

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra telekomunikačnej techniky

System pre ovládanie vlakovej klimatizácie

Vedúci práce: Ing. Pavel Troller, CSc.

Externý vedúci práce: Ing. Michal Sarnovský

Autor: Bc. Andrej Dorinec

Máj 2016

Čestné prehlásenie

Prehlasujem, že som zadanú diplomovú prácu spracoval sám s prispením vedúceho práce a konzultanta a používal som len literatúru v práci uvedenú. Ďalej prehlasujem, že nemám námietky proti požičovaniu alebo zverejňovaniu mojej diplomovej práce alebo jej časti so súhlasom katedry.

Dátum: 27. 5. 2016

.....

Andrej Dorinec

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra telekomunikační techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Andrej Dorinec**

Studijní program: Komunikace, multimédia a elektronika
Obor: Síť elektronických komunikací

Název tématu: **Systém pro ovládání vlakové klimatizace**

Pokyny pro vypracování:

Navrhněte a zrealizujte ovládací a diagnostický software pro vlakovou klimatizaci a vytápění, který bude propojený s klimatizačním a vytápěcím vlakovým zařízením sběrnici CAN předem definovaným protokolem. Software zrealizujte pro systém s mikroprocesorem STM32F4 třídy ARM Cortex-M3 s připojeným grafickým dotykovým displejem. Obsluha musí být schopna prostřednictvím zařízení jednoduchými dotyky ovládat funkce klimatizace a vytápění vlaku, sledovat stav a diagnostikovat celé zařízení. Permanentní záznamové médium ani obvody reálného času nejsou požadovány, jediné propojení se systémem je prostřednictvím zmiňované sběrnice CAN.

Seznam odborné literatury:

- [1] Skalický. P.: Přístrojové aplikace mikropočítačů. Vydavatelství ČVUT, Praha 2004. ISBN: 80-01-03111-X.
- [2] Yiu, J.: The Definite Guide to ARM Cortex-M3 and Cortex-M4 processors, Third Edition, ARM Ltd. Cambridge, UK, 2014. ISBN: 978-0-124080-829.
- [3] Firemní materiály firmy STMicroelectronics. Dostupné na: <http://www.st.com> [on-line]
- [4] Dokumentace dostupná na stránce <http://www.ucsimply.cz> [on-line]
- [5] Dokumentace k mikrokontroléru STM32F429.

Vedoucí: Ing. Pavel Troller, CSc.

Platnost zadání: do konce letního semestru 2016/2017



prof. Ing. Boris Šimák, CSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 21. 12. 2015

Anotácia

System pre ovládanie vlakovej klimatizácie

Táto práca sa zaoberá návrhom ovládacieho a diagnostického softvéru pre vlakovú klimatizáciu a vyhrievanie, ktorý je prepojený s klimatizačným a vyhrievacím vlakovým zariadením zbernicou CAN (Controll Area Network) s dopredu definovaným protokolom. Softvér je implementovaný pre systém s mikroprocesorom STM32F4 triedy ARM Cortex – M4 s pripojeným grafickým dotykovým displejom. Obsluha je pomocou grafického rozhrania schopná riadiť či zobrazovať aktuálny stav vlakovej klimatizácie. V prvej časti diplomovej práce sa zaoberám teoretickou rovinou architektúry ARM Cortex – M4, mikroprocesor STM32F429, FreeRTOS (Free Real-Time Operating System), displej TFT LCD (Thin-Film-Transistor Liquid-Crystal Display), zbernica CAN. Druhá časť je venovaná hardvérovému návrhu zdrojov pre nový a väčší displej. V tretej časti je popísaný zdrojový kód a jeho funkcia.

Kľúčové slová: ARM, STM32F429, FreeRTOS, TFT-LCD

Anotation

System for Control of Train Air conditioning

This thesis deals with design of control and diagnostic software for train air conditioning and heating system. The connection with the system is established over CAN (Controller Area Network) bus with predefined protocol. The Software is implemented for system with STM32F4 class ARM Cortex - M4 with TFT LCD touch panel. Service personnel can manage or observe diagnostic outputs of train air conditioning by a graphical user interface. The first part of my diploma thesis describes basic HW and SW components of the solution - ARM Cortex - M4 architecture, STM32F4 microcontroller, FreeRTOS (Free Real-Time Operating System), TFT LCD (Thin-Film-Transistor Liquid-Crystal Display) display, CAN bus. The second part is dedicated to the hardware design of power supply for the display. The third part contains software design - it includes particular drivers and functions.

Index Terms: ARM, STM32F429, FreeRTOS, TFT-LCD

Obsah

| | | |
|-------|------------------------------------|----|
| 1 | Úvod..... | 1 |
| 2 | Teoretická časť..... | 2 |
| 2.1 | STM32F429 Discovery kit..... | 2 |
| 2.1.1 | Obsah discovery kitu | 2 |
| 2.1.2 | STM32F429ZIT6 mikrokontrolér..... | 5 |
| 2.2 | Mikroprocesory ARM..... | 8 |
| 2.3 | FreeRTOS..... | 9 |
| 2.3.1 | Real-Time aplikácie..... | 9 |
| 2.3.2 | Multitasking..... | 9 |
| 2.3.3 | Mechanizmus manažovania úloh | 10 |
| 2.3.4 | Semafor | 11 |
| 2.3.5 | Fronta | 12 |
| 2.4 | TFT-LCD displej..... | 14 |
| 2.4.1 | LCD..... | 15 |
| 2.4.2 | TFT | 16 |
| 2.4.3 | LCD-TFT Controller (LTDC) | 17 |
| 2.4.4 | Synchrónne časovanie displeja..... | 18 |
| 2.4.5 | Odporová dotyková plocha | 20 |
| 2.5 | CAN zbernica..... | 22 |
| 2.5.1 | Fyzická vrstva..... | 23 |
| 2.5.2 | Úrovne signálu | 23 |
| 2.5.3 | Konektor | 24 |
| 2.5.4 | Metóda prístupu..... | 24 |
| 2.5.5 | Typy rámcov | 25 |
| 2.5.6 | Štruktúra dátového rámca..... | 25 |

| | | |
|-------|--|----|
| 2.5.7 | Arbitrácia | 27 |
| 2.6 | Basic Extended Controller area network | 28 |
| 2.6.1 | Filtre | 28 |
| 2.6.2 | Prijímanie správ | 30 |
| 2.7 | STemWin | 31 |
| 3 | Praktická časť | 33 |
| 3.1 | Návrh hardvéru | 33 |
| 3.1.1 | Napät'ové zdroje | 33 |
| 3.1.2 | Spínaný menič s MC34063 | 34 |
| 3.1.3 | Lineárny menič LM317 | 36 |
| 3.2 | Návrh softvéru | 37 |
| 3.2.1 | Moduly | 38 |
| 3.2.2 | Modul CAN driver | 38 |
| 3.2.3 | Modul LCD – TFT driver | 39 |
| 3.2.4 | Modul MainApp | 40 |
| 3.2.5 | Modul STemWin5 | 40 |
| 3.2.6 | Modul STMPE811 driver | 41 |
| 3.3 | Grafické rozhranie | 41 |
| 3.3.1 | Úvodná obrazovka aplikácie | 41 |
| 3.3.2 | Menu aplikácie | 42 |
| 3.3.3 | Obrazovka chlazení | 43 |
| 3.3.4 | Obrazovka topení | 43 |
| 3.3.5 | Obrazovka kalendára | 44 |
| 3.3.6 | Obrazovka grafov | 46 |
| 3.3.7 | Obrazovka porúch | 46 |
| 3.3.8 | Obrazovka informácií | 47 |

| | | |
|---|----------------------------------|----|
| 4 | Záver | 48 |
| 5 | Zoznam obrázkov..... | 49 |
| 6 | Zoznam tabuliek | 51 |
| 7 | Zoznam použitej literatúry | 52 |

Zoznam použitých symbolov a skratiek

| | |
|------------------|--|
| AHB | High-performance Bus |
| APB | Advanced Peripheral Bus |
| ARM | Advanced RISC Machine |
| CAN | Controlling Area Network |
| CSMA/CR | Carrier Sense Multiple Access/Collision Resolution |
| DMIPS | Dhrystone Milion instructions Per Second |
| EXID | Extended ID |
| FIFO | First in first out |
| HBP | Horizontal Back Porch |
| HFP | Horizontal front porch |
| HSYNC | Height synchronization |
| HVAC | Heating, ventilating and air conditioning |
| I ² C | Inter-Integrated Circuit |
| IDE | Integrated Development Environment |
| LCD | Liquid crystal display |
| Master | Ten, ktorý riadi komunikáciu na zbernici |
| RISC | Reduced instruction set computer |
| RTC | Real time clock |
| RTOS | Real time operating system |
| Slave | Ten, ktorý je riadený pri komunikácii na zbernici |
| SPI | Serial Peripheral Interface Bus |
| STID | Standard ID |
| TFT | Thin-film-transistor |
| UART | Universal asynchronous receiver/transmitter |
| USB | Universal |
| VBP | Vertical Back porch |
| VFP | Vertical front porch |
| WSYNC | Width synchronization |

Zoznam príloh

Priložené DVD obsahuje diplomovú prácu vo formáte .pdf a priečinky:

[A] – Schéma zapojenia USB OTG

[B] – Obsahuje fotky predrátovania vývojového kitu, zdrojov a displeja

[C] – Tabuľka zapojenia pinov vo formáte .xlsx .

[D] – Zdrojový kód

[E] – Literatúra použitá pri riešení diplomovej práce

1 Úvod

Železničná doprava je aj v dnešnej dobe, kedy je letecká doprava veľmi lacná a rozšírená, veľmi používaným dopravným prostriedkom. Denne vlaky prepravujú veľké množstvo cestujúcich nielen na krátke vzdialenosti ale aj medzištátne, kde cesta trvá aj niekoľko desiatok hodín. Pre cestujúcich je pri takto dlhých cestách a rôznych vonkajších podmienkach nutné zabezpečiť vhodnú teplotu a čerstvý vzduch. O vlaky, ktoré sú vystavené nízkym teplotám je nevyhnutné zabezpečiť vyhrievanie na určitú teplotu aj počas odstavenia. Dôvodom je, že vyhriať premrznutý vlak je energeticky náročnejšie než ho udržiavať pri určitej teplote nad nulou. Z tohto dôvodu sa dnes inštalujú klimatizačné a vyhrievacie zariadenia, ktoré majú zabezpečiť vyššie zmienené podmienky.

Vlaky teda obsahujú klimatizácie a vyhrievacie zariadenia, ktoré sú primárne riadené z centrálnej jednotky. Jednotlivé vagóny by mali disponovať určitým riadiacim systémom s grafickým rozhraním, ktorým by obsluha mohla ovládať alebo ovplyvňovať určité parametre klimatizácie. Dôležité pre obsluhu je aj registrácia poruchy takýchto zariadení.

Účelom je navrhnuť systém, vrátane jeho grafického rozhrania, ktorý bude komunikovať s klimatizáciou a vyhrievacím zariadením. Tento systém by mal byť schopný ovládať teplotu vo vagóne, ovládať prívod čerstvého vzduchu, oznámiť poruchu funkčnosti alebo zobrazovať aktuálny stav veličín nameraných vlakovými čidlami. Prepojenie medzi riadiacim systémom a klimatizáciou s vyhrievacím zariadením bude realizované zbernicou CAN (Controlling Area Network), ktorá je vhodná pre komunikáciu v zarušenom prostredí a je rozšírená v takýchto aplikáciách .

2 Teoretická časť

V tejto časti sú zhrnuté teoretické znalosti nutné k pochopeniu základných blokových funkcií a pre pochopenie celého systému. Sú to znalosti nevyhnutné k realizácii diplomovej práce.

2.1 STM32F429 Discovery kit

Realizácia vlastnej dosky, jej obvodový návrh a osadenie by viedlo k náročnému a zdĺhavému vývoju. Preto som hľadal alternatívu, ktorá mi umožní rýchlo a jednoducho vytvoriť vlastnú aplikáciu. Na trhu je množstvo vývojových dosiek, ktoré sa dajú použiť. Pre moju prácu som zvolil Discovery kit STM32F429.

STM32F429 discovery kit je vývojová doska rady STM32, umožňujúca vyvíjať aplikácie jednoducho a efektívne. Svojím bohatým vybavením je vhodná pre široké využitie. Firmware je postavený na operačnom systéme FreeRTOS, o ktorom sa zmienim v samostatnej kapitole.

2.1.1 Obsah discovery kitu

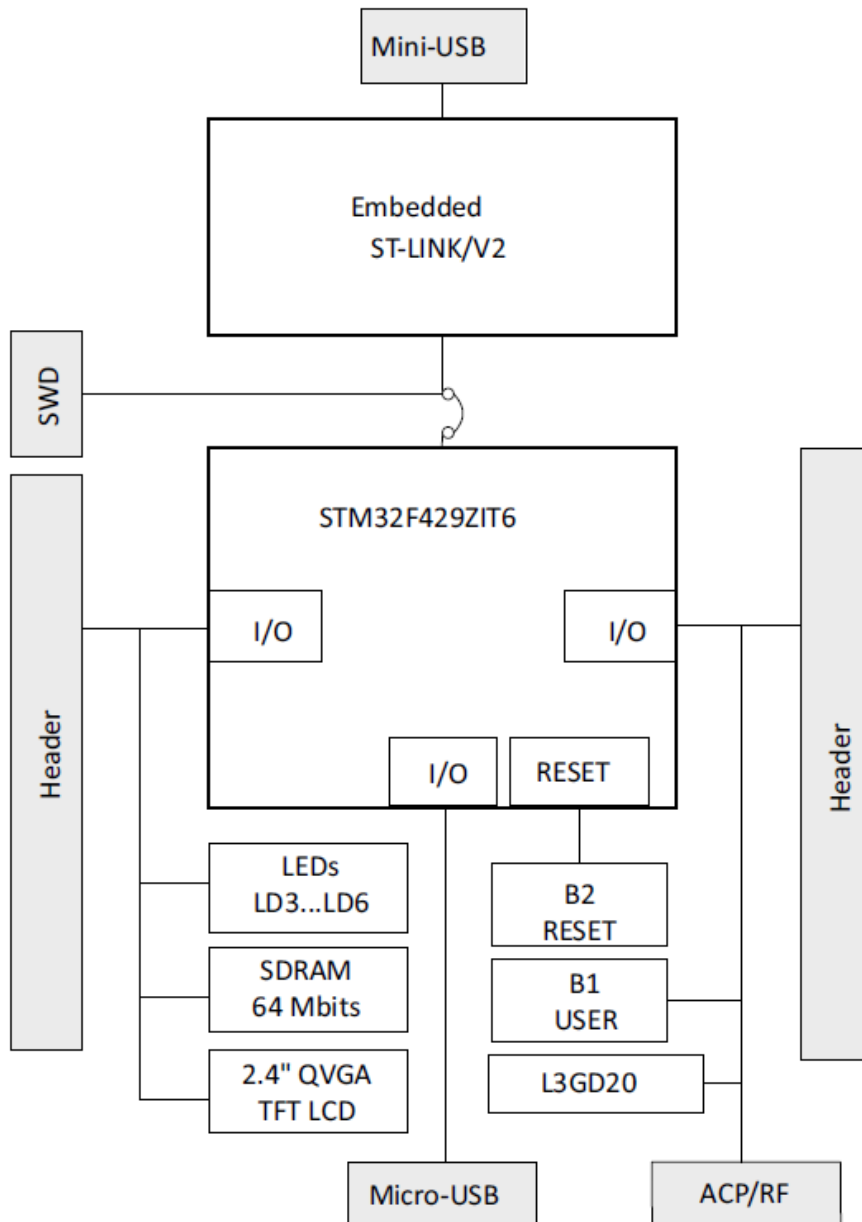
- mikrokontrolér STM32F429ZIT6 s 2 MB Flash pamäte a 256 KB pamäte RAM v LQFP144 púzdre (Low Profile Quad Flat Package 144). 2 MB flash pamäť je využitá na 70%, skompilovaný program má necelé 2 MB, takže bolo nutné s pamäťou zaobchádzať šetrne.
- ST-LINK/V2 je debugger a programátor pre mikroradiče STM32. Vďaka programátoru som programoval discovery kit jednoduchým pripojením do USB.
- napájanie na doske cez zbernicu USB (3V a 5V).
- pohybový senzor L3GD20 MEMS, gyroskop a akcelerometer- TFT LCD 240 x 320, 262K farieb RGB. Pohybové senzory je možné využiť pre budúci vývoj dosky. Displej TFT LCD som nahradil displejom s väčším rozmerom a rozlíšením, no s rovnakou technológiou.
- SDRAM 64 Mbit. Pamäť je nevyhnutná pre vykresľovanie obrazu pre nový displej s rozmerom 800 x 460.
- led diody, 2 pre užívateľa, 3 pre USB komunikáciu a 1 pre napájanie

- dve tlačítka – jedno pre reštartovanie mikroprocesoru a druhé pre spustenie testovacieho režimu. Ak je pri reštarte stlačené druhé tlačítko, discovery kit spustí testovací režim.
- USB OTG micro konektor. Po analizovaní obvodu som z dôvodu kolidovania s farebnými signálmi musel odpájať súčiastky U7, U8, C53 a R67 (viz príloha na CD [A]).
- externé pinheady pripojené k LQFP144 I/O k rýchlemu a jednoduchému pripojeniu



Obrázok 1: Discovery kit STM32F429 (prebrané z [1])

Nasledujúci obrázok popisuje vzájomné prepojenie jednotlivých častí discovery kitu. Srdcom je mikrokontrolér STM32F429ZIT6, ktorému sa budem venovať v nasledujúcej kapitole.



Obrázok 2: Hardvérový blokový diagram (prebrané z [1])

2.1.2 STM32F429ZIT6 mikrokontrolér

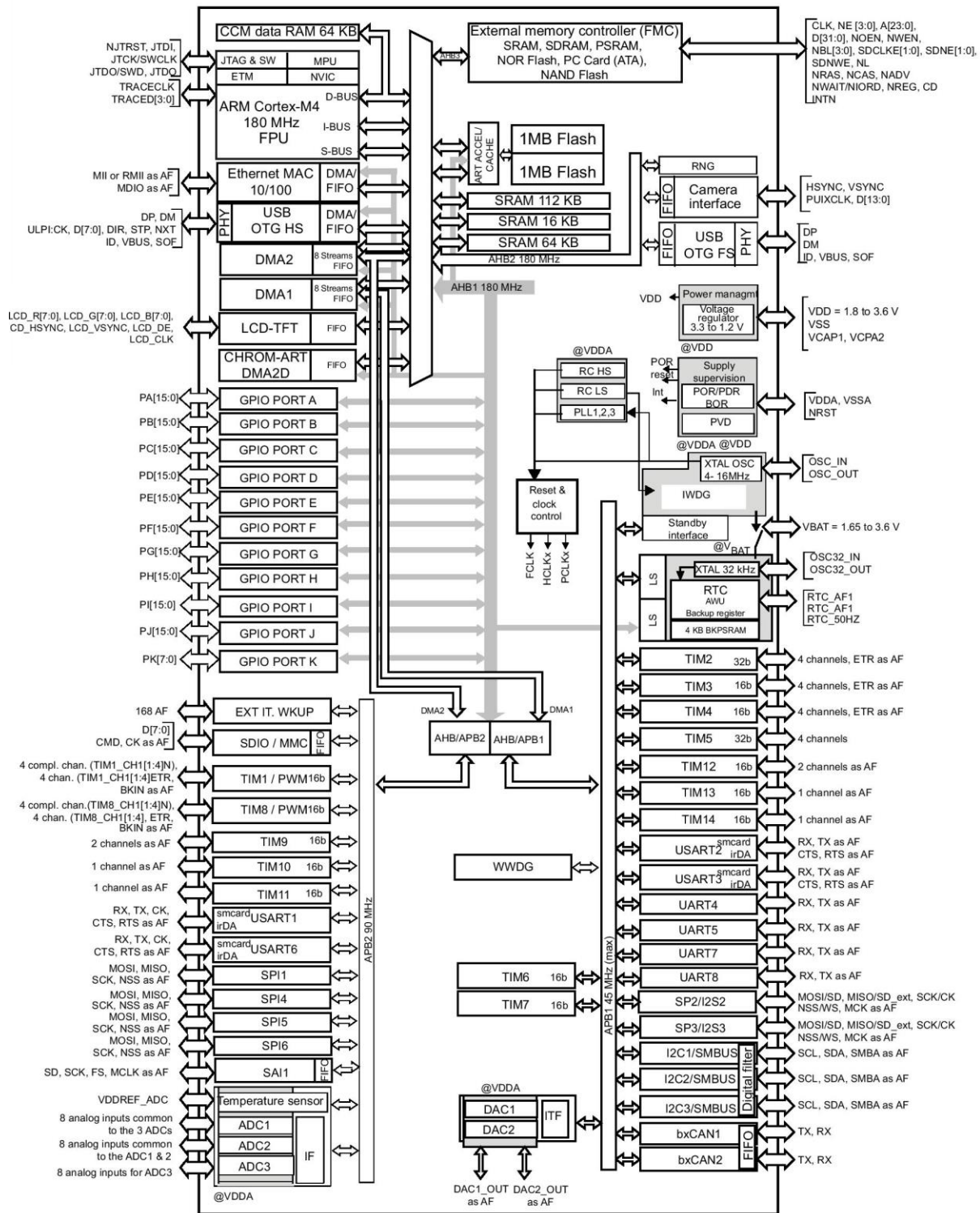
Mikrokontrolér STM32F429ZIT obsahuje jadro ARM Cortex – M4 32 – bitový procesor s FPU má 225 DMIPS. Jeho vlastnosti a výhody sú popísané v tabuľke 1.

