

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

Fakulta informatiky a informačných technológií

Evidenčné číslo: FIIT-5220-56340

Bc. Michal Valiček

NÁVRH A IMPLEMENTÁCIA SYSTÉMU NA ZBER ÚDAJOV

Diplomová práca

Študijný program: softvérové inžinierstvo

Študijný odbor: 9.2.5. softvérové inžinierstvo

Školiace pracovisko: Ústav počítačového inžinierstva a aplikovanej informatiky

Vedúci záverečnej práce: doc. Ing. Tibor Krajčovič, PhD.

Máj 2016

Čestné prehlásenie

Čestne prehlasujem, že diplomovú prácu som vypracoval samostatne,
pod odborným dohľadom vedúceho diplomovej práce.

Bratislava, máj 2016

Bc. Michal Valiček

PodĎakovanie

Ďakujem môjmu školiťovi doc. Ing. Tiborovi Krajčovičovi, PhD. za vždy zaujímavé, vecné konzultácie a nikdy nekončiaci pozitivizmus v akokoľvek kritickej situácii. Podobne aj Ing. Petrovi Pištekov, PhD. za umožnenie spolupráce v projekte vypúšťania stratosférického balóna, Alexandrovi Kuťkovi a celému tímu Slovenskej organizácie pre vesmírne aktivity za poskytnutie cenných skúseností z početných vypustení balónov do stratosféry a príprav prvej slovenskej družice. Neposledne aj Nadácii Tatrabanky za prostriedky na realizovanie finančne náročného projektu.

ANOTÁCIA

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Návrh a implementácia systému na zber údajov

Študijný program: Softvérové inžinierstvo

Autor: Bc. Michal Valiček

Vedúci diplomovej práce: doc. Ing. Tibor Krajčovič, PhD.

Máj, 2016

Systém na zber údajov je elektronické zariadenie slúžiace na zaznamenávanie výstupov zo senzorov experimentu. Takéto zariadenia sa masívne rozšírili s príchodom lacných mikropočítačov a pamätí. DAQ systémy dokážu autonómne spracovávať výstupy z experimentu 24 hodín denne, 7 dní v týždni, bez nutnosti kontroly operátorom.

V tejto práci navrhujeme univerzálny systém na zber údajov. Zariadenie spracováva analógové a digitálne výstupy zo senzorov. Na údaje aplikuje sériu konfigurovateľných pravidiel a pridá časové známky. Takto spracované údaje sú uložené na pamäťovú kartu a prípadne odosielané cez bluetooth alebo ethernet na spracovanie. Zariadenie má charakter systému reálneho času, preto firmvér modelujeme metódou RTSAD, ktorá primárne zobrazuje toky údajov medzi procesmi a funkciami.

Súčasťou práce je implementácia navrhnutého systému na hardvéri riadenom mikropočítačom na báze jadra Cortex-M3. Systém verifikujeme v laboratórnych podmienkach metódou, ktorú sme publikovali na konferencii IIT.SRC 2015 a aj praktickými testami. Praktické testy simulujú reálne nasadenie, jedným z testov je záznam zo senzorov stratosférického balónu.

ANNOTATION

Slovak University of Technology Bratislava

Faculty of Informatics and Information Technologies

Design and Implementation of Data Acquisition System

Degree Course: Software Engineering

Author: Michal Valiček

Supervisor: Assoc. Prof. Tibor Krajčovič

2016, May

Data acquisition system is a system used to acquire and record real world physical variables as a part of an experiment. Such systems have been rapidly expanded at time of cheap microcomputers and memories arrival on market. Researchers benefit of the ability to perform experiments 24 hours a day, 7 days a week, without a need of human control.

In this diploma thesis we are designing universal data acquisition system. The device collects digital and analogue data out of connected sensors. The series of configurable rules are applied on the collected data (trigger subsystem) and the result is tagged with time stamps. Such data are stored on Secure Digital memory card and optionally sent through Bluetooth serial line or ethernet to external system to be used online. Proposed system has specific timing constrains, thus firmware is designed with the RTSAD method, which is visualizing data flows between processes and functions.

The output of the thesis includes implemented prototype on hardware driven by Cortex-M3 microcontroller. We are verifying the prototype in laboratory environment with the method we have presented at IIT.SRC 2015 conference. The prototype has been used in series of real-life experiments, including data acquisition at stratospheric balloon.

Obsah

1 Úvod	11
1.1 Štruktúra dokumentu	11
2 Analýza	13
2.1 Existujúce riešenia	13
2.1.1 Oregon Scientific WMRS2000	13
2.1.2 Dataq DI-161	14
2.1.3 National Instruments USB-6003	15
2.1.4 DAQ v experimente ATLAS	17
2.2 RT analýza a modelovanie	19
2.2.1 CORE	19
2.2.2 JSD – Jackson System Design	20
2.2.3 RTSAD, DARTS, YSM	20
2.2.4 Objektové modely	20
2.2.5 Metóda porovnania rôznych prístupov	21
2.3 Hardvér	21
2.3.1 AD prevodník	22
2.3.2 Riadiaci obvod	26
2.3.2.1 Mikropočítače s jadrom AVR	26
2.3.2.2 Mikropočítače s jadrom ARMCortex-M3	26
2.3.2.3 Obvody FPGA	27
2.3.2.4 Záver	28
2.3.3 Komunikačné rozhrania	28
2.3.3.1 Káblové pripojenie	29
2.3.3.2 Bezdrôtové rozhrania	30
2.3.3.3 Obvod ENC28J60	31
2.3.3.4 Obvod W5100	32
2.3.3.5 Obvod BC417	32
2.3.4 Správa napájania	32
2.3.5 Zapojenie hardvéru	34
2.4 Operačný systém	34

2.5	Zabezpečenie komunikácie	35
3	Návrh riešenia	37
3.1	RTSAD model.....	37
3.1.1	Funkčná dekompozícia.....	38
3.1.2	Funkčná dekompozícia druhej úrovne	39
3.1.3	Analýza stavových prechodov	41
4	Špecifikácia prototypu	43
4.1	Funkcionálne požiadavky prototypu	43
4.2	Nefunkcionálne požiadavky.....	44
5	Implementácia prototypu	45
5.1	Vstupno-výstupne piny.....	45
5.2	AD prevodník.....	46
5.3	SPI rozhranie a pamäťová karta	47
5.4	SPI rozhranie a sieťová konektivita	49
5.5	Spúšťačí podsystem, trigger.....	51
5.6	UART rozhranie a Bluetooth.....	53
6	Overenie riešenia.....	54
6.1	Overenie v laboratórnom prostredí	54
6.1.1	Experimentálne získané hodnoty	55
6.1.2	Teorém o hranici využitia CPU	56
6.2	Overenie v cieľovej doméne - kvalita spánku.....	58
6.3	Overenie v cieľovej doméne - radiačné pozadie	60
6.4	Overenie v cieľovej doméne - stratosférický balón	62
6.5	Overenie v cieľovej doméne - let lietadlom	64
	Záver.....	66
	Slovník termínov a skratiek	68
	Zoznam použitej literatúry	69
	Prílohy.....	71

1 Úvod

Systémy na zber údajov - DAQ systémy patria k základnej výbave vedcov a technikov. Slúžia na okamžité a aj dlhodobé merania, overovanie, testovanie v laboratórnom, prípadne v externom prostredí. Definovať takýto systém je najvhodnejšie na základe toho, čo vykonáva. DAQ systémy zachytávajú fyzikálne veličiny, ktoré sú reprezentované senzormi v podobe elektrického napätia alebo prúdu v takom formáte, aby bolo možné neskôr tieto hodnoty prechádzať a analyzovať. Tieto merania sú vykonávané automaticky, po konfigurácii, bez zásahu človeka, čo kladie vysoké nároky na spoľahlivosť a odolnosť systému. Situáciu komplikuje charakter experimentov, ktoré často prebiehajú v externom prostredí, mimo laboratória a zariadenia sú vystavené širokému spektru teplôt, vlhkosti, nestabilnému alebo dokonca nedostupnému zdroju napájania [20].

DAQ systémy sa skladajú zo siedmich elementov:

- zber údajov,
- meranie (prevod analógovej hodnoty do digitálnej informácie),
- časovanie a spúšťanie záznamu vzhľadom na stav prostredia (trigger),
- hodiny reálneho času,
- riadenie systému,
- komunikačné rozhrania (záznamové média alebo externé systémy dostupné prostredníctvom siete),
- záznam a archivácia údajov [19].

Ak je implementovaných všetkých 7 funkcií, jedná sa o plnohodnotný DAQ systém, ak nejaké chýbajú, hovoríme o module DAQ systému, v prípade ak sú implementované ďalšie funkcie, takýto systém je komplexným systémom, ktorý zahŕňa DAQ.

1.1 Štruktúra dokumentu

Dokument sa skladá zo šiestich hlavných sekcií. Prvá kapitola sa venuje úvodu do problematiky, definuje systém na zber údajov. Druhá kapitola sa venuje analýze existujúcich riešení, analýze požadovanej funkcionality kladenej na takýto systém a analýze jednotlivých komponentov systému a metodológie použitej pri vývoji systému. Tretia kapitola predstavuje vytvorený model metódou RTSAD. Štvrtá kapitola

sa zaoberá návrhom riešenia hardvéru a softvéru, ktoré vychádza z analýzy a špecifikácie. V piatej kapitole je uvedený súhrn najzaujímavejších častí implementácie a v šiestej kapitole uvádzame verifikovanie systému v laboratórnom prostredí a pri praktických testoch.

2 Analýza

V tejto kapitole sa zameriavame na rôzne hardvérové a softvérové prvky DAQ systému. Prinášame viaceré alternatívy a vyberáme najvhodnejšie riešenie na implementáciu. Zaoberáme sa aj modelovaním systému metodikou, ktorá je najvhodnejšia pre špecifickú oblasť vnorených systémov.

2.1 Existujúce riešenia

Na trhu existuje veľké množstvo zariadení s rôznymi parametrami. V tejto oblasti je možné uvažovať o celom spektre od amatérskych meteostaníc, ktoré zaznamenávajú údaje z pripojených senzorov v intervaloch rádovo minút. Meteostanice získané hodnoty odosiľajú na pripojený počítač, ktorý je vybavený softvérom na kumulovanie a analyzovanie dát alebo ich odosiela do databázy na internete, z ktorej čerpajú webové aplikácie. Na opačnej strane spektra sú komplexné laboratórne zariadenia, ktoré majú zložitý podmieňovací podsystém a sú určené a vyvinuté na konkrétny experiment – senzor, detektor. Príkladom takéhoto systému je DAQ experimentu ATLAS, ktorého len podmieňovací podsystém obsahuje 250 tisíc procesorov.

2.1.1 Oregon Scientific WMRS2000

WMRS2000 (obr. 1.) je bežne dostupný a medzi amatérskymi meteorológmi obľúbený systém na zber údajov dodávaný aj so senzormi (dokumentácia je dostupná na webe výrobcu <http://weather.oregonscientific.com/>). Systém bezdrôtovo pripája senzory rýchlosti a smeru vetra, teploty a vlhkosti vzduchu, zrážok. Bezdrôtové komunikačné linky sú čoraz obľúbenejšie v oblasti inteligentných domácností, pretože inštalácia je neinvazívna. Na druhej strane prináša obmedzenia v oblasti napájania, čo je v tomto prípade riešené veľmi prepracovaným riadením napájania. Meteostanice sú spravidla typu data logger, teda k akvizícii dát dochádza v dlhých periódach, rádovo jednotky až desiatky minút. Podobnosť s laboratórnymi vedeckými prístrojmi si ponecháva aj v nutnosti pripojenia na počítač, ktorý cez USB rozhranie číta výstup meteostanice a slúži na jeho uchovávanie, resp. odosielanie do meteoslužieb na internete.

Oregon Scientific WMRS2000 patrí k zaujímavým príkladom dostupným pre amatérov. Zariadenie sme do analýzy uviedli, pretože jeho charakter spôsobuje

obmedzenia v oblasti napájania a spracováva aj analógové (teplota, vlhkosť) a digitálne signály (početnosti impulzov v anemometri a zrážkovom senzore).



Obr. 1.: Dedikovaný systém na zber údajov.

2.1.2 Dataq DI-161

DI-161 (obr. 2.) spracováva signály z akýchkoľvek senzorov s digitálnym výstupom, ide teda o všeobecný systém na zber údajov (dokumentácia je dostupná na webe výrobcu <http://www.dataq.com/products/>). Podobne ako v predchádzajúcom prípade, zaznamenáva údaje v nízkych frekvenciách, konkrétne udalosti, čas kedy k nim došlo, ich trvanie a početnosť. Má implementovaných 8 samostatných digitálnych kanálov. Pripojenie je prostredníctvom ethernetu a USB zbernice k počítaču, ktorý zabezpečuje uchovávanie dát, ich agregáciu a vyhodnocovanie.



Obr. 2.: Všeobecný systém pre zber digitálnych údajov.

