájení. Pin DQ slouží k přenosu dat a je propojen s pinem mikrokontroléru zajišťujícím komunikaci prostřednictvím protokolu 1-Wire. Poslední pin GND slouží k připojení zemního potenciálu.



Obr. 4-1 Připojení signálů na piny teplotního čidla (převzato z [3])

[10] Převzato z [3]

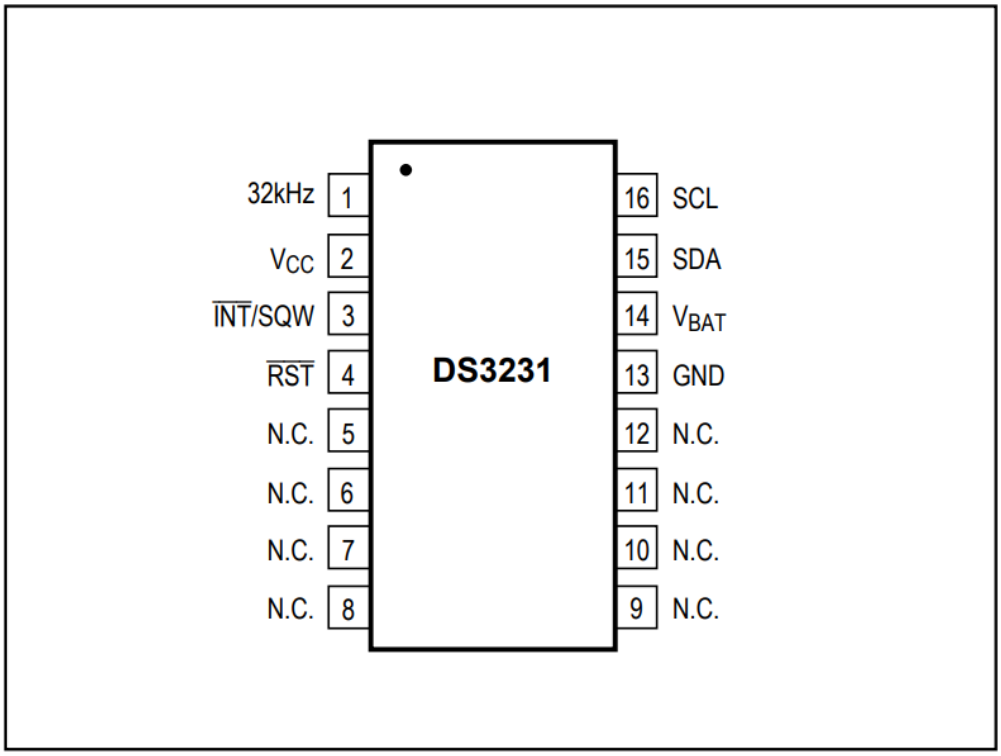
5. KALENDÁŘOVÝ OBVOD DS3231

DS3231 je přesný kalendářový obvod komunikující po sběrnici I2C. Pro zvýšení přesnosti je interní oscilátor teplotně kompenzován. Kalendářový obvod umožňuje udržovat informace o hodinách, minutách, sekundách, dnech, datu, roku a dnu v týdnu. Pro měsíce, které obsahují méně než 31 dnů je datum automaticky přizpůsobeno. Kalendářový obvod obsahuje také korekci pro přestupný rok. Formát uložených hodin je možno přepínat mezi 24 a 12 hodinovým formátem. Při použití 12 hodinového formátu je uložena značka pro indikaci AM/PM. Na výstupním pinu obvodu lze odebírat obdélníkový signál o frekvenci 32 kHz, na druhý výstupní pin lze softwarově připnout obdélníkový signál o určené frekvenci. Frekvence tohoto signálu jsou pevně dané, lze mezi nimi vybírat, ale nelze je libovolně nastavovat. Obvod umožňuje nastavení až dvou programovatelných přerušení. Tato přerušení se vyvolají v okamžiku, kdy je nastavený čas roven času běžícím v kalendářovém obvodu. Ve své práci tato přerušení nevyužívám.[11]

5.1 Vstupy a výstupy obvodu

Napájení kalendářového obvodu se uskutečňuje pomocí dvou rozdílných pinů. Na pin VCC se připojuje napájecí napětí, zatímco na pin VBAT se připojuje záložní napětí baterie. Napájení ze záložní baterie umožňuje běh kalendářového obvodu ve chvílích, kdy není připojeno napájecí napětí VCC. Pomocí hodinového signálu, připínaném na pin SCL a datového vodiče připínaném na pin SDA, probíhá komunikace podle standardů I2C. Na pinech 32 kHz a INT/SQW se nachází obdélníkový signál, který je možno využívat pro další aplikace. Poslední zapojený pin obvodu RST slouží k resetování obvodu. Zbytek pinů zůstává nezapojen.

[11] Převzato z [9]



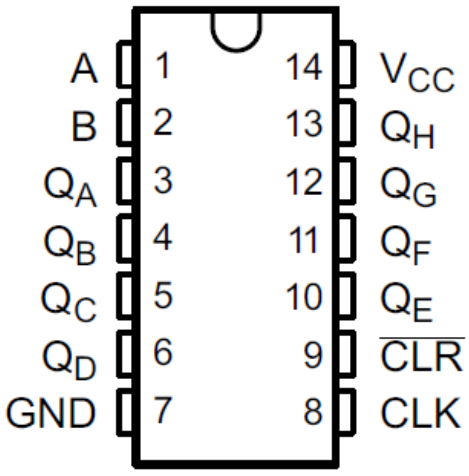
Obr. 5-1 Připojení signálů na piny kalendářového obvodu (převzato z [9])

6. POSUVNÝ REGISTR SN74HC164

8-bitový posuvný registr obsahuje sériově řazené synchronní klopné obvody, které obsahují na vstupu obvod AND. Asynchronní vstup CLR sloužící k resetování klopných obvodů je aktivní v nízké logické úrovni. Vzorkování příchozích dat probíhá v průběhu vzestupné hrany na vstupu CLK. Vystavený bit prostřednictvím signálu MOSI je přiveden na vstupní AND obvod a nastavuje nebo resetuje první klopný obvod. Data původně uložená v klopném obvodu jsou přepisována do následujících klopných obvodů. Výstupní signál z posledního klopného obvodu je přiváděn zpět do mikrokontroléru a slouží jako signál MISO.[12]

6.1 Vstupy a výstupy obvodu

Pouzdro posuvného registru obsahuje celkově 14 pinů, z nichž všechny jsou využity. Napájení probíhá pomocí pinu VCC. Na vstupech obvodu A a B je připojen obvod AND. Z toho důvodu je vstupní signál MOSI připojován současně na oba piny. Výstupy jednotlivých klopných obvodů jsou vyvedeny na piny QA až QH. Pin CLR slouží k připojení resetovacího signálu a v této úloze není využit. Obvod je synchronizován pomocí hodin připojovaných na pin CLK. Na poslední pin posuvného registru GND se připojuje zemní potenciál.

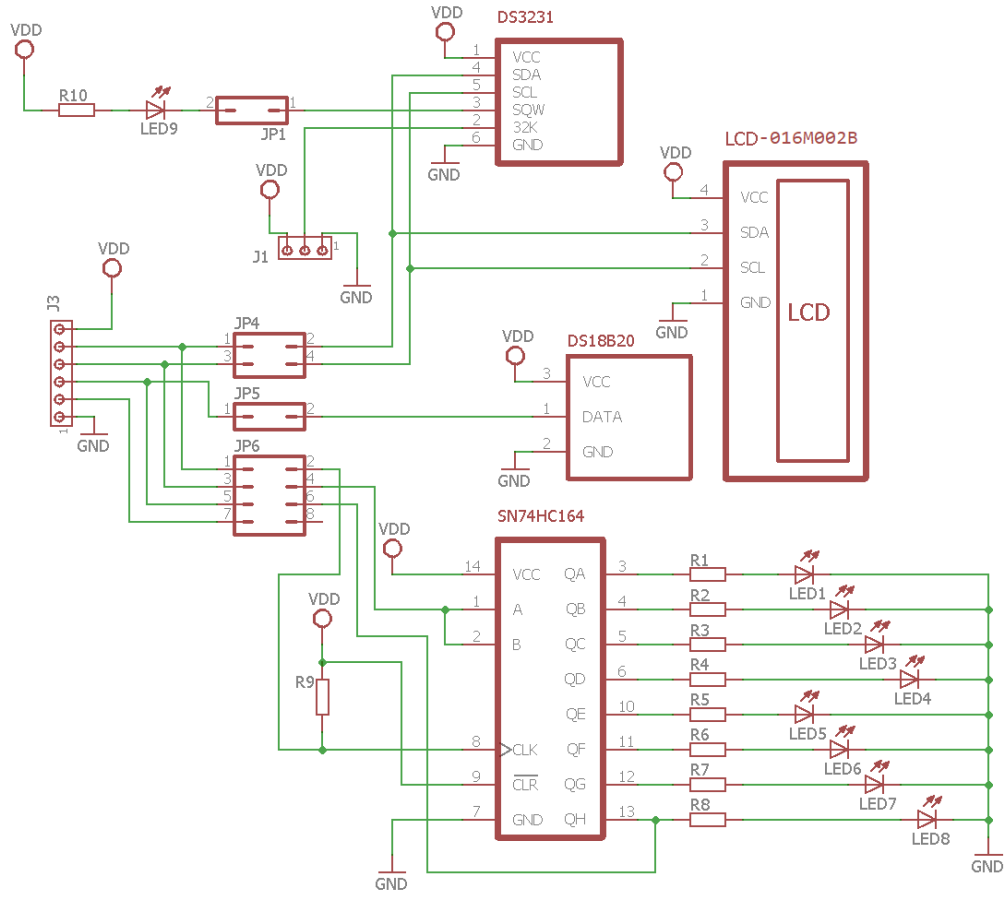


Obr. 6-1 Vstupy a výstupy posuvného registru (převzato z [10])

[12] Převzato z [10]

7. PŘIPOJENÍ PERIFERIÍ K VÝVOJOVÉMU SYSTÉMU

Tato kapitola pojednává o připojení signálů na vývojovou desku TWR-S08LH64 a hardwarové realizací zadaných úloh.