| Vlastnosti | Výhody |
|--|---|
| <p>Vysoký výkon:</p> <ul style="list-style-type: none"> - až 180 MHz/225 DMIPS Cortex – M4 - CoreMark score: 608 na 180MHz - CoreMark/MHz: 3,37 | <ul style="list-style-type: none"> - CPU je taktované na maximálnu frekvenciu tj. 180 MHz a to z dôvodu náročnosti spracovania obrazu. Táto hodnota frekvencie SYSCCLK nám taktiež umožnila nastaviť baudrate 250 kbps na CANe. |
| <p>Maximálna integrácia:</p> <ul style="list-style-type: none"> - Až 2MB Flash pamäte na čipe - Až 256 KB pamäte SRAM - Reštartovací obvod, interné RC, PLL, WLCSP | <ul style="list-style-type: none"> - Pamäť je nevyhnutná pre program, ktorý má po kompilácii necelé 2 MB. Najväčšiu záťaž programovej pamäte tvorí grafika (knihnica StemWin) |
| <p>Navrhnutý pre vysoký výkon a ultra rýchly prenos dát:</p> <ul style="list-style-type: none"> - ART akcelerátor (akcelerátor pamäte) - Chrom-ART akcelerátor (grafický akcelerátor) | <ul style="list-style-type: none"> - Výkon ekvivalentný s nulovým čakáním vybavenia z pamäte Flash. - Grafický obsah je vytváraný dvakrát rýchlejšie ako nezávisle z CPU |
| <ul style="list-style-type: none"> - 32 bitová, 7 vrstvová matica AHB zbernice s 10 masters a 8 slaves zahrňujúca 3 bloky SRAM - Multi DMA kontroléry: 2 všeobecné, 1 pre USB HS, 1 pre Ethernet | <ul style="list-style-type: none"> - Súbežné vykonávanie a prenos dát |
| <ul style="list-style-type: none"> - Flexibilné pamäťové rozhranie s SDRAM, až do 90 MHz, 32-bit paralelne | <ul style="list-style-type: none"> - vysoká šírka pásma pre externé pamäte |
| <p>Výborná energetická účinnosť:</p> <ul style="list-style-type: none"> - 260uA/MHz na 180 MHz pri bežiacom CoreMark z pamäte Flash (s vypnutými perifériami) - RTC <1 uA - VDD z 3,6 V na 1,7 V - 1,2 V regulátor s možnosťou škálovania energie | <ul style="list-style-type: none"> - Veľmi flexibilné na redukovanie spotreby energie pre aplikácie bežiacie s rýchlym spracovaním a nízkym energetickým výkonom počas behu na nízke napätie alebo na dobíjacej batérii. |
| <p>Periféria a konektivita:</p> <ul style="list-style-type: none"> - konektivita: rozhranie pre kameru, krypto/hash HW procesor s AES GCM a CCM podporou, a SHA-256 - Ethernet MAC10/100 s podporou IEEE 1588 v2, dva USB OTG (jeden s podporou HS) - až 20 komunikačných rozhraní (zahrňujúce 4x USART + 4xUART, 6xSPI, 3xI2C s digitálnym filtrom, 2xCAN, SDIO) - USART na 11,25 Mbit/s, SPI na 45Mbit/s | <ul style="list-style-type: none"> - Nové možnosti pripojiť a komunikovať s vysokorýchlostnými zariadeniami napr. Ethernet kde by bolo možné využiť kryptovanie alebo hash. V mojej práci používam hlavne CAN, pre komunikáciu s klimatizáciou a zbernica I2C je používaná dotykovou plochou displeja. |
| <p>Audio:</p> <ul style="list-style-type: none"> - jednocelový PLL, 2xI2C a 1x SAI s podporou TDM | <ul style="list-style-type: none"> - Vysoká kvalita viac-kanálového audia. |
| <ul style="list-style-type: none"> - LCD TFT kontroler - Až SVGA formát (800 x 600) | <ul style="list-style-type: none"> - Podpora rôznych displejov. V mojom prípade použitý displej s rozmermi 800 x 480. |

| Vlastnosti | Výhody |
|---|--|
| - Až 24 – bitový RGB paralelný výstup - podpora dvoch vrstiev | |
| Analóg: - 2x 12-bitový DAC, 3x 12-bitový ADC dosahujúci - Až 17 timerov: 16 a 32 bitových, bežiacich na 180 MHz | - Väčšia precíznosť vďaka vysokému rozlíšeniu. |
| Integrácia: WLCSP143 4,5 x 5,5 mm, 2 MB Flash a 256 KB SRAM pamäte | - Menšia doska umožňuje menšie aplikácie. |
| Rozsiahle nástroje a softvérové riešenia: - Rôzne IDE, knižnice, RTOS, atp. | - široké možnosti pre vývoj vlastnej aplikácie |

Tabuľka 1: Výhody a vlastnosti discovery kitu

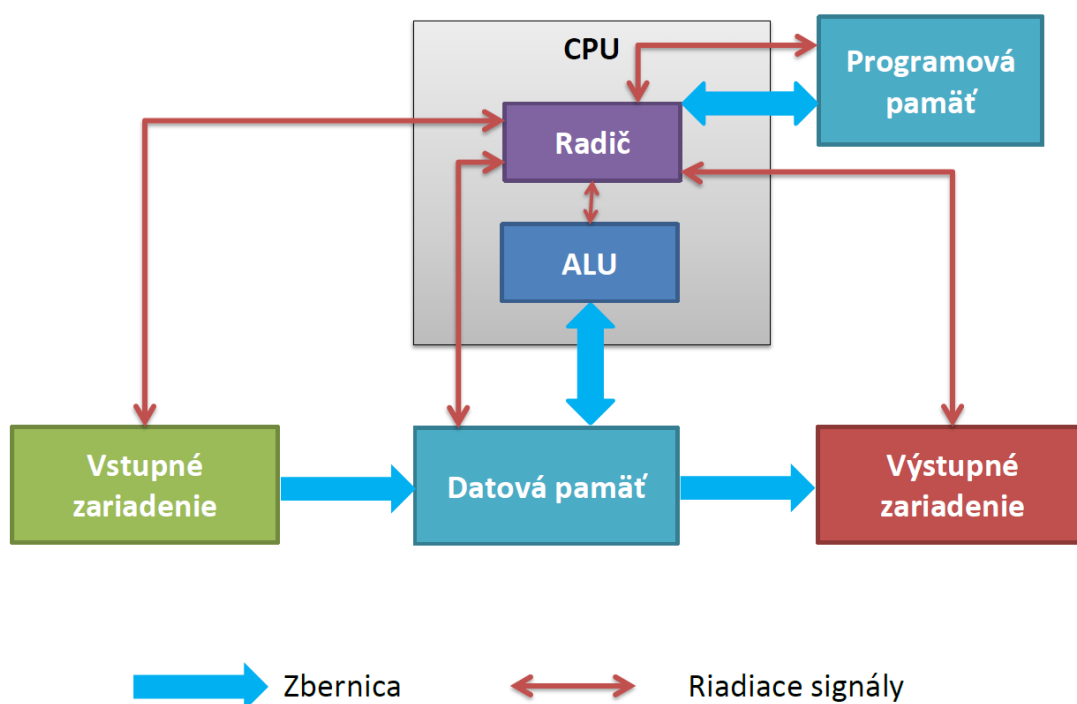
Na obrázku 3 je znázornený blokový diagram procesoru. Je tam vidieť prepojenie jednotlivých zberníc a periférií, ktoré som používal pri práci s discovery kitom.



Obrázok 3: Blokový diagram STM32F429ZIT6 (prebrané z [1])

2.2 Mikroprocesory ARM

ARM je rodina procesorov RICS od firmy ARM Holdings s veľmi nízkou spotrebou elektrickej energie používaných hlavne v mobiloch a tabletoch. Od svojho vzniku spoločnosť ARM Holdings vytvorila mnoho architektúr s rôznymi jadrami, jednou z najperspektívnejších je Cortex M4. Procesor ARM Cortex M4 je 32-bitový procesor s Harvardskou architektúrou. Harvardská architektúra sa líši od Von Neumanovskej hlavne tým, že má pamäť rozdelenú na programovú a dátovú časť tak, ako je zobrazené na obrázku 4.



Obrázok 4: Harvardská architektúra

V programovej pamäti sú uložené napríklad inštrukcie, konštanty a kód ktorý sa z pamäte vykonáva. Naopak v dátovej pamäti sú uložené premenné, tabuľky, výsledky, atď. To vedie k možnosti paralelného spracovania inštrukcií aj dát.

Podrobné informácie tejto problematiky nie sú obsahom tejto diplomovej práce. Viac informácií ohľadom procesorov ARM Cortex-M4 nájdeme v datasheete k procesoru Cortex-M4 [3].

2.3 FreeRTOS

Free RTOS (Real-Time Operating System) je jedným z popredných na trhu, komerčný a kompletne voľný operačný systém pre embedded zariadenia. Podporuje 35 architektúr, je striktné kontrolovaný, profesionálne vyvíjaný, dnes je veľmi rozšírený RTOS pre mikrokontroléry.

2.3.1 Real-Time aplikácie

Hlavný rozdiel medzi štandardnou aplikáciou a real-time aplikáciou je v časovom vymedzení pre vykonanie činnosti. V real-time aplikáciach vieme deterministicky určiť čas počas ktorého bude aplikácia vykonávaná na základe znalostí hardvéru a softvéru. Typicky, aplikácie tohto typu obsahujú hard a soft real-time požiadavky.

Soft real-time požiadavky – udávajú čas ukončenia, ale porušenie tohto termínu nevedie k stavu, kedy je systém nepoužiteľný. Typickým príkladom je pomalá reakcia na úder do klávesy. Takýto stav môže byť nežiadúci a nepríjemný, ale systém sa dá naďalej používať.

Hard real-time požiadavky – udávajú čas ukončenia, ale porušenie tohto termínu vedie k stavu, kedy je systém nepoužiteľný. Napríklad airbag v aute je neužitočný, ak by bola jeho reakčná doba na senzor nárazu príliš pomalá.

Free RTOS je trieda real-time operačných systémov navrhnutých pre mikrokontrolery.

2.3.2 Multitasking

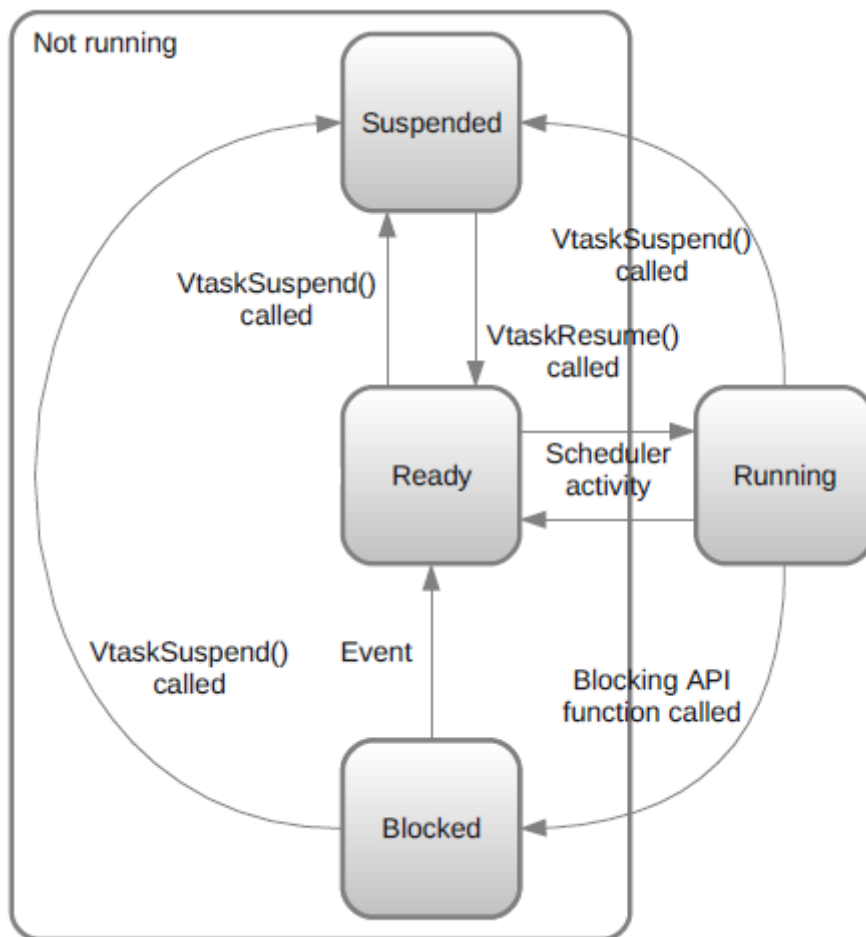
Jedná sa o bežnú vlastnosť dnešných operačných systémov. Embedded systémy majú väčšinou prístup len k jednému procesoru, ktorý má mnoho vstupov a výstupov. Real-time operačný systém musí zabezpečiť, aby sa určitý čas venoval rôznym aktivitám. Z toho dôvodu sa vo FreeRTOS tvoria tzv. úlohy (Tasks), ktoré majú určité stavy, štruktúry a priority podľa ktorých sa systém rozhodne, ktorú úlohu vykoná.

Samotný systém má vytvorené určité Tasky ktoré pravidelne vykonáva. Medzi ne som vytvoril ďalšiu, CAN_PROCESS TASK. Vzhľadom na to, že na zbernici CAN sa neustále prenášajú správy a je nežiadúce, aby sa systém k správam nedostal načas, som nastavil prioritu dostatočne vysokú na to, aby sa k procesu často vracal.

V mojom prípade nie je procesor vyťažovaný natoľko, aby sa k vytvoreným úlohám nedostal. Počas spracovávania samostatného programu sa CPU nedostane ani cez 10% výkonu.

2.3.3 Mechanizmus manažovania úloh

Free RTOS umožňuje manipulovať s viacerými úlohami, no len jedna úloha môže byť vykonávaná v danom čase (s jednojadrovým procesorom). Systém preto potrebuje plánovač úloh, ktorým rozhodne ktorá úloha sa má vykonávať. Tento plánovač je v jadre FreeRTOS a vyberá úlohy podľa priority a aktuálneho stavu úlohy. Rozdielne stavy úloh sú zobrazené na obrázku 5.



Obrázok 5: Stavy úloh vo FreeRTOS (prebrané z [2])

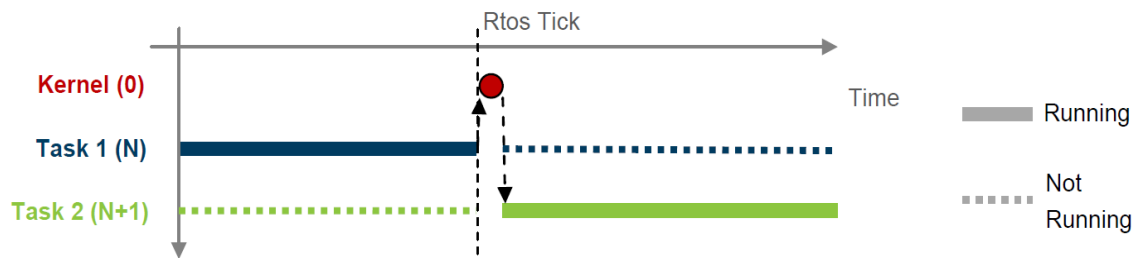
Na aplikačnej úrovni sú dva možné stavy, v ktorých sa môže úloha nachádzať. Sú to stavy „Running“ a „Not running“. Plánovač si stav „Not running“ delí na ďalšie tri stavy:

Suspend: úloha bola deaktivovaná aplikáciou.

Blocked: úloha je blokovaná a čaká na synchronizačnú udalosť.

Ready: úloha je pripravená na vybavenie ale úloha s vyššou prioritou je stále vykonávaná.

Cieľom plánovania úloh je rozhodnúť, ktorá z úloh v stave „Ready“ musí byť vykonaná v daný čas. FreeRTOS dosahuje tento cieľ prioritami, ktoré sú úlohám pridelené pri ich vytváraní. Priorita je jediný element, ktorý plánovač úloh uvažuje pri rozhodovaní, ktorá úloha má byť práve spustená.

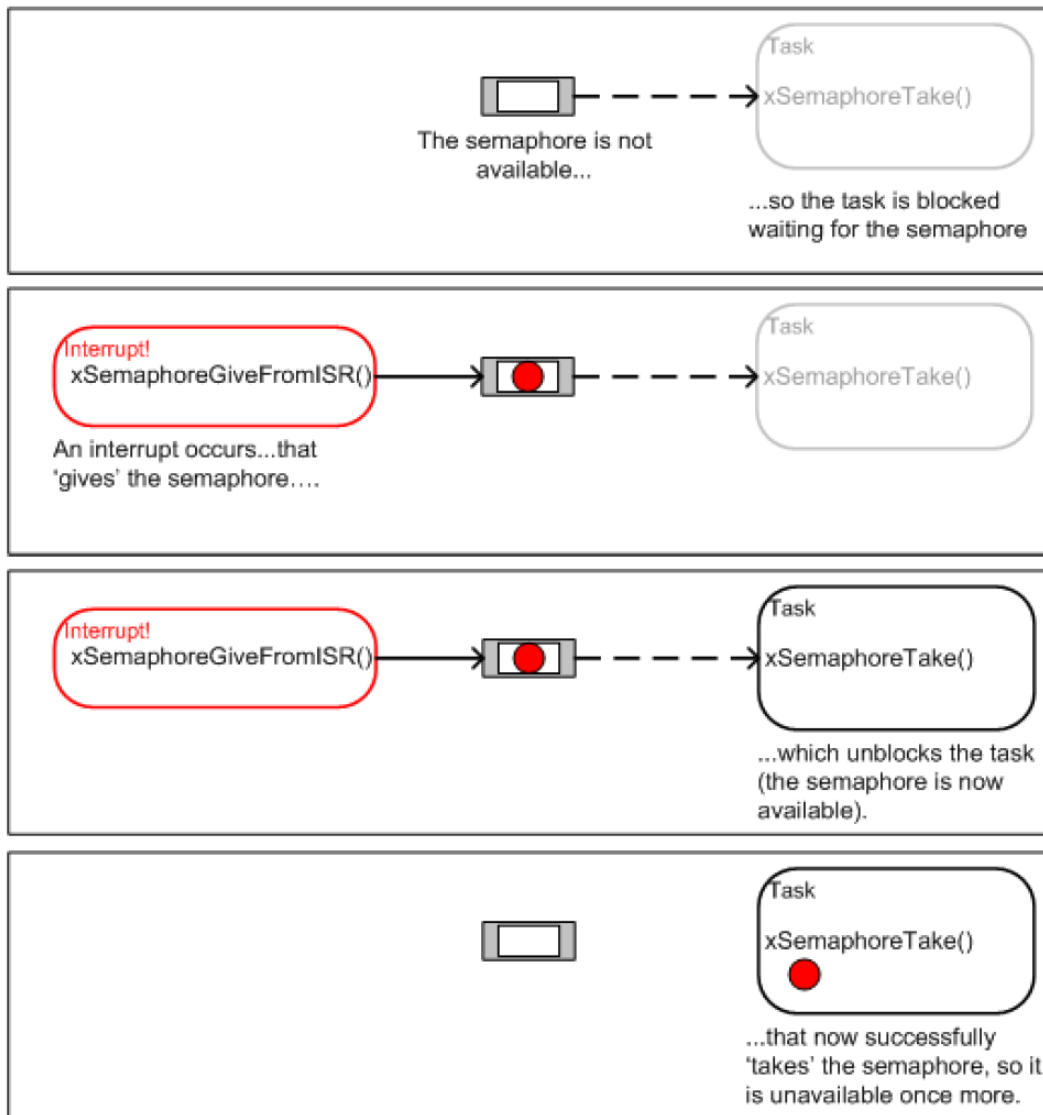


Obrázok 6: Grafické zobrazenie vykonávania úloh (prebrané z [2])

2.3.4 Semafor

S cieľom synchronizovať rôzne úlohy, FreeRTOS kernel poskytuje semafor. Semafor si môžeme predstaviť ako synchronizačný token, ktorý si môžu úlohy medzi sebou vymeniť.

V prípade synchronizácie úlohy s inou úlohou, alebo prerušením, si úloha požiada o semafor. Ak semafor nie je dostupný, úloha prejde do stavu „blocked“ a bude čakať na dostupnosť semaforu. Aktuálna úloha, ktorá je v stave „ready“ sa spustí. Po ukončení úlohy je semafor opäť dostupný a úloha, ktorá čakala na dostupnosť sa odblokuje, prevezme semafor a vykoná sa. Proces môžeme vidieť na obrázku 7.