Napriek všeobecnému využitiu je zariadenie uspokojené špeciálne na kvantifikáciu OEE (aj. overall equipment effectiveness), teda celkovú efektívnosť zariadenia. Charakter získavaných údajov umožňuje výpočet efektívnosti výrobných linky, kedy sa zohľadňuje celkový čas produkcie (dĺžka trvania impulzu), okamžitý počet vyrobených zariadení, celkový počet vyrobených zariadení a počet produktov, ktoré neprešli výstupnou kontrolou (v posledných troch parametroch zariadenie zaznamenáva početnosť impulzov). Viacero DI-161 zariadení dokáže spolupracovať a ukladať informácie do jednej databázy, čím sa dá získať okamžitý a dlhodobý prehľad o efektívnosti celej továrne. Každý z kanálov je konfigurovateľný a dokáže zaznamenávať udalosti, ktoré sú označované časom a dátumom. Tieto udalosti môžu byť reprezentované senzormi ako nábežné, dobežné hrany (napríklad dvere boli otvorené o 10:40 21.9.2015). Podobne je možné zaznamenať čas zotrvania vstupu na špecifikovanej logickej úrovni (napríklad kompresor bol zapnutý v čase od 13:51 21.9.2015 do 14:02 21.9.2015) a početnosť v danej časovej perióde (napríklad počet dokončených kusov bol 14 v čase od 8:00 21.9.2015 do 9:00 21.9.2015).

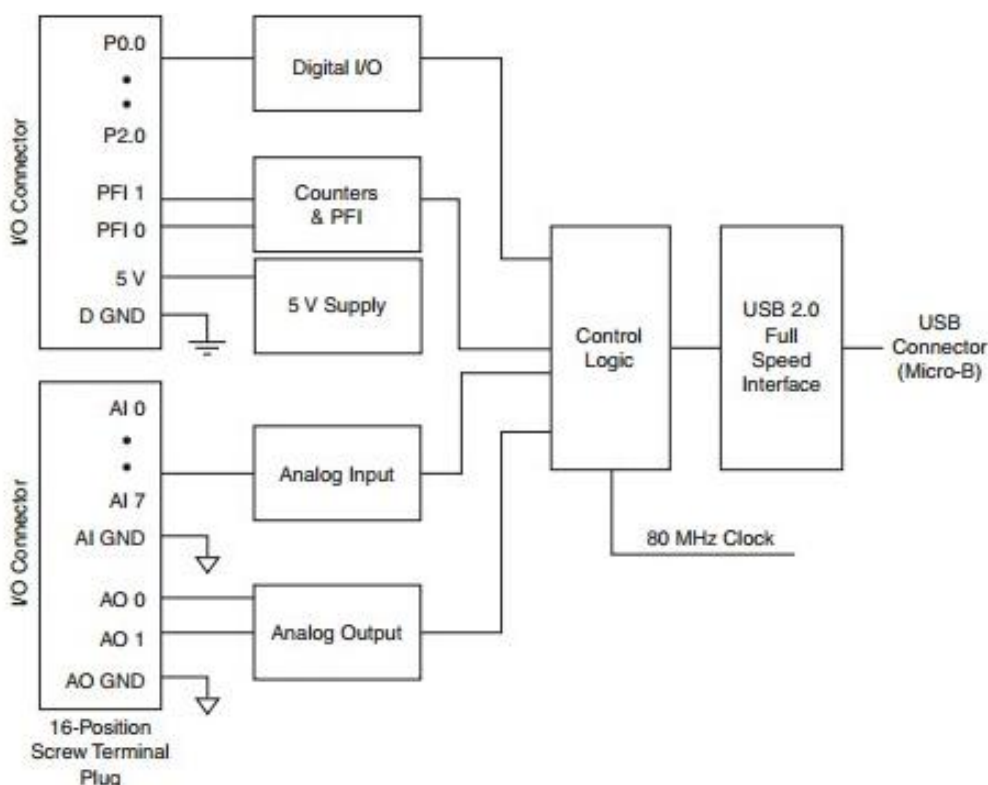
DATAQ používa hardvérové moduly pre každý kanál. Na obrázku 2 sú viditeľné ako žlté 5-pinové kvádríky nad vstupnými svorkami. Vymeniteľnosť modulov umožňuje zariadeniu získavať údaje zo širokého spektra v priemysle používaných napäťových úrovní logických signálov. Napríklad modul IN5S umožňuje na pripojenom kanáli čítať logiku 24 VDC.

Zariadenie sme do analýzy uviedli, pretože plní veľmi elegantne funkciu spracovávania digitálnych signálov a je uspokojené na rýchle a aj pomalé udalosti. Zaznamenáva udalosti v periódach od 5 milisekúnd až po 198 dní. Zaujímavé je tiež prispôbovanie vstupných kanálov pre rozdielne logiky vymeniteľnými hardvérovými modulmi. Dôležitá je dostupnosť rozhrania ethernet popri štandardnému USB. Cena zariadenia na trhu je približne 630 dolárov a je daná aj špičkovým priemyselným vyhotovením.

2.1.3 National Instruments USB-6003

Zariadenie je určené na základné laboratórne merania, číta analógové údaje na 8 kanáloch s rozlíšením AD prevodníka 16 bitov. Maximálna agregovaná rýchlosť na všetkých kanáloch je 100 tisíc vzoriek za sekundu. Dáta sa odosielať prostredníctvom zbernice USB do počítača s možnosťou dočasného uchovávania vo FIFO zásobníku s kapacitou 2047 vzoriek. Na 13 digitálnych vstupoch je umožnené jednoduché počítanie

hrán impulzov. Zariadenie nemá riešenú komplexnú správu napájania a je nevhodné na napájanie batériami pri meraní v externom prostredí mimo laboratória, čo by ostatne komplikovala aj nutnosť pripojenia na plnohodnotný počítač. Vstupný rozsah napätia na vstupoch je 10 VAC a nie je možné ho akokoľvek zmeniť. Okrem tejto základnej funkcionality spracovávaní analógových signálov a počítania hrán digitálnych signálov, je zariadenie vybavené aj DA prevodníkom, ktorý dokáže na výstupe nastaviť analógovú hodnotu. Bloková schéma pripojenia vstupno-výstupných svoriek a základne prvky, ktoré spracovávajú signály sú na obrázku 3.



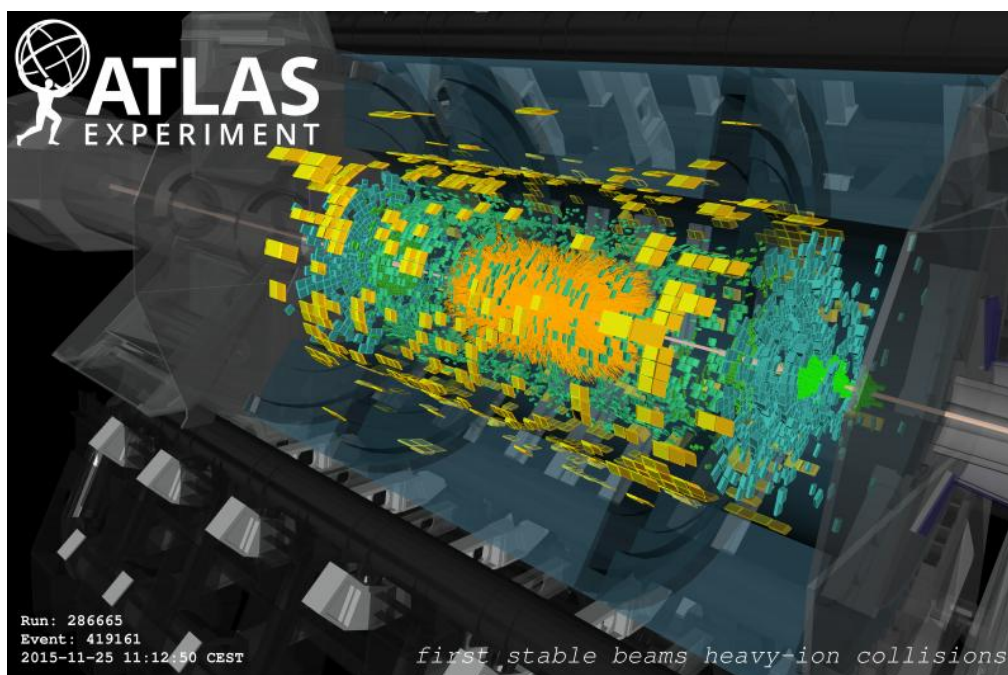
Obr. 3.: Bloková schéma zariadenia USB-6003.

Zariadenie má dve svorkovnice, jedna agreguje digitálne a druhá analógové vstupy signálov. Údaje sú riadiacou logikou spracované, doplnené o možnosť podmieňovania spracovávaní jednoduchým triggerovacím vstupom a následné údaje prechádzajú USB rozhraním do počítača, kde sú vizualizované.

Zariadenie sme do analýzy vybrali, pretože poskytuje základnú funkcionality spracovávaní analógových a digitálnych signálov v prostredí jednoduchých laboratórnych meraní. Zariadenie nie je vhodné na automatizované merania v priemysle alebo exteriéri. Cena zariadenia je približne 550 eur, ale reálna cena je spravidla vyššia, nakoľko je nutné zakúpiť softvér na osobný počítač, ktorého hodnota môže byť násobne vyššia.

2.1.4 DAQ v experimente ATLAS

Experiment ATLAS je jeden zo štyroch hlavných detektorov výskumnej organizácie CERN. Ide o jeden z najkomplexnejších a najväčších strojov aké človek doteraz vytvoril. Jeho úlohou je zaznamenávať špecifikované typy udalostí, zrážok častíc, ktoré prebiehajú na mieste kríženia trubíc kolajderu LHC. LHC pozostáva z dvoch kruhových urýchľovačov, ktoré protibežne vedú častice v rýchlostiach blížiacich sa rýchlosti svetla. Od poslednej významnej rekonštrukcie ktorá bola ukončená v prvej polovici roku 2015 sa násobne zväčšila početnosť udalostí, ktoré musí DAQ na experimentoch vyhodnotiť a v prípade splnenia podmienok zaznamenať.



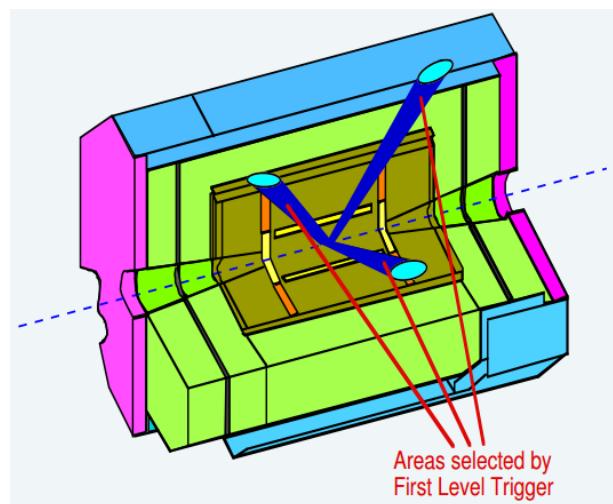
Obr. 4.: Vizualizácia udalosti v detektore ATLAS.

Na obrázku 4 je vizualizácia prvkov detektora, ktoré generovali signál počas zrážky číslo 419161, dňa 25.11.2015, kedy došlo k prvým stabilným zväzkom ťažkých jónov olova od rozsiahlej údržby a rekonštrukcie. Úlohou DAQ v ATLASe je zaznamenať produkty zaujímavých zrážok. Pre vývojárov tohto systému išlo o mimoriadne náročnú úlohu, nakoľko:

- Perióda medzi udalosťami je iba 25 ns. Je to dokonca kratšia doba než čas za ktorý produkty preletia detektorom pri maximálnej rýchlosti a tak viaceré udalosti prebiehajú paralelne.
- Každá zrážka generuje približne 25 interakcií v detektore.

- Úložný priestor je vo svojej podstate konečný a je potrebné selektívne vybrať len časť udalostí, ktoré sme schopní zaznamenať na trvalé úložisko. Pri frekvencii interakcií 10^9 je potrebné vybrať a zaznamenať len rádovo stovky.
- Zaujímavé udalosti sa dejú veľmi zriedkavo a problém odhadnúť ich je netriviálny. Príkladom je produkcia higgsového bozónu, ku ktorému podľa teoretického modelu dochádza približne v jednej z 10^{13} zrážok. Ide o mimoriadne zriedkavý jav a úlohou DAQ je ho zaznamenať s vysokou pravdepodobnosťou, napriek významnému šumu ostatných udalostí.

Zber údajov je podmienený trojstupňovým rozhodovacím podsystemom. Podmieňovanie prvej úrovne (Level-1 trigger) v prvom rade identifikuje moment, kedy došlo k zrážke častíc v detektore. Následne po zachytení údajov hľadá kandidátov zaujímavých produktov v obmedzenej časti detektora s odozvou len 2 μ s. Nízka odozva rozhodovania je kľúčová, nakoľko všetky údaje musia ostať dostupné v zásobníkoch počas celej doby vykonávania, až do rozhodnutia, či budú zahodené alebo spracované ďalšími úrovňami. Kľúčom k nízkej latencii je rozhodovanie, nie v kontexte celého detektora, ale len vo vybraných zónach záujmu. Príklad takýchto zón u náhodnej zrážky je na obrázku 5. Počet udalostí, ktoré prechádzajú do Level-2 triggeru sa zníži o 4 rády na 10^5 [1].



Obr. 5.: Príklad výberu zón záujmu Level-1 triggerom.

Druhá úroveň rozhodovania kandidátov na zaujímavé udalosti už prebieha v kontexte celého detektora. Rozhodovanie trvá 10 ms a ako v minulom prípade, aj teraz musia údaje ostať dostupné počas celej doby vyhodnocovania. Značne dlhšia odozva zložitejšej logiky je možná, nakoľko množstvo informácií je znížené desaťtisíc

násobne prvou úrovňou oproti surovým dátam. Výstup druhej úrovne má periodicitu už len 1 ms.