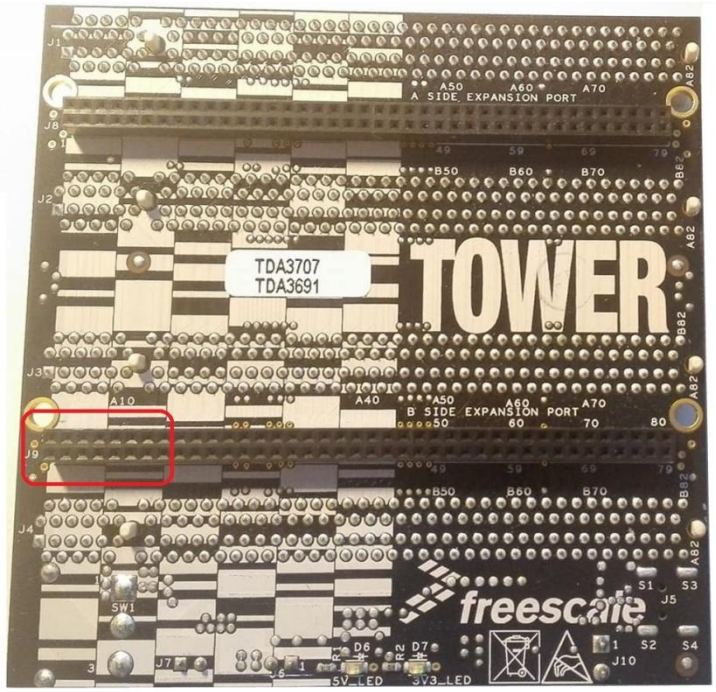


Obr. 7-1: Schéma připojení periferií k vývojovému systému

7.1 Signály mikrokontroléru

Vývojovou desku TWR-S08LH64 osazenou mikrokontrolérem MC9S08LH64 lze rozšířit modulem TOWER. Výstupy ovládané pomocí vnitřního modulu obsluhy sběrnice SPI a I2C lze odebírat pouze na rozšiřujícím portu modulu TOWER. Z toho důvodu je jeho připojení nezbytné při využití vnitřního modulu. Při použití softwarové obsluhy není připojení modulu TOWER nezbytně vyžadováno, protože díky možnosti definování výstupních pinů pro připojení dané sériové sběrnice lze teoreticky ke komunikaci využít jakékoliv dostupné piny.

Obr. 7-2 Základní vyvedení signálů vývojové desky



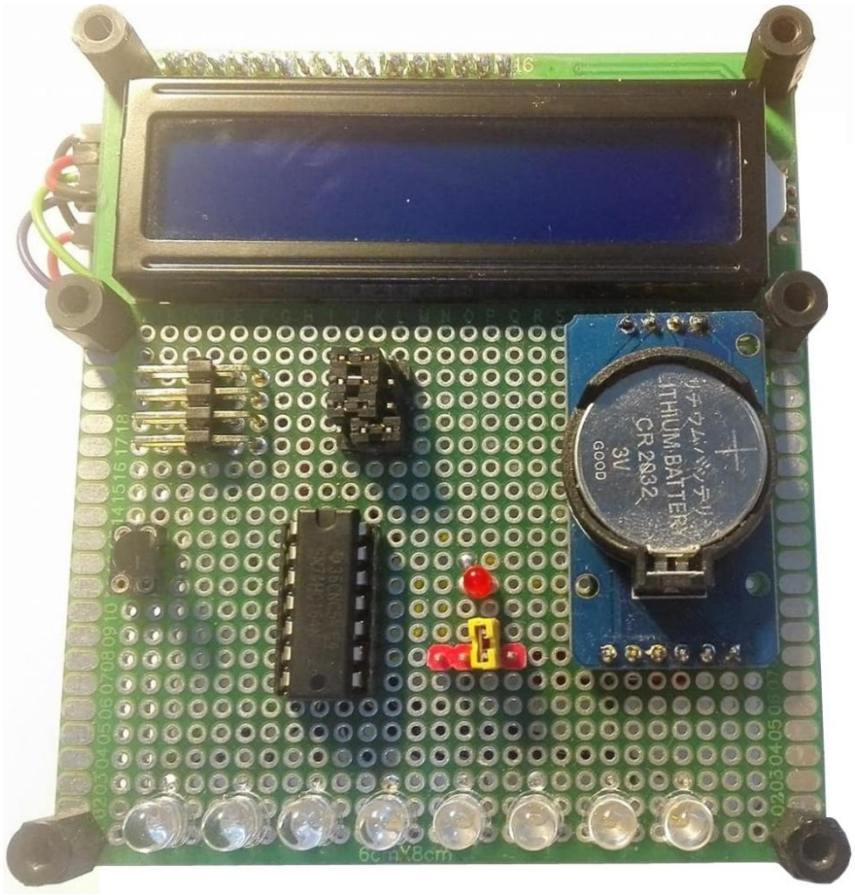
Obr. 7-3 Základní vyvedení signálů modulu TOWER

Z důvodu snadného přístupu k signálům jsem pro výchozí připojení signálů zvolil piny uvedené v následující tabulce. Napájecí a zemnící potenciál je připojen pomocí pinů J3-9 a J3-10.

Tab. 7-1 Připojení signálů

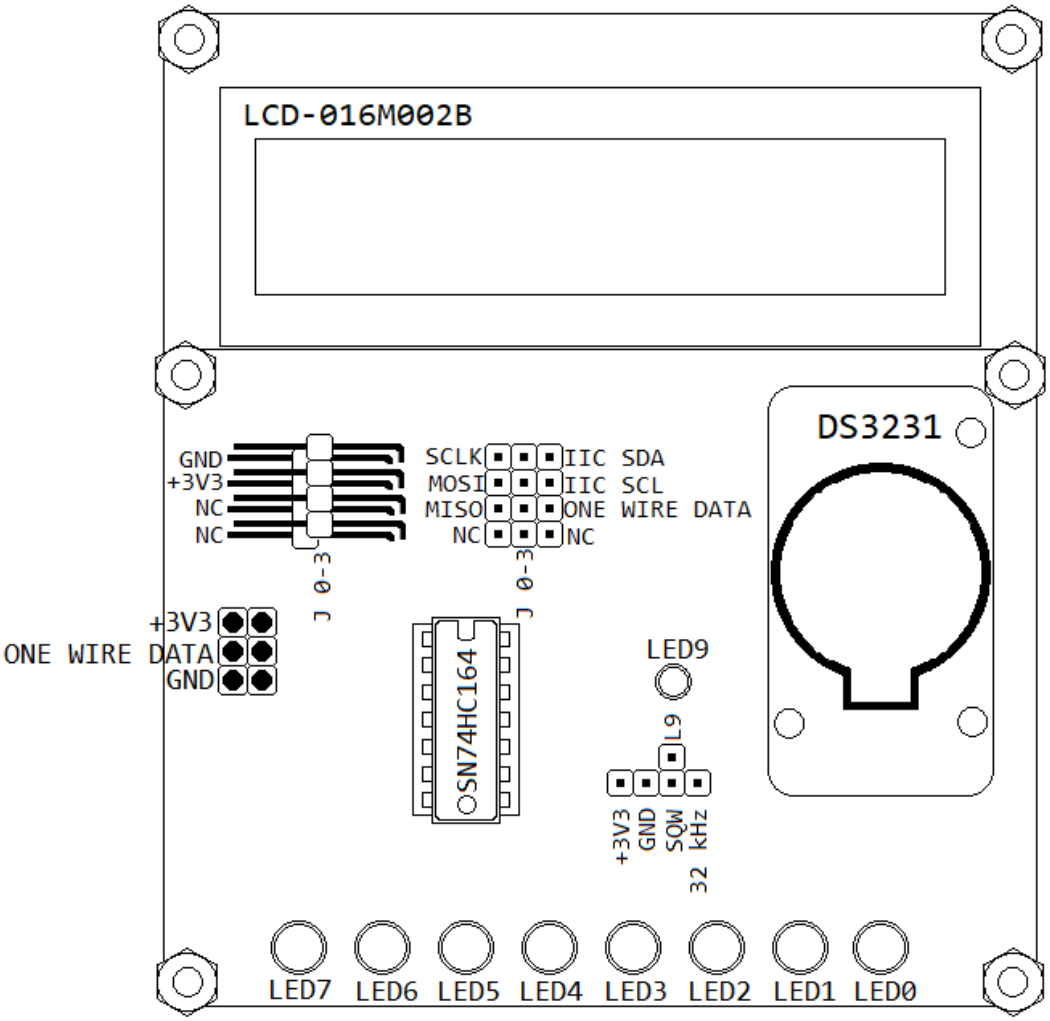
	Softwarová obsluha sběrnic	
	Výstupní pin mikrokontroléru	Výstupní pin vývojové desky
1-Wire data	PTC3	JP11-3
I2C SCL	PTC4	JP11-5
I2C SDA	PTC5	JP11-7
SPI SCLK	PTC5	JP11-7
SPI MOSI	PTC4	JP11-5
SPI MISO	PTC3	JP11-3
	Využití vnitřních modulů	
	Výstupní pin mikrokontroléru	Výstupní pin modulu TOWER
1-Wire data	-	-
I2C SCL	PTB5	J9-10
I2C SDA	PTB4	J9-11
SPI SCLK	PTB6	J9-7
SPI MOSI	PTB5	J9-10
SPI MISO	PTB4	J9-11

7.2 Hardwarové zapojení periferií



Obr. 7-4 Pájivé pole s externími obvody

Pro účely sjednocení všech zadaných úloh jsem vytvořil přípravek z univerzálního pájivého pole skládající se ze čtyř hlavních prvků. Komunikační signály k jednotlivým externím obvodům lze připojovat a odpojovat pomocí propojek. Vývody signálů jsou zobrazeny na následujícím obrázku.



Obr. 7-5 Rozmístění součástek a vyvedení signálů na pájivém poli

Deska obsahuje čidlo teploty DS18B20 umístěné v patici. Pro možnost připojení více čidel je deska vybavena další neobsazenou paticí. Vývody jsou připojené na napájecí a zemnící napětí, datový vodič je přiveden na pin ONE WIRE DATA, jenž je možno propojkou připojit na výstupní konektor desky.

Display LED-016M002B a kalendářový obvod DS3231 shodně komunikují po sběrnici I2C. Z toho důvodu jsou oba prvky připojeny na tuto sběrnici. Vývody obdélníkových signálů kalendářového obvodu jsou vyvedeny na piny SQW a 32 kHz. Pomocí propojky lze propojit pin SQW s pinem L9 a spínat tak LED diodu LED9 k potenciálu země pomocí kalendářového obvodu.