Obrázok 7: Predávanie semaforu medzi úlohami (prebrané z [2])

2.3.5 Fronta

Fronty sa vo FreeRTOS používajú na komunikáciu a synchronizáciu medzi úlohami. Sú dôležitý element, nevyhnutný pre implementovanie komplexnej aplikácie s úlohami, ktoré medzi sebou interagujú. Úlohou fronty je akumulovať konečné množstvo dát, ktoré medzi sebou interagujú. Úlohou fronty je akumulovať konečné množstvo dát, ktoré medzi sebou interagujú. Fronty majú byť prístupné pre zápis aj čítanie všetkým úlohám, nemajú patriť konkrétnej úlohe. Fronta je typu FIFO, takže elementy sú čítané z fronty v poradí, v akom boli zapísané. Sú dve metódy zápisu do fronty, na začiatok alebo koniec fronty.

Možnosť fronty som využil pri prijímaní správ zo zbernice CAN. Vytvoril som frontu CAN_rxQueue pre prijaté správy. Správa sa po prijatí dostane do fronty a až sa dostane na rad Task CAN_Proccss, vykoná sa kontrola prijatých správ. Po spracovaní správy sa správa z fronty vymaže.

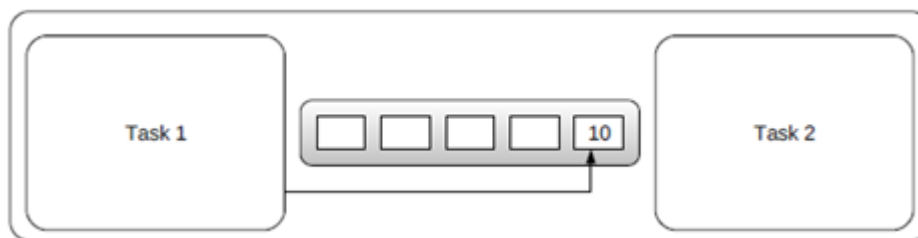
Ilustrácia používania fronty

Aby sme umožnili komunikáciu medzi úlohou 1 a úlohou 2, vytvoríme frontu. Fronta môže obsahovať 5 hodnôt a pri jej vytvorení neobsahuje žiadne hodnoty, takže je prázdna:



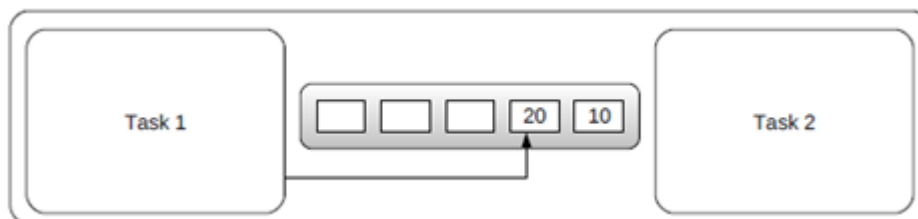
Obrázok 8: Vytvorenie prázdnej fronty (prebrané z [2])

Úloha 1 zapíše hodnotu do fronty. Hodnota je uložená na koniec tejto fronty. Keďže bola fronta prázdna, je momentálne hodnota prvá, aj posledná:



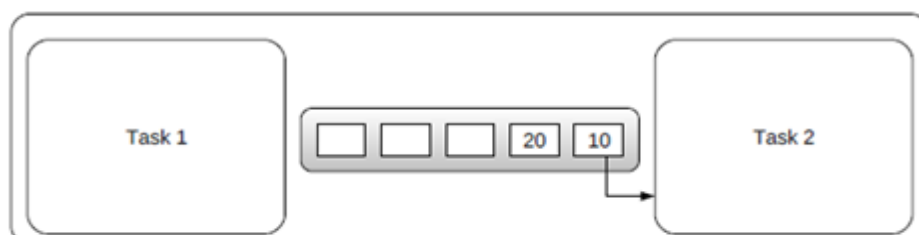
Obrázok 9: Vloženie hodnoty do fronty (prebrané z [2])

Úloha 1 zapíše ďalšiu hodnotu. Fronta obsahuje hodnotu zapísanú predtým a novú hodnotu. Predchádzajúca hodnota ostáva zatiaľ na začiatku fronty, nová hodnota ostáva na konci fronty. Tri miesta máme stále voľné:



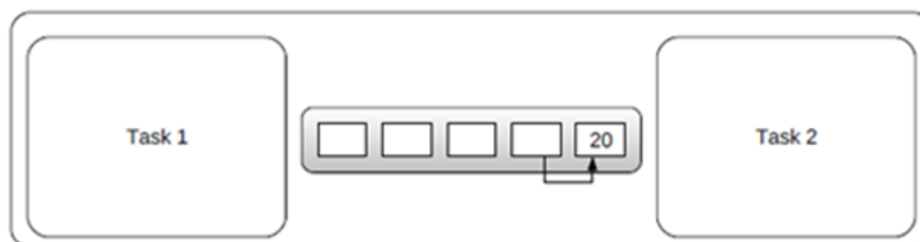
Obrázok 10: Vloženie druhej hodnoty do fronty (prebrané z [2])

Úloha 2 prečíta hodnotu z fronty. Prijme hodnotu na prvom mieste fronty:



Obrázok 11: Prečítanie hodnoty z fronty (prebrané z [2])

Úloha 2 vytiahla hodnotu z fronty a druhá hodnota sa posunie z druhého miesta na prvé miesto fronty. Uvoľní sa miesto vo fronte, máme 4 voľné miesta:



Obrázok 12: Uvoľnenie miesta vo fronte (prebrané z [2])

2.4 TFT-LCD displej

Discovery kit STM32F429 disponuje 2,4 palcovým LCD-TFT displejom. Vzhľadom na malý rozmer displeja som sa rozhodol nahradiť ho 7 palcovým LCD-TFT displejom. LCD-TFT je varianta LCD používajúca TFT technológiu za cieľom zvýšenia kvality obrazu.

Nový a väčší displej DEM 800480J1 TMH-PW-N (A-Touch) disponoval vývodmi, ktorými pôvodný displej nedisponoval. TFT technológia si vyžaduje niekoľko zvláštnych napájacích napätí a tento konkrétny typ nemá integrované meniče, ktorými by si z jedného napájacieho napätia vytvoril. Na všetky potrebné napätia má ale vyvedené vodič, na ktoré je napätie potreba priviesť. Trochu nám to zkomplikovalo stavbu prototypu, ale poradili sme si, viz. kapitola 3.1 Návrh hardvéru.

Ďalej sa jednalo o piny konfiguračné a v poslednom rade piny, po ktorých sa prenášali samotné dáta obrazu. V nasledujúcej tabuľke je zoznam všetkých 50 pinov s popisom ich významu.

| Číslo pinu | Názov | Funkcia |
|------------|-------|--|
| 1-2 | VLED+ | Podsvietenie (anóda) |
| 3-4 | VLED- | Podsvietenie (katóda) |
| 5 | GND | Zem |
| 6 | VCOM | Referencia napätia pre spodnú potenciálovú elektródu (3,4 V) |
| 7 | DVDD | Digitálne napájanie (3,3 V) |
| 8 | MODE | Pin pre zvolenie módu DE (MODE="1") alebo SYNC(MODE="0"). Displej negeneruje synchronizačné pulzy a preto som použil som SYNC mód. |
| 9 | DE | Data enable |
| 10 | VS | Vertikálny synchronizačný vstup. |
| 11 | HS | Horizontálny synchronizačný vstup. |

| | | |
|-------|-------|---|
| 12-19 | B7-B0 | Modré dáta |
| 20-27 | G7-G0 | Zelené dáta |
| 28-35 | R7-R0 | Červené dáta |
| 36 | GND | Zem |
| 37 | DCLK | Hodiny pre displej |
| 38 | GND | Zem |
| 39 | L/R | Vykresľovanie zľava doprava alebo naopak. Podľa orientácie displeja som zvolil L/R. |
| 40 | U/D | Vykresľovanie zhora dolu alebo naopak. Podľa orientácie displeja som zvolil U/D. |
| 41 | VGH | Napätie Gate ON (+18V) |
| 42 | VGL | Napätie Gate OFF (-8V) |
| 43 | AVDD | Analógové napájanie (10,3V) |
| 44 | RESET | Signál reset |
| 45 | NC | Nepripojené |
| 46 | VCOM | Referencia napätia pre spodnú potenciálovú elektródu |
| 47 | DITHB | 6 bitové alebo 8 bitové rozlíšenie. Zvolil som 6 bitové rozlíšenie. |
| 48 | GND | Zem |
| 49-50 | NC | Nepripojené |

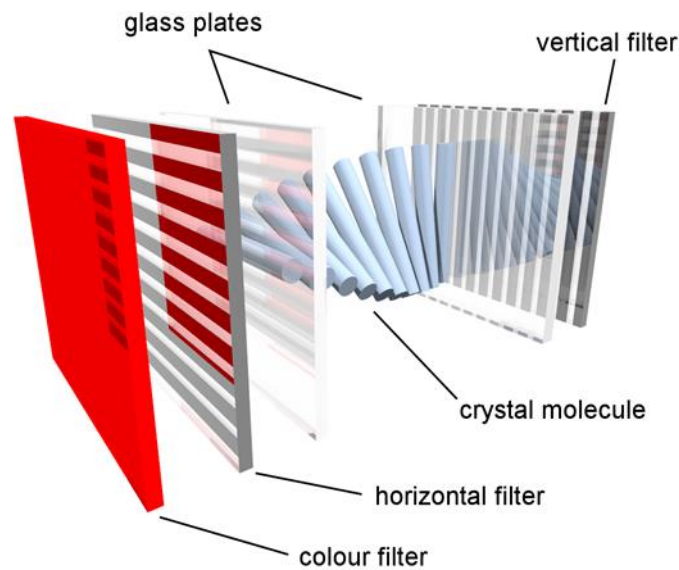
Tabuľka 2: Piny displeja DEM 800480J1 TMH-PW-N (prebrané z [3])

2.4.1 LCD

Skratka LCD je akronym a znamená Liquid Crystal Displej. Tento typ displeja je veľmi tenký a používa sa v tenkých panelových monitoroch. Displej je tvorený farebným filtrom, dvoma polarizačnými filtermi, dvoma vodivými doskami a tekutými kryštálmi tak, ako je zobrazené na obrázku 13.

Polarizačné filtre sú navzájom pootočené o 90 stupňov. Nepolarizované biele svetlo generované diódovým podsvietením (vodiče LED+ a LED-) prejde cez prvý filter, ktorý vytvorí polarizované svetlo. To je druhým filtrom neprepustené a to práve z dôvodu opačnej polarizácie. Privedením napätia na vodivé dosky sa tekuté kryštály otáčajú (usporiadajú) a polarizácia prechádzajúceho svetla sa mení v závislosti na pootočených kryštáloch. Čím je väčšia zmena polarizácie voči vstupnému filteru, tým viac svetla prejde na farebný filter. Pixel displeja je tvorený tromi takýmito konštrukciami, tzv. subpixel, s

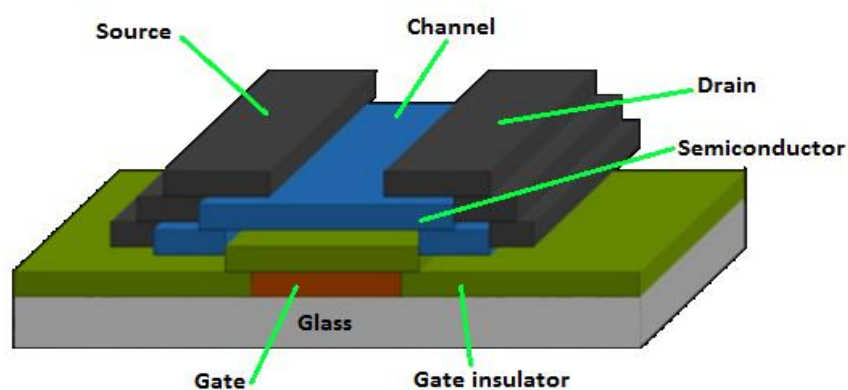
farebným filtrom červenej, modrej a zelenej farby. Výsledná farba je tvorená intenzitami jednotlivých farebných príspevkov.



Obrázok 13: Liquid Crystal Displej (prebrané z [4])

2.4.2 TFT

Skratka TFT je akronym pre Thin-Film Transistor. Jedná sa o aktívnu maticu tranzistorov ovládajúcich jednotlivé subpixely (na rozdiel od pasívnej matice, ktorá nemá tranzistor pre každý subpixel). Na obrázku č. 14 je znázornený tranzistor ktorý sa môže v takejto matici používať.

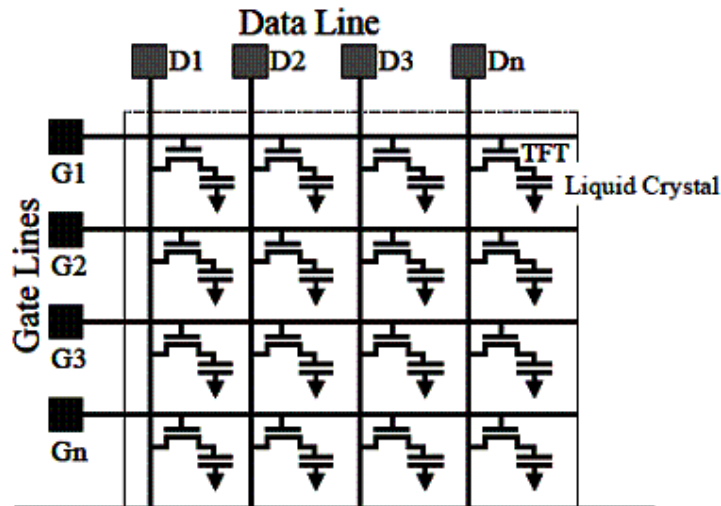


Obrázok 14: Štruktúra Thin-Film Transistor (TFT) (prebrané z [5])

Na obrázku č. 15. je vidieť názorné zapojenie takejto matice. V jednoduchom TFT, napríklad N-kanálový TFT, je za účelom zapnutia privedené pozitívne napätie na gate,

napätie z dátového vodiča je privedené na source tranzistoru a drain spolu s LC (Liquid Crystal) kapacitorom sa nabije na napätie s rovnakou hodnotou.

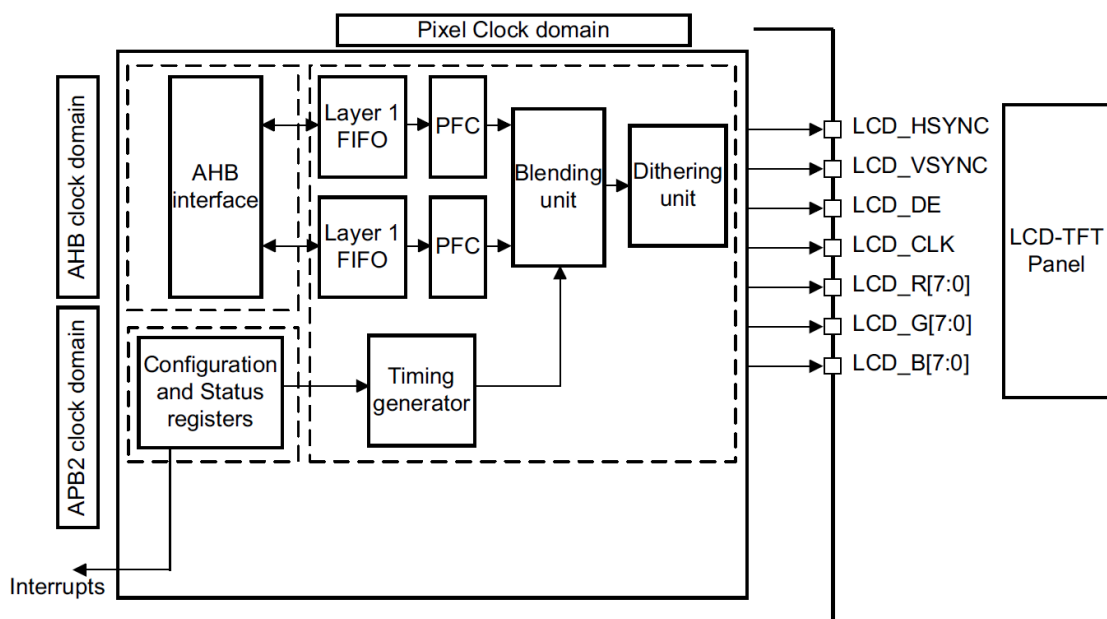
Na vodiče G (Gate Line) sa privádza napätie GATE ON a GATE OFF. Napätie VCOM je privedené na kapacitory.



Obrázok 15: Zapojenie matice LCD TFT displeja (prebrané z [6])

2.4.3 LCD-TFT Controller (LTDC)

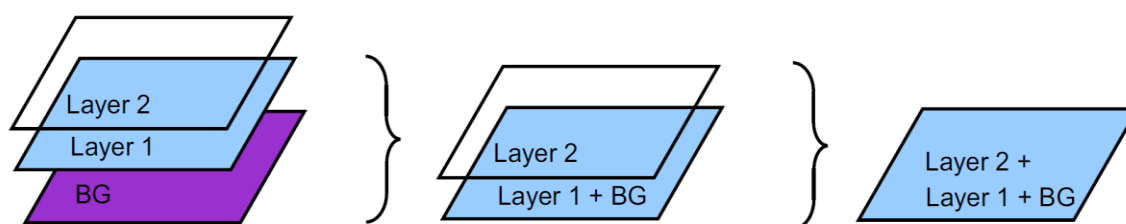
Kontroler displeja LCD-TFT riadi vykresľovanie obrazu na displej. Má 24 bitový paralelný RGB výstup, 8 bitov na pixel, podporuje XGA rozlíšenie (1024x768), programovateľné časovanie vykresľovania obrazu pre rôzne typy displejov, programovateľnú polaritu synchronizačných signálov, podporuje až 8 farebných formátov a iné.



Obrázok 16: Blokový diagram LTDC (prebrané z [8])

Periféria podporuje dve vrstvy, Layer 1 a Layer 2. Tieto vrstvy je možné nezávisle konfigurovať. Obrazové informácie pre tieto vrstvy sú prenášané zbernicou AHB, do registrov sa pristupuje pomocou zbernice APB2. Z jednotlivých vrstiev informácie pokračujú do bloku PFC. Tento blok zabezpečuje prevod vstupného formátu vrstvy na slová, ktoré sú ďalej predávané bloku Blending unit.

Blok blending unit jednoducho spája všetky vrstvy, aj s pozadím, do jedného obrazu, ktorý sa ďalej spracuje ako celok v bloku Dithering unit.



Obrázok 17: Spájanie vrstiev s pozadím (prebrané z [8])

Dithering je proces, ktorý sa používa pri zobrazovaní obrazov na zariadení s nižšou farebnou hĺbkou. Cieľom je, aby pri redukcii ostala zachovaná vizuálna informácia obrazu. Dithering sa snaží túto stratu kompenzovať. Typickým príkladom sú farebné fotografie zobrazené na displeji, ktorý zobrazuje len čiernu a bielu farbu. Svetlejšie časti obrazu majú väčšiu hustotu bielych pixelov a tmavšie časti obrazu majú väčšiu hustotu čiernych pixelov. Tento proces vytvára ilúziu, že máme viac odtieňov šedej farby aj napriek tomu, že zobrazovací displej zobrazuje len čiernu a bielu farbu.