Level-3 trigger sa nepozera na udalosť v kontexte hľadania kandidátov častíc, ktoré môžu byť predzvest'ou zaujímavej udalosti, ale v kontexte celej zrážky, t.j. všetkých produktov. Na spracovanie jednej udalosti je potrebný čas približne 1 s, pri procesore s výkonom 1000 MIPS. Používané procesory majú výkon 100 až 200 MIPS a preto neprekvapí fakt, že konečný počet procesorov v triggerovacom podsysteme je 250 000, ako sme uviedli vyššie [1].

Detektor ATLAS sme uviedli v analýze, aby sme ukázali ako komplexné môžu byť DAQ systémy, čo je viditeľné na viacúrovňovom podmieňovaní záznamu údajov.

2.2 RT analýza a modelovanie

Vzhľadom na to, že softvér je vo svojej podstate nehmotný a ťažko uchopiteľný, je dobrým zvykom pred samotným alebo počas vývoja inkrementálne tvoriť model systému. Vďaka modelu je možné celú problematiku vývoja daného systému lepšie pochopiť a vhodnejšie spracovať. Z toho plyní vo výsledku lepšia udržiavateľnosť kódu, menšia chybovosť a pod. [6] V súčasnosti sú dobre popísané viaceré spôsoby vývoja softvéru, vnorených systémov a modelovania. Odlišujú sa v rôznych prístupoch a ich výstupy sledujú spravidla rôzne ciele. Je teda dôležité vybrať si metódu modelovania podľa druhu navrhovaného systému. Oblasť systémov pracujúcich v reálnom čase má špecifické požiadavky vzhľadom na časové špecifikácie, ktoré musia byť dodržané. Ich nedodržanie je vnímané ako zlyhanie systému. Metódy, ktoré sa využívajú na dokázanie, či ten ktorý systém je SRČ, spravidla analyzujú funkcie, medzi ktorými prebiehajú toky údajov a preto je vhodné využiť metódy modelovania, ktoré dekomponujú navrhovaný systém do podsystémov a následne funkcií [12].

Podľa Anthonyho Stainesa sa medzi hlavné metódy modelovania systémov s konkurenčnými úlohami a systémov reálneho času zaraďujú CORE, YSM, MASCOT, JSD, DARTS, RTSAD, ADARTS, CODARTS, HOOD, HRT-HOOD, ROOM, UML a UML-RT. Prvých 6 menovaných používa prístup riadený údajmi k riešeniu problému. Na druhej strane posledných 7 využíva objektovú notáciu [12].

2.2.1 CORE

CORE metóda bola navrhnutá v Anglicku, s cieľom poskytnúť postupy na analýzu požiadaviek pri vývoji vnorených systémov v letectve. Pozostáva z 5 krokov

a má neformálnu notáciu pozostávajúcu z diagramov blokov a požiadaviek. Zameriava sa hlavne na definovanie požiadaviek a nie na samotný dizajn, model. Nevýhody teda plynú z ukončenia procesu pred zodpovedaním, dôležitých otázok kladených pri vývoji SRČ. Nie je doriešené časovanie úloh, ich paralelnosť a synchronizácia. Výstupom nie je architektonický dizajn.

2.2.2 JSD – Jackson System Design

Je veľmi špecifická metodológia, ktorá model priebežne transformuje do systémov implementácie. Hlavnou myšlienkou je fakt, že dizajn by mal v prvom rade modelovať realitu. Každá entita v modeli je tak teda paralelná úloha v realite.

2.2.3 RTSAD, DARTS, YSM

Príbuzné metódy, ktoré modelujú systém z pohľadu tokov udalostí a riadenia, ako napríklad tok diskretných, spojitých údajov, spúšťač signál, povoľovacia (aj. enable) a zakazujúca (aj. disable) riadiaca udalosť. Práve prístup z pohľadu toku údajov a riadenia je veľmi zaujímavý pri modelovaní systému na zber údajov, ktorého ústrednou entitou sú toky údajov získavaných z prostredia. Nevýhodou je nemožnosť použiť metódu na prototypovanie, kroky vývoja softvéru musia byť vykonané sekvenčne, vodopádovo.

Metódy CODARTS a ADARTS vychádzajú z rovnakých princípov ako táto trojica (konkrétne z DARTS, pričom CODARTS prináša nové triedy správ a ADARTS metódu špecifikuje s použitím jazyku ADA).

2.2.4 Objektové modely

Hlavným predstaviteľom je UML. Opäť z podstaty DAQ systémov a jednoduchosti riadiaceho elementu (jednoduchosť vychádza zo špecifikácie, ktorú uvádzame v kapitole 4), nepredpokladáme využiteľnosť objektových modelov. Tieto modely sú vhodné na návrh softvéru z viacerých pohľadov (biznis model, správanie a štruktúra systému apod.). V špecifickom kontexte vnoreného systému sa zameriavame primárne na tok údajov a procesmi nad nimi a práva model toku údajov nie je súčasťou UML modelu.

2.2.5 Metóda porovnania rôznych prístupov

Staines ďalej vyberá 3 oblasti v ktorých všetkých 13 metód porovnáva. Ide o použiteľnosť, ktorá hovorí o jednoduchosti nasadenia tej ktorej metódy a dostupnosť CASE nástrojov. Pozitívne je brané, ak je presne špecifikovaná notácia. V tejto oblasti jasne dominuje UML, ktorej nasadeniu pomáha rozsiahle portfólio CASE nástrojov a priamočiarosť notácie, ktorá je navyše veľmi často používaná aj mimo RT systémov a je preto pravdepodobné, že vývojári sa s ňou už stretli. Druhá oblasť je konzistentnosť metódy, kde sa pozitívne hodnotí čo najviac podobná notácia naprieč procesom modelovania všetkými úrovňami. Práve všeobecnosť UML a teda rôznorodosť notácii rôznych diagramov môže byť zdrojom horšieho pochopenia a výslednom slabom orientovaní sa naprieč modelom. RTSAD, DARTS a z nich vychádzajúce metódy sú od vzniku koncipované na riešenie oblasti RT systémov a tomu zodpovedá aj maximálne jednoduchá notácia. Treťou oblasťou je schopnosť metódy reflektovať špecifickosť RT systémov. Paralelnosť úloh, časovanie, riadiace a údajové toky. UML na naplnenie tejto oblasti vyžaduje rozšírenia definované v UML-RT. Lepšie sú na tom ADARTS a RTSAD, ktoré ale tiež na úplne pokrytie potrieb RT systémov vyžadujú rozšírenia [12].

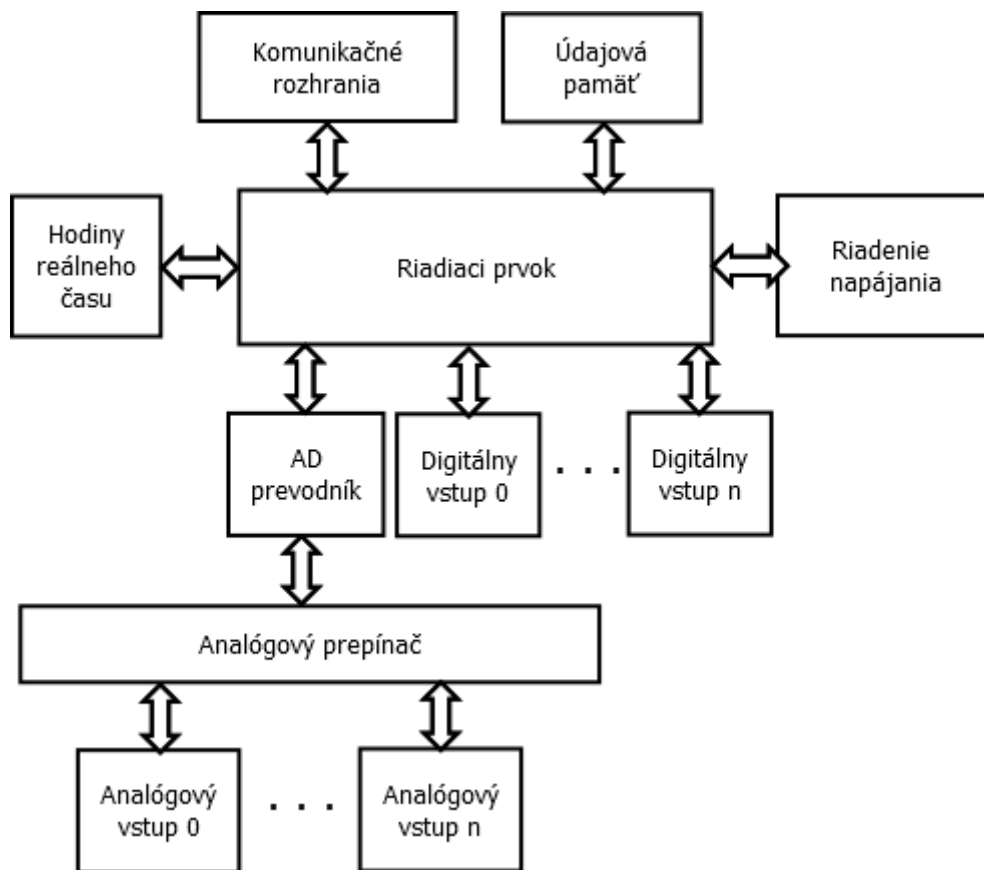
Vzhľadom na strednú komplexnosť navrhovaného systému na zber údajov, jeho špecifické RT požiadavky a ústrednú úlohu tokov údajov sme sa rozhodli použiť metódu RTSAD, ktorá je veľmi dobre špecifikovaná, poskytuje presné postupy na dekomponovanie problému a jednoduchú notáciu, ktorá zbytočne neodvádza pozornosť od kľúčových aspektov systému.

Pre metódu RTSAD sme sa priklonili aj z dôvodu používania v minulosti pri implementácii experimentálneho zariadenia na správu serverov [príloha C].

2.3 Hardvér

Na obrázku 6 je znázornená architektúra navrhovaného systému. Viaceré bloky sú základnými prvkami DAQ systémov, niektoré boli doplnené na zvýšenie univerzálnosti zariadenia. Ústrednou entitou je riadiaci obvod, ktorý môže byť reprezentovaný počítačom, mikropočítačom, systémom na čipe, FPGA obvodom apod. Riadiaci prvok prostredníctvom periférií vykonáva samotné úlohy spojené so zberom údajov. Údaje zo senzorov s digitálnym výstupom číta prostredníctvom n digitálnych vstupov. Obdobne analógové údaje sú čítané AD prevodníkom, ku ktorému sú

prepínačom prepájané rôzne kanály. Po prečítaní a spracovaní sú doplnené časovou značkou, ktorú získava riadiaci prvok z hodín reálneho času a sú zaznamenané do údajovej pamäte, prípadne odosielané okamžite cez komunikačné rozhrania. V neposlednom rade zariadenie má prehľad o stave napájania vďaka špecializovaného bloku riadenia napájania.



Obr. 6.: Blokova schéma DAQ systému.

2.3.1 AD prevodník

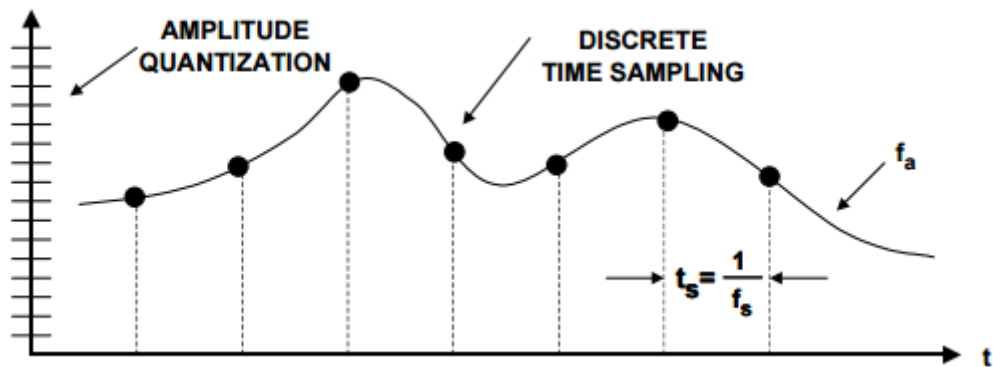
Systém DAQ spracováva vstupy zo svojho okolia, kde je väčšina signálov spojitých. Preto je nevyhnutné doceliť, aby ich dokázal spracovávať. Nakoľko riadiaci prvok je digitálny, musíme použiť moduly, ktoré zabezpečujú transformáciu analógového signálu na digitálnu informáciu tak, aby mikropočítač, ASIC alebo FPGA obvod vedeli takúto fyzikálnu veličinu čítať, uchovávať a manipulovať s ňou.

AD prevodník vzorkuje a kvantuje analógovú informáciu a transformuje ju zo spojitý na diskretnú. Kľúčové parametre sú:

- Vzorkovacia frekvencia - určuje koľkokrát za sekundu sa zaznamená vzorka,

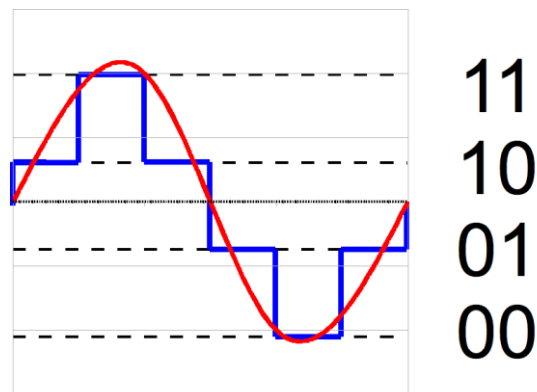
- Bitová hĺbka (rozlíšenie vzorkovania) - značí do koľkých intervalov, úrovni môže byť analógová hodnota začlenená. Nakoľko jeden bit môže nadobudnúť dve logické úrovne, je množstvo kvánt rovný hodnote 2^n , kde n je práve bitová hĺbka, teda počet bitov výslednej digitálnej informácie [19].