Posuvný registr SN74HC164 je propojen s piny SCLK, MOSI a MISO, pomocí kterých probíhá komunikace dle standardů SPI. Výstupy posuvného registru jsou přivedeny k LED diodám LED0 – LED7.

Výstupní 8-pinový konektor desky obsahuje dva nevyužité piny. Dále pak 4 piny J 0-3, které jsou vyvedeny na prostřední čtveřici propojovacích pinů. Pomocí propojek lze tedy na výstupní konektor připojit signály sběrnice SPI nebo současně sběrnic I2C a 1-Wire.

8. SOFTWAROVÁ REALIZACE

Tato kapitola popisuje softwarovou realizaci komunikace s externími obvody. Součástí kapitoly je popis knihoven sloužících ke komunikaci po sběrnicích 1-Wire, I2C a SPI. Dále následuje popis knihoven zprostředkovávajících komunikaci s obvody DS18B20, DS3231 a SN74HC164. Knihovny I2C a SPI umožňují softwarovou obsluhu sběrnice i využití vnitřních modulů SPI a I2C. Důvodem softwarové obsluhy těchto sběrnic je možnost využití knihovny v zařízeních bez vnitřních modulů a jednoduchá vizualizace složitosti těchto komunikačních protokolů při softwarovém ovládání binárních pinů.

8.1 Knihovna 1-Wire

Teplotní čidlo DS18B20 firmy Maxim, které je využito v této práci, komunikuje prostřednictvím sběrnice 1-Wire. Pro vedení komunikace s teplotním čidlem jsem vytvořil knihovnu, která využívá funkce připravené v knihovně 1WIREmaster.h. Funkce, jež jsou součástí této knihovny, jsou použitelné pro jakékoliv zařízení komunikující prostřednictvím sběrnice 1-Wire. Mikrokontrolér HCS08 neobsahuje vnitřní modul pro obsluhu komunikace, z toho důvodu jsou funkce knihovny obsluhovány pouze softwarově. V následující části jsou popsané funkce knihovny 1-Wire.

8.1.1 Inicializace komunikace

Inicializační funkce void ONEWIRE_Init(void) nastavuje pouze frekvenci mikrokontroléru na 40 MHz. Pro dosažení přesného časování datových signálů na sběrnici je tato rychlost je důležitou součástí obsluhy komunikace sběrnice 1-Wire.

8.1.2 Časování komunikace

Během komunikace prostřednictvím sběrnice 1-Wire není využíván žádný hodinový signál. Pro výměnu dat je tedy klíčové správné časování. Mikrokontrolér HCS08 neobsahuje knihovní funkce zprostředkující zpoždění. Z toho důvodu jsem vytvořil dvě funkce realizující definovatelné zpoždění při komunikaci. Funkce void ONEWIRE_Time1us(void) je pouze interní funkcí knihovny a obsahuje pouze assemblerovský příkaz NOP. Obsluha volání a samotné provedení funkce trvají při frekvenci mikrokontroléru 40 MHz přibližně 1,5 μs. Tento čas byl změřen pomocí osciloskopu. Samotná obsluha čítačů trvá déle než 1,5 μs, proto je nelze u takto

krátkého časování využívat. Tato funkce je volaná funkcí ONEWIRE_delay, dále pak funkcemi ONEWIRE_BitWrite a ONEWIRE_BitRead.

Funkce void ONEWIRE_delay(byte val) volá ve smyčce funkci ONEWIRE_Time1us. Počet cyklů smyčky je určen pomocí parametru funkce. Obsluha volání a samotná funkce s parametrem 1 trvá 46 cyklů mikrokontroléru. Funkce je v jiných knihovních funkcích využita pro tvoření zpoždění 8 až 500 µs.

8.1.3 Funkce Reset

Začátek komunikace na sběrnici 1-Wire je zahájen vyvoláním Reset pulsu mikrokontrolérem, jež je následován Presence pulsem generovaným zařízením slave. Pro vyvolání Reset pulsu slouží funkce byte ONEWIRE_Reset(void). Funkce stáhne sběrnici do nízké logické úrovně a setrvá v ní 500 µs. Poté sběrnici uvolní a čeká 100 µs. V tomto čase zařízení slave odpovídá, proto v tomto čase dochází ke čtení stavu sběrnice. Návratovou hodnotou je presence pulse zařízení. Pokud zařízení odpovědělo, vrací funkce log. 0, v opačném případě log.1.

8.1.4 Zápis bitu

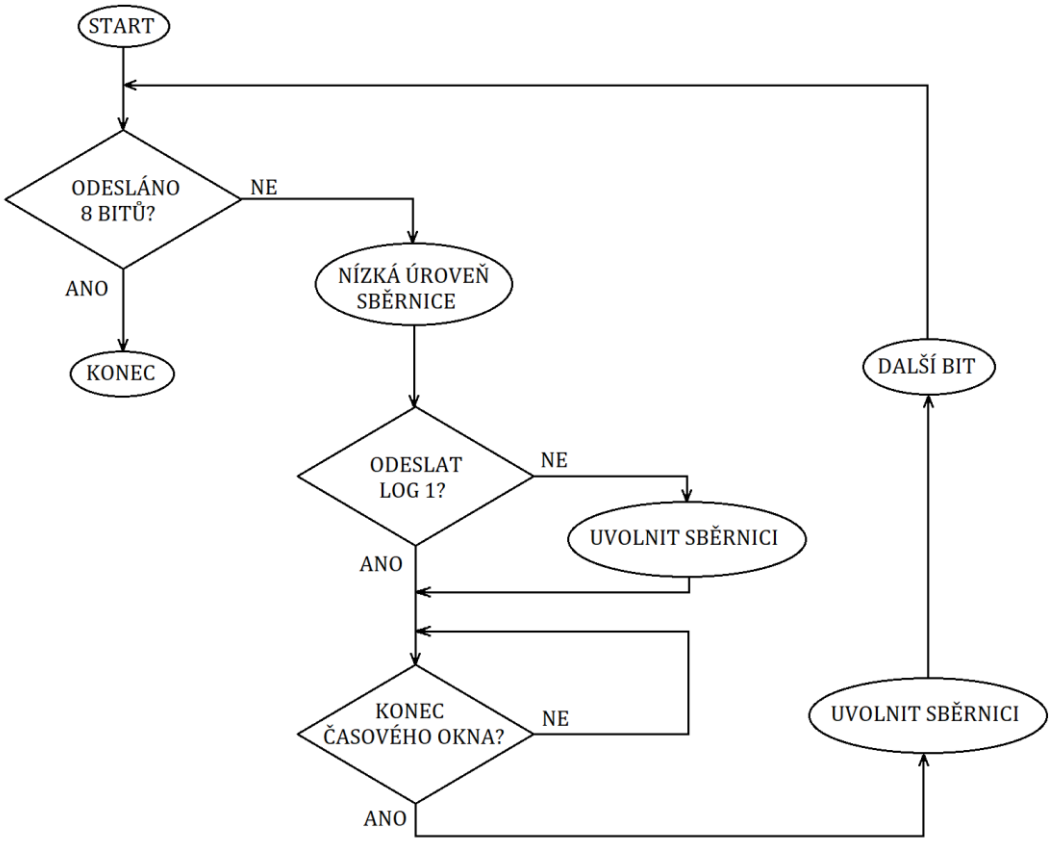
Zápis jednoho bitu je základem výměny dat se zařízením slave. Samotný bit je umístěn v časovém okně o předepsané délce popsané v teoretickém rozboru 1-Wire sběrnice. Mikrokontrolér zahajuje časové okno stáhnutím sběrnice do nízké logické úrovně po dobu 1,5 µs. Po této časové prodlevě sběrnici uvolní, pokud má být zapisována log. 1. V opačném případě sběrnici neuvolní a ta setrvává v nízké logické úrovni. Následuje časová prodleva 80 µs. V této chvíli nastává konec časového okna a mikrokontrolér sběrnici uvolní. Funkce void ONEWIRE_BitWrite(byte bit) provádí zápis jednoho bitu, jenž je jejím parametrem.

8.1.5 Čtení bitu

Stejně jako zápis bitu i čtení bitu je základem výměny dat se zařízením slave. Bit je opět umístěn v časovém okně předepsaných rozměrů. Na začátku komunikace stáhne mikrokontrolér sběrnici do nízké logické úrovně. Tím vyvolá začátek časového okna. V nízké logické úrovni setrvává 1,5 µs, poté sběrnici uvolní a čeká 8 µs. V tomto čase je zařízením slave na sběrnici vystavena hodnota přenášeného bitu. Mikrokontrolér čte sběrnici, a pokud má vysokou logickou úroveň, ukládá do paměti log. 1. V opačném případě ukládá log. 0. Následuje prodleva 90 µs do konce časového okna. Funkce byte ONEWIRE_BitRead(void) vykonává čtení jednoho bitu. Návratovou hodnou je přečtený bit.