2.4.4 Synchronné časovanie displeja

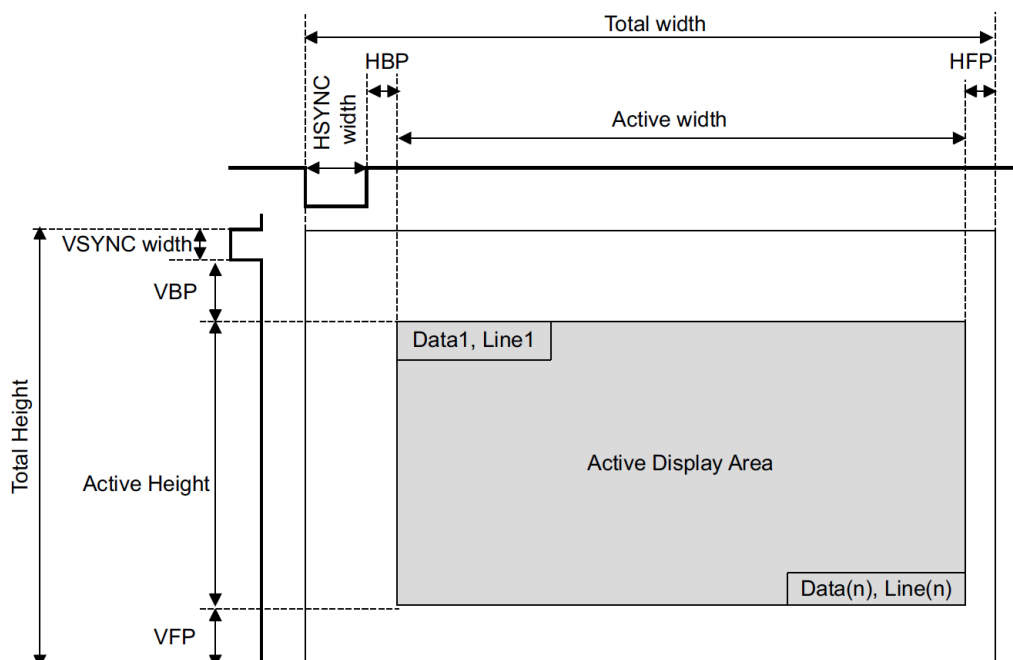
K správne chodu displeja je potrebné mať správne nastavené synchronizačné pulzy. Každý displej má tieto informácie v datasheete. Pre displej použitý v mojej diplomovej práci sú hodnoty zobrazené v nasledujúcej tabuľke.

| Item | Symbol | Min | Typ | Max | Unit |
|-------------------------|--------------|-----|-----|------|------|
| Horizontal display area | Active width | - | 800 | - | DCLK |
| Dclk frequency | Fclk | - | 30 | 50 | MHz |
| One horizontal line | Total width | 889 | 928 | 1143 | DCLK |

| | | | | | |
|-----------------------|---------------|-----|-----|-----|-------------|
| HS pulse width | HSYNC | 1 | 48 | 255 | DCLK |
| HS blanking | AHBP | - | 88 | - | DCLK |
| HS front porch | HFP | 1 | 40 | 255 | DCLK |
| Vertical display area | Active height | - | 480 | - | Total width |
| VS period time | Total height | 513 | 525 | 767 | Total width |
| VS pulse width | VSYNC | 3 | 3 | 255 | Total width |
| VS blanking | AVBP | - | 32 | - | Total width |
| VS front porch | VFP | 1 | 13 | 255 | Total width |

Tabuľka 3: Tabuľka synchronizačných pulzov displeja (prebrané z [3])

Podľa datasheetu displeja [3] je jeho veľkosť 800 x 480 pixelov, táto hodnota však popisuje len aktívnu oblasť displeja, na ktorej sa vykresľuje obraz. Z hľadiska softvérového, je displej väčší, a to práve o oblasť, tzv. blanking display area, ktorá je vytvorená horizontálnymi a vertikálnymi synchronizačnými pulzami.



Obrázok 18: Znáznornenie časovania displeja (prebrané z [8])

Základnú frekvenciu Dclk použitého displeja som najskôr nastavil na 30 MHz, takže perióda hodinového signálu DCLK je 33,33 ns(1).

$$DCLK = \frac{1}{DCLK} = \frac{1}{30 * 10^6} = 33,33 \text{ ns (1)}$$

Ostatné parametre som nastavil na typické podľa tabuľky uvedenej v datasheete [3] tabuľka 3. Do registrov displeja je nutné vložiť súčty jednotlivých hodnôt parametrov displeja. Každú hodnotu je nutné po jej výpočte ešte dekrementovať o 1. Výpočty a vzorce sú zobrazené v nasledujúcej tabuľke.

| Premenná | Výpočet | Hodnota |
|-----------------------------------|---------------------------------------|---------|
| Horizontal Sync | HSYNC - 1 | 47 |
| Vertical Sync | VSYNC - 1 | 2 |
| Accumulated horizontal back porch | AHBP - 1 = HSYNC + HBP - 1 | 87 |
| Accumulated vertical back porch | AVBP - 1 = VSYNC + VHP - 1 | 31 |
| Accumulated active width | Active width + HSYNC + HBP - 1 | 887 |
| Accumulated active height | Active height + VSYNC + VBP - 1 | 511 |
| Total width | Active width + HSYNC + HBP + HFP - 1 | 927 |
| Total height | Active height + VSYNC + VBP + VFP - 1 | 524 |

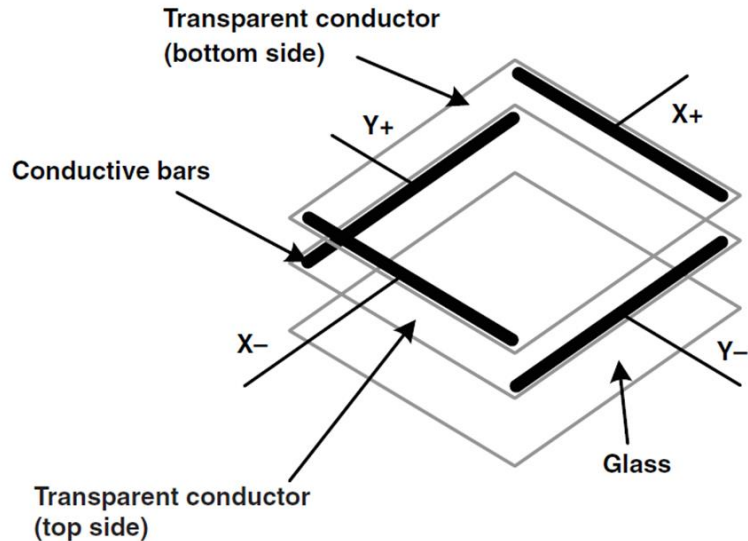
Tabuľka 4: Vypočítané hodnoty pre správny beh displeja

Všetky hodnoty som podľa tabuľky tabuľky 4 zvolil v správnom rozmedzí, po tomto nastavení displej funguje správne. Všetky hodnoty sú v zdrojovom kóde uložené ako makrá, takže pri použití iného displeja je zmena parametrov jednoduchá.

2.4.5 Odporová dotyková plocha

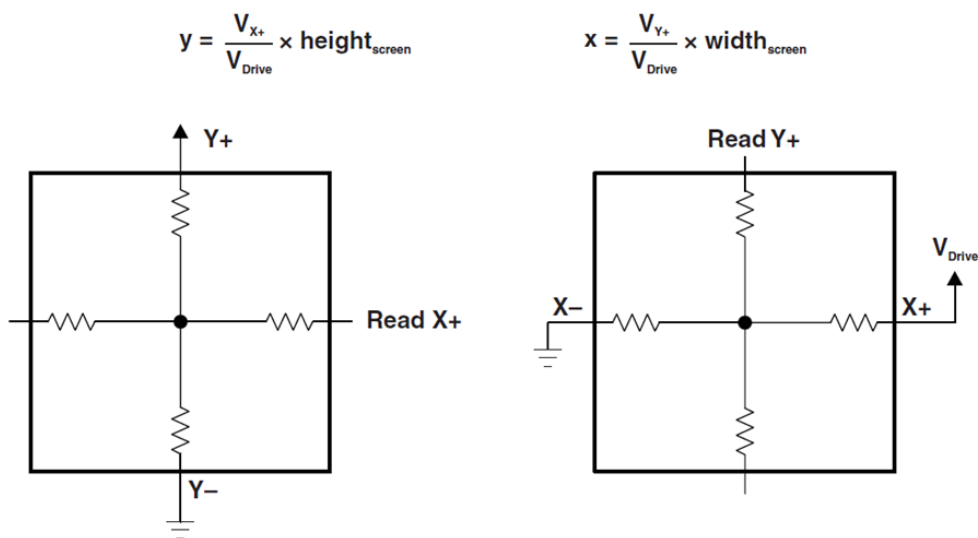
Pre odporovú dotykovú plochu displeja je vyvedený páskový kábel pre 4 vodiče. Ten je privedený do radiča STMPE811QTR, ktorý sa nachádza na doske discovery kitu. Tento radič umožňuje až 8 vodičové pripojenie, avšak vzhľadom k nášmu 4 vodičovému dotykovému panelu sú 4 vodiče privedené na zem. Štvorvodičový odporový dotykový displej je zobrazený na obrázku 19.

STMPE811 periodicky vyčítava súradnice dotykovej matrice a STM32 ich periodicky cez I²C vyčítava (práca STMPE driveru).



Obrázok 19: Konštrukcia odporového dotykového displeja (prebrané z [8])

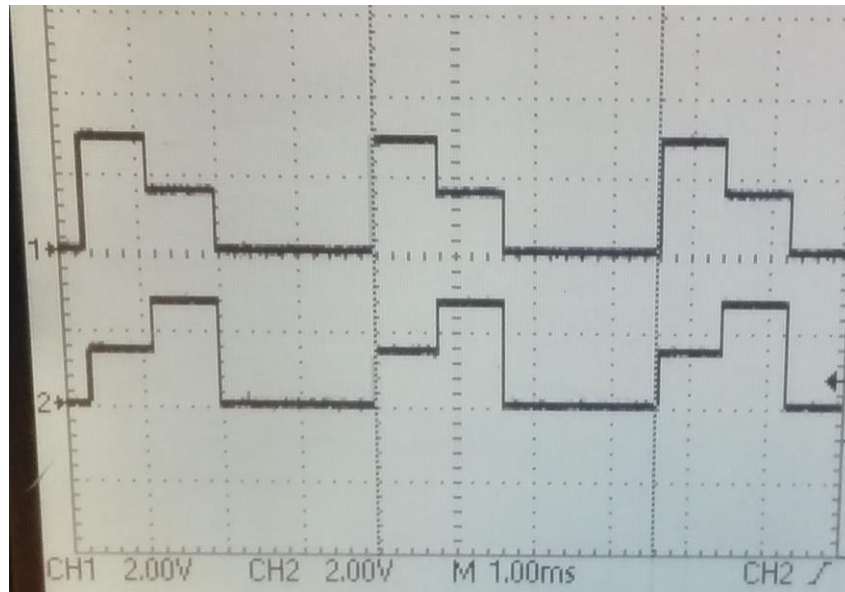
Princíp fungovania odporovej plochy je jednoduchý. Je tvorený dvomi odporovými vrstvami, ktoré sú vzájomne odizolované a je medzi nimi nekonečný, alebo veľmi veľký odpor (rádovo mega ohmy). Pri dotyku sa tieto dve vrstvy spoja, vzniká vodivý prepoj. Tým sa umožní merať napätie v danom mieste. Z toho sa ďalej odvodí miesto dotyku tak, ako je zobrazené na obrázku 20.



Obrázok 20: Čítanie súradníc z 4 vodičového dotykového displeja (prebrané z [8])

Súradnice X a Y z odporového dotykového displeja sú čítané v dvoch krokoch. V prvom kroku je na Y+ privedené napätie a Y- privedené na zem. Na vodiči X+ sa zmeria napätie. Pomer zmeraného napätia a napätia privedeného na Y+ nám určí, kde na dotykovom poli došlo k spojeniu spodnej a hornej vrstvy. X súradnica je vypočítaná analogicky, na X+ je privedené napätie a X- je privedené na zem. Na vodiči Y+ sa zmeria

napätie a pomerom zmeraného napätia a napätia privedeného na vodič X+ sa vypočíta, kde došlo k dotyku. Tento proces aj s výpočtom súradníc je zobrazený na obrázku 20.



Obrázok 21: Signály 4 vodičového dotykového displeja

Na obrázku 21 sú zobrazené signály, ktoré sú prenášané vodičmi 4 vodičového dotykového displeja. Prenos je rozdelený do 4 častí. Keďže v našom prípade používame len 2 páry zo 4, tak sú napätia zobrazené len v prvých dvoch štvrtinách.

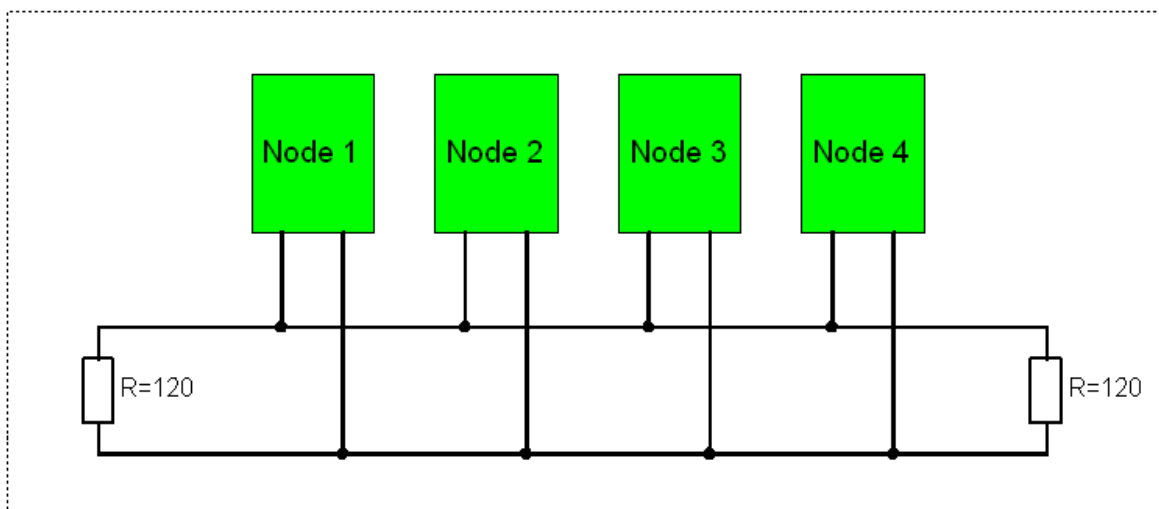
V pravej štvrtine procesu merania polohy dotyku sa na X pár privedie napätie približne 3,3 V. Na Y páre sa meria napätie v mieste dotyku. Ak je dotyk úplne vľavo tak na Y nameriame 3,3 V, úplne vpravo nameriame 0 V.

V druhej štvrtine je proces analogický. Na Y pár sa privedie napätie 3,3 V. Na X páre sa meria napätie v mieste dotyku. Ak je dotyk úplne dole tak nameriame 3,3 V, úplne hore nameriame 0 V.

2.5 CAN zbernica

CAN zbernica sa začala používať v 80. rokoch hlavne v automobilovom priemysle. Je to typ multimaster zbernice, kde sú v systéme všetky uzly rovnocenné. Prijímacie a vysielačie uzly nemajú vlastnú adresu. Rámce, niekedy označené aj ako správy, ktoré posielajú na zbernicu, majú jedinečný identifikátor, ktorý identifikuje obsah danej správy. Po prijatí správy prebieha u každého zariadenia filtrácia práve na základe tohto identifikátoru. V závislosti na filtrácii je správa ďalej spracovaná alebo zahodená. Takto sa ku každému uzlu dostanú len správy, ktoré sú pre neho významné.

2.5.1 Fyzická vrstva



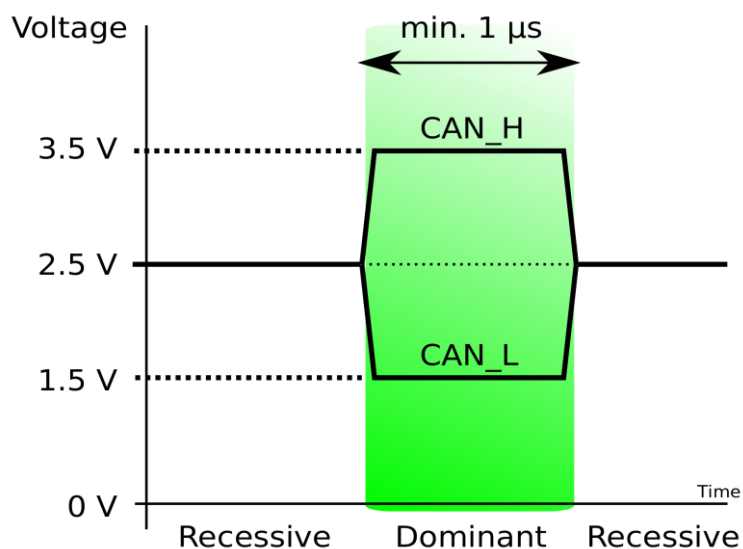
Obrázok 22: Zapojenie uzlov do zbernice (prebrane z [9])

Ako je vidieť na obrázku 22, k prenosu dát sa používa symetrické vedenie s charakteristickou impedanciou 120 ohm. Aby sa zabránilo odrazom, sú na oboch koncoch vedenia pripojené zakončovacie 120 ohmové odpory.

2.5.2 Úrovne signálu

Pôvodná špecifikácia zbernice CAN neobsahovala žiadne požiadavky na prenosové médium ani na napäťové úrovne. Sú definované len dve formálne hodnoty napäťových úrovní a to recesívna a dominantná.

Recesívna, kľudová úroveň znamená, že na oboch vodičoch je 2,5 V. Dominantná úroveň znamená, že jeden vodič je na 1,5 V a druhý na 3,5 V.

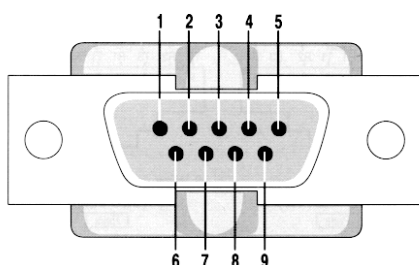


Obrázok 23: Recesívna a dominantná úroveň signálu (prebrané z [10])

Ak sú k zbernici pripojené viaceré stanice, a aspoň jedna z nich vysiela dominantnú úroveň, zatiaľ čo ostatné vysielajú recesívnu úroveň, na zbernici sa zobrazí dominantná úroveň. Ak všetky stanice vysielajú recesívnu úroveň, zostáva na zbernici recesívna úroveň.

2.5.3 Konektor

Vzhľadom na to, že k prenosu potrebujeme len 2 dráty, CAN_H a CAN_L, tak konektor môže byť rôzneho druhu. Najčastejšie sa však stretáme s 9 pinovým CANNON konektorom no môžu sa používať aj iné, ako napríklad konektor RJ 45. Na obrázku 24 je znázornené zapojenie jednotlivých pinov na konektore.



Obrázok 24: 9 Pinový konektor CANNON (prebrané z [11])

| Pin | Signál | funkcia |
|-----|------------|-------------------|
| 1 | - | Rezervované |
| 2 | CAN_L | Dominantný (low) |
| 3 | CAN_GND | Zem |
| 4 | - | Rezervované |
| 5 | (CAN_SHLD) | Voliteľné |
| 6 | GND | Voliteľné |
| 7 | CAN_H | Dominantný (high) |
| 8 | - | Rezervovaný |
| 9 | (CAN_V+) | Voliteľné |

Tabuľka 5: Pripojenie pinov konektoru CANNON

2.5.4 Metóda prístupu

Na zbernici CAN sa používa metóda prístupu CSMA/CR. Princíp metódy spočíva v tom, že každý uzol neustále sníma stav zbernice. V okamihu, keď detekuje kľudový stav zbernice, pokúsi sa vysielať. Môže nastať prípad, kedy sa o vysielenie pokúsi 2 a viac uzlov súčasne, vtedy dochádza k tzv. arbitrácii. Arbitrácia je proces, pri ktorom sa rozhodne, kto získa prístup k zbernici a s tým možnosť vysielať.

2.5.5 Typy rámcov

V komunikačnom protokole CAN sa rámce delia na 4 typy:

Data frame

Dátový rámec zabezpečuje prenos informácií z vysielaného uzla na zbernicu pre všetky ostatné uzly pripojené k zbernici.

Remote frame

Žiadosť o rámec má obdobný formát ako dátový rámec, no neobsahuje dátové pole a RTR bit je v logickej 1, recesívny stav (v dátovom rámci má dominantný stav). Účel tohto rámca je žiadosť iného uzla na zbernici o vyslanie dátového rámca s totožným identifikátorom, aký je v žiadosti.

Error frame

Chybový rámec sa skladá z dvoch častí, ERROR FLAG a ERROR DELIMITER. Uzol, ktorý identifikuje chybu prijímaných bitov začne vysieľať za sebou 6 dominantných bitov, čím poruší štruktúru rámca. Ostatné uzly začnú vysieľať tiež 6 dominantných bitov a tak celková dĺžka ERROR FLAGu môže byť od 6 do 12 bitov. Za ERROR FLAG nasleduje ERROR DELIMITER ktorý sa skladá z 8 recesívnych bitov.

Overload frame

Tento rámec má analogickú štruktúru ako chybový rámec. Uzol tento rámec vysieľa hlavne z dôvodu nedostatku času na spracovanie predchádzajúcej správy.