Vzorkovanie spočíva v zachytení úrovne amplitúdy analógového signálu v diskrétnych časoch (Obr. 7).



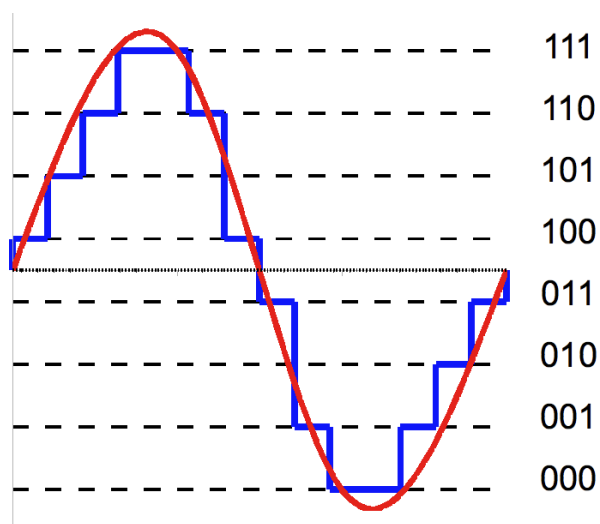
Obr. 7.: Príklad vzorkovania analógovej informácie.

Kvantovanie na druhej strane znamená zaradenie zachytenej amplitúdy do jedného z kvantovacích intervalov. Na obrázku 8 kvantujeme signál v hĺbke 2 bitov, ktoré vytvárajú 2^2 rôznych intervalov. Červený priebeh zobrazuje analógový signál a modrý jeho digitalizovanú podobu. Rozdiely medzi priebehmi sú chybou digitalizácie a predstavujú stratu informácie.



Obr. 8.: Digitalizácia s 2-bitovým rozlíšením.

Na obrázku 9 sme zväčšili hĺbku kvantovania o jeden bit a zdvojnásobili frekvenciu. Digitalizovaná reprezentácia tak vernejšie popisuje pôvodný signál a straty informácie sú nižšie.



Obr. 9.: Digitalizácia s 3-bitovým rozlíšeným vo vyššej frekvencii.

Z vyššie uvedeného vyplýva, že na čo najpresnejší záznam analógovej informácie, ju musíme digitalizovať s čo najvyššou frekvenciou a kvantovať ju s maximálnou bitovou hĺbkou. Prakticky ale pre frekvenciu platí vzorkovací teorém (aj. Nyquist–Shannon sampling theorem), ktorý hovorí, že vzorkovacia frekvencia musí byť minimálne dvojnásobkom maximálnej frekvencie, ktorá sa vyskytne vo vzorke [20].

V tabuľke 1 sú rôzne príklady bitovej hĺbky a prislúchajúceho rozlíšenia AD prevodníka pri referenčnom napätí 5 V.

Tabuľka 1.: Rôzne bitové hĺbky a rozlíšenie AD prevodníka pri $V_{ref} = 5\text{ V}$.

Bitová hĺbka	Počet kvánt	Rozlíšenie ADC
4	16	321,5 mV
8	256	19,5 mV
12	4096	1,22 mV
16	65536	76,3 μV
20	1048576	4,8 μV

Okrem vyššie spomenutých rozdielnych vlastností AD prevodníkov, súvisiacich hlavne s precíznosťou prevodu, pre rôzne aplikácie sa ukazuje rôzne kriticky doba prevodu. Doba prevodu a maximálna frekvencia prevodníkov je výrazne ovplyvnená ich konštrukciou:

- AD prevodníky s postupnou aproximáciou – prevod prebieha iteratívne, v n krokoch. V prvom kroku sa určí hodnota najvýznamnejšieho bitu – MSB, na

základe ktorého sa nastaví prvý rezistor zo sady, cez ktorú sa pripája referenčné napätie porovnávacieho člena. Prevod pokračuje ďalším zaradovaním rezistorov, až pokiaľ rozdiel v napätiach nie je väčší ako kvantovací krok – rozlíšenie prevodníka. Doba prevodu sa spravidla pohybuje rádovo v desiatkach mikrosekúnd.

- Paralelné prevodníky – prevod prebieha vo viacerých porovnávacích členoch naraz, paralelne. Referenčné napätia sa získavajú z rezistorovej siete. Najväčšou nevýhodou je komplikovanosť takéhoto zapojenia, pri ktorom sa používa $2^n - 1$ porovnávacích členov. Výhodou je minimálny počet krokov, v ktorom sa informácia digitalizuje.
- Rôzne kombinácie postupnej aproximácie a paralelných prevodníkov – za cenu predĺženia času prevodu je možné napríklad vo viacerých n krokoch previesť informáciu paralelnými porovnávacími členmi, ktorých bude $2n$ -násobne menej [20].

Pri rôznych experimentoch môžu byť nároky na precíznosť prevodníka rôzne. Pri tejto práci predpokladáme použitie AD prevodníkov integrovaných v púzdre riadiaceho člena, ale existujú rôzne dôvody, pre ktoré má zmysel uvažovať o externých obvodoch zabezpečujúcich prevod. V prvom rade je informácia v analógovej forme náchylná na skreslenie. Mikropočítač je komplikované zariadenie, ktoré v púzdre obsahuje často viaceré zdroje hodinových signálov a na ne zapojené periférie. Tieto periférie môžu negatívne ovplyvniť kvalitu prevodu, aj keď existujú metódy minimalizovania ich dopadu. Pri implementácii síce použijeme integrovaný prevodník (dôvodom je minimalizovanie hmotnosti zariadenia pri teste v kapitole 6.4), jeho výstup sa ale pokúsime ošetriť softvérovým filtrom, prípadne aplikovaním spánkových režimov, ktoré znižujú aktivitu mikropočítača počas prevodu.

Pre prípadné ďalšie hardvérové verzie zariadenia sme identifikovali nasledovné externé obvody, ktoré sú vhodné pre rôzne domény použitia a bolo by vhodné ich implementovať ako vymeniteľné moduly:

- ADS115 - 16-bitové rozlíšenie pri frekvencii 860 Hz a 4 kanáloch, pripojenie prostredníctvom zbernice TWI,
- LTC2225 - 12-bitové rozlíšenie pri frekvencii až 10 MHz, pripojenie prostredníctvom 12-bitovej paralelnej zbernice.

2.3.2 Riadiaci obvod

Jadrom navrhovaného systému je riadiaci obvod, ktorý prostredníctvom periférií vykonáva všetky funkcie DAQ. Takéto systémy bývajú často riešené ako periférie k osobnému počítaču. Takáto architektúra ale nespĺňa viaceré z kritérií, ktoré sme si určili v cieľoch tejto práce. Systém má byť schopný samostatnej prevádzky v externom prostredí, ktoré môže byť ťažko dostupné a nehostinné (teplota, vlhkosť, problémy s napájaním). Odolný priemyselný počítač významne predražuje riešenie a môže byť motiváciou na odcudzenie. Z vyššie uvedených dôvodov sa pri výbere riadiaceho elementu orientujeme na mikropočítače a obvody FPGA.

2.3.2.1 Mikropočítače s jadrom AVR

Spoločnosť Atmel v roku 1996 uviedla na trh radu mikropočítačov s modifikovanou harvardskou architektúrou. Jadro AVR má šírku údajovej zbernice 8 bitov a môže byť taktované na frekvenciu 16 MHz. Väčšina inštrukcií je vykonaných v jednom hodinovom takte, čo vo výsledku predstavuje výkon až do 16 MIPS. Ceny základných vývojových dosiek od čínskych dodávateľov sa pohybuje v rozpätí od 5€ do 7€.

Z hľadiska dostupnosti a ceny sme sa rozhodli vybrať ako reprezentatívny obvod mikropočítač ATmega32, ktorý disponuje 32 kB programovej pamäte typu flash, 1,5 kB údajovej pamäte typu SRAM a 3 kB permanentnej pamäte typu EEPROM. Sada implementovaných periférií je pomerne zaujímavá, aj keď z každého typu je integrovaná len jedna fyzická zbernica. Na čipe je umiestené SPI, USART, TWI (I²C). Súčasťou je AD prevodník s rozlíšením 10 bitov a maximálnou frekvenciou 15 kHz [2].

Za hlavnú výhodu tohto obvodu považujeme veľmi komplexnú správu napájania s množstvom rôznych režimov. To predstavuje výhodu hlavne v zariadeniach s obmedzeným zdrojom napájania. Spotrebu obvodu sme experimentálne namerali na úrovni 1 mA na 1 MHz v režime running. V režime extended standby spotreba klesne až tisícnásobne.

2.3.2.2 Mikropočítače s jadrom ARM Cortex-M3

V roku 2003 sa na trh dostali mikropočítače s 32-bitovými jadrami ARM Cortex-M3. Tieto obvody sa vyznačujú špičkovým pomerom ceny a výkonu. Obvody v cene rádovo jednotiek dolárov poskytujú 32 bitov širokú zbernicu pri

hodinovej frekvencii až 180 Mhz a výkonom podľa syntetického testu Dhrystone až na úrovni 225 MIPS na maximálnej taktovacej frekvencii [13].

Vzhľadom na dostupnosť, cenu a dokumentáciu sme ako reprezentatívny obvod vybrali STM32F103C8R8. Základné vývojové dosky, ktoré obsahujú minimum súčiastok potrebných na prevádzku je možné objednať od dodávateľov z Číny v cenách od 6 do 8€.

Tento obvod rady STM32 má na čipe 128 kB programovej pamäte typu flash a 20 kB údajovej pamäte typu SRAM. Jadro je 32-bitové a obsahuje redukovanú inštrukčnú sadu s integrovaným radičom DMA. Hlavnou devízou je množstvo hardvérovo implementovaných periférií, menovite 3x USART s riadiacimi signálmi, 2xTWI (resp. I²C), 2xSPI, CAN zbernica. Za veľmi výraznú výhodu v doméne DAQ systému je precízny AD prevodník s rozlíšením 12 bitov, ktorý dokáže pracovať na frekvenciách až do 1 MHz. Súčasťou je analógový prepínač-multiplexor, ktorý dokáže prepojiť na prevodník jeden z 10 pinov.

Spotrebu obvodu sme experimentálne namerali na úrovni 0,7 mA na 1 MHz v režime running. Použitím správy napájania vieme spotrebu znížiť na približne tretinu.

2.3.2.3 Obvody FPGA

Obvody FPGA, teda programovateľné hradlové polia sú čipy, ktorých funkcionality je daná návrhárom až po výrobe. Na programovanie sa využívajú hardvér opisné jazyky, HDL. Výhodou je variabilita, ktorú získava vývojár a možnosť prototypovania. Cieľovou doménou obvodov FPGA sú zariadenia vyrábané v malých sériách, ktoré nie je cenovo výhodné navrhovať ako obvody ASIC (zákaznícke čipy) a na ktorých komplexnosť je použitie konvenčného mikropočítača nedostatočné. Ďalšou doménou je prototypovanie ASIC obvodov, prípadne v čase rekonfigurovateľné zariadenia. Takýto riadiaci obvod by si vedel aktualizáciou vytvoriť nové, efektívnejšie bloky, vytvoriť nové komunikačné rozhrania a pod. [9]

V posledných rokoch je trendom nepoužívať homogénne hradlové polia logických členov, ale poskytnúť viaceré stavebné bloky na čipe. Integruje sa SRAM, registre a iné podporné obvody, čo pri rozsiahlejších čipoch umožňuje syntetizovanie aj zložitejších mikropočítačov a mikroprocesorov.

Ako referenčný obvod sme vzhľadom na dostupnosť u nášho dodávateľa vybrali Xilinx XC3500, ktorý obsahuje 500 000 hradiel. Ekvivalent 0,5 milióna hradiel je

dostatočný na syntetizovanie jednoduchého OpenRISC jadra a niektorých periférií, napr. ethernetu. Cena obvodu sa pohybuje okolo 30€. Odhad spotreby obvodu FPGA je náročným procesom, ktorého výsledok je len približný. Vo fáze analýzy sa môžeme spoľahnúť len na výrobcom uvádzaný maximálny odber na úrovni 106 mA v komerčnom prevedení [23].

2.3.2.4 Záver

Zariadenie, ktoré v tejto práci navrhujeme a implementujeme musí byť schopné dlhodobo pracovať na batérie. Nakoľko ale nie je pravdepodobné prenášanie zariadenia, je možné spoľahnúť sa na priebežné dobíjanie fotovoltackým panelom pomerne veľkej batérie. AVR jadrá poskytujú relatívne vysoký výkon s veľmi dobrou správou napájania, nedostatočná je ale údajová pamäť SRAM (zásobníky pre ethernet, pamäťovú kartu a namerané údaje).

Pre významné množstvo experimentov považujeme 12 bitový AD prevodník (obvody radu STM32F10x) ako dostatočný, výhodné je jeho integrovanie v čipe, čo môže prispieť k zvýšeniu spoľahlivosti. V prípade ak by precíznosť 12 bitov bola nepostačujúca, je možné veľmi jednoducho ADC vlastnosti DAQ systému na báze takéhoto mikropočítača vylepšiť pripojením externého obvodu na jedno z mnohých poskytovaných rozhraní.