8.1.6 Zápis bytu

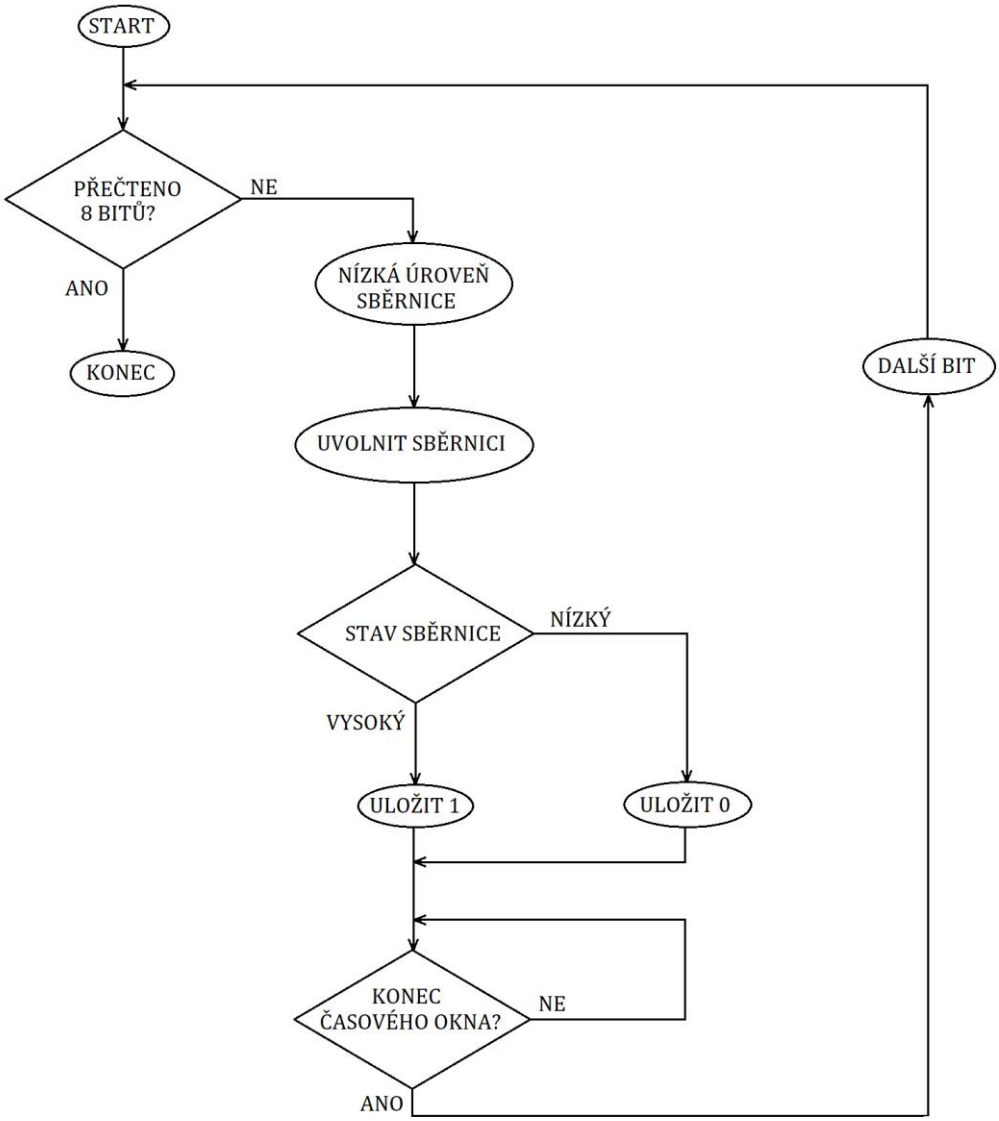
Funkce void ONEWIRE_ByteWrite(byte data) realizuje zápis bytu předaného v parametru funkce. Pro samotný zápis se využívá funkce ONEWIRE_BitWrite, která je volaná 8x ve smyčce. V průběhu smyčky je třeba provádět logickou operaci shift pro přípravu následujícího bitu, aby došlo vždy k odeslání požadovaného bitu. První se zasílá LSB, poslední MSB.



Obr. 8-1 Vývojový diagram - odesílání bytu 1-Wire

8.1.7 Čtení bytu

Funkce byte ONEWIRE_ByteRead(void) realizuje čtení bytu. Pro vyčítání jednoho bytu dat využívá funkci ONEWIRE_BitRead, která je volána 8x ve smyčce. V průběhu smyčky je využito logické operace posun bitů v přijímaném bytu, aby byl vyčtený bit přiřazen na správnou pozici v rámci přijímaného bytu. Prvním vyčteným bitem je LSB, posledním MSB.



Obr. 8-2 Vývojový diagram - přijímání bytu 1-Wire

8.2 **Knihovna I2C**

Z důvodu realizace komunikace s obvody, jež využívají sběrnici I2C je vhodné využívat knihovnu, která uskutečňuje samotnou komunikaci podle standardů I2C. Mikrokontrolér HCS08 firmy NXP obsahuje vnitřní hardwarový modul komunikace po tomto typu sběrnice. Při použití tohoto modulu lze ovšem využívat jen piny pro komunikaci určené. Proto jsem vytvořil i softwarovou obsluhu komunikace I2C. Díky

tomu je možno využít pro komunikaci jakékoli vstup-výstupní piny, které lze ovládat uživatelem. Přepínání mezi softwarovou obsluhou a obsluhou pomocí vnitřního modulu se provádí v hlavičkovém souboru I2Cmaster.h. Pokud je uživatelem definováno IICHW, je využit vnitřní modul I2C, při definování IICSW se pak využívá softwarová obsluha sběrnic.

Samotná knihovna obsahuje pouze základní funkce sloužící pro vedení komunikace. V následující části jsou jednotlivé funkce popsány s ohledem na softwarovou obsluhu nebo obsluhu pomocí vnitřního modulu.

8.2.1 Inicializace komunikace

Pro inicializaci komunikace slouží funkce void I2C_init(void). Při definování softwarové obsluhy funkce pouze nastavuje frekvenci mikrokontroléru. Jelikož je softwarová obsluha sběrnice pomalejší než obsluha pomocí vnitřního modulu, je vhodné využít frekvenci 40 MHz.

Pokud je definována obsluha komunikace pomocí vnitřního modulu, nastavuje funkce výstupní piny, baudrate a připojuje hodinový signál do I2C modulu. Na začátku inicializace je vhodné modul zakázat a opět povolit, tím dojde ke smazání dat, která se vyskytují v registrech modulu a k nastavení výchozích hodnot.

8.2.2 Začátek komunikace

Podle standardů I2C začíná komunikace příkazem START, který je definován jako sestupná hrana na datovém vodiči v okamžiku, kdy má vodič s hodinovým signálem vysokou logickou úroveň, po tomto příkazu následuje adresa zařízení, se kterým bude vedena komunikace. Tato posloupnost je vždy stejná. Z toho důvodu jsem spojil tyto dva úkony do jedné funkce. Knihovní funkce int I2C_start(byte address) vyvolá příkaz START následovaný adresou zařízení, která je funkci předána jako parametr. Návratovou hodnou funkce je ACK bit.

Při využití vnitřního modulu je příkaz START proveden nastavením bitu MST v registru IICC1. Následné vyslání adresy zařízení se vykoná jejím vložením do registru IICD. Vložením dat do registru IICD, v okamžiku kdy je bit TX registru IICC1 nastaven, je zahájeno jejich vysílání. Odeslání celého bytu je signalizováno nastavením bitu IICIF registru IICS.

Začátek komunikace pomocí softwarové obsluhy vyžaduje manuální provedení zmiňovaného úkonu. Jelikož má hodinový vodič SCL vysokou logickou úroveň v okamžiku, kdy se nevyužívá, stačí pro příkaz START stáhnout datový vodič SDA na nízkou logickou úroveň. Pro následné vyslání adresy se využívá funkce I2C_write popsána níže.

8.2.3 Opětovný začátek komunikace

V průběhu komunikace se slave obvody je třeba vykonávat opětovný začátek komunikace. Tento řídicí příkaz je definován stejně jako příkaz START. Na rozdíl od příkazu START, který se může objevovat až po příkazu STOP, se příkaz REPEAT START vysílá v průběhu komunikace mezi příkazy START a STOP. Značí opětovný začátek komunikace v okamžiku, kdy předchozí komunikace ještě neskončila. Využívá se ve chvílích, kdy je třeba opakovaně obvodu slave zaslat adresu, která má rozdílný poslední bit. Tímto způsobem lze změnit směr toku dat. Jelikož je příkaz REPEAT START vždy následován adresou zařízení slave, jsou opět oba úkony sloučeny do jediné funkce int I2C_rep_start(byte address), které se adresa volaného zařízení předává pomocí parametru. Návratovou hodnotou této funkce je ACK bit.

Při využití vnitřního modulu I2C se příkaz REPEAT START provádí nastavením bitu RSTA v registru IICC1. Zároveň je potřeba nastavit mód na vysílání, to se provede nastavením bitu TX registru IICC1. Vyslání adresy zařízení je inicializováno zapsáním příslušné hodnoty do registru IICD.

U softwarové obsluhy je třeba provést příkaz START. Jelikož není zřejmé, jestli se na vodiči SDA vyskytuje vysoká logická úroveň, je třeba ji zajistit přepnutím pinu SDA na vstup. Aby nebyl vytvořen příkaz STOP, je napřed potřeba stáhnout vodič SCL do nízké logické úrovně, poté můžeme uvolnit vodič SDA. Pull-up rezistor zajistí vysokou logickou úroveň na vodiči. Následně je potřeba stejným způsobem uvolnit vodič SCL. V této chvíli lze vodič SDA stáhnout do nízké logické úrovně pro vytvoření příkazu REPEAT START. Následné odeslání adresy slave zařízení je provedeno pomocí funkce I2C_write, která je popsána níže.

8.2.4 Ukončení komunikace

Standardy protokolu I2C vyžadují ukončení komunikace příkazem STOP. Ten je definován jako náběžné hrana na datovém vodiči SDA v okamžiku, kdy má hodinový vodič SCL vysokou logickou úroveň. Po ukončení komunikace přestává původně oslovené zařízení slave reagovat, dokud nedojde k příkazu START následovaném jeho adresou. Ukončení komunikace se provádí pomocí volání funkce void I2C_stop(void).

Generování signálu STOP se při využití vnitřního modulu provádí pomocí zapsání 0 do bitu MST v registru IICC1. Tímto úkonem je automaticky generován STOP signál.