2.5.6 Štruktúra dátového rámca

Štruktúra rámca je pre verzie 2.0 A a 2.0 B takmer úplne rovnaká. Rozdiel je len v dĺžke identifikátoru. Pri verzii 2.0 A je dĺžka identifikátoru 11 bitov a pri verzii 2.0 B, tzv. rozšírená verzia (extended version), je dĺžka identifikátoru 29 bitov. Mimo rozdielnej dĺžky identifikátoru, sú štruktúry 2.0 a 2.0 B totožné. Rámec obsahuje tiež dátové pole, ktoré má dĺžku od 0 do 64 bitov. V tabuľke 6 je znázornená štruktúra rámca verzie 2.0 A.

| | | | | | | | | | | | |
|-----|------|-----|-----|-----|-----|--------|------|------|------|------|-----|
| 1 b | 11 b | 1 b | 1 b | 1 b | 4 b | 0-64 b | 15 b | 1 b | 1 b | 1 b | 7 b |
| SOF | AF | RTR | IDE | RB | DLC | DATA | CRC | CRCD | ACKS | ACKD | EOF |

Tabuľka 6: Štruktúra rámca 2.0 A

V prvom riadku sú popísané dĺžky jednotlivých polí a v druhom riadku sú skratky konkrétnych polí. Ich význam je popísaný v tabuľke 7.

| Názov poľa | Funkcia |
|---|---|
| SOF – Start Of Frame | Určuje začiatok rámca/správy. Vždy uložená logická 0(dominantný bit). |
| AF – Arbitration Field | Arbitračné pole obsahuje identifikátor vysielanej správy. Na základe hodnoty v tomto poli sa rozhoduje, ktorý uzol vyhrá arbitráž. |
| RTR – Remote Transmission Request | Logická 0 (dominantný stav) identifikuje dátový rámec. |
| IDE – Identifier Expresion | Rozlišuje základný (2.0 A) alebo rozšírený (2.0 B) formát správy. |
| RB – Reserved Bit | Rezervovaný bit, pre prípadné využitie budúcich štandardov. |
| DLC – Data Length Code | Hodnota určuje koľko bajtov bude v dátovom poli. |
| DATA - Data | Dátové pole o veľkosti 0 až 8 bajtov. |
| CRC – Cyclic Redundancy Code | 15 kontrolných bitov cyklického redundantného kódu pri zahrnutí všetkých predchádzajúcich polí. |
| CRCD – Cyclic Redundanci Code Delimeter | Ohraničenie poľa CRC logickou 1 (recesívny bit) |
| ACKS – Acknowledge Slot | Potvrdzujúce pole. Vysielač vysielal tento bit ako recesívny. Ak aspoň jeden uzol prijal správu bez chyby, prepíše tento bit na dominantný, čím sa potvrdí prijatie správy vysielaču. |
| ACKD - Acknowledge Delimeter | Ohraničenie poľa ACKS logickou 1 (recesívny bit) |
| EOF – End Of Frame | Koniec rámca sa skladá z najmenej 7 logických 1 (recesívnych bitov) |

Tabuľka 7: Polia správ a ich funkcie

Medzi jednotlivými správami nasledujú po poli EOF najmenej 3 bity pre ukľudnenie všetkých vysielačov. V tejto dobe môžu prijímacie uzly informovať vysielačie uzly o chybách prenosu.

Protokol v tejto diplomovej práci pozostáva zo 7 správ ktoré sa prenášajú po zbernici. Dve z nich sú v smere od VCU, ID 0x101 a ID 0x111. Ostatných 5 správ je od HVAC, ID 0x181, 0x191, 0x1A1, 0x1B1 a ID 0x1C1. Podrobný popis bitov a správ v práci nie je uvedený, pretože sa jedná o interný dokument spoločnosti POLL s.r.o.

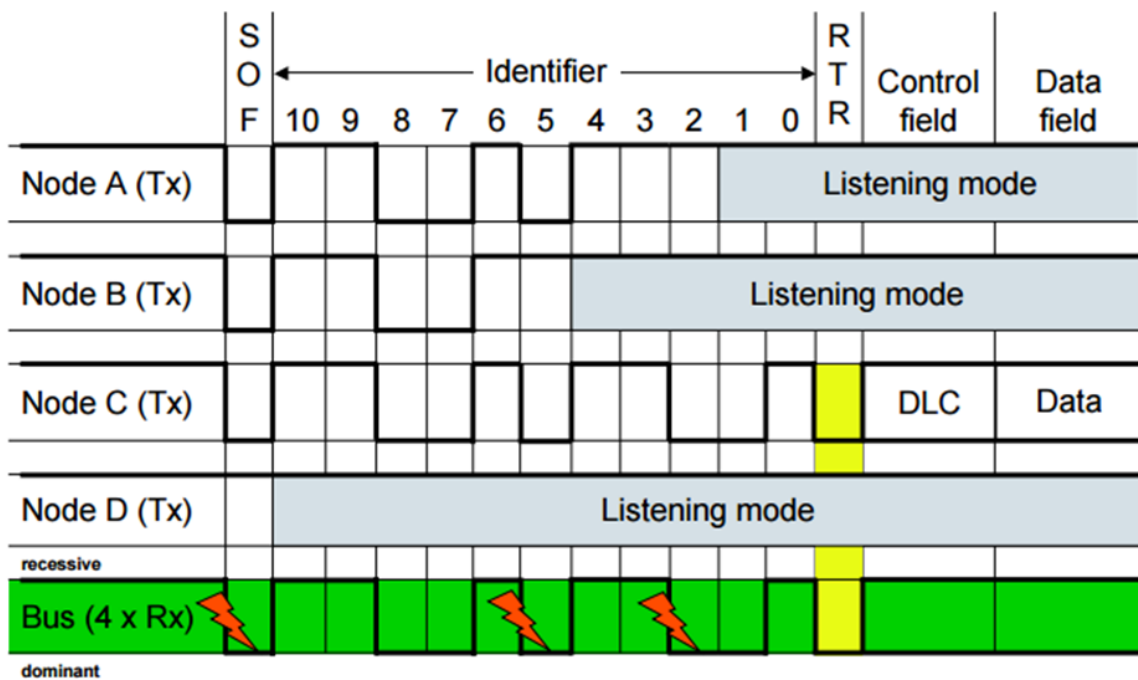
Tieto správy sú vysielané v oboch smeroch, v pravidelných intervaloch. Tým sa zabezpečuje nielen pravidelný a aktuálny stav systému ale tiež informácia, či zariadenie neprestalo komunikovať. Program je navrhnutý tak, aby po neprijatí 10 správ po sebe nasledujúcich reprezentoval stav HVAC ako offline. Tým zariadenie indikuje, že je komunikácia s klimatizačnou jednotkou prerušená.

2.5.7 Arbitrácia

Ako je spomenuté už vyššie, uzly kontrolujú stav zbernice a v prípade, že nedetekujú prenos po zbernici sa pokúsia vyselať správu. Za takýchto okolností môže nastať stav, kedy sa rozhodnú vyselať správu viaceré uzly súčasne.

Vtedy nastáva proces arbitrácie. Počas vysielania bitov, dominantných a recesívnych úrovní, každý uzol sleduje stav zbernice. Ak uzol, ktorý vysielala na zbernicu recesívnu úroveň ale detekuje na zbernici úroveň dominantnú, musí zastaviť proces vysielania, prehráva arbitráciu. Takýmto procesom sa zabezpečí to, že dôležitejšia správa, tzn. správa s vyšším identifikátorom, dostáva prednosť prenosu po zbernici.

Takáto arbitračná metóda vyžaduje, aby všetky uzly pripojené na zbernicu boli synchronizované a pripravené k vzorkovaniu v rovnakú dobu. Z tohto dôvodu je niekedy zbernica CAN nazývaná synchrónnou zbernicou. Takýto termín však nieje úplne presný, pretože všetky dáta sú odoslané bez hodinového signálu.



Obrázok 25: Príklad arbitrácie (prebrané z [12])

Každý uzol generuje interný hodinový signál, ktorý je synchronizovaný s vysielajúcou stanicou pomocou dátového signálu. Z tohto dôvodu je po 5 po sebe nasledujúcich bitoch rovnakej logickej hodnoty vložený jeden dodatočný bit opačnej logickej hodnoty. Podobný princíp sa používa napríklad v USB zbernici.

2.6 Basic Extended Controller area network

Basic Extended CAN, skrátene bxCAN, je periféria discovery kitu ktorá podporuje protokol CAN. Periféria je navrhnutá tak, aby bola schopná prijať veľké množstvo správ s minimálnym zaťažením CPU. Medzi hlavné vlastnosti bxCAN patrí podpora verzie 2.0A a 2.0B, maximálna rýchlosť je 1 Mbit/s, obsahuje 3 mailboxy, možnosť konfigurácie priority, dve prijímacie FIFO fronty a škálovateľné filtre.

2.6.1 Filtre

Po prijatí správy sa uzol musí rozhodnúť, v závislosti na hodnote identifikátoru správy, či je správa smerovaná uzlu alebo nie. V prípade, že je správa smerovaná uzlu, je skopírovaná do pamäte SRAM. V opačnom prípade je správa zmazaná bez zásahu softvéru.

Pre splnenie vyššie spomínaných požiadaviek je bxCAN vybavený 28 konfigurovateľnými a škálovateľnými bankami filtrov. Takáto filtrácia na hardvérovej úrovni značne šetri CPU, pretože by inak musela prebiehať na softvérovej úrovni. Každá filtračná banka pozostáva z dvoch registrov, CAN_FxR0 a CAN_FxR1.

Pre optimalizáciu a prispôsobenie filtrov potrebám aplikácie môže byť veľkosť každej filtračnej banky konfigurovaná samostatne. V závislosti na veľkosti filtračnej banky je:

- Jeden 32 bitový filter
- Dva 16 bitové filtre

Tieto filtre môžu byť konfigurované ako maska alebo identifikátor. Pri porovnávaní s filtrom sa porovnávajú polia identifikátoru (2.0 A a 2.0 B), IDE a RTR.

Mód masky

Maska je 32 bitové alebo 16 bitové číslo ktoré určuje, ktoré bity sú pre nás validné. Princíp je podobný ako pri práci s maskami v IP sieťach. Logickým súčinom bitov prijatej správy s nami definovanou maskou získame bitové vyjadrenie, ktoré bxCAN ďalej porovnáva s nami preddefinovanou hodnotou identifikátoru, IDE a RTR. Ak sa hodnoty zhodujú, správa bude prijatá, v opačnom prípade je zahodená.

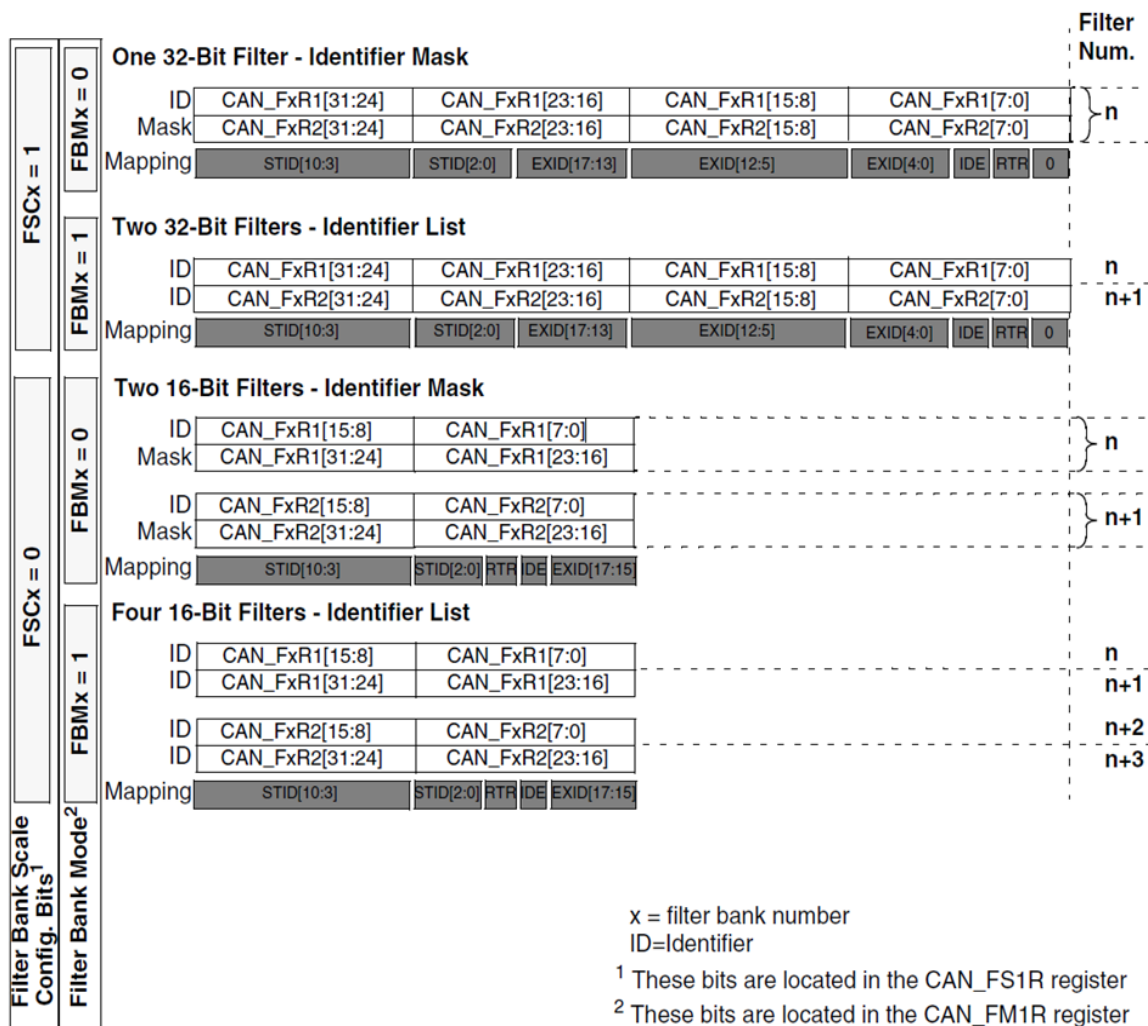
Táto metóda je vhodná pre vyčlenenie určitého spektra správa, ktoré chceme prijímať.

Mód identifikátoru

Mód identifikátoru funguje na princípe porovnávania všetkých bitov prijatej správy s bitmi nášho filtru. V prípade, že sú všetky bity správy zhodné s bitmi nášho identifikátoru, bude správa akceptovaná.

Táto metóda je vhodná pre určenie konkrétnej správy, ktorú chceme prijať.

Na nasledujúcom obrázku sú znázornené jednotlivé módy a mapovanie bitov v registroch.

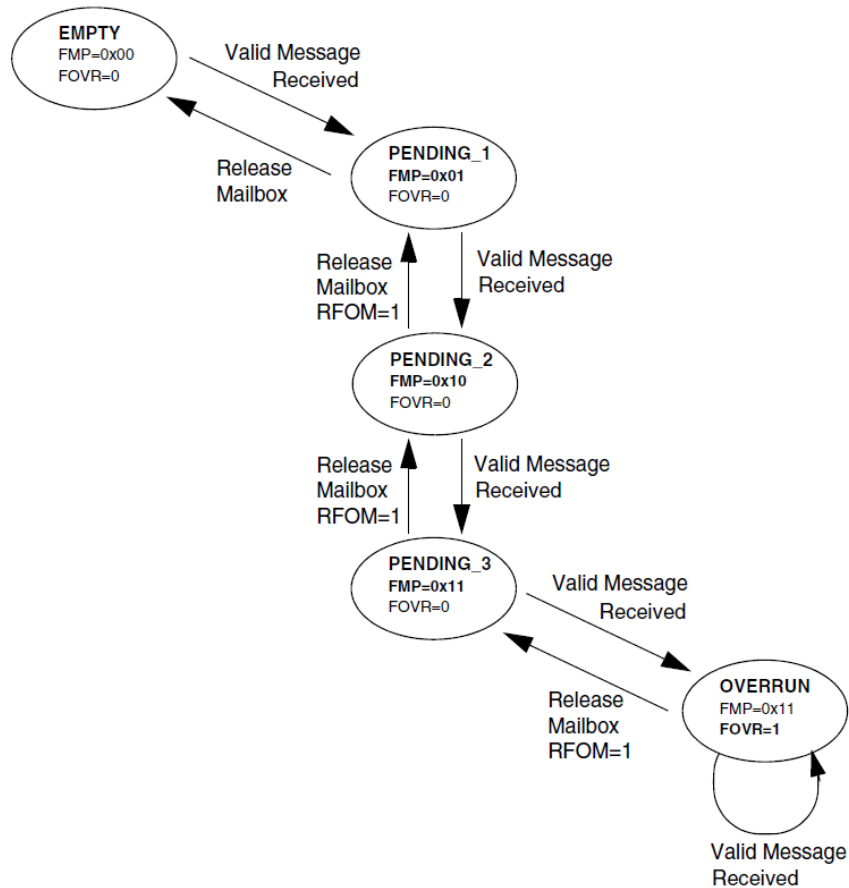


Obrázok 26: Konfigurácia filtrov – organizácia registrov (prebrané z [8])

2.6.2 Prijímanie správ

Prijím CAN správ je použitá FIFO fronta, zložená z troch mailboxov. S cieľom šetrenia výkonu CPU, jednoduchosti softvéru a garantovania konzistencie je FIFO riadené kompletne hardvérovo. Aplikácia pristupuje k správam uloženým vo FIFO fronte cez výstupný mailbox.

Mailbox si môžeme predstaviť ako miesto vo fronte, na ktorom môže byť správa uložená. FIFO fronta obsahuje 3 mailboxy, v ktorých môže byť uložená správa. Po prijatí správy sa tieto mailboxy plnia. Pri spracovaní sa správa z mailboxu vyberie a fronta sa posunie. V prípade, že sú všetky mailboxy zaplnené, dochádza k zahodeniu správy. Na obrázku 27 je znázornená FIFO fronta s 3 mailboxami.



Obrázok 27: Príjímacie stavy FIFO fronty (prebrané z [8])

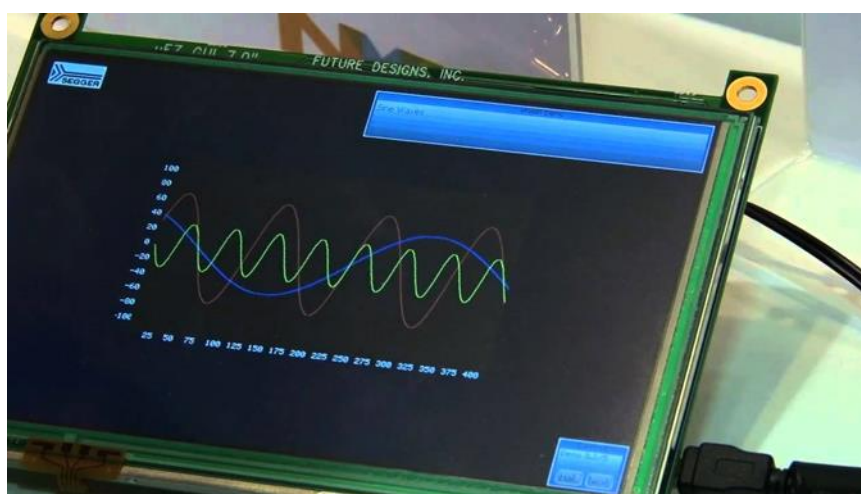
2.7 STemWin

K vytvoreniu grafického rozhrania (GUI) bola použitá knižnica STemWin. Táto profesionálna knižnica umožňuje vytvárať GUI s akýmkoľvek STM32, LCD/TFT displejom a LCD/TFT kontrolérom. Táto knižnica je poskytovaná v skompilovanej forme (C knižnica, pre náš účel sme použili libSTemWin522_4x9i_CM4_OS_GCC.a ktorá je predkompilovaná pre použitie s operačným systémom (v našom prípade FreeRTOS) pre prekladač GCC) pre akéhokoľvek STM32 zákazníka zadarmo.

Knižnica ponúka rozsiahle riešenie s množstvom vlastností ako JPG, GIF a PNG dekodovanie, množstvo widgetov (ako napríklad zaškrťavacie políčka, tlačítka, lišty, kalendáre, meradlá) a taktiež nástroj GUIBuilder, ktorý umožňuje jednoduché vytváranie obrazoviek.



Obrázok 28: Ukážka GUI vytvoreného knižnicou StemWin (prebrané z [13])



Obrázok 29: Ukážka vykreslenia grafu knižnicou StemWin (prebrané z [14])

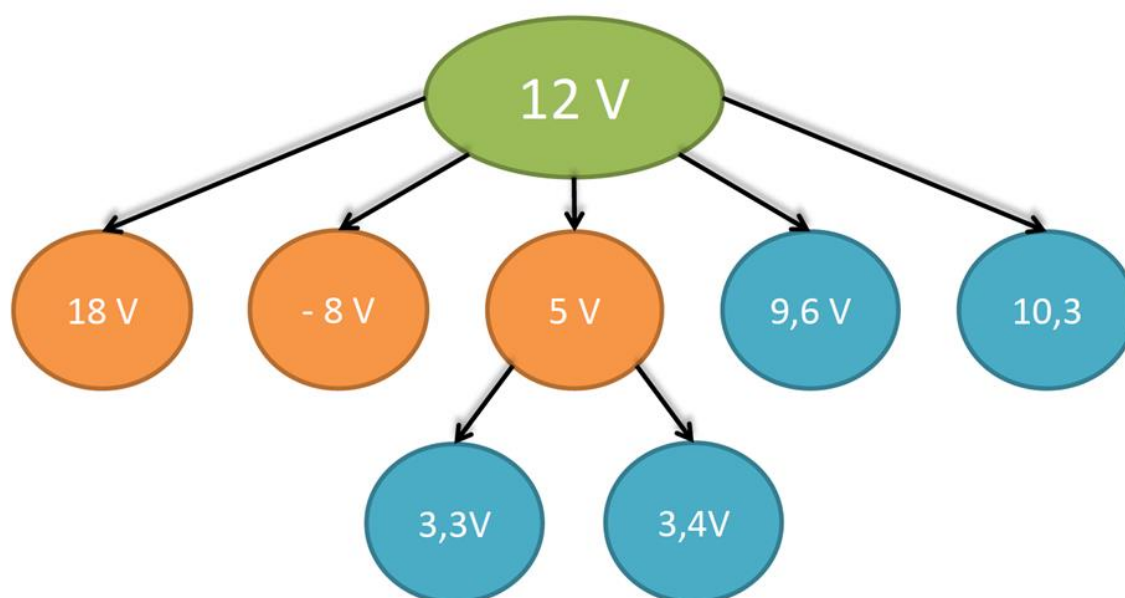
3 Praktická časť

3.1 Návrh hardvéru

Pripojenie nového displeja viedlo nielen k prepojeniu jednotlivých pinov z discovery kitu na displej ale tiež realizáciu zdrojov, ktoré boli nevyhnutné k správne mu behu displeja. Predávajú sa čipy, ktoré generujú nami požadované napätia avšak ich cena je veľmi vysoká. Návrh a zostrojenie zdrojov nie je časovo ani finančne náročné. Cena všetkých súčiastok sa pohybuje okolo 350 Kč čo je asi pätina ceny čipu, ktorý by takéto signály generoval.