Vzhľadom na to, že mikropočítače s jadrom Cortex spĺňa naše požiadavky (pomerne nízka spotreba, dostatok periférií a údajovej pamäte), nevidíme významný dôvod na použitie FPGA procesorov, s ktorým sa spájajú dodatočné náklady a zvýšená komplexnosť implementácie.

2.3.3 Komunikačné rozhrania

Ak systém pre zber údajov plánujeme navrhnuť ako univerzálne zariadenie, je dôležité zvážiť aj on-line odosielanie zaznamenaných údajov, nakoľko mnohé praktické aplikácie takýto prístup vyžadujú. Vzhľadom na dostupnosť internetu je tak možné údaje analyzovať priebežne a ušetriť náklady na výjazdy zamestnancov a zbieranie pamäťových médií. Rovnako pri zničení alebo ukradnutí zariadení používateľ nestratí merania, ktoré mohli byť ťažko opakovateľné.

2.3.3.1 Káblové pripojenie

Na vytvorenie komunikácie medzi vnoreným systémom a osobným počítačom sa v praxi používa viacero rozhraní, medzi najčastejšie používané uvádzame:

- RS232 – v minulosti veľmi obľúbený štandard na pripájanie meracích prístrojov a priemyselných zariadení. Veľmi jednoduchá duplexná, sériová zbernica využíva logiku $\pm 12\text{ V}$ a je vhodná bez modifikácii aj na väčšie vzdialenosti. V súčasnosti je vytlačaná USB zbernicou a ethernetom podľa aplikačnej domény. RS232 s inými napäťovými úrovňami je bežnou súčasťou aj najlacnejších mikropočítačov – UART, USART, konverziu zabezpečuje veľmi spoľahlivý a známy obvod MAX232.
- USB – sériové rozhranie, ktoré vytlačilo množstvo predchádzajúcich štandardov, okrem zmieňovaného univerzálneho RS232 aj špecifické PS2, Gameport. USB je často implementované hardvérovo do mikropočítačov, prípadne existujú knižnice na jednoduchú softvérovú implementáciu aj do jednoduchších obvodov.
- Ethernet – dnes najčastejšie rozhranie na pripojenie počítačov do komunikačnej siete. Výhodou je možnosť zapojenia do už spravidla vybudovanej ethernetovej siete a možnosť napájať zariadenie prostredníctvom PoE – Power over Ethernet. Nevýhoda je zložitejšie zabezpečenie po zdieľanej zbernici. Hardvérová implementácia je skôr sporadická a len na mikropočítačoch vyššej triedy, zabezpečiť sa ale dá rôznymi obvodmi, ktoré potom s μP komunikujú prostredníctvom jemu známej zbernice, napríklad SPI [3].

V minulosti by bolo možné uvažovať aj nad LPT rozhraním, ktoré je univerzálne, ale zlyháva vo veľmi voľnej štandardizácii naprieč výrobcami (hlavne priebehy a časovanie signálov), zložitým riadením, nízkymi prenosovými rýchlosťami a samozrejme vytrácaním sa zo základných dosiek počítačov [1].

Z pohľadu kontextu nasadenia systému na zber údajov sme sa rozhodli implementovať rozhranie ethernet. Hlavnými výhodami je konfigurovateľnosť prostredníctvom webového prehliadača, možnosť umiestniť zariadenie kdekoľvek na sieti, čo je výhodné hlavne v priemysle, kedy je možné zbierať údaje na viacerých miestach a spracovávať jediným serverom.

2.3.3.2 Bezdrôtové rozhrania

Systémy pre zber údajov môžu slúžiť na automatizovanie meraní priamo v teréne. Často nie je možné použiť drôtové, pevné pripojenie a podmienky experimentu vyžadujú spracovávanie v reálnom čase. Vtedy sa dá s výhodou využiť bezdrôtové, neinvazívne pripojenie, ktoré sa v posledných rokoch stáva čoraz viac preferované a teda aj náklady na implementáciu drasticky klesajú.

Na bezdrôtové pripojenie uvažujeme nasledovné možnosti:

- Wi-Fi – pokrytie bezdrôtovými wi-fi sieťami v osídlených oblastiach je čoraz intenzívnejšie. Problémom je spravidla súkromné vlastníctvo týchto sietí a teda vzniká len veľmi málo príležitostí, kedy by sa systém na zber údajov mal použiť na experiment v teréne na mieste, ktoré je pokryté sieťou, do ktorej má prístup používateľ. Nevýhodou je rovnako ako pri ethernet zdieľaný kanál, ktorý v prípade bezdrôtovej siete je ešte viac otvorený. Výkon vnorených systémov je z definície obmedzený a implementácia komplexnejších zabezpečovacích algoritmov by mohla vyčerpať významné množstvo zdrojov.
- Mobilná sieť – pokrytie územia Slovenskej republiky mobilnou sieťou najväčšieho operátora sa na koncu roka 2014 blížila k 90%. To vytvára dobrý predpoklad na vyžitie tohto komunikačného rozhrania k priebežnému odosielaniu nameraných údajov z prostredia mimo laboratória. Systémy na zber údajov, ktoré sa anglicky označujú ako *data logger* navyše generujú veľmi nízky dátový tok a preto ani finančná náročnosť takéhoto riešenia nie je vysoká, dokonca ani pri dlhodobých experimentoch. Výhodou je možnosť využívania pri experimentoch vo vozidlách a všeobecne v pohybe.
- Bluetooth – bluetooth vo forme priemyselných modulov so smerovými anténami poskytuje veľmi dobré možnosti na prepájanie zariadení v rádovo kilometroch. Nevýhoda je nutnosť priamej viditeľnosti a pri väčších vzdialenostiach potreba budovania opakovačov. Neposledne v porovnaní s inými technológiami vyššia cena celého riešenia. Natol'ko diskutovaná bezpečnosť je vo verzii BT 2.0 riešená vynucovaním šifrovania a obmenou kľúčov ešte skôr ako je ich možné prelomiť.
- Rádio – rádiové pripojenie historicky rozhodne patrí do výčtu možností bezdrôtového prenosu. Vzhľadom na neustále znižovanie finančnej náročnosti ostatných riešení ho nepovažujeme za vhodné na implementáciu do

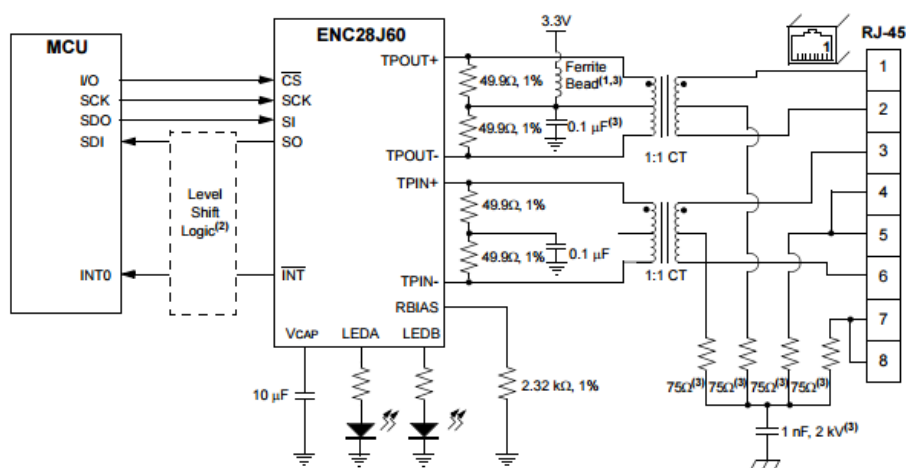
všeobecného zariadenia. Nevýhoda je tiež náročnosť na napájacie zdroje a nutnosť budovania si vlastnej siete opakovačov.

Komunikačné rozhrania navrhovaného DAQ systému sú ethernet a bluetooth modul. Na komunikáciu po krútenej dvojlinke použijeme obvod ENC28J60. Tento modul je známy a veľmi obľúbený, knižnice s rutinami sú súčasťou väčšiny RTOS. Na zabezpečenie bezdrôtovej komunikácie sme na testovacie účely tejto práce vybrali veľmi dobre dostupný modul JY-MCU s obvodom BC417.

2.3.3.3 Obvod ENC28J60

ENC28J60 je ethernetový radič s rozhraním SPI, vyžadujúci minimum ďalších pripojených súčiastok. Je dodávaný v DIL púzdrach, ktoré sú vhodné na výrobu prototypov a aj QFN do produkcie. Má hardvérovo implementovanú fyzickú vrstvu 10-BaseT a MAC vrstvu. Obvod obsahuje 8 kB zásobník na vysielané a prijímané rámce. Obvod sa skladá zo 7 hlavných blokov:

1. SPI rozhranie na pripojenie s riadiacim obvodom.
2. Kontrolné registre na nastavovanie a monitorovanie stavu obvodu.
3. Pamäť typu SRAM, ktorá dočasne uchováva prijaté a odoslané rámce.
4. Riadiaca a dátová zbernica.
5. Blok riadenia prístupu do SRAM pri využívaní DMA.
6. MAC modul, ktorý implementuje funkcionality IEEE 802.3
7. Modul fyzickej vrstvy, ktorý dekoduje a kontroluje prijaté bity.



Obr. 10.: Zapojenie rozhraní ethernet a SPI k obvodu ENC28J60.

Obvod je veľmi jednoduchý a pokrýva iba prvé dve fyzické vrstvy modelu TCP/IP. Tento obvod je ale veľmi populárny a existuje množstvo knižníc na

zabezpečenie implementácie všetkých nutných protokolov na komunikáciu v prostredí TCP/IP sietí. Obrázok 11 znázorňuje zapojenie obvodu k zberniciam SPI a ethernet. Na zabezpečenie základnej funkcionality sa pripája k obvodu len transformátormi oddelený konektor RJ45.

2.3.3.4 Obvod W5100

Obvod W5100 má hardvérovo implementovaný úplný TCP/IP zásobník, teda značne zjednodušuje vývoj aplikácie a samozrejme aj zvyšuje predvídateľnosť celého systému, čo je v prípade RT systémov kľúčové. Oproti ENC28J60 poskytuje 4-násobok kapacity zásobníka na prijímanie a vysielanie rámcov. Väčší zásobník umožní predĺžiť periódy spúšťania úloh zabezpečujúcich sieťovú komunikáciu a zabezpečiť lepšiu flexibilitu odozvy kľúčových úloh, tj. tých ktoré zabezpečujú samotné zbieranie úloh. K mikropočítaču sa obvod pripája opäť prostredníctvom rozhrania SPI, teda po rýchlejšej, duplexnej, synchronizovanej, sériovej zbernici. Jediným záporom oproti ENC28J60 je horšia dostupnosť obvodu a priemyselných modulov založených na tomto čipe.

2.3.3.5 Obvod BC417

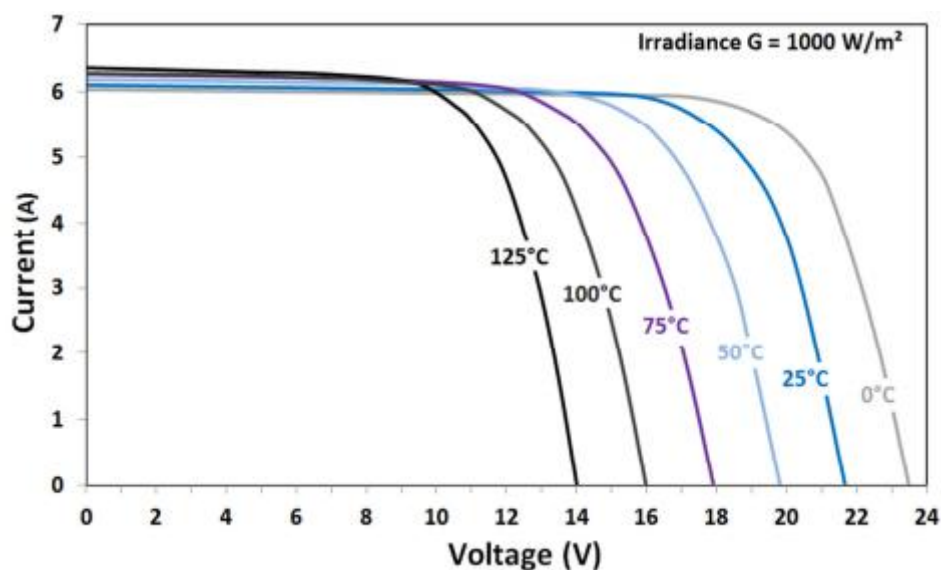
MC417 je veľmi obľúbený integrovaný obvod na zabezpečenie bluetooth komunikácie v moduloch. Zjednodušene poskytuje náhradu za káblové prepojenie medzi UART rozhraniami. Na základnú funkcionality nie je nutné robiť žiadne špeciálne úkony mimo tradičných rutín nad UART. Komplexnejšia funkcionality je dostupná cez dobre zdokumentované AT príkazy.

2.3.4 Správa napájania

Vzhľadom na to, že systém navrhujeme ako univerzálny, medzi scenármi použitia je vykonávanie experimentov mimo laboratória, často v priestoroch s obmedzenou dostupnosťou napájania. Z tohto dôvodu implementujeme aj systém správy a dohľadu napájania z akumulátora a jeho dobíjanie prostredníctvom fotovoltického panelu. Touto cestou dokážeme minimalizovať nutnosť výjazdov k zariadeniu a zlepšiť plánovanie údržby vďaka dostupnosti dôležitých informácií o stave batérie a panelu.