Složitější je generování STOP signálu pomocí softwarové obsluhy. Zde je třeba provést náběžnou hranu signálu SDA v okamžiku popsaném výše. Nejprve je zapotřebí dostat nízkou logickou úroveň na vodiči SCL. Poté co je nastavena, je možné do nízké

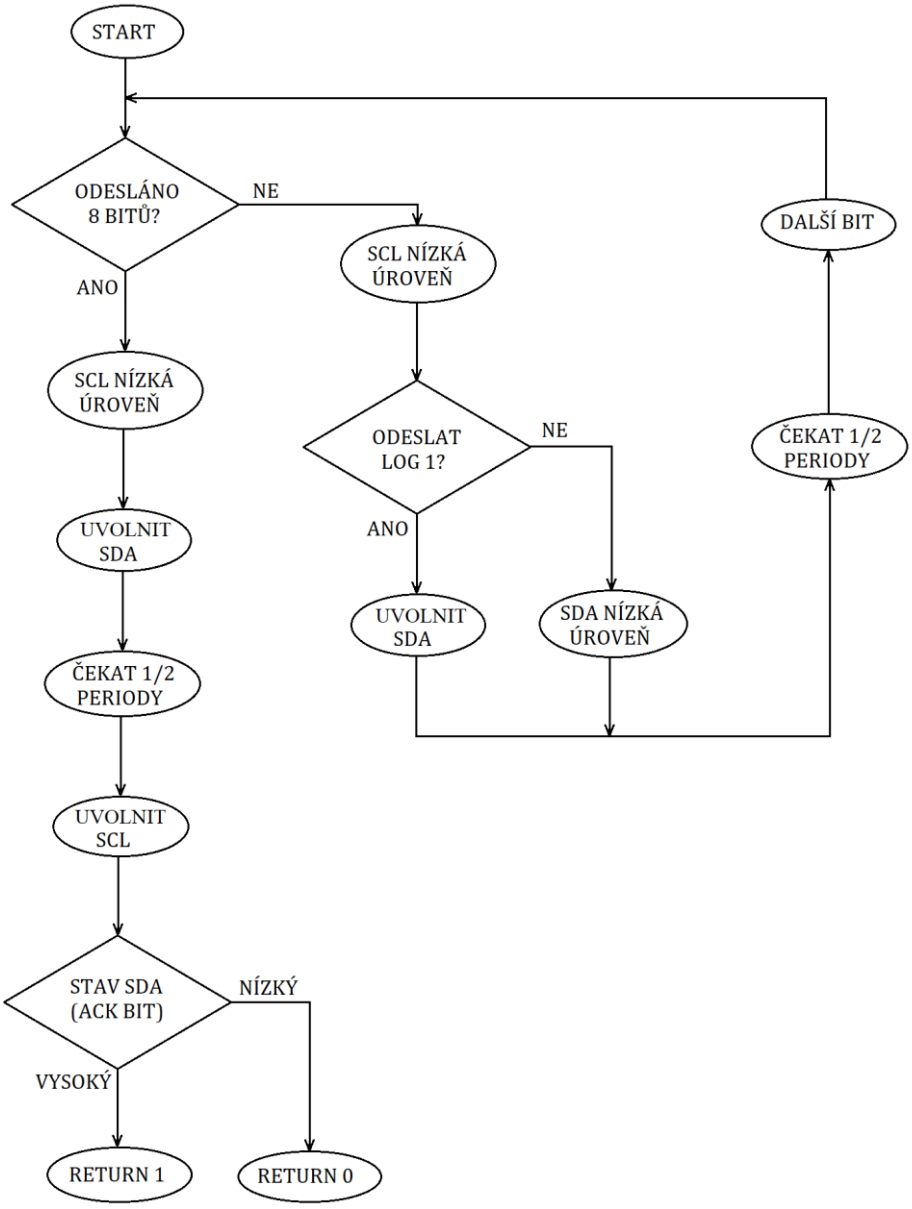
logické úrovně stáhnout i datový vodič SDA. Pokud by v tomto okamžiku nebyla na vodiči SCL nízká logická úroveň, způsobila by změna stavu na vodiči SDA nežádoucí příkaz START. Následně stačí uvolnit vodič SCL, pull-up odpor vyvolá vysokou logickou úroveň. Nyní je nutno uvolnit vodič SDA. Pull-up odpor i zde způsobí zvýšení logické úrovně, to vyvolá vzestupnou hranu potřebnou pro generování příkazu STOP.

8.2.5 Zápis bytu

Během komunikace je potřeba zapisovat byty do slave zařízení. Tento zápis musí probíhat podle standardů komunikace I2C. Zápis je opět možno provádět pomocí softwarové obsluhy komunikace nebo vnitřního modulu. Zapsání bytu se provádí pomocí funkce int I2C_write(byte data). Parametrem této funkce jsou data k zápisu, návratovou hodnotou je ACK bit zařízení slave.

Při využití vnitřního modulu je zapotřebí definovat, zda se bude provádět zápis nebo čtení dat. Tato informace je uložena v bitu TX registru IICC1. Pro zápis dat se bit TX nastavuje. Po nastavení bitu TX se ukládají data do registru IICD. Uložením dat do tohoto registru, v okamžiku kdy je TX nastaven, je inicializován přenos. Dokončení přenosu je signalizováno nastavením bitu IICIF registru IICS. Po dokončení odeslání bytu dat je zařízením slave vystaven ACK nebo NACK bit. Tento bit je mikrokontrolérem uložen do bitu RXAK v registru IICS, odkud je možné bit číst.

Pokud je komunikace ovládána softwarově, je třeba generovat hodinový signál a ve správných okamžicích vystavovat na sběrnici jednotlivé bity zasílaného bytu. Pro odeslání jednoho bitu je potřeba stáhnout vodič SCL do nízké logické úrovně a následně stáhnout nebo uvolnit vodič SDA pro vystavení požadovaného bitu. Po vystavení následuje zpoždění času o půl periody signálu SCL a uvolnění vodiče SCL. V tomto okamžiku zařízení slave čte stav sběrnice. Pro správné časování hodinového signálu následuje další zpoždění času o půl periody. Tímto způsobem se ve smyčce odesílá všech 8 bitů. Po posledním bitu odesílaného bytu odpovídá zařízení slave devátým bitem ACK. Z toho důvodu je třeba generovat devátou periodu signálu SCL a ve vhodném okamžiku číst stav sběrnice SDA.



Obr. 8-3 Vývojový diagram - zápis bytu I2C

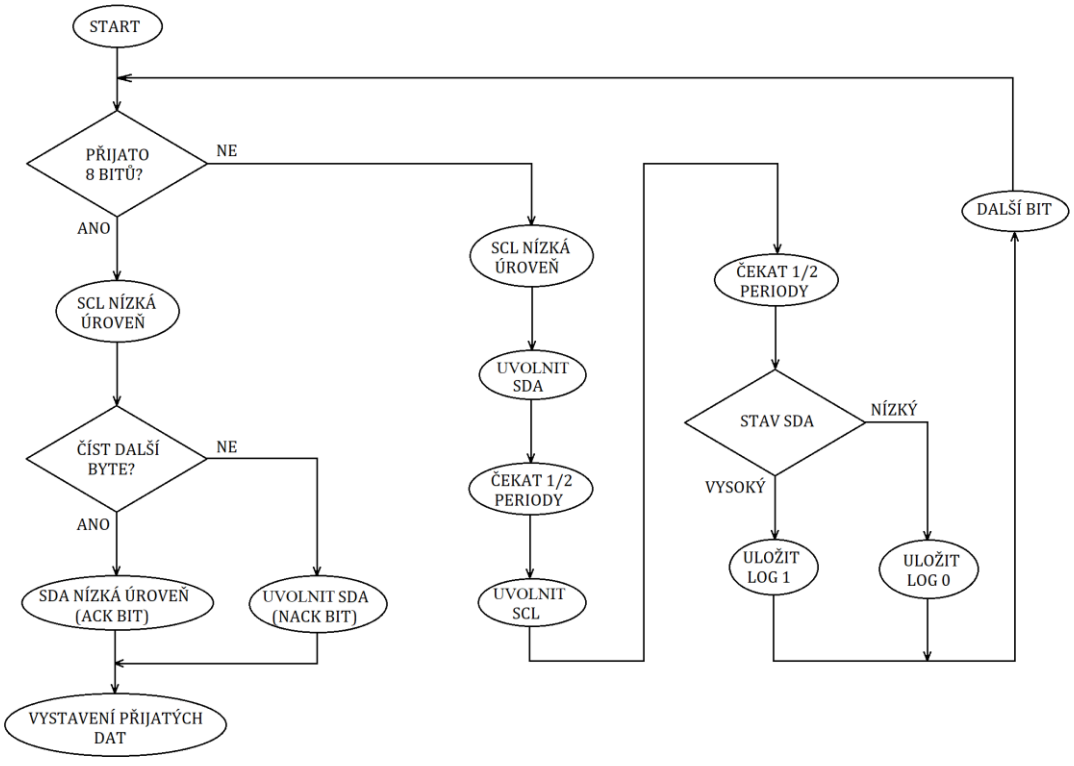
8.2.6 Čtení bytu

U čtení bitu přenášeného ze slave zařízení mohou nastat dvě rozdílné situace. Jednou z těchto situací je vyčítání bytu, za kterým následuje ukončení komunikace, druhou je potom vyčítání bytu, za kterým se pokračuje ve vyčítání dalších bytů. Tyto dvě situace se liší pouze v odesílaném devátém bitu. Pokud je jako devátý bit odeslán NACK, komunikace končí. V případě odeslání bitu ACK komunikace pokračuje. Z toho

důvodu obsahuje knihovna dvě funkce, které řeší tyto dvě situace a liší se pouze v devátém bitu. Pokud má být provedeno odeslání ACK bitu, využívá se funkce byte I2C_readCONTINUE(void), v případě odeslání NACK bitu se využívá funkce byte I2C_readEND(void). Návratovou hodnotou těchto funkcí je přečtený byte.

Čtení bytu se při využití vnitřního modulu provádí opět pomocí registru IICD. Před samotným čtením je nutné resetovat bit TX registru IICC1. Resetování bitu značí režim čtení. Nastavením nebo resetováním bitu TXAK registru IICC1 je určeno, zda bude po přečtení bytu vystaven na sběrnici bit ACK nebo NACK. Vyslání tohoto bitu je provedeno automaticky po dokončení čtení. V tomto případě resetování bitu TXAK značí zaslání ACK bitu. Po zapsání těchto dvou bitů se pokračuje v samotném přijímání dat. Přijímání je inicializováno „falešným“ přečtením registru IICD. Jakmile vnitřní modul dokončí čtení jednoho bytu, je nastaven bit IICIF registru IICS. Před přečtením dat z registru IICD je potřeba nastavit resetovaný bit TX. Pokud by se tak nestalo, čtením registru by došlo k zahájení čtení dalšího bytu. Na závěr funkce je resetován flag IICIF zapsáním log.1.

U softwarové obsluhy komunikace se čtení bytu provádí podobným způsobem jako zápis bytu s tím rozdílem, že zápis se neprovádí a vodič SDA se vzorkuje ve chvílích vysoké logické úrovně vodiče SCL. Po přijmutí 8 bitů se na vodič SDA vystavuje bit ACK nebo NACK stažením nebo uvolněním vodiče SDA podle použité knihovní funkce.