3.1.1 Napätové zdroje

Displej potreboval 6 zdrojov napájania -8V; 3,3 V; 3,4 V; 9,6 V; 10,3 V a 18 V, ktoré som odvodil z 12 V adaptéra. Z adaptéra som odvodil tiež 5 V, ktoré som použil pre discovery kit, ktorá bola počas vývoja napájaná cez USB. Celkovo som teda zrealizoval 3 spínané zdroje a 4 lineárne.



Obrázok 30: Odvodenie napätí z adaptéra

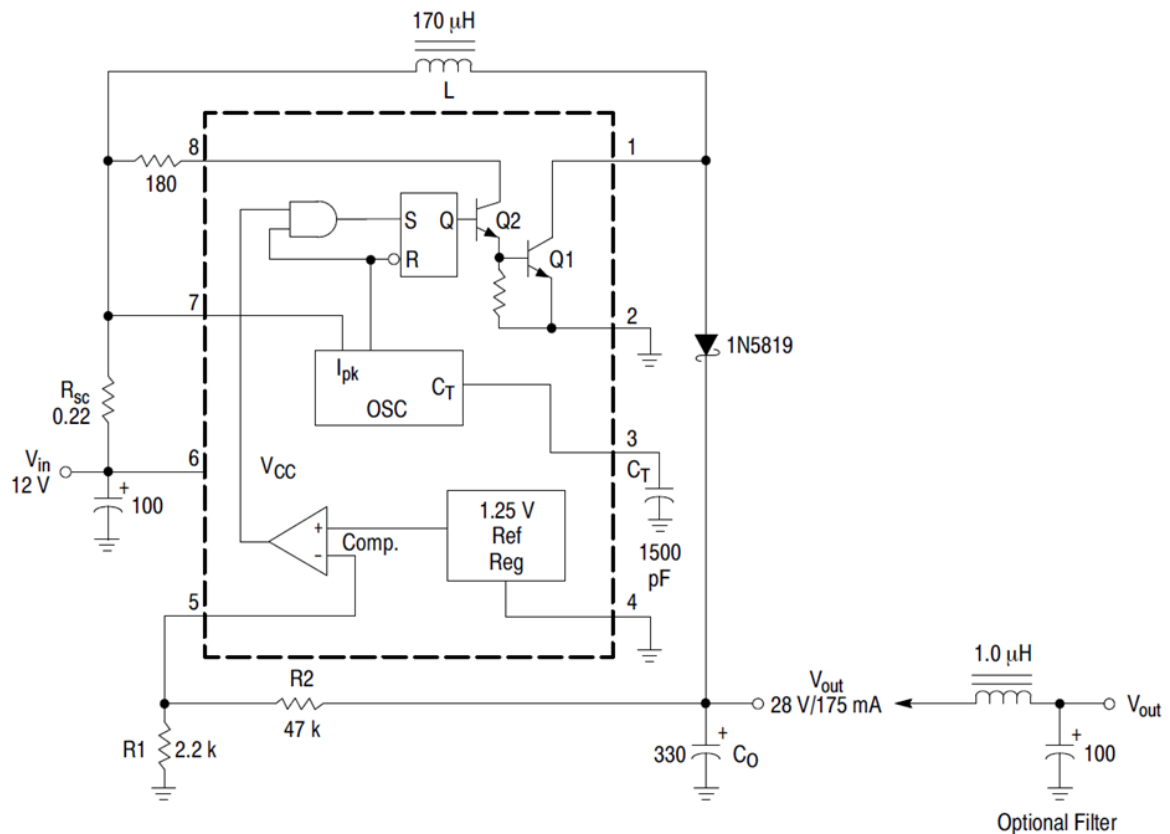
Na obrázku je znázornené, ako som odvodil napätia z adaptéra. V zelenej bubline je 12 V, ktoré získavam z adaptéra. Z 12 V sú odvedené 3 spínané zdroje (oranžové) a 2 lineárne zdroje (modré). Zo spínaného 5 V zdroja som odvodil ešte dva lineárne.

Napätia - 8 V, 5 V a 18 V sú získané spínaným meničom s obvodom MC34063. Napätia 3,3 V; 3,4 V; 9,6 V a 10,3 V sú získané lineárnym meničom s obvodom LM317.

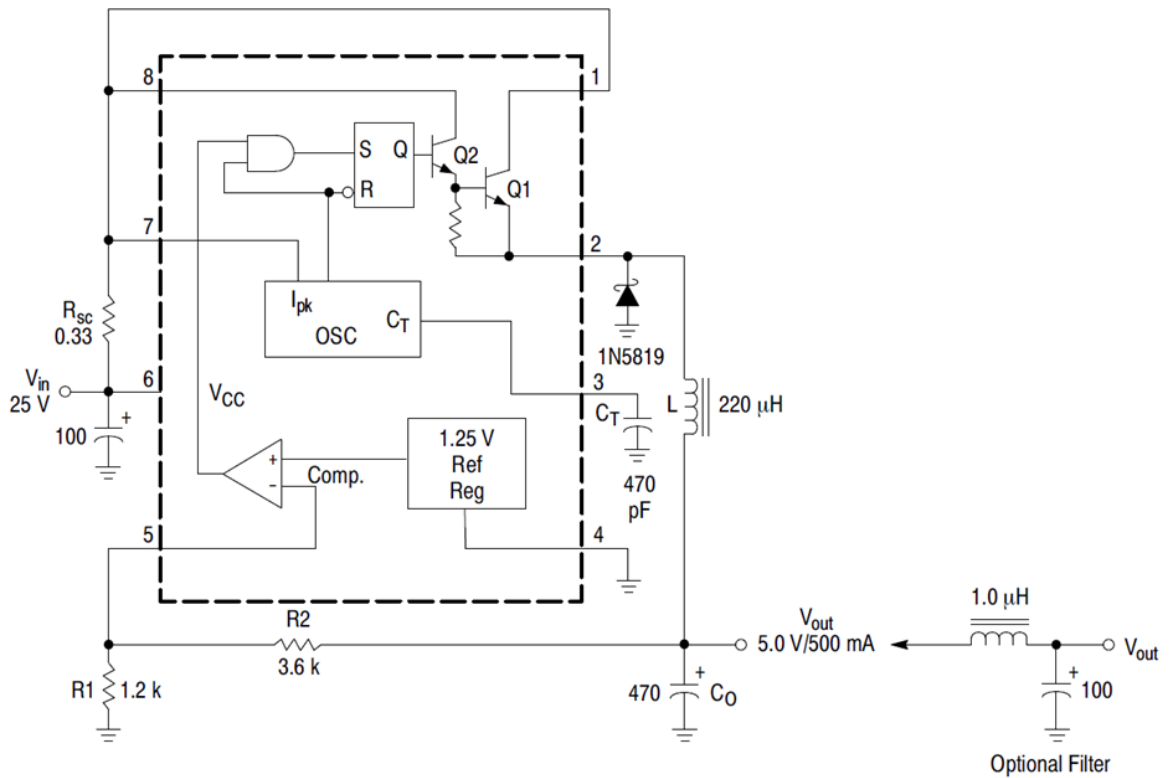
Na výstupe zdrojov je zapojená zelená dióda ako indikátor, že zdroje fungujú.

3.1.2 Spínaný menič s MC34063

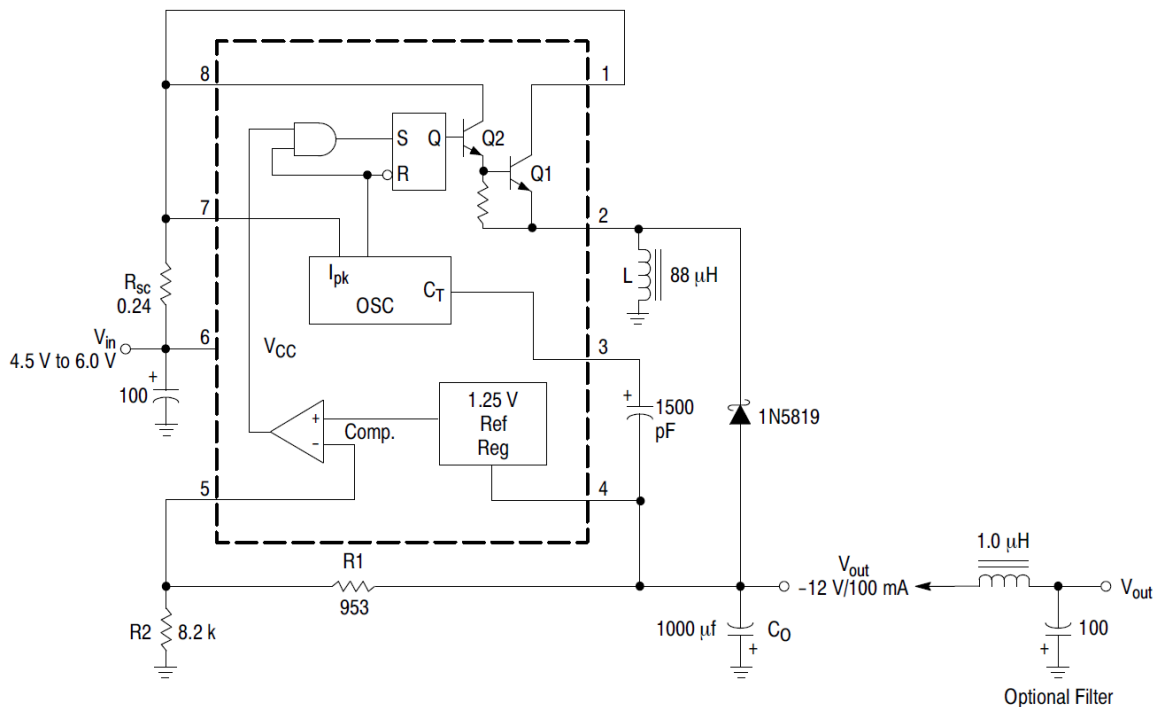
K získaniu napätí som použil zapojenie s obvodom MC34063. Vzhľadom na napätia som musel použiť 3 rôzne zapojenia a to rastúci prevodník pre získanie 18 V z 12 V, klesajúci prevodník pre získanie 5 V z 12 V a v poslednom rade aj invertujúci prevodník na získanie - 8 V z 12 V. Zapojenia na nasledujúcich obrázkoch vychádzajú z katalógového zapojenia, ktoré označuje aj jeho vnútorné zapojenie.



Obrázok 31: Schéma zapojenia rastúceho prevodníku s obvodom MC34063 (prebrané z [15])

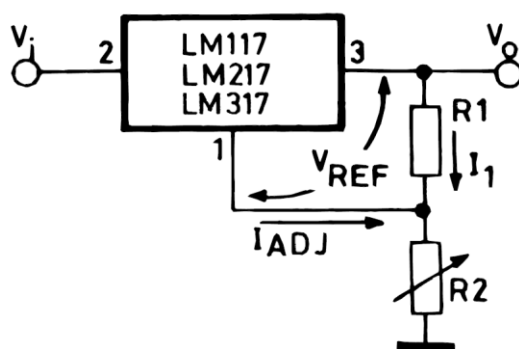


Obrázok 32: Schéma zapojenia klesajúceho prevodníku s obvodom MC34063 (prebrané z [15])



Obrázok 33: Schéma zapojenia invertujúceho prevodníku s obvodom MC34063 (prebrané z [15])

3.1.3 Lineárny menič LM317



Obrázok 34: Schéma zapojenia s LM317 (prebrané z [16])

LM317 je integrovaný stabilizátor napätia s maximálnym zaťažovacím prúdom 1,5 A. Má 3 vývody. Výstupné napätie je nastaviteľné pomerom odporov R1 a R2 a rozdiel vstupného a výstupného napätia nemôže byť väčší než 40 V. Výstupné napätie sa počíta podľa nasledujúceho vzorca:

$$V_0 = V_{ref} \left(1 + \frac{R2}{R1} \right) + I_{adj} R2 \quad V \quad (2)$$

kde V_{ref} je 1,25 V.

Keďže neboli všetky súčiastky dostupné podľa schém zapojení pre obvody MC34063 a LM317, musel som ich nahradiť súčiastkami, alebo kombináciami súčiastok, s blízkymi hodnotami a z toho dôvodu výstupné napätia nie sú úplne presné.

| Očakávané napätie [V] | Namerané napätie [V] |
|-----------------------|----------------------|
| - 8 | - 8,06 |
| 3,3 | 3,29 |
| 3,4 | 3,33 |
| 5 | 4,92 |
| 9,6 | 9,65 |
| 10,3 | 10,36 |
| 18 | 17,9 |

Tabuľka 8: Porovnanie očakávaných a nameraných hodnôt zdrojov

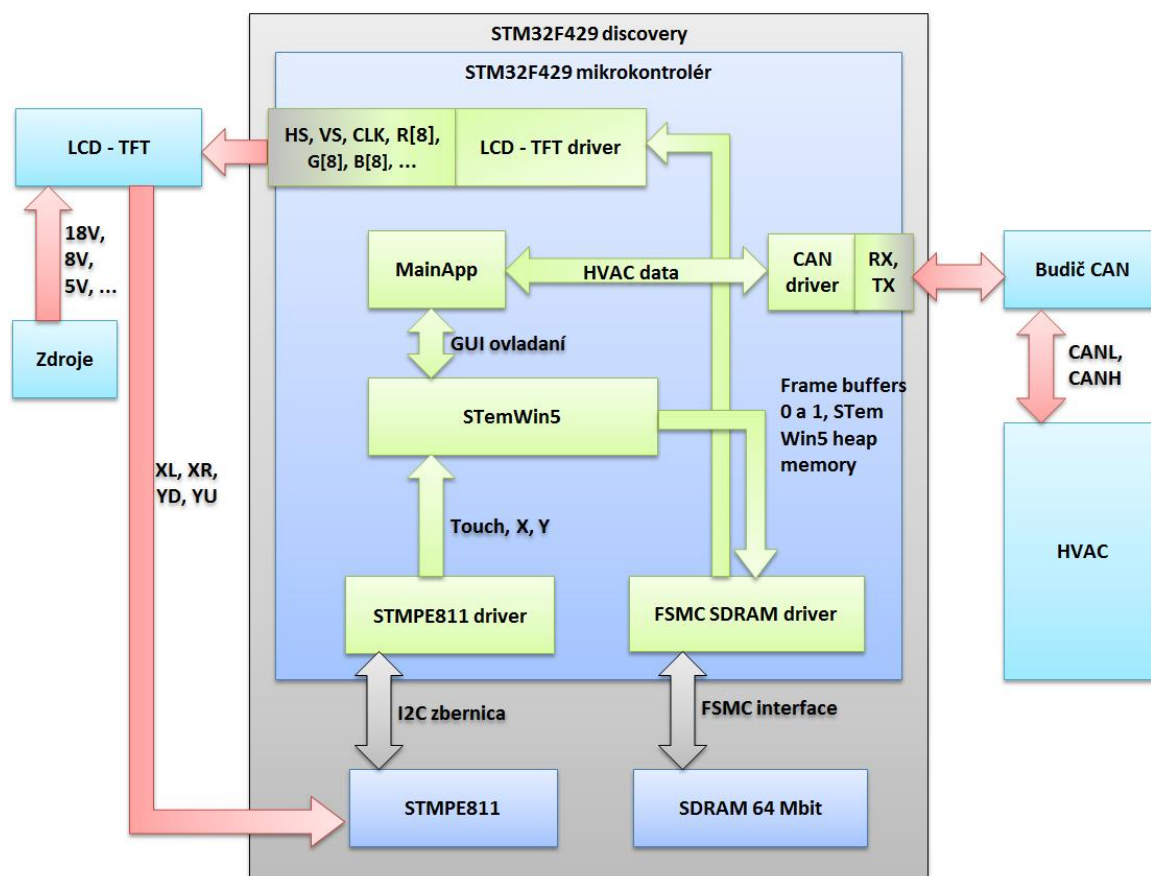
Z tabuľky je vidieť, že rozdiel v očakávaných a nameraných hodnotách napätia je minimálny a správnej funkcii displeja to neprekáža.

3.2 Návrh softvéru

V tejto časti vysvetlím ako fungujú jednotlivé bloky programu a ich funkciu. Bloková schéma programu je znázornená na obrázku 33: Bloková schéma programu. Program je písaný v jazyku C. Program som vyvíjal vo vývojovom prostredí Eclipse a kompilované GCC prekladačom.

Moduly, ktoré aplikácia obsahuje, majú jeden alebo dva súbory. Zdrojový súbor (s príponou .c) obsahuje zdrojový kód modulu. Hlavičkový súbor (s príponou .h) obsahuje definície makier, konštánt, používaných štruktúr, pripojenia ďalších potrebných modulov, deklaráciu vonkajších funkcií a iné.

Callback je ukazovateľ na funkciu s presne definovanými vstupnými a výstupnými parametrami. Pojem callbacková funkcia si môžeme predstaviť ako ukazovateľ na funkciu. Modul takúto funkciu volá v prípade, že zaregistruje určitú udalosť. V callbackovej funkcii je napísané ako a či na danú udalosť reagovať.



Obrázok 35: Bloková schéma programu

3.2.1 Moduly

Aplikácia sa skladá z jednotlivých modulov, ktorých úlohou je riadiť zobrazovanie obrazu na LCD – TFT displej (LCD – TFT driver) spracovanie dotykov na displeji (STMPE811 driver), riadenie zbernice CAN (CAN driver) alebo beh samotného programu (MainApp). Blok STemWin5 obsahuje knižnicu, ktorá slúži k interpretovaniu grafického rozhrania.

3.2.2 Modul CAN driver

Tento modul slúži na prijímanie a odosielanie rámcov, niekedy označovaných aj ako správy, medzi HVAC a riadiacim zariadením. Každá správa obsahuje 11 bitový špecifický identifikátor, ktorý určuje okrem priority rámca aj to, komu táto správa patrí. Ďalej obsahuje dáta a tiež bity, určujúce ďalšie vlastnosti správy. Viac informácií o jednotlivých poliach je v kapitole 2.5.6 Štruktúra dátového rámca.

Identifikátory ktoré sa majú prenášať po zbernici smerom k aplikácii majú adresy v hexadecimálnej hodnote 0x181, 0x191, 0x1A1, 0x1B1 a 0x1C1.

V budúcnosti je pravdepodobné, že sa k zbernici CAN pripoja ďalšie zariadenia, ktoré budú vysielat správy, ktoré nemusia byť určené pre HVAC alebo mnou vyvinutý riadiaci systém. Periféria bxCAN discovery kitu umožňuje konfigurovať filtre, ktoré filtrujú prijaté správy. Odvodenie masky filtru bolo jednoduché, a to z dôvodu, že adresy, ktoré majú byť prijaté sa nachádzajú v jednom pásme za sebou.

| Adresa | | | | | | | | | | | |
|---------------|---------|----|----|----|----|----|----|----|----|----|----|
| Hexadecimálne | Binárne | | | | | | | | | | |
| | 10. | 9. | 8. | 7. | 6. | 5. | 4. | 3. | 2. | 1. | 0. |
| 0x181 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0x191 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0x1A1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0x1B1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0x1C1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| Maska | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

Obrázok 36: Odvodenie masky filtra

Na obrázku 36 je vidieť, že bity 0-3 a 7-10 sa zhodujú pre všetky správy ktoré majú byť prijaté. Na týchto bitoch nám teda záleží, a preto v maske majú hodnoty týchto bitov logickú1. Bity ktoré sa menia, v našom prípade 4-6, sú pre nás nepodstatné a majú hodnotu logickej 0.

V tomto module je tiež vytvorenie tasku CAN, ktorý sa pravidelné volá aby vykonal príjem a odoslanie správ. Tiež je vytvorená fronta, do ktorej sa plnia prijaté správy.