Na meranie nabíjacieho a odoberaného prúdu, napätia akumulátora používame AD prevodník integrovaný v riadiacom mikropočítači. Na dobíjanie akumulátora prostredníctvom FV panelu existujú dva základné prístupy:

- PWM – akumulátor je cez spínací tranzistor pripájaný k panelu vždy ak je napätie na jeho svorkách pod definovanú úroveň. Nevýhodou je úbytok napätia na úroveň batérie, nakoľko je FV panel veľmi mäkký zdroj, naopak výhodou je jednoduchosť zapojenia.
- MPPT – metóda ktorá využíva nastaviteľný menič napätia, ktorého strieda je nastavovaná tak, aby FV panel pracoval v bode maximálneho výkonu. Nevýhoda je komplexnosť zapojenia, nutnosť implementovať algoritmus hľadania optimálneho pracovného bodu, úbytok napätia na meniči a komplikovaná VA charakteristika panelu, ktorá môže pri čiastočnom zatemnení obsahovať lokálne maximum.



Obr. 11.: Úbytok výkonu a napätia na FV paneli [17].

Boli publikované viaceré štúdie [19], z ktorých vyplýva, že komplikovaný prístup dobíjania prostredníctvom MPPT neprináša zvýšenie odoberaného výkonu, hlavne z dôvodu úbytkov napätia, ktoré môžu dokonca v horúcich dňoch obmedziť dobíjanie pre príliš nízke napätie panelu (negatívna závislosť napätia a teploty). Obrázok 12 ukazuje rôzne výkonové krivky a úbytky napätia pri teplotách blížiacich sa 100 °C, ktoré sú reálne dosiahnuteľné počas horúcich letných dní.

Vďaka využívaniu integrovaných obvodov s pokročilou správou napájania sme sa rozhodli pre technologicky jednoduchší a v konečnom dôsledku spoľahlivejší prístup dobíjania prostredníctvom PWM. Jednoduchosť algoritmu dokonca nevyklučuje riadenie hlavným riadiacim mikropočítačom (úloha riadenia nabíjania má krátku a konštantnú dobu behu), ktorý tak vie veľmi jednoducho pričleniť údaje správy napájania k nameraným údajom.

2.3.5 Zapojenie hardvéru

V tejto práci sa nezameriavame na podrobný návrh hardvéru. Využívame prototypovacie moduly, ktoré značne zjednodušujú a zrýchľujú návrh zariadenia. Ich pripojenie spočíva v privedení napájania na napájacie piny a prepojenie komunikačných rozhraní. Mimo modulov bolo nutné pripojiť minimum ďalších súčiastok:

- Zenerove a schottkyho diódy na digitálnych a analógových vstupoch. Zenerova dióda chráni vstup pred vyšším napätím, aké dovoľuje výrobca priviesť na vstup mikropočítača. Rýchla schottkyho dióda v závernom smere chráni vstup pred záporným napätím. Obdobne sú chránené napájacie vstupy.
- Bočník a operačný zosilňovač na nepriame meranie nabíjacieho prúdu.
- Spínací MOSFET tranzistor na pripojenie akumulátora k fotovoltaiickému panelu.
- Stavové LED diódy na výstupne piny mikropočítača. Svetelné diódy vyžadujú prúdový zdroj, preto boli pripojené sériovo s rezistorom, ktorý limituje maximálny prúd podľa katalógového listu.

2.4 Operačný systém

Pri implementácii systému pracujúceho v reálnom čase ma vývojár spravidla tri možnosti. Jednou z nich je použitie hotového RT operačného systému, ktorý okrem plánovania úloh v systéme často ponúka knižnice na prácu s perifériami, prípadne zabezpečuje funkcie zvyšovania spoľahlivosti kontrolovaním operácií s pamäťou apod. Druhou možnosťou je zostavenie si vlastného plánovača, ktorý môže byť navrhnutý priamo na cieľovú aplikáciu, čím sa dá získať veľmi vysoká efektivita na tradične limitovanom obvode. Na druhej strane stojí samozrejme náročnosť a čas potrebný na implementáciu a v prípade kritických aplikácií aj netriviálne testovanie kódu [17] [22].

Poslednou často vyberanou alternatívou je riadenie volaní periodických úloh prerušeniami cez čítače, v prípade aperiodických úloh, prerušeniami z podsystémov mikropočítača. Zjavnou výhodou je minimálna réžia na riadenie toku programu, pochopiteľne nevýhoda je minimálny dosah na plánovanie a možnosť vyhľadovania systému príliš často plánovanou aperiodickou úlohou.

V článku, ktorý sme publikovali na konferencii IIT.SRC v roku 2015 je predstavená zdokonalená metóda na overovanie RT vlastností akéhokoľvek operačného systému použiteľného vo vnorených systémoch [príloha C]. V praktickej časti sme

porovnali najpopulárnejší RT operačný systém FreeRTOS a používateľský plánovač, ktorý vznikol doimplementovaním doplnkových funkcií do RIOS (mimoriadne úsporná implementácia round robin plánovača, ktorá vznikla na University of California). Merania prebehli na architektúre AVR a vyplynulo z nich, že FreeRTOS systém bez zásadných modifikácií len veľmi ťažkopádne a nepredvídateľne reaguje na aperiodické stimuly.

Ďalšou limitáciou v našom prípade je potreba dosiahnuť minimálnu hmotnosť pri použití v stratosférickom balóne. Z toho vyplýva požiadavka na minimalizovanie spotreby (menšie batérie a teda aj hmotnosť). Samotný operačný systém zvyšuje nároky na dodávaný prúd, popri zvýšených nárokoch na pamäť a strojový čas, ktoré môžu poskytnúť len vyššie rady procesorov. Tie sú spravidla dodávané na väčších a zložitejších moduloch, ktoré sa vyznačujú vyššou hmotnosťou.

Na základe vyššie uvedených dôvodov sme sa rozhodli pre spúšťanie úloh na základe pravidelných prerušení čítača. Aperiodickou úlohou je napríklad zápis na kartu a komunikácia. V oboch prípadoch ale vieme veľmi dobre odhadnúť ich najkratšiu možnú periódu pripravenosti a tak nehrozí vyhľadovanie systému.

2.5 Zabezpečenie komunikácie

Naš prototyp komunikuje dvomi rozhraniami, prostredníctvom bluetooth a ethernetu, ktorý sa využíva aj na konfiguráciu vo webovom rozhraní.

Bluetooth sa bežne používa ako komunikačné bezdrôtové rozhranie medzi telefónmi a perifériami. Je teda očakávané, že rieši aj úroveň zabezpečenia. Bezpečnostný úrad NSA považuje zabezpečenie za dostatočné (dokument Bluetooth Security - the National Security Agency https://www.nsa.gov/ia/_files/wireless/I732-016R-07.pdf), aj keď s medzerami, ktoré spočívajú v spoliehaní sa na úsudok používateľa. Práve najslabším článkom je potvrdzovanie párovania. V momente keď používateľ potvrdí spárovanie zariadenia, zariadenia si vymenia šifrovacie kľúče a potenciálne nebezpečná periféria má často neobmedzený prístup k službám. Významný krok ku zvýšeniu bezpečnosti prišiel s verziou 2.1, ktorá vynucuje šifrovanie, pokiaľ ho obe zariadenia podporujú. Nebezpečenstvo vieme znížiť prepnutím do neviditeľného režimu, kedy komunikačný modul nereaguje na scan žiadosti a jeho adresa BT_ADDR nevyhnutná k párovaniu ostáva skrytá. V takomto prípade neostáva iná možnosť ako pokúsiť sa adresu uhádnuť skúšaním. Jej veľkosť je 8 bajtov, teda počet možností na

vyskúšanie je 2^{64} . Prostredníctvom AT príkazov vieme nastaviť len špecifické adresy, ktorým sa umožní párovanie. Počet kombinácií sa nám zvýši na 2^{128} , čo je približne $2 \cdot 10^{38}$ kombinácií zdrojovej a cieľovej adresy modulu. Ako posledný spôsob zabezpečenia používame zakazovanie služieb, pokiaľ nie sú nevyhnutné. Konfigurácia je tak dostupná len v prípade inicializácie zariadenia, počas samotného zberu údajov by rekonfigurácia znehodnotila výstupný súbor a preto nie je dôvod, aby ostala poskytovaná a predstavovala tak bezpečnostné riziko.

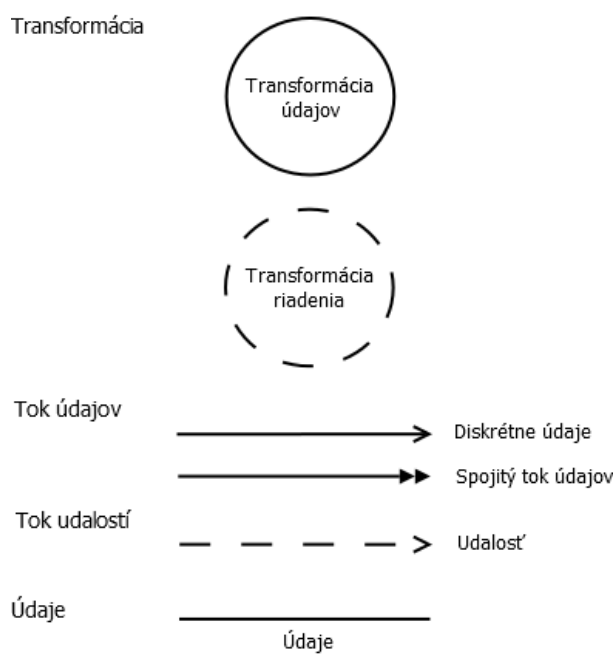
Zabezpečenie údajov prechádzajúcich rozhraním ethernet je výrazne náročnejšou úlohou. Na architektúru STM32 existujú komplexné riešenia, ktoré ponúkajú protokolový zásobník vrátane SSL implementácie v HTTPS. Problémom je pamäťová náročnosť. Obľúbená kombinácia, ktorá je podporovaná aj výrobcom pozostáva z modulov FreeRTOS, lwIP a PolarSSL. Celková náročnosť na operačnú pamäť je výrobcom uvádzaná na 76484 bajtov [11]. Rozhodli sme sa tak šifrovať údaje nad protokolom HTTP, v zmysle šifrovania bloku používateľských údajov metódou DES, ktorá je hardvérovo akcelerovala na jadrách Cortex-M3. Výmenu kľúčov v našej aplikácii neriešime, zariadenie dostane symetrický kľúč ako súčasť firmvéru.

V prípade zvýšených požiadaviek na zabezpečenie je možné využiť pokročilejší algoritmus 3DES, ktorý je rovnako podporovaný hardvérom a jeho funkcionality je podobne podporovaná štandardnými STM32 knižnicami. V takomto prípade je nevyhnutné počítať so zvýšenými nárokmi na výpočtový výkon a teda predĺženie doby vykonávania komunikačných úloh.

3 Návrh riešenia

V tejto kapitole prostredníctvom metódy RTSAD postupne dekomponujeme systém. Prechádzame kontextovou úrovňou, ktorá znázorňuje komponenty systému. Pokračujeme úrovňou podsystémov a tie rozkladáme na funkcie, toky údajov a úložiská údajov. Na záver prostredníctvom stavového diagramu vizualizujeme viaceré aspekty systému.

Model je úzko naviazaný na požiadavky ktoré má zariadenie plniť a postupne ich dekomponuje na úlohy, toky údajov medzi nimi a dátové úložiská. Dekompozícia na úlohy tak už v prvotných fázach umožňuje vykonať RT analýzu a dokázať tak, že systém v konečnom dôsledku bude plniť špecifikované doby odoziev [6]. Obrázok 12 obsahuje použitú notáciu výsledných modelov RTSAD.



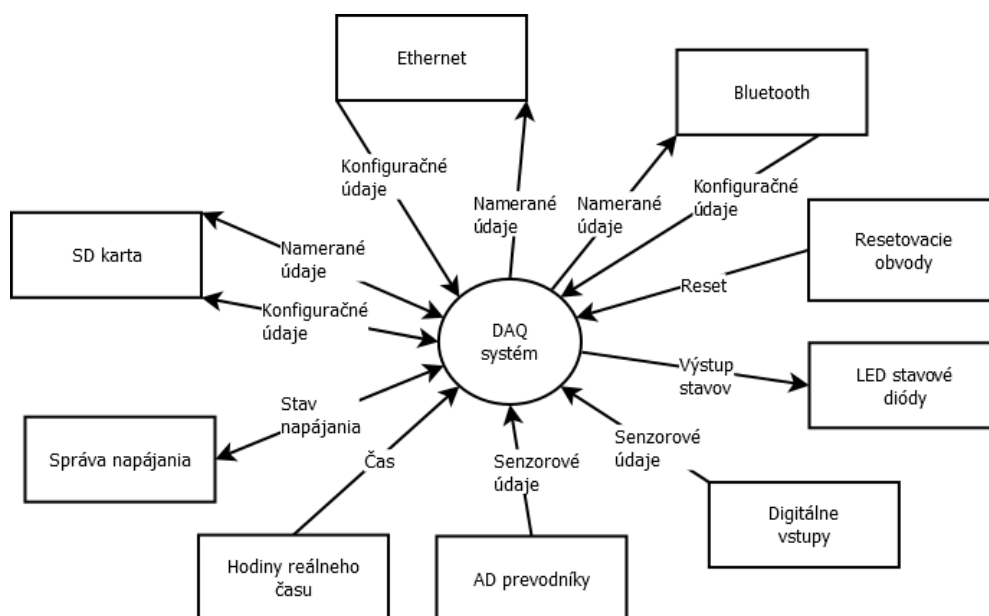
Obr. 12.: Notácia RTSAD.