Obr. 8-4 Vývojový diagram - čtení bytu I2C

8.3 Knihovna SPI

Součástí řešení zadaného úkolu je komunikace s posuvným registrem. Ten komunikuje prostřednictvím sběrnice SPI. Pro jednodušší vedení komunikace jsem vytvořil knihovnu pro daný posuvný registr, která využívá funkce této knihovny. Součástí této knihovny jsou pouze základní dvě funkce sloužící k vedení komunikace podle standardů SPI. Jelikož mikrokontrolér HCS08 firmy NXP obsahuje vnitřní modul SPI komunikace, tento modul je využit pro funkce knihovny. Signály, jež vznikají při použití vnitřního modulu SPI, jsou vyvedeny na piny, jejichž pozice je pevně dána a nedají se tak využívat jiné volné vývody mikrokontroléru. I z toho důvodu je součástí knihovny softwarová obsluha komunikace SPI, která řeší tento problém. Přepínání mezi obsluhou pomocí vnitřního modulu a softwarovou obsluhou komunikace se provádí v hlavičkovém souboru SPImaster.h pomocí definování SPIHW pro použití vnitřního modulu SPI, nebo SPISW pro použití softwarové obsluhy.

8.3.1 Inicializace knihovny

Před zahájením samotné komunikace po sběrnici SPI, je potřeba provést prvotní inicializaci. Pro ni se využívá funkce void SPI_Init(void), která nastavuje mikrokontrolér. Při využití softwarové obsluhy se v inicializační funkci nastavují pouze hodnoty signálů do klidového stavu, kdy neprobíhá komunikace.

Pokud je uživatelem nastaveno použití vnitřního modulu SPI, nastavuje funkce požadované výstupy signálů, dále pak prvky, jež jsou u komunikace SPI nastavitelné, jako například baudrate, polaritu hodin, fázi hodin nebo první odesílaný bit. Pro dosažení vyšší přenosové rychlosti je také během inicializace přepnuta frekvence mikrokontroléru na 40 MHz.

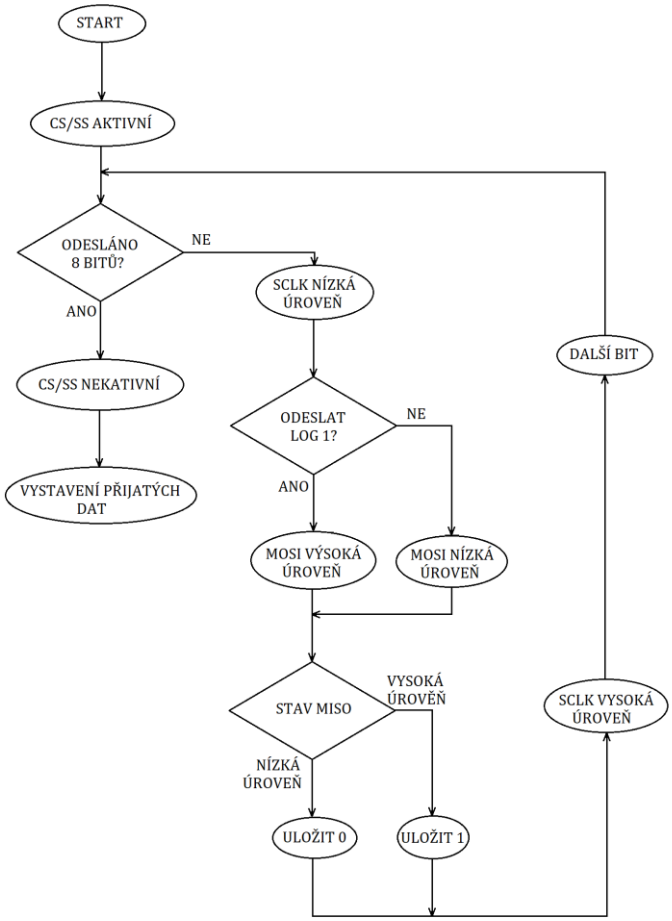
8.3.2 Čtení a zápis jednoho bytu

Dle standardů komunikace SPI jsou ke komunikaci využívány čtyři signály. Na dvou z těchto čtyř signálů se vyskytují přenášená data ve směru od mikrokontroléru do zařízení slave a od zařízení slave do mikrokontroléru. Jednosměrný tok dat na dvou vodičích má za následek výskyt dvou rozdílných dat přenášených v jednom okamžiku. Kvůli této vlastnosti protokolu SPI je v knihovně vytvořená jediná funkce, která zároveň zapisuje a čte data. Knihovní funkce byte SPI_1byteRW(byte write) zapisuje byte předaný v parametru funkce do zařízení slave a vrací byte vyčtený ze zařízení.

Při využití vnitřního modulu je nutné číst nastavený bit SPTEF registru SPIS. Pokud by došlo k zapsání dat určených k odeslání před přečtením 1 uložené v bitu SPTEF, zapsaná data by byla ignorována. Tento příznakový bit je resetován ve chvíli, kdy je přečten s hodnotou 1. Po přečtení nastaveného bitu SPTEF dochází k uložení dat do registru SPID. Uložením dat je zahájen jejich přenos, zároveň jsou přijímána data vysílaná zařízením slave. Tato data se ukládají také do registru SPID. Jakmile je přenos dokončen, je nastaven příznakový bit SPRF v registru SPIS. Tento bit je resetován, pokud je čten jako nastavený. Přijatá data jsou přečtena a funkce je vrací jako návratovou hodnotu.

Pokud je využita softwarová obsluha sběrnice, je potřeba generovat hodinový signál na vodiči SCLK. Odeslání jednoho bitu je zahájeno stažením signálu SCLK do nízké logické úrovně. Poté je signál MOSI vytažen do vysoké logické úrovně, pokud se odesílá log. 1. V opačném případě je signál MOSI stažen do nízké logické úrovně. Po nastavení úrovně signálu MOSI je odečten stav signálu MISO a uložen. Odesílání jednoho bitu končí vytažením signálu SCLK do vysoké logické úrovně. Odesílání bytu probíhá pomocí odesílání bitu ve smyčce.

Softwarová obsluha sběrnice řeší pouze případ, kdy je polarita hodin i fáze hodin 1 a prvním odesílaným bitem je MSB. Toto nastavení komunikace je vhodné pro posuvný registr SN74HC164.



Obr. 8-5 Vývojový diagram - čtení a zápis bytu SPI

8.4 Knihovna DS18B20

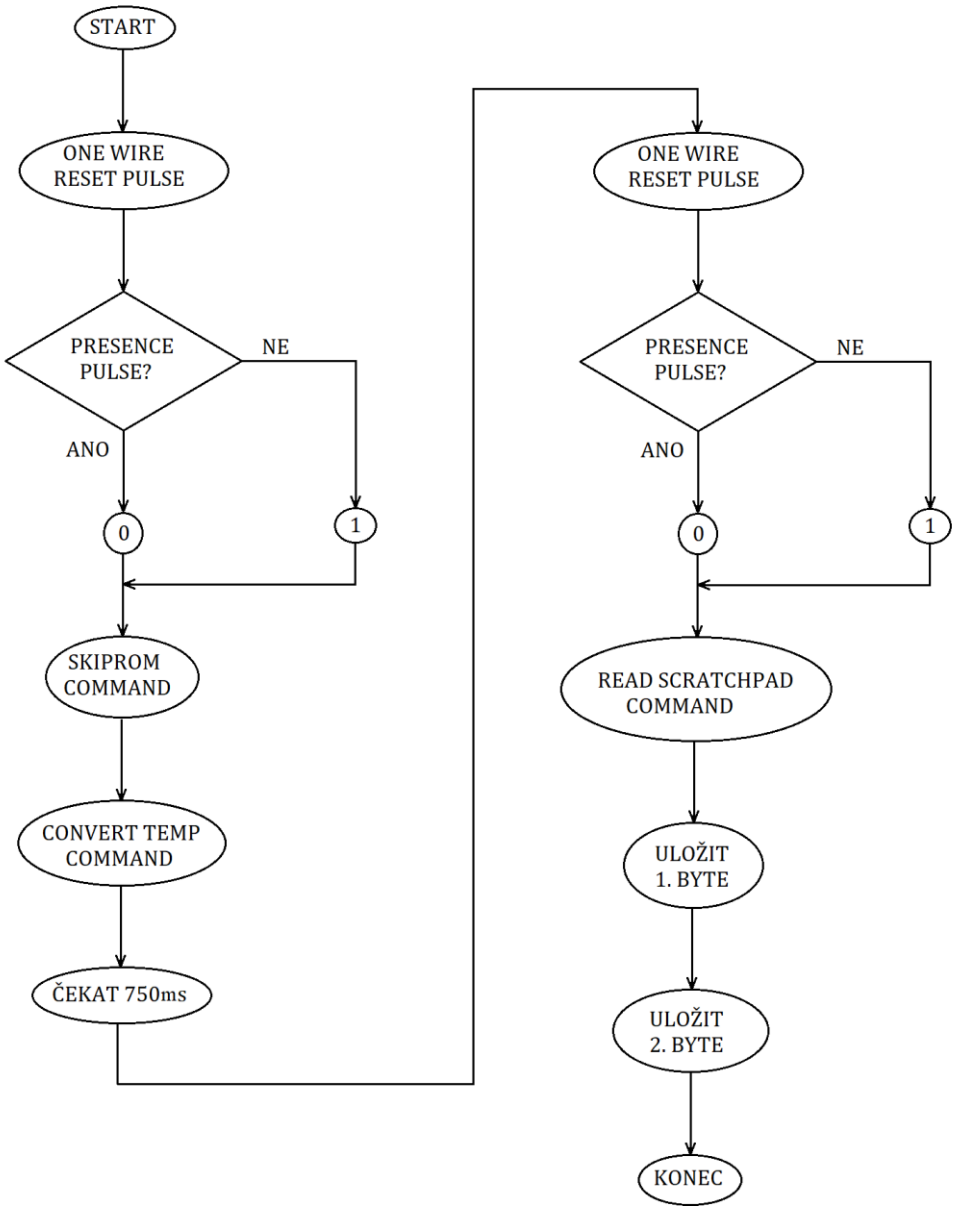
Knihovna DS18B20_ONEWIRE.h obsahuje základní funkce pro vedení komunikace s jedním teplotním čidlem. Funkce zprostředkovávají vyčítání teplotních dat z čidla ve formě uložené v teplotním čidle, vyčítání teplotních dat a následnou konverzi a vyčítání adresy čidla z paměti ROM.

8.4.1 Čtení teploty

Funkce void DS18B20_ReadTemperature(byte *raw0, byte *raw1) slouží k vyčtení dat teploty uložených v teplotním čidle. Následně se využívá ve funkci DS18B20_ReadAndConvertTemperature popsané níže. Parametrem funkce jsou adresy dvou bytových proměnných, do kterých je uložena vyčtená teplota. Do první proměnné je uložena nepřepočítaná hodnota prvního bytu, do druhé nepřepočítaná hodnota druhého bytu teploty. Jelikož je komunikace s teplotními čidly DS18B20 shodná

s komunikací se staršími čidly DS1820, lze i u nich využít tuto funkci. Následný přepočet dat získaných z čidla DS1820 je odlišný a není součástí knihovních funkcí knihovny DS18B20.