Úlohou tohto modulu je komunikovať s modulom MainApp. Dáta tu bežia obojsmerne. Modul MainApp posielajú CAN driveru správy, ktoré chce vysielajú a ten ich posielajú budiču CAN po vodičoch RX a TX. V prípade, že driver CAN prijme správu po RX a TX od budiču CAN, správu spracuje a pošle do modulu MainApp.

uint16_t CAN_getErrCool(Void) – funkcia vracia blok bitov určujúcich poruchy chladenia. Funkcie tohto typu sú štyri.

void CAN_init(void) – inicializačná funkcia. V tejto funkcii sa nastavuje periféria CAN2, spúšťajú sa hodiny, nastavuje filter či rýchlosť prenosu. Táto funkcia tiež obsahuje vytvorenie tasku CAN.

void CAN_process(void *p) – v tejto funkcii prebieha pravidelné odosielanie správ pre HVAC a ukladanie prijatých správ do predom vytvorených štruktúr.

void CAN2_RX0_Handler(void) – handler periférie CAN.

void CAN_setVentSet(int vent) – funkcia nastavuje konkrétny bit v správe, ktorá sa odosiela. V tomto prípade nastavuje setpoint ventilácie.

3.2.3 Modul LCD – TFT driver

Modul obsahuje kompletne nastavenie displeja. Polaritu synchronizačných signálov (HSYNC, VSYNC), frekvencia displeja a jeho rozmery, ktoré som spočítal a odvodil pomocou datasheetu. Vzhľadom na to, že discovery kit disponuje od výroby malým 2,4 palcovým displejom, tak je tento súbor len modifikovaný tak, aby parametre vyhovovali novému, 7 palcovému displeju. Podrobný popis odvedenia a vypočítaných hodnôt je v kapitole 2.4.4 Synchronne časovanie displeja.

Úlohou tohto modulu je zobraziť na displeji grafické rozhranie ktoré som navrhol. Vytvorené GUI je uložené do pamäte SDRAM z ktorej modul LCD – TFT jednotlivé obrázky vyberie a reprezentuje ich pomocou pinov HS, VS, CLK, RGB a ďalších. Driver sa nezaobera

tým, či vykresľuje graf, tlačidlo alebo skrolovacie okno. Jeho úlohou je vykresliť obraz, ktorý som navrhol.

3.2.4 Modul MainApp

Modul MainApp zabezpečuje, aby sa správy smerujúce k HVAC posielali periodicky. Identifikátory ktoré sa majú prenášať po zbernici smerom k HVAC majú adresy v hexadecimálnej hodnote 0x101 a 0x111.

| Line | Arrival Time [ms] | ID [Hex] | Dir | Data [0, 1, 2, 3, 4, 5, 6, 7] | Description |
|-------|-------------------|----------|-----|---------------------------------|-------------|
| 1 | 18:16:43.866 | 101 | Rx | 1B 00 00 00 00 00 00 00 | |
| 2 | 18:16:43.866 | 111 | Rx | 00 00 00 00 00 00 00 00 | |
| 3 | 18:16:44.794 | 101 | Rx | 1C 00 00 00 00 00 00 00 | |
| 4 | 18:16:44.794 | 111 | Rx | 00 00 00 00 00 00 00 00 | |
| 5 | 18:16:45.722 | 101 | Rx | 1D 00 00 00 00 00 00 00 | |
| 6 | 18:16:45.722 | 111 | Rx | 00 00 00 00 00 00 00 00 | |
| 7 | 18:16:46.650 | 101 | Rx | 1E 00 00 00 00 00 00 00 | |
| 8 | 18:16:46.650 | 111 | Rx | 00 00 00 00 00 00 00 00 | |
| Count | Average Time[ms] | ID [Hex] | Dir | Data [0, 1, 2, 3, 4, 5, 6, 7] | Description |
| 114 | 928,0 | 101 | Rx | 8B 00 00 00 00 00 00 00 | |
| 113 | 928,0 | 111 | Rx | 00 00 00 00 00 00 00 00 | |

Obrázok 37: Periodické odosielanie správ do HVAC

Správy sa musia periodicky odosielať (obrazok 37) a to z dvoch dôvodov. Tým prvým je, že HVAC má vždy aktuálne hodnoty nastavenia parametrov a je informovaná o každej zmene v najbližšej správe. Druhý dôvod je prostý, slúži to ako informácia, že je spojenie v poriadku. V prípade, že by HVAC neprijímala správy, odpojila by sa od zbernice a začala byť autonómna. Rovnako tak v prípade, že by riadiaci systém neprijímal žiadne správy z HVAC, považoval by spojenie za prerušené. Táto vlastnosť systému je použiteľná napríklad v prípade, keď sa zobrazuje graf nameraných hodnôt teploty. V prípade prerušenia spojenia nie je žiadúce, aby graf vynášal poslednú nameranú hodnotu.

3.2.5 Modul STemWin5

Knižnica StemWin umožňuje vytvoriť grafické rozhranie obsahujúce rôzne prvky ako grafy, tlačítko alebo posuvný slajder. Toto GUI je uložené v SDRAM, z ktorej sa to ďalej interpretuje na displej pre užívateľa. Modul STemWin5 má v sebe uložené dáta o tom, kde je aký prvok vložený a čo sa má v prípade jeho stlačenia, posunutia alebo inej aktivity vykonať.

Knižnica dostane súradnice akcie (dotyku na displeji) z modulu STMPE811. Na základe súradníc vie knižnica určiť, čo sa na danom mieste nachádza a podľa toho rozhodne, či sa má napríklad tlačidlo stlačiť alebo zaškrtnúť zaškrťavacie políčko. Po zmene, respektíve reakcie modulu na dotyk, je generovaná správa, ktorá určuje aký úkon bol vykonaný (stlačenie tlačidla, pustenie tlačidla, zaškrtnutie políčka, zmena hodnoty, atp...). Ku každej takejto správe je naprogramovaná aktivita, ktorá sa má v takomto prípade vykonať.

3.2.6 Modul STMPE811 driver

STMPE811 je 4 drátový odporový kontrolér dotykového displeja, ktorý prostredníctvom I2C zbernice prevádza signály dotyku do driveru STMPE811. Hlavnou úlohou tohto driveru je spracovať signály tak, aby pri dotyku na displeji správne určil miesto dotyku. Do driveru sa pri dotyku posielajú hodnoty, ktoré musia byť v prípade x-ovej súradnice delené číslom 4,875 a v prípade y-ovej súradnice deliť číslom 7,2708.

Výsledné hodnoty sa ďalej podľa obr. č. 32 spracujú v knižnici STemWin5, ktorá na základe bodu dotyku vyhodnotí či a aké ďalšie úkony sa budú vykonávať.

3.3 Grafické rozhranie

V tejto kapitole rozoberiem mnou navrhnuté grafické rozhranie, jeho vzhľad, možnosti a funkcie. Aplikácia obsahuje úvodnú obrazovku, menu so šiestimi ikonami (Chlazení, Topení, Kalendář, Grafy, Poruchy a Informace). Ku každej ikone je pridelená obrazovka, ktorá sa zobrazí po kliknutí na ikonu.

3.3.1 Úvodná obrazovka aplikácie

Táto obrazovka sa zobrazí pri spustení systému. Obsahuje logo spoločnosti POLL s.r.o. a tiež 9 bodov na ktorých sa indikuje načítavanie systému. Počas procesu načítania systému je vždy jeden bod väčší než ostatné. Tento bod sa posúva doprava a až sa dostane na koniec, tak sa presunie na začiatok.



Obrázok 38: GUI - Úvodná obrazovka

3.3.2 Menu aplikácie

Po načítaní systému sa zobrazí menu aplikácie, ktoré obsahuje šesť ikon. V hornej časti okna je lišta ktorá sa nachádza v každom okne aplikácie. V ľavom hornom rohu je zobrazené malé logo spoločnosti POLL s.r.o a tiež využitie CPU v percentách. V pravom hornom rohu beží čas.



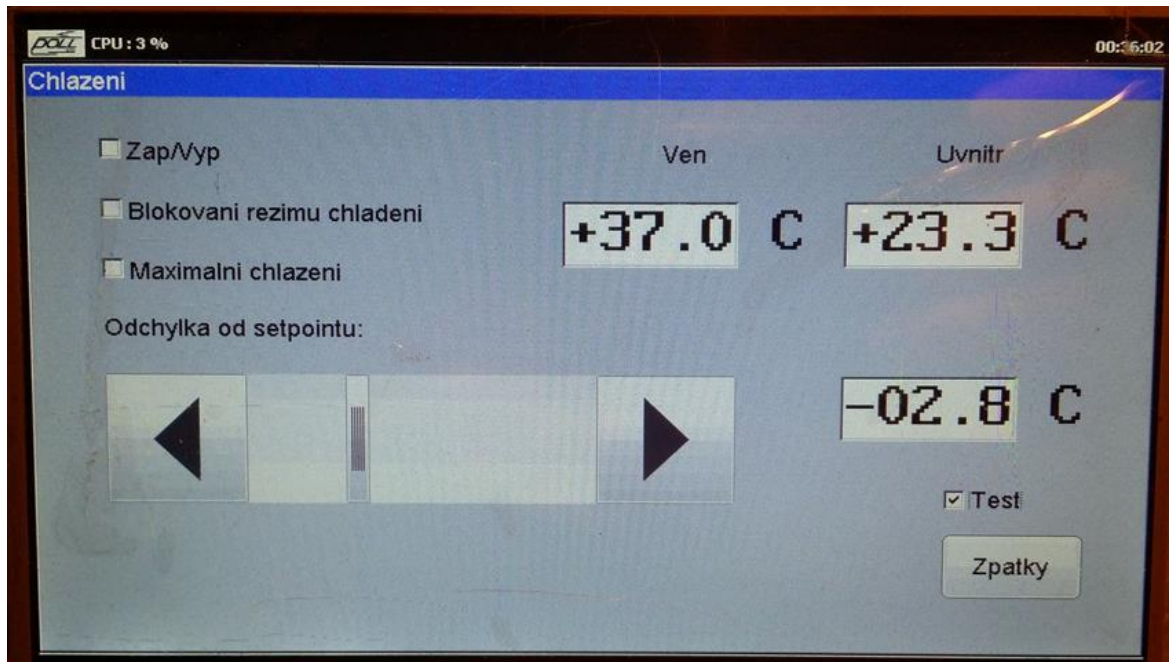
Obrázok 39: GUI – Menu aplikácie

3.3.3 Obrazovka chlazení

Po kliknutí na ikonu „chlazení“ sa zobrazí okno, v ktorom sa nastavujú parametre chladenia. Obrazovka obsahuje štyri zaškrťavacie políčka, ktorými je možné zapnúť alebo vypnúť HVAC, blokovať chladenie, zapnúť maximálne chladenie alebo spustiť test. Na obrazovke sú tiež zobrazené teploty vo vnútri a teploty von. Posuvnou lištou sa nastavuje hodnota odchyľky od setpointu, ktorá je zobrazená na pravo od posuvnej lišty. V posledom rade obsahuje tlačidlo „zpátky“, ktorým sa užívateľ dostane naspäť do menu aplikácie.

Vonkajšia teplota je zpriemerovaná teplota ktorú namerajú teplotné čidlá na streche vlaku. HVAC na základe tejto teploty vyhodnotí, aká teplota má byť vo vozni. Odchýľka od setpointu umožňuje túto hodnotu v určitom rozmedzí meniť a tým prinútiť HVAC aby chladila viac alebo menej než vyhodnotila.

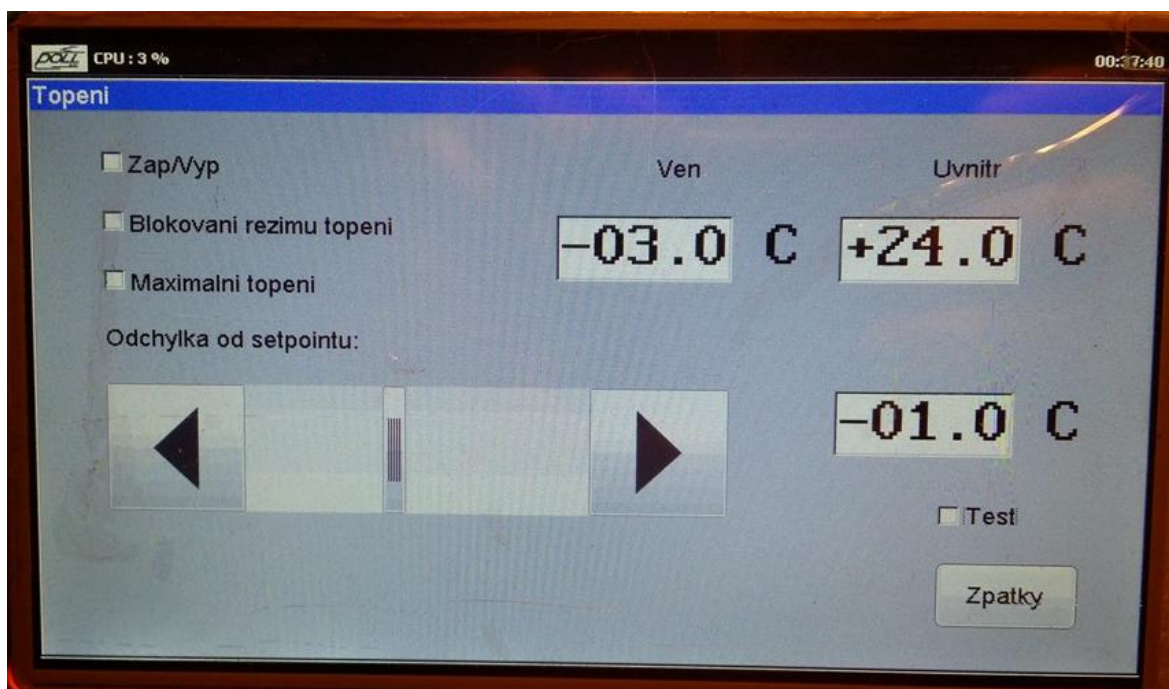
Test aplikácie je zavedený len za účelom prezentovania procesu bez pripojenia k HVAC. Test simuluje vonkajšiu a vnútornú teplotu. V závislosti na nastavení odchyľky od setpointu sa vnútorná teplota vagónu zníži alebo zvýši.



Obrázok 40: GUI – Obrazovka chlazení

3.3.4 Obrazovka topení

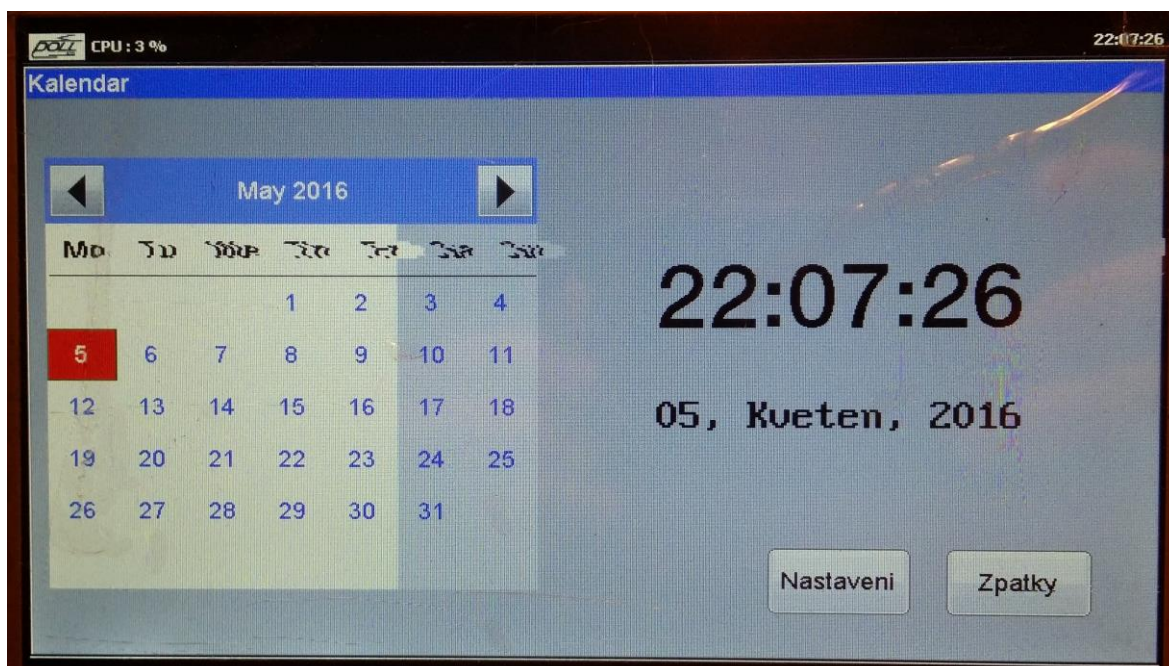
Obrazovka „topení“ sa zobrazí po kliknutí na druhú ikonu v poradí. Jej návrh a význam jednotlivých komponentov je analogický ako pri obrazovke „chlazení“.



Obrázok 41: GUI – Obrazovka topení

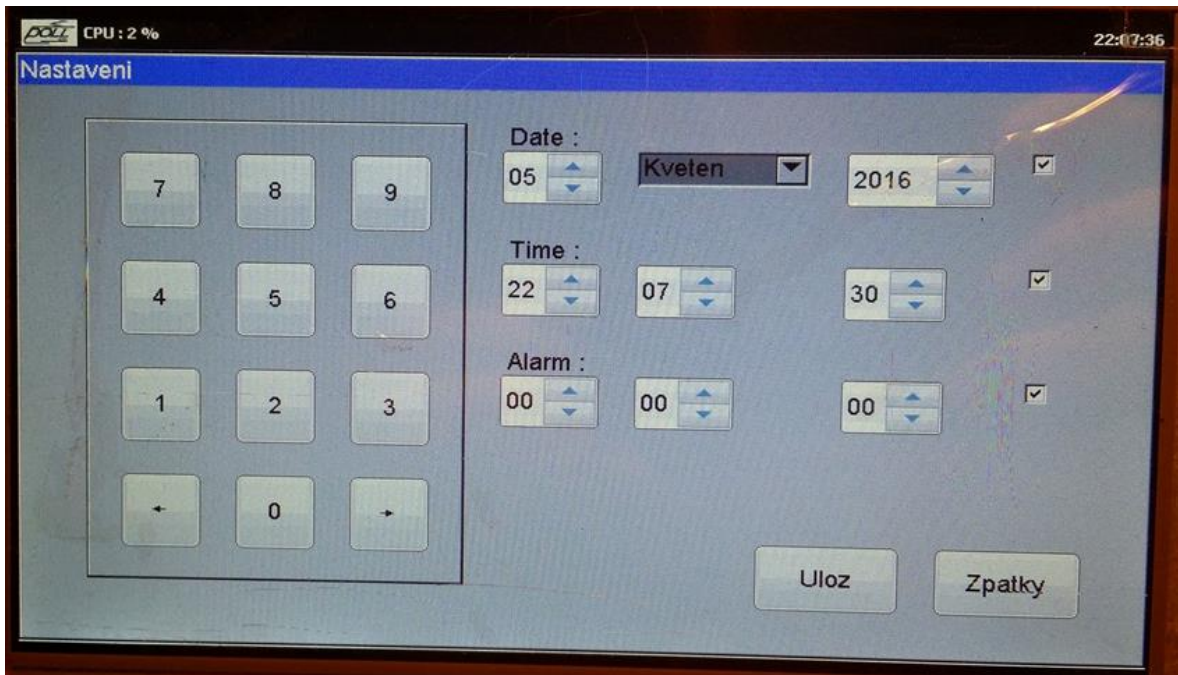
3.3.5 Obrazovka kalendára

Pod treťou ikonou sa nachádza obrazovka „kalendára“. Na obrazovke je zobrazený kalendár, čas a dátum. Obrazovka obsahuje dve tlačidlá. Tlačidlom „Zpatky“ sa užívateľ dostane späť do menu. Tlačidlo „Nastavení“ otvorí ďalšie okno(obrazok 43), ktoré obsahuje rozsiahle nastavenia.

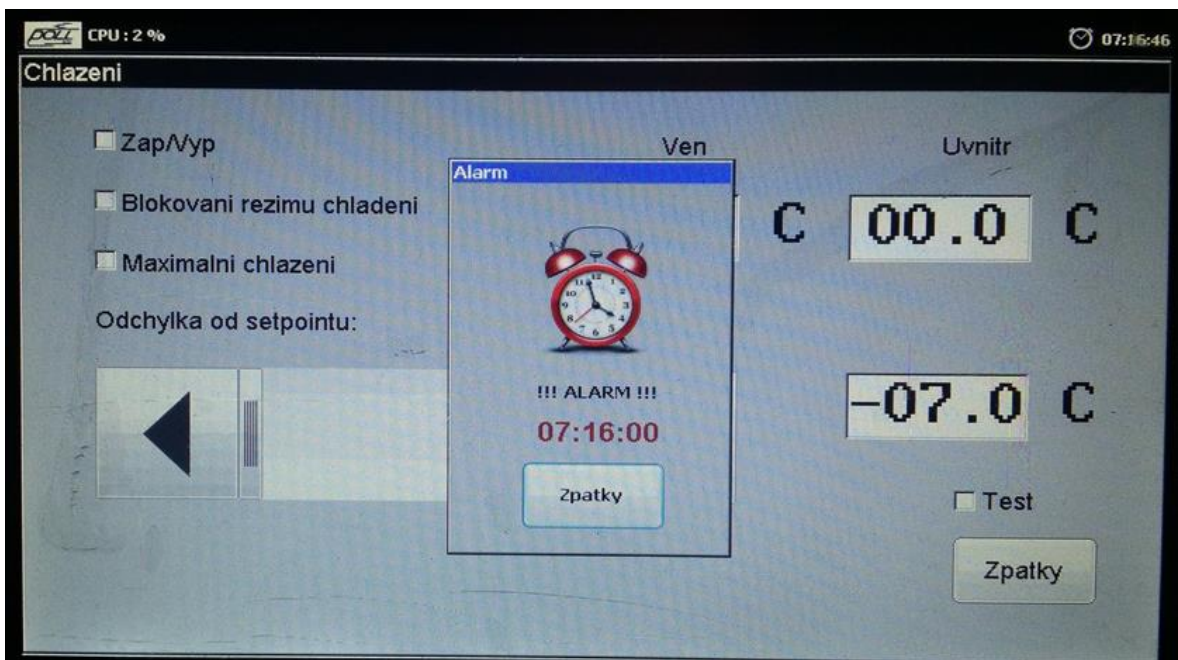


Obrázok 42: GUI – Obrazovka kalendára: dátum a čas

V obrazovke nastavenia sa nastavuje aktuálny dátum, čas a alarm. Po dosiahnutí času alarmu vyskočí na obrazovku dialógové okno Alarm (obrazok 44). Zaškrtnutými políčkami sa určuje, ktoré hodnoty chceme uložiť.