Po dekompozícii a tvorbe modelu je výsledkom množina funkcií, ktoré sa nazývajú transformáciami alebo procesmi a rozhraní medzi nimi, ktoré tvoria toky údajov alebo riadiacich signálov

3.1 RTSAD model

Na obrázku 13 je znázornený kontextový diagram. Je to prvá úroveň reprezentácie modelu, ktorá zobrazuje všetky hlavné entity systému. Z obrázku vyplýva, že DAQ systém obsahuje komunikačné rozhrania ethernet a bluetooth. Obe rozhrania umožňujú odosielať namerané údaje mimo systém a prijímať konfiguráciu do systému.

Namerané údaje sú ukladané na pamäťovú kartu, odkiaľ môžu byť aj čítané v prípade dlhšej nedostupnosti pripojenia prostredníctvom komunikačných rozhraní. Na karte sú uložené aj konfiguračné údaje, aby systém v prípade zlyhania a resetu dokázal pokračovať v nastavenom experimente bez zásahu obsluhy. Resetovacie obvody reprezentuje resetovací obvod a príbuzné systémy zvyšujúce spoľahlivosť systému. Menovite watchdog, ktorý resetuje riadiaci prvok bez odozvy a brown-out detection obvod, ktorý resetuje obvod v prípade poklesu napájania, kedy hrozí nekorektné vykonávanie inštrukcií. Informácie o stave zariadenia sú vyobrazované svetelnými diódami. Surové údaje zo senzorov prichádzajú z AD prevodníkov a digitálnych vstupov. Časové značky vracajú hodiny reálneho času a nakoniec informácie o stave akumulátora a jeho nabíjania získavame z bloku správy napájania.



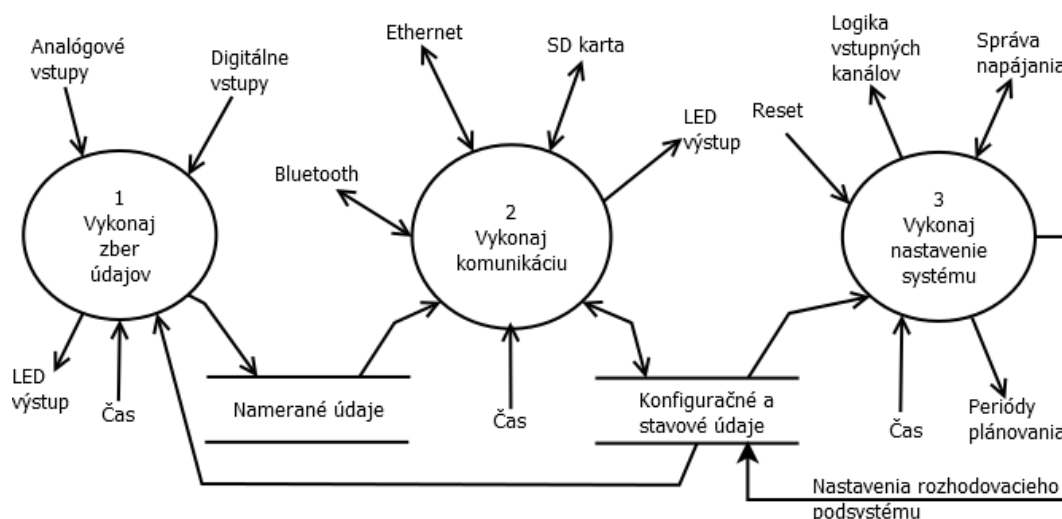
Obr. 13.: Kontextový diagram DAQ systému.

3.1.1 Funkčná dekompozícia

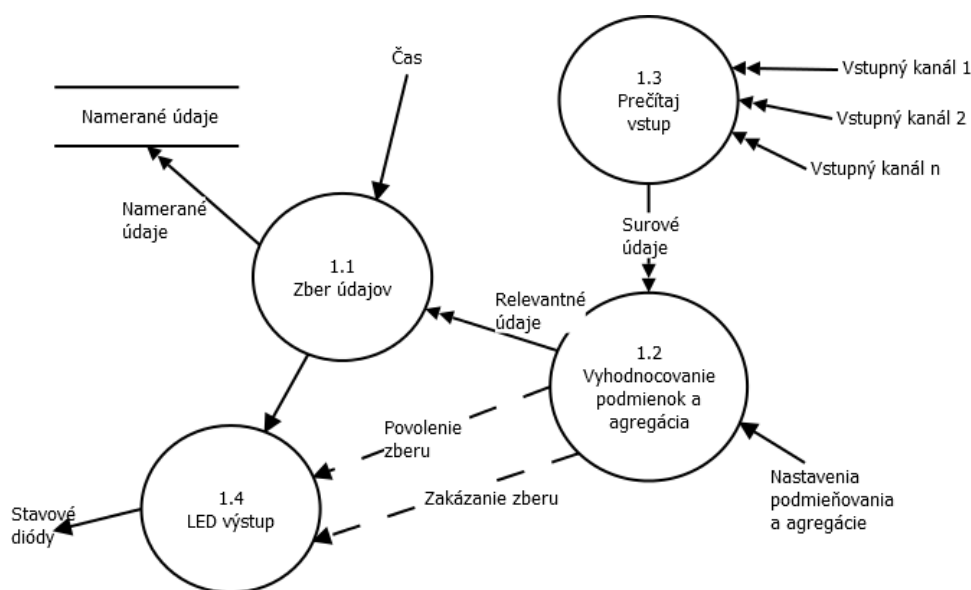
Obrázok 14 znázorňuje funkčnú dekompozíciu. Aplikácia systému reálneho času je takto rozložená do kľúčových funkcií. Výsledkom je trojica relatívne samostatných transformácií, procesov – 1 Vykonaj zber údajov, 2 Vykonaj komunikáciu a 3 Vykonaj nastavenie systému. Táto úroveň modelovania prináša najvyššiu úroveň procesov a dátových tokov medzi nimi. Slovo vykonaj na začiatku každej funkcie v tomto modeli vychádza z konvencie RTSA, z ktorej vychádza RTSAD.

Z povahy navrhovaného systému vieme, že kritickou z pohľadu časovania je transformácia 1 Vykonaj zber údajov. Transformácia musí zbierať údaje v čo

najpravidelnejších intervaloch a správne ich označovať časovými známkami. Pri ďalších úrovniach dekompozície ostanú kritické úlohy len v jej podriadených transformáciách.



Obr. 14.: Dekompozícia na podsystémy.



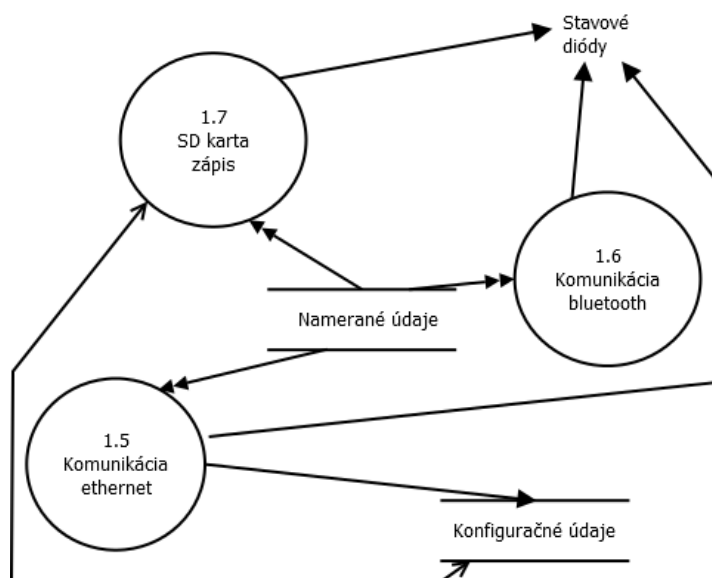
Obr. 15.: Vykonaj DAQ - transformácia.

3.1.2 Funkčná dekompozícia druhej úrovne

Na obrázku 15 je ďalšia dekompozícia prvej transformácie z obrázku 14. Model zobrazuje tok údajov z jednotlivých kanálov vstupných senzorov, ktoré sú načítavané transformáciou 1.3 Prečítaj vstup. Následne surové údaje prechádzajú cez transformáciu 1.2 Vyhodnocovanie podmienok a agregácia, ktorá vyhodnocuje podmienky na zahájenie zberu údajov a agregovanie údajov a následné vyhodnocovanie ich priemeru, miním, maxím. Výsledky agregácie a údaje vyhovujúce podmienkam prechádzajú do

1.1 Zber údajov, v ktorej sú doplnené o časovú známku a sú ukladané do dočasného zásobníka nameraných údajov. Tak Zber údajov ako aj Vyhodnocovanie podmienok a agregácia môžu zmeniť nastavenie svetelných diód, z ktorých je možné odpozorovať aktuálny stav zariadenia.

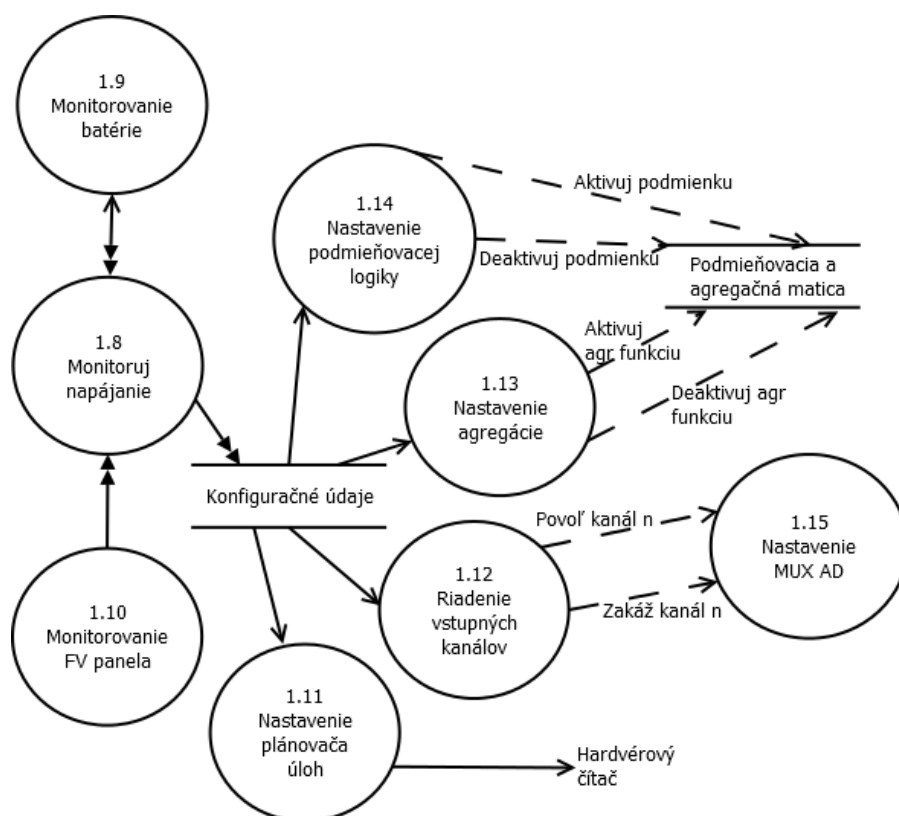
Transformácia 2 Vykonaj komunikáciu je podrobnejšie rozkreslená na obrázku 16. Ústrednou entitou je úložisko nameraných údajov, kde sa ukladá výsledok procesu zberu údajov predchádzajúcou transformáciou. Toto úložisko je v princípe zásobník, ktorý je priebežne napĺňaný a po dosiahnutí určitej úrovne sa údaje z neho presúvajú na SD kartu (1.7 SD karta zápis), prípadne komunikačné rozhrania (1.5 Komunikácia ethernet, 1.6 Komunikácia bluetooth) tak, aby tieto vstupno-výstupné operácie boli v maximálnej možnej miere efektívne. Navyše všetky tri transformácie môžu meniť stav výstupných LED diód.



Obr. 16.: Vykonaj komunikáciu – transformácia.

Špeciálne prípady sa dajú vyčítať z prepojenia transformácie 1.7 SD karta zápis a úložiska Konfiguračné údaje, ktoré slúži na uchovávanie nastavenia v prípade neočakávaného resetu zariadenia. Nastavenie tak môže byť obnovené a systém sa zotaví po obnovení korektných podmienok. Posledným prepojením je 1.5 Komunikácia ethernet na konfiguračné údaje, čo značí konfigurovateľnosť prostredníctvom siete.

Transformácia 3 Vykonaj nastavenie systému v diagrame toku údajov na obrázku 16 zabezpečuje hlavne dohľad nad napájaním, nastavovanie podmieňovacieho a agregáčného podsystému a výber aktívnych vstupných kanálov.