Čtení teploty je zahájeno voláním funkce ONEWIRE_Reset, která realizuje resetovací puls. Následuje ROM příkaz SKIP ROM 0xCC, který lze využít, jelikož je připojeno pouze jedno teplotní čidlo. Po zapsání následuje příkaz CONVERT TEMPERATURE, který má podobu čísla 0x44. Příkaz spustí měření teploty v teplotním čidle. Výrobce udává maximální dobu měření 750 ms. Po uběhnutí této časové prodlevy, pro kterou se využívá funkce ONEWIRE_delay, následuje opět volání funkce ONEWIRE_Reset. Nyní je teplota již konvertována a uložena v paměti teplotního čidla. Následným příkazem SKIP ROM je opět přeskočeno adresování čidla a příkazem READ SCRATCHPAD 0xBE je zahájeno vyčítání dvoubytové hodnoty, která je ukládána na adresy proměnných vložených jako parametr funkce. Zapsání všech příkazů je prováděno pomocí funkce ONEWIRE_ByteWrite, která je součástí knihovny 1WIREmaster.



Obr. 8-6 Vývojový diagram - vyčtení teploty z DS18B20

8.4.2 Čtení a konverze teploty

Funkce void DS18B20_ReadAndConvertTemperature(int *whole, int *decimal) slouží k vyčtení teploty z čidla DS18B20 a následné konverzi čísla. Funkce je navržena pro přepočet teploty u čidla, jež měří pokojovou teplotu. Z toho důvodu se nepředpokládá měření teplot pod bodem mrazu a funkce proto neprovádí konverzi čísla ze záporných hodnot teploty.

Na začátku běhu funkce je volána funkce DS18B20_ReadTemperature popsaná výše. Navrácené hodnoty jsou uloženy a dál je prováděna konverze hodnot. Podoba

navrácených dat je uvedena v technické dokumentaci teplotního čidla, z toho důvodu se zde nezabývám samotným přepočtem získaných hodnot.

8.4.3 Čtení jedinečné adresy čidla

Zařízení připojená ke sběrnici 1-Wire mají svá jedinečná čísla. Ta slouží k adresaci konkrétního slave zařízení v případě, kdy je na tuto sběrnici připojen více než jeden slave. Snímač teploty DS18B20 má své jedinečné číslo uložené v ROM paměti. Jedná se o 64-bitovou adresu, kde prvních 8 bitů tvoří hodnota 0x28, kód rodiny DS18B20, poté následuje 48 bitů obsahujících unikátní sériové číslo a 8 bitů kontrolní součet CRC. Funkce slouží ke zjištění ROM čísla čidla, které by mohlo být následně použito při realizaci komunikace s více čidly.

Vyčítání dat ze snímače probíhá prostřednictvím ROM příkazu Read ROM, který následuje po volání funkce ONEWIRE_Reset. Příkaz Read ROM, reprezentovaný hodnotou 0x33, lze použít pouze pokud je na sběrnici přítomno jen jedno teplotní čidlo. V případě připojení více čidel by docházelo ke kolizi dat. Ta by byla způsobena odpověďmi všech čidel ve stejném okamžiku. Pro vyčítání adresy z paměti ROM teplotního čidla se 8x ve smyčce volá funkce ONEWIRE_ByteRead. Získaná adresa čidla je uložena do pole bytů. Adresa prvního prvku pole je předávána funkci void DS18B20_ReadROM(byte *rom) jako parametr.

8.4.4 Zadání úlohy k otestování knihovny

Zobrazte měřenou teplotu a adresu teplotního čidla DS18B20 komunikujícího po sběrnici 1-Wire na LCD displej přípravku komunikujícího po sběrnici I2C. Pro vyčítání teploty, adresy a následné zobrazení na displeji prostudujte a využijte připravené knihovny. Ukládejte do paměti minimální a maximální hodnotu teploty.

Přepínání mezi maximem, minimem a aktuální teplotou realizujte pomocí tlačítka, které vyvolává přerušení. Pomocí druhého tlačítka provádějte vymazání minimální a maximální uložené teploty.

Aktuální nebo minimální nebo maximální teplotu zobrazte na prvním řádku LCD displeje, na druhém řádku zobrazte unikátní adresu teplotního čidla.

Po sekundě posunujte zobrazovaný text na 2. řádku displeje. Pro generování 1 s využijte modul TOD.

8.5 Knihovna DS3231

Pro zjednodušení práce s kalendářovým obvodem jsem vytvořil knihovnu DS3231_I2C.h. Jelikož obvod komunikuje po sběrnici I2C, byly k realizaci komunikačních rutin využity funkce knihovny I2Cmaster.h. Knihovna obsahuje funkce pro zapsání času a data, pro vyčtení času a data, připínání a odepínání obdélníkového signálu na výstupní pin obvodu. V následující části jsou popsány způsoby vyčítání a zápisu dat, jež jsou využity v samotných knihovních funkcích

8.5.1 Nastavení bytu

Pro zapsání například časového údaje minuty je potřeba na správnou adresu kalendářového obvodu zaslat data ve správném formátu. Formát uložených dat se pro různé údaje mírně odlišuje, většina z nich je ovšem uložena ve formátu BCD. Zapsání jednoho bytu je využito například v knihovní funkci DS3231_SetDay, která uloží do paměti kalendářového obvodu zadaný den v týdnu. Pro správné zapsání jednoho bytu na určitou adresu se postupuje následujícím způsobem:

1) Začátek komunikace obsahující příkaz START
2) Vystavení 7-bitové adresy zařízení následované 0 pro zápis dat.
3) Vyslání adresy registru, ve kterém jsou uložena žádaná data. Vysláním adresy dojde k posunu adresového pointeru na žádanou pozici.
4) Vyslání bytu ve správném formátu, který bude uložen v dosaženém registru. Zařízení odpovídá vystavení ACK bitu.
5) Ukončení komunikace pomocí příkazu STOP.

8.5.2 Nastavení 2 a více bytů

Kalendářová data jsou uložena v registrech obvodu za sebou, pro využití co nejméně paměti jsou některá z těchto dat rozdělena do více registrů. Z tohoto důvodu je v různých případech vhodné zapisovat do paměti kalendářového obvodu více než jeden byte dat. Zapsáním více bytů dat v jednom bloku je také možné nastavit různá data bez nutnosti opakovaného adresování kalendářového obvodu. Zápisu více než jednoho bytu dat je využito v knihovní funkci DS3231_SetTime, která zapíše do paměti kalendářového obvodu 3 datové byty, a to sekundy, minuty a hodiny. Zapsání bloku bytů se provádí následujícím způsobem:

1) Začátek komunikace obsahující příkaz START
2) Vyslání 7-bitové adresy zařízení následované 0 pro zápis dat.
3) Vyslání adresy registru, od kterého hodlám zapisovat data.

4) Vyslání prvního bytu ve správném formátu, adresový pointer se po přijetí automaticky posune na následující adresu. Z toho důvodu je možno ihned zasílat další byte. Zařízení odpovídá vysláním ACK bitu.
5) Zaslání dalších bytů ve správném formátu. Zařízení pokaždé znovu odpovídá vysláním ACK bitu.
6) Ukončení komunikace pomocí příkazu STOP.

8.5.3 Vyčítání bytu

Vyčítání bytu z paměti kalendářového obvodu se provádí podobným způsobem jako zápis. Pro vyčtení je třeba ze správné adresy číst uložená data. Pro vyčítání dat se využívá I2C příkaz REPEAT START, který ve vhodném okamžiku znovu vyšle příkaz start následován adresou zařízení. Vyčítání jednoho bytu dat z paměti kalendářového obvodu se využívá například v knihovní funkce DS3231_TellDay, která vrací přečtený den v týdnu. Vyčtení bytu z paměti obvodu se provádí následovně:

1) Začátek komunikace obsahující příkaz START
2) Vyslání 7-bitové adresy zařízení následované 0 pro zápis dat.
3) Vyslání adresy registru, ze kterého budou vyčítána data. Zasláním adresy dojde k posunu adresového pointeru na požadovanou pozici.
4) Vyslání příkazu REPEAT START
5) Vyslání 7-bitové adresy zařízení následované 1 pro vyčítání dat
6) Přečtení dat z registru, na který aktuálně ukazuje adresový pointer a následné vystavení NACK bitu.
7) Ukončení komunikace pomocí příkazu STOP.

8.5.4 Vyčítání 2 a více bytů

Některá data jsou z důvodu uspoření místa v paměti kalendářového obvodu rozdělena do více registrů, v tom případě je potřeba vyčítat více než jeden byte. Vyčítání více bytů je vhodné i v případech, kdy požadujeme například vyčtení sekund a minut zároveň, jelikož v tomto případě odpadá nutnost opakovaného adresování obvodu. Vyčítání více bytů v jednom volání kalendářového obvodu využívá knihovní funkce DS3231_TellTime. Tato funkce vyčítá hodiny, minuty a sekundy a následně je uloží ve správném formátu na adresy zadané v parametru funkce. Vyčítání bloku dat probíhá následujícím způsobem:

1) Začátek komunikace obsahující příkaz START.
2) Vyslání 7-bitové adresy zařízení následované 0 pro zápis dat.
3) Vyslání adresy registru, od kterého budou vyčítána data. Odesláním adresy dojde k posunu adresového pointeru na požadovanou pozici.

4) Vyslání příkazu REPEAT START
5) Vystavení 7-bitové adresy zařízení následované 1 pro čtení dat.
6) Přečtení bytu z registru, na který aktuálně ukazuje adresový pointer a následné vystavení ACK bitu. Po odeslání bytu se adresový pointer obvodu slave automaticky posunuje na následující registr. Zasláním ACK bitu dávám obvodu příkaz pro zaslání dalšího bytu dat.
7) Vyčtení zbytku požadovaných dat, vždy je třeba vystavit ACK bit.
8) Po vyčtení posledního požadovaného bytu je třeba vyslat NACK bit. Zasláním NACK bitu dojde k ukončení vysílání ze strany obvodu slave.
9) Ukončení komunikace pomocí příkazu STOP.