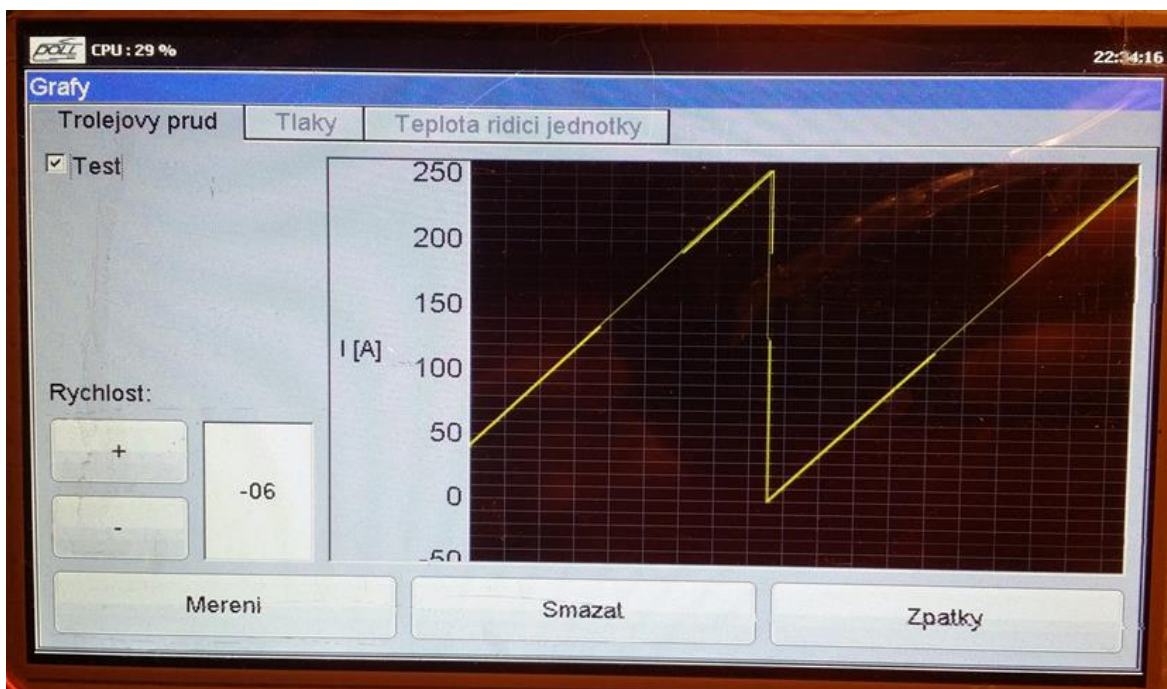
Obrázok 43: GUI – Obrazovka kalendára: nastavenia



Obrázok 44: GUI – Okno alarmu

3.3.6 Obrazovka grafov

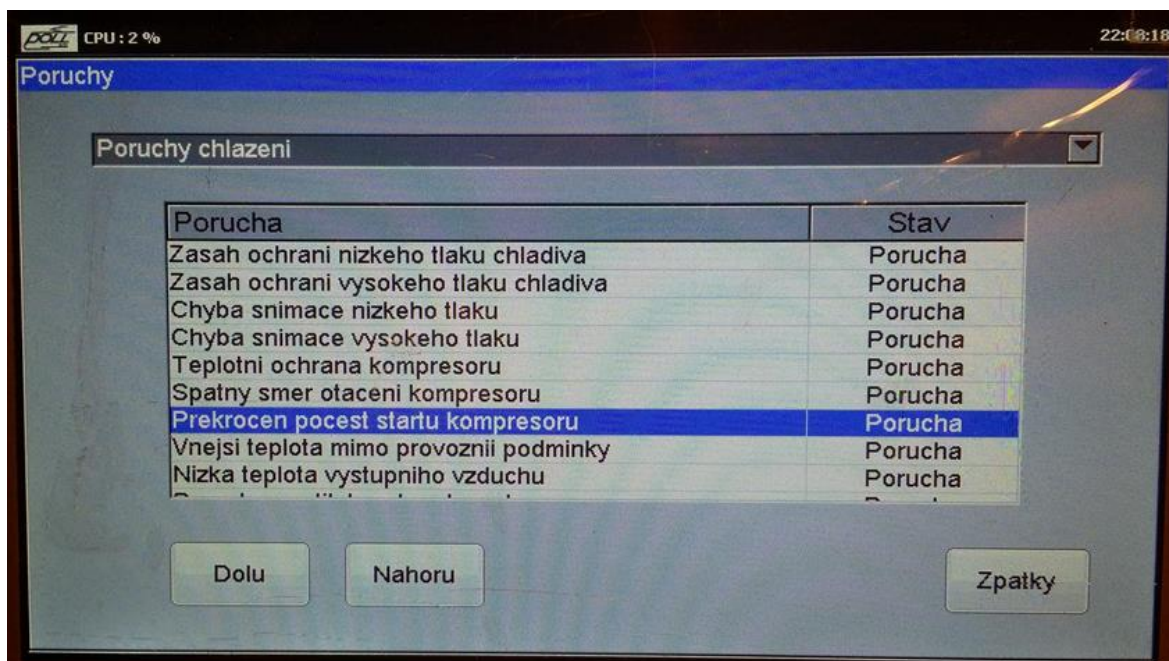
Na tejto obrazovke sa zobrazujú grafy hodnôt. Je možnosť výberu karty podľa toho, či chceme sledovať prúd, tlak alebo teplotu. Pre ilustráciu som vygeneroval pilovitý signál, ktorý sa pri zaškrtnutí políčka test interpretuje do grafu. Tlačidlami plus a mínus sa mení rýchlosť vykreslovania. Tlačidlom zmazať sa zmaže všetko, čo je vykreslené na grafe.



Obrázok 45: GUI – Obrazovka grafov

3.3.7 Obrazovka porúch

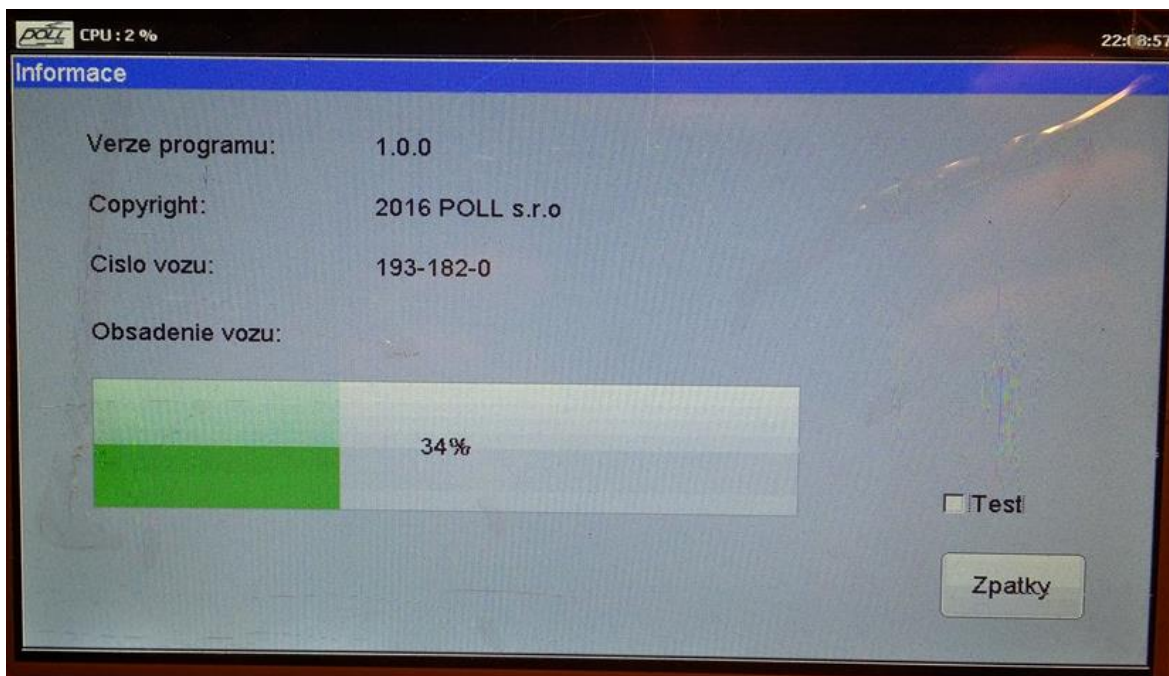
Obsahom tejto obrazovky je rozbaľovacie okno, v ktorom si užívateľ môže vybrať typy poruchy, ktoré chce zobrazovať. V tabuľke pod rozbaľovacím oknom sa zobrazí do riadku názov poruchy a vedľa jej stav. Tlačidlami Dolu a Nahoru sa užívateľ môže pohybovať po poruchách, ktoré sú vypísané.



Obrázok 46: GUI – Obrazovka porúch

3.3.8 Obrazovka informácií

Pod poslednou ikonou sa nachádzajú základné informácie o programe a tiež percentuálne vyjadrenie obsadenia vozu.



Obrázok 47: GUI – Obrazovka informácií

4 Záver

Na koniec práce pridávam zhrnutie postupu a splnenia cieľov mojej diplomovej práce.

V prvom kroku som sa pokúsil pripojiť nový displej k discovery kitu. To viedlo k analýze jednotlivých pinov kitu, aby som správne pripojil farebné, synchronizačné a napájacie vodiče displeja. Po predrátovaní som zistil, že discovery kit nie je schopný energeticky užiť displej čo viedlo k vývoju zdrojov. Vzhľadom na to, že sú na internete dosky, ktoré poskytujú presné, nami potrebné napätia, považoval som to za prvotnú alternatívu riešenia. Po porovnaní ceny takejto dosky s cenami súčiastok potrebných na realizáciu spínaných a lineárnych zdrojov som sa rozhodol zrealizovať zdroje sám. S realizáciou vlasných zdrojov som nepočítal a tak sa pripojenie displeja značne skomplikovalo. Po oživení a zmeraní všetkých zdrojov som displej pripojil a po úprave konfigurácie v operačnom systéme discovery kitu sa stal displej plne funkčný.

V druhom kroku som sa venoval realizácii zbernice CAN. Discovery kit disponuje dvoma CAN perifériami kde piny prvej periférie sa prekrývali s pinami farebných vodičov. Táto periféria sa stala týmto faktom nepoužiteľná a preto som musel použiť perifériu druhú. Tá mala čiastočne spoločné obvody s USB a bolo nutné odpájať kondenzátor, ktorý rušil CANovský prenos. Po upravení obvodu a naprogramovaní driveru sa dáta prenášali zbernicou, čo som otestoval pomocou USB – CAN adaptéru pripojeného do PC.

V tretej etape som sa venoval návrhu grafického rozhrania. Zoznámil som sa s knižnicou STemWin, programom pre vytváranie grafického rozhrania GUIBuilder. Po uvážení a rozložení jednotlivých ikon na ploche a všetkých obrazoviek som sa pokúsil GUI vytvoriť. Program GUIBuilder však umožňuje vytvoriť len kostru kódu obrazovky a tak bolo nakoniec jednoduchšie vytvárať obrazovky priamo vo vývojovom prostredí eclipse. GUIBuilder som použil len pre vygenerovanie základného okna a funkciou pomocou ktorej sa obrazovka v programe volá.

Posledná etapa súvisela s prepojením grafického rozhrania so zbernicou CAN a tiež naprogramovaním callbackových funkcií pre každú obrazovku.

5 Zoznam obrázkov

| | |
|---|----|
| Obrázok 1: Discovery kit STM32F429 (prebrané z [1])..... | 3 |
| Obrázok 2: Hardvérový blokový diagram (prebrané z [1])..... | 4 |
| Obrázok 3: Blokový diagram STM32F429ZIT6 (prebrané z [1])..... | 7 |
| Obrázok 4: Harvardská architektúra..... | 8 |
| Obrázok 5: Stavby úloh vo FreeRTOS (prebrané z [2]) | 10 |
| Obrázok 6: Grafické zobrazenie vykonávania úloh (prebrané z [2]) | 11 |
| Obrázok 7: Predávanie semaforu medzi úlohami (prebrané z [2]) | 12 |
| Obrázok 8: Vytvorenie prázdnej fronty (prebrané z [2])..... | 13 |
| Obrázok 9: Vloženie hodnoty do fronty (prebrané z [2]) | 13 |
| Obrázok 10: Vloženie druhej hodnoty do fronty (prebrané z [2]) | 13 |
| Obrázok 11: Prečítanie hodnoty z fronty (prebrané z [2]) | 13 |
| Obrázok 12: Uvoľnenie miesta vo fronte (prebrané z [2]) | 14 |
| Obrázok 13: Liquid Crystal Displej (prebrané z [4])..... | 16 |
| Obrázok 14: Štruktúra Thin-Film Transistor (TFT) (prebrané z [5])..... | 16 |
| Obrázok 15: Zapojenie matice LCD TFT displeja (prebrané z [6]) | 17 |
| Obrázok 16: Blokový diagram LTDC (prebrané z [8]) | 17 |
| Obrázok 17: Spájanie vrstiev s pozadím (prebrané z [8])..... | 18 |
| Obrázok 18:Znázornenie časovania displeja (prebrané z [8]) | 19 |
| Obrázok 19: Konštrukcia odporového dotykového displeja (prebrané z [8]) | 21 |
| Obrázok 20: Čítanie súradníc z 4 vodičového dotykového displeja (prebrané z [8])..... | 21 |
| Obrázok 21: Signály 4 vodičového dotykového displeja | 22 |
| Obrázok 22: Zapojenie uzlov do zbernice (prebrane z [9]) | 23 |
| Obrázok 23: Recesívna a dominantná úroveň signálu (prebrané z [10]) | 23 |
| Obrázok 24: 9 Pinový konektor CANNON (prebrané z [11]) | 24 |
| Obrázok 25: Príklad arbitrácie (prebrané z [12])..... | 28 |
| Obrázok 26: Konfigurácia filtrov – organizácia registrov (prebrané z [8]) | 30 |
| Obrázok 27: Príjímacie stavy FIFO fronty (prebrané z [8]) | 31 |
| Obrázok 28: Ukážka GUI vytvoreného knižnicou StemWin (prebrané z [13]) | 32 |
| Obrázok 29: Ukážka vykreslenia grafu knižnicou StemWin (prebrané z [14]) | 32 |
| Obrázok 30: Odvodenie napätí z adaptéra..... | 33 |

| | |
|---|----|
| Obrázok 31: Schéma zapojenia rastúceho prevodníku s obvodom MC34063 (prebrané z [15]) | 34 |
| Obrázok 32: Schéma zapojenia klesajúceho prevodníku s obvodom MC34063 (prebrané z [15]) | 35 |
| Obrázok 33: Schéma zapojenia invertujúceho prevodníku s obvodom MC34063 (prebrané z [15]) | 35 |
| Obrázok 34: Schéma zapojenia s LM317 (prebrané z [16])..... | 36 |
| Obrázok 35: Bloková schéma programu | 37 |
| Obrázok 36: Odvodenie masky filtra | 38 |
| Obrázok 37: Periodické odosielanie správ do HVAC | 40 |
| Obrázok 38: GUI - Úvodná obrazovka | 42 |
| Obrázok 39: GUI – Menu aplikácie | 42 |
| Obrázok 40: GUI – Obrazovka chlazení | 43 |
| Obrázok 41: GUI – Obrazovka topení | 44 |
| Obrázok 42: GUI – Obrazovka kalendára: dátum a čas | 44 |
| Obrázok 43: GUI – Obrazovka kalendára: nastavenia | 45 |
| Obrázok 44: GUI – Okno alarmu..... | 45 |
| Obrázok 45: GUI – Obrazovka grafov | 46 |
| Obrázok 46: GUI – Obrazovka porúch | 47 |
| Obrázok 47: GUI – Obrazovka informácií | 47 |

6 Zoznam tabuliek

| | |
|--|----|
| Tabuľka 1: Výhody a vlastnosti discovery kitu..... | 6 |
| Tabuľka 2: Piny displeja DEM 800480J1 TMH-PW-N (prebrané z [3]) | 15 |
| Tabuľka 3: Tabuľka synchronizačných pulzov displeja (prebrané z [3]) | 19 |
| Tabuľka 4: Vypočítané hodnoty pre správny beh displeja | 20 |
| Tabuľka 5: Pripojenie pinov konektoru CANNON | 24 |
| Tabuľka 6: Štruktúra rámca 2.0 A..... | 25 |
| Tabuľka 7: Polia správ a ich funkcie..... | 26 |
| Tabuľka 8: Porovnanie očakávaných a nameraných hodnôt zdrojov | 36 |

7 Zoznam použitej literatúry

- [1] Discovery kit with STM32F429ZI MCU, 2016. STMicroelectronics. URL: http://www2.st.com/content/st_com/en.html (accessed June 06, 2016).
- [2] AT04056: Getting Started with FreeRTOS on Atmel SAM, 2014. Atmel Corporation - Microcontrollers, 32-bit, and touch solutions. URL: <http://www.atmel.com/> (accessed June 06, 2016).
- [3] DEM 800480J1 TMH-PW-N (A-TOUCH), 2015. Transfer Multisort Elektronik - Electronic components. URL: <http://www.tme.eu/gb/> (accessed June 06, 2016).
- [4] subpixel.png (300×240), 2006. Bucaro TechHelp. URL: <http://bucarotechelp.com/> (accessed June 06, 2016).
- [5] tft-1.jpg, 2006. The World of Liquid Crystal Displays. URL: <http://www.personal.kent.edu/~mgu/LCD/tft.htm> (accessed June 06, 2016).
- [6] image020.gif. Liquid Crystal Displays: An overview. URL: http://www.battlesnake.co.uk/_uni/lcd.htm (accessed June 06, 2016).
- [7] DM00031020, 2015. STMicroelectronics. URL: http://www2.st.com/content/st_com/en.html (accessed June 06, 2016).
- [8] slaa384a.pdf, 2010. 4-Wire and 8-Wire Resistive Touch-Screen Controller Using the MSP430 (Rev. A). URL: <http://www.ti.com/lit/an/slaa384a/slaa384a.pdf> (accessed June 06, 2016).
- [9] can1.gif, 2015. ESC32 CAN Bus | AutoQuad. URL: <http://autoquad.org/wiki/wiki/aq-esc32/esc32-general-functions/can-bus/> (accessed June 06, 2016).
- [10] Canbus_levels.svg, 2013. Wikimedia. URL: https://commons.wikimedia.org/wiki/File:Canbus_levels.svg (accessed June 06, 2016).
- [11] 2000db9PIN.jpg, 2013. Technical Page. URL: <http://www.camradio.net/TS2Ktech.html> (accessed June 06, 2016).
- [12] can.png. andthinker: Automotive Interview Questions CAN INTERVIEW QUESTIONS. URL: <http://www.andthinker.com/2016/02/automotive-interview-questions-can1.html> (accessed June 06, 2016).

- [13] FDI_uEZGUI-2478-70WVE_screen.JPG. GUI 7in Touch Screen LCD Kit - Future Designs, Inc. (FDI) | Mouser. URL:
<http://eu.mouser.com/new/futuredesignsinc/fdiguitouchscreen/> (accessed June 06, 2016).
- [14] NXP Segger emWin Graphic Library - ARM TechCon 2011 - YouTube. URL:
<https://youtu.be/5mCoQKhtndQ?t=106> (accessed June 06, 2016).
- [15] MC34063A.pdf. SparkFun Electronics. URL:
<https://www.sparkfun.com/datasheets/IC/MC34063A.pdf> (accessed June 06, 2016).
- [16] lm317t_st.pdf, 2008. DATASHEET SEARCH SITE | WWW.ALLDATASHEET.COM. URL:
http://www.ges.cz/sheets/l/lm317t_st.pdf (accessed June 06, 2016).