8.5.5 Zadání úlohy k otestování knihovny

Prostudujte si funkce knihovny DS3231_I2C.h a pomocí vhodně zvolených funkcí zobrazte na první řádek LCD displeje aktuální čas, na druhý řádek zobrazte datum a den vyčtený z RTC. Pokuste se vytvořit režim SET, ve kterém se bude nastavovat čas, datum a blikání červené LED diody. Prostudujte si v datasheetu mikrokontroléru přerušení IRQ a KBI. Následně pomocí tlačítek vyvolávajících tato přerušení realizujte přepínání mezi režimy a samotné nastavování času.

8.6 Knihovna ShiftRegister

Knihovna ShiftRegister_SPI využívá funkce knihovny SPImaster. Funkce obsažené v této knihovně využívají logické operace pro práci s posuvným registrem a umožňují nastavovat výstupní piny obvodu.

8.6.1 Nastavení bytu

Funkce int ShiftReg_SetByte(byte data) slouží k zapsání jednoho bytu dat do posuvného registru. Zapisovaný byte je předávaný funkci prostřednictvím parametru funkce. Návratová hodnota funkce je byte přečtený v průběhu zápisu. Pro realizaci funkcionality je využívána funkce SPI_1ByteRW knihovny SPImaster.

8.6.2 Změna pinu

Pro změnu hodnoty jednoho pinu slouží funkce int ShiftReg_ChangePin(byte pin). Parametrem funkce je číslo bitu, jež požadujeme změnit. V průběhu funkce je vyčten uložený byte v posuvném registru zapsáním bytu 0x00. Následně je u vyčteného bytu

provedena logická operace XOR s požadovaným bitem, pozměněná data jsou opět zapsána do posuvného registru. Během zapisování pozměněných dat je z posuvného registru vyčten byte, který je funkcí navrácen.

8.6.3 Nastavení pinu

Nastavení nebo resetování pinu se vykonává prostřednictvím funkce int ShiftReg_SetPin(byte pin, byte value). Parametrem funkce je číslo bitu, který bude nastaven nebo resetován a nastavovaná hodnota 0 nebo 1. Pokud je zadána jiná hodnota, provede se resetování pinu. V průběhu funkce je vyčtena hodnota bytu uloženého v posuvném registru zapsáním hodnoty 0x00. Následně je požadovaný bit nastaven nebo resetován a změněná data jsou do posuvného registru opět zapsána. Vyčtená data během zápisu jsou funkcí navrácena.

8.6.4 Rotace vlevo a rotace vpravo

Funkce int ShiftReg_ShiftL(void) a int ShiftReg_ShiftR(void) realizují logickou operaci bitová rotace. Je proveden vždy posuv o jednu pozici doleva nebo doprava, v závislosti na použité funkci. Při rotaci doleva se sedmý bit posunuje na pozici nultého bitu, při rotaci doprava pak nultý bit na pozici sedmého bitu.

Obě funkce v prvním kroku vyčtou byte uložený v posuvném registru zapsáním hodnoty 0x00. Vyčtená data jsou následně logicky upravena a nahrána zpět do posuvného registru. Funkce vrací hodnotu vyčtenou z posuvného registru při zapisování upravené hodnoty.

8.6.5 Bitová inverze

Pomocí funkce int ShiftReg_Inverse(void) lze vykonat bitovou inverzi. Bitová inverze spočívá v náhradě 0 za 1 a naopak.

Funkce v prvním kroku zapisuje do posuvného registru byte 0x00. Zapsáním nového bytu dojde k vyčtení původního. Pro vykonání bitové inverze je využita funkce XOR pro všechny bity vyčteného bytu. Upravený byte je opět zapsán do posuvného registru. Zapsáním do posuvného registru je vyčten byte, který je funkcí navrácen.

8.6.6 Zadání úlohy k otestování knihovny

Realizujte různé režimy zapínání a vypínání LED diod pomocí posuvného registru. Ke komunikaci využijte připravenou knihovnu ShiftRegister_SPI.h, která umožňuje

komunikaci mezi mikrokontrolérem a posuvným registrem po sběrnici SPI. Pro přepínání mezi režimy použijte tlačítka, která vyvolají přerušení. Časování zapínání LED diod realizujte pomocí modulu TOD a jeho přerušení. Toto časování pro jednotlivé režimy vhodně zvolte.

Režimy:

0) Postupně zapínejte LED diody zprava doleva, po zapnutí všech je vypínejte zleva doprava. Pravou LED diodu nechte stále svítit. Využijte funkci ShiftReg_ChangePin.
1) Zobrazujte binárně čísla 0x00 – 0xFF. Pro binární zobrazení využijte osmici LED diod.
2) Pomocí funkce ShiftReg_ShiftL a ShiftReg_ShiftR posunujte vhodně zvoleným počtem zapnutých LED diod doleva a doprava.
3) Pseudonáhodně zapínejte LED diody pomocí funkce ShiftReg_SetPin, po vhodně zvolené časové prodlevě proveďte negaci, vypněte zapnuté a zapněte vypnuté LED diody, a pokračujte v pseudonáhodném zapínání.

9. ZÁVĚR

Předložená práce se zabývá realizací laboratorní úloh pro mikrokontrolér MC9S08LH64 demonstrující využití sériových synchronních sběrnic SPI a I2C a sériové asynchronní sběrnice 1-Wire. Pro realizaci laboratorních úloh byl navržen a realizován jednoduchý přípravek na pájivém poli. Na tomto poli je umístěn kalendářový obvod komunikující po sběrnici I2C, posuvný registr s 8 vysoce svítivými LED diodami a teplotní snímač Maxim Integrated DS18B20.

Dále jsem vytvořil jednotlivé knihovny, které realizují samostatnou komunikaci prostřednictvím sběrnic I2C, SPI a 1-Wire. Tyto knihovny obsahují funkce, které zprostředkovávají odesílání a přijímání dat podle standardů jednotlivých komunikačních protokolů se všemi náležitostmi. Součástí knihoven SPI a I2C je možnost využívat softwarovou obsluhu sběrnic nebo vnitřní modul SPI a I2C. Softwarová obsluha sběrnic slouží pro demonstraci jednotlivých kroků potřebných při odesílání dat pomocí ovládání binárních pinů a je ji možno využít v případě, kdy mikrokontrolér neobsahuje vnitřní moduly SPI a I2C. Funkce všech těchto knihoven jsem dále využil v knihovnách DS18B20, LCD, DS3231 a ShiftRegister.

Knihovna DS18B20 obsahuje funkce pro přečtení teploty z teplotního čidla, čtení 2 bytů uložených v paměti teplotního čidla a načtení 64-bitové adresy teplotního čidla z paměti ROM. Pro realizaci těchto funkcí jsou využity funkce knihovny 1-Wire. Další knihovnou je DS3231. Ta využívá funkce knihovny I2C, pomocí kterých nastavuje čas a datum kalendářového obvodu nebo vyčítá čas a datum. Pro zobrazení přečtených dat jsem navíc vytvořil knihovnu LCD, která umožňuje ovládání LCD displeje a zobrazování vyčtených dat. LCD display, který je součástí přípravku, taktéž komunikuje prostřednictvím sběrnice I2C. Poslední je knihovna ShiftRegister, která využívá funkce knihovny SPI a nastavuje nebo mění výstupy posuvného registru a tím ovládá osmici LED diod.

Na závěr jsem vytvořil 3 laboratorní úlohy se zadáním, které demonstrují využití mých knihoven a periferií v praxi. Uvedené úlohy byly odladěny a jejich zdrojové kódy se nachází společně se zadáním úloh na přiloženém CD.

10. SEZNAM POUŽITÉ LITERATURY

[1] NXP SEMICONDUCTOR. MC9S08LH64 Reference Manual. NXP Semiconductors [online]. 2012 [cit. 2018-03-08]. Dostupné z: https://www.nxp.com/docs/en/data- sheet/MC9S08LH64.pdf

[2] NXP SEMICONDUCTOR. TWR-S08 USER GUIDE. NXP Semiconductors [online]. 2012 [cit. 2018-03-12]. Dostupné z: https://www.nxp.com/docs/en/user-guide/TWRS08LH64UG.pdf

[3] DS18B20 Programmable Resolution 1-Wire Digital Thermometer. Maxim Integrated [online]. Maxim Integrated Products, 2015 [cit. 2018-03-15]. Dostupné z: https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf

[4] GOOK, Michael. Hardwarová rozhraní: průvodce programátora. Brno: Computer Press, 2006. Hardware (Computer Press). ISBN 80-251-1019-2.

[5] Understanding the I2C Bus. Www.ti.com [online]. Texas Instruments Incorporated, 2015 [cit. 2018-03-16]. Dostupné z: http://www.ti.com/lit/an/slva704/slva704.pdf

[6] MATOUŠEK, David. Práce s mikrokontroléry ATMEL. 2. vyd. Praha: BEN - technická literatura, 2006. ISBN 80-730-0209-4.

[7] BASICS OF THE SPI COMMUNICATION PROTOCOL. Circuit Basics [online]. [cit. 2018-03-20]. Dostupné z: http://www.circuitbasics.com/basics-of-the-spi-communication-protocol/

[8] Serial Peripheral Interface (SPI). Sparkfun [online]. Niwot, Colorado: SparkFun Electronics [cit. 2018-03-22]. Dostupné z: https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi

[9] Maxim Integrated Products. DS3231 Extremely Accurate I2C-Integrated RTC/TCXO/Crystal [online]. 2015, 20 [cit. 2018-03-25]. Dostupné z: https://datasheets.maximintegrated.com/en/ds/DS3231.pdf

[10] Texas Instruments. SNx4HC164 8-Bit Parallel-Out Serial Shift Registers [online]. 2017, 35 [cit. 2018-04-01]. Dostupné z: http://www.ti.com/lit/ds/symlink/sn74hc164.pdf

11. SEZNAM PŘÍLOH NA CD

- Laboratorní úloha realizující komunikaci po sběrnici 1-Wire mezi snímačem teploty DS18B20 a mikrokontrolérem NXP MC9S08LH64

- Laboratorní úloha realizující komunikaci po sběrnici I2C mezi kalendářovým obvodem DS3231 a mikrokontrolérem NXP MC9S08LH64

- Laboratorní úloha realizující komunikaci po sběrnici SPI mezi posuvným registrem SN74HC164 a mikrokontrolérem NXP MC9S08LH64

- Zadání laboratorní úlohy pro sběrnici I2C

- Zadání laboratorní úlohy pro sběrnici 1-Wire

- Zadání laboratorní úlohy pro sběrnici SPI