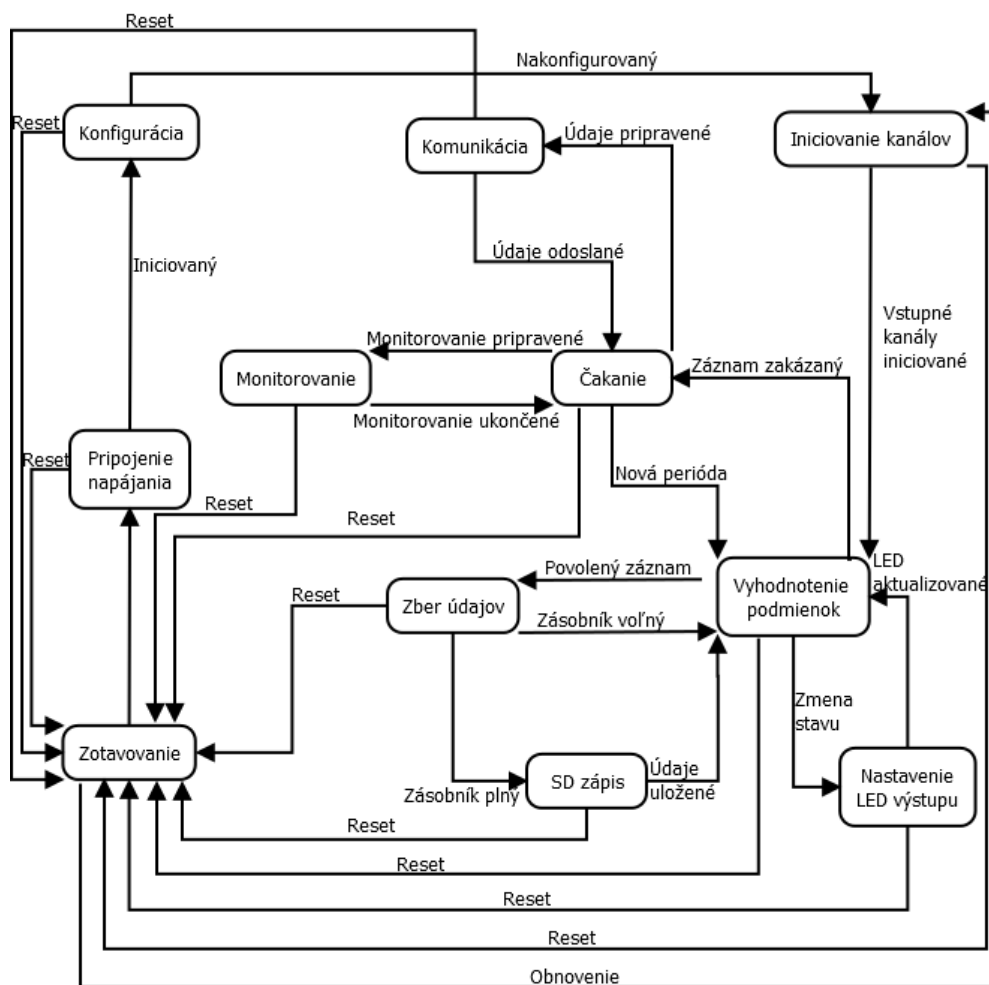
Obr. 17.: Vykonaj systémové nastavenie a kontrolu – transformácia.

Transformácia 1.8 prostredníctvom 1.9 Monitorovanie batérie a 1.10 Monitorovanie FV panela zisťuje aktuálny stav nabitia a kondíciu zdroja, povoľuje a rozhoduje o nabíjaní z fotovoltaičského panela a meria nabíjací prúd. Výstup z nej sa zaraďuje medzi konfiguračné údaje, čo prostredníctvom 2 Vykonaj komunikáciu umožňuje alarmovať používateľa, ktorý môže vopred zabezpečiť výjazd technikov v prípade zlej kondície akumulátora, poškodenia alebo zanesenia povrchu FV panela. 1.13 Nastavenie agregácie a 1.14 Nastavenie podmieňovacej logiky prostredníctvom povoľovacích a zakazovacích príkazov modifikujú maticu funkcií a kanálov. Jednotlivé kanály môžu byť vyradené alebo priradené transformáciami 1.12 Riadenie vstupných kanálov a 1.15 Nastavenie MUX AD. V neposlednom rade dochádza k nastavovaniu hardvérového čítača, ktorý periodicky vyvoláva funkciu čítania kanálov funkciou 1.11 Nastavenie plánovača úloh.

3.1.3 Analýza stavových prechodov

Na obrázku 18 je znázornený diagram stavov a prechodov medzi nimi. Zariadenie po pripojení napájania vykoná inicializáciu a čaká na konfiguráciu, kedy má aktívne sieťové rozhrania, ale neprebíha zbieranie údajov. Po konfigurácii sa nastaví vstupné kanály, čo spočíva v ich povolení v rutine prepínania multiplexora AD

prevodníku a matice podmieňovania a agregácie. V nasledujúcom stave už prebieha vyhodnocovanie podmienok záznamu a agregovanie funkcií (stav Vyhodnotenie podmienok). Ak sú podmienky splnené, zariadenie vykoná zber údajov na vstupe a uloží ich do zásobníka, prípadne zásobník vyprázdni na pamäťovú kartu SD.. Podmieňovací systém v prípade zmeny svojho stavu môže v určitých intervaloch vyvolať nastavenie stavových LED diód. V prípade, že podmienky na záznam splnené nie sú, prechádza sa do stavu čakania, v ktorom sa v daných intervaloch vyvolá monitorovacia rutina. Podobne sa v stave čakania vyvoláva aj komunikačná rutina. Zo stavu čakania systém vychádza po prerušení generovaného pretečením čítača.



Obr. 18.: Diagram stavových prechodov.

V ktoromkoľvek stave je možné resetovanie systému tlačidlom, watchdogom alebo brown-out detekciou. V takom prípade zariadenie vyvoláva zálohu konfigurácie z pamäťovej karty a autonómne prechádza na iniciovanie kanálov a zber údajov.

4 Špecifikácia prototypu

V predchádzajúcich kapitolách sme analyzovali a navrhli systém na zber údajov. V tejto kapitole špecifikujeme funkcionálne a nefunkcionálne požiadavky kladené na implementáciu prototypu. Opis implementácie a následného overovania je uvedený v nasledujúcich kapitolách.

4.1 Funkcionálne požiadavky prototypu

Naším cieľom je vytvoriť zariadenie schopné zbierať údaje zo senzorov s analógovým a digitálnym výstupom. Zariadenie je použiteľné na široké spektrum experimentov, pri ktorých je nevyhnutný záznam vo vyšších frekvenciách, rádovo desiatkách kHz. Na druhej strane často nemá zmysel všetky veličiny sledovať tak často, získali by sme tak údaje ktoré sú duplicitné, nemajú výpovednú hodnotu. Príkladom je meranie teploty, pri ktorom sa prejavuje tepelná zotrvačnosť materiálu senzoru a aj doba potrebná na tepelnú výmenu medzi prostredím a púzdrom senzoru. Preto zariadenie musí poskytovať nástroje na selekciu údajov, čím poskytnuté záznamy budú maximálne relevantné. Podobne pri experimente nemusí používateľa zaujímať akýkoľvek stav sledovaného subjektu. Experiment môže špecifikovať špecifické udalosti, pri ktorých má dôjsť k zberu údajov. V oboch prípadoch je vhodné, ak má DAQ zariadenie implementovaný rozhodovací podsystém, ktorý vyberá z množiny zaznamenaných údajov len tie zaujímavé.

Pri vývoji kladieme dôraz na nízku cenu zariadení, nakoľko chceme vytvoriť podmienky aj pre paralelné nasadenie systémov v rôznych lokalitách. V skratke, chceme vytvoriť univerzálne zariadenie, ktoré pomôže používateľom, výskumníkom pri tvorení zaujímavých experimentov. Funkcionálne požiadavky sme definovali nasledovne:

- Systém prijíma digitálne signály na štyroch kanáloch vo frekvencii 2 kHz.
- Systém implementuje digitálny podmieňovací podsystém (trigger), ktorý umožňuje:
 - uchovať časovú značku zachytenia nábežnej hrany,
 - uchovať časovú značku zachytenia dobežnej hrany,
 - uchovať časovú značku zmeny logickej úrovne (nábežná alebo dobežná hrana),

- uchovať početnosť impulzov v špecifikovanej perióde,
- uchovávať hodnoty na vstupe v špecifikovanej perióde.
- Systém prijíma analógové signály na štyroch kanáloch vo frekvencii 2 kHz.
- Systém implementuje analógový podmieňovací podsystem (trigger), ktorý umožňuje:
 - zaznamenať údaje v prípade prekročenia nastavenej hodnoty amplitúdy na analógovom vstupe,
 - zaznamenať údaje v prípade detekcie hrany (nábežnej, dobežnej, akejkolvek) na konfigurovateľnej úrovni amplitúdy,
 - zaznamenať početnosť hrán impulzov na konfigurovateľnej úrovni amplitúdy,
 - uchovávať hodnoty na vstupe v špecifikovanej perióde,
 - hľadať minimálnu, maximálnu a priemernú hodnotu v sade údajov s konfigurovateľnou mohutnosťou.

Zariadenie je testované pri frekvencii zberu údajov vo frekvencii 2 kHz z dôvodu pripravovaného vypúšťania stratosférického balóna. Prototyp vyvíjame a testujeme pri nižšej frekvencii tak, aby sme zabezpečili nižšiu spotrebu a vyššiu odolnosť voči poruchám (spôsobené hlavne rušením).

4.2 Nefunkcionálne požiadavky

Na systém sú okrem funkcionálnych požiadaviek kladené aj rôzne doplnkové špecifikácie, ktoré sme stanovili nasledovne:

- Konfigurácia prostredníctvom štandardných nástrojov (webový prehliadač),
- Možnosť spracovania výstupov v aj bežne dostupných nástrojoch (napr. MS Excel),
- Spotreba neprekračujúca 100 mA (umožňuje napájanie z akumulátora s dobíjaním malým FV panelom),
- Systém zaznamenáva údaje na SecureDigital kartu (vhodné kapacity a výborná cena za jednotku kapacity),
- Systém umožňuje priebežné odosielanie nameraných hodnôt na vzdialený server,
- Systém umožňuje ochranu integrity odosielaných údajov zabezpečením.

5 Implementácia prototypu

V tejto kapitole sa venujeme kľúčovým častiam samotnej implementácie na základe analýzy, vytvoreného modelu a špecifikácie v predchádzajúcich kapitolách.

5.1 Vstupno-výstupne piny

Mikropočítač STM32F103Cx má integrovaných 37 univerzálnych vstupno-výstupných pinov. Ich konfigurácia spočíva v pripojení hodinového signálu, ich frekvencie a jedného z 8 režimov uvedených v tabuľke 2.

Tabuľka 2.: rôzne funkcie GPIO na mikropočítačoch STM32.

Funkcia	Označenie v knižnici
Alternatívna funkcia (open-drain)	GPIO_Mode_AF_OD
Alternatívna funkcia (push-pull)	GPIO_Mode_AF_PP
Analógový vstup	GPIO_Mode_AIN
Digitálny vstup (floating)	GPIO_Mode_IN_FLOATING
Digitálny vstup (pull-down)	GPIO_Mode_IPD
Digitálny vstup (pull-up)	GPIO_Mode_IPU
Digitálny výstup (open-drain)	GPIO_Mode_Out_OD
Digitálny výstup (push-pull)	GPIO_Mode_Out_PP

Alternatívna funkcia pin pripája k integrovanému podsystému obvodu. V prípade konfigurácie analógového vstupu, pin sa pripája na multiplexor jedného z prevodníka. Digitálny vstup je možné čítať v jednej z dvoch logických úrovní a naopak digitálny výstup je možné nastaviť na jednu z dvoch logických úrovní. V prípade alternatívnej funkcie a digitálneho výstupu je možné konfigurovať pin v režime push-pull, kedy tranzistor podľa nastavenia pripája pin na zem resp. na napájanie alebo open-drain, kedy tranzistor pin pripája len na zem a v prípade nastavenia na logickú jednotku je pin odpojený. V druhom prípade sa očakáva pripojenie externého rezistoru, ktorý "vyťahuje" logickú úroveň nahor. Ďalšie nastavenia digitálneho výstupu sú pull-down, ktorý pripája pin interným rezistorom na zem, pull-up, obdobne s interným rezistorom, ale pripojením na napájanie. Treťou alternatívou je floating, bez pripojenia digitálneho vstupu [13].

Časť použitej konfigurácie prostredníctvom štandardnej STM32 knižnice je uvedený nižšie:

```
// * * * * * GPIO init block * * * * *
GPIO_InitTypeDef gpio;
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE);
gpio.GPIO_Pin = GPIO_Pin_0 | GPIO_Pin_1 | GPIO_Pin_2 | GPIO_Pin_3; //analogove DAQ
gpio.GPIO_Speed = GPIO_Speed_50MHz; //rychlost portu
gpio.GPIO_Mode = GPIO_Mode_AIN; //rezim analogovy vstup
GPIO_Init(GPIOA, &gpio); //struktura na konfiguracne registre
gpio.GPIO_Pin = GPIO_Pin_4 | GPIO_Pin_5 | GPIO_Pin_6 | GPIO_Pin_7; //dig. vstupy DAQ
gpio.GPIO_Speed = GPIO_Speed_50MHz;
gpio.GPIO_Mode = GPIO_Mode_IPU; //dig. vstup pull-up
GPIO_Init(GPIOB, &gpio);
gpio.GPIO_Pin = GPIO_Pin_0; //debug vystup DAQ
gpio.GPIO_Speed = GPIO_Speed_50MHz;
gpio.GPIO_Mode = GPIO_Mode_Out_PP; //push-pull vystup
GPIO_Init(GPIOB, &gpio);
```

5.2 AD prevodník

Mikropočítače série STM32 poskytujú 2 až 3 AD prevodníky s precíznosťou 12 bitov a maximálnou frekvenciou 1 MHz. Typ prevodu je postupná aproximácia (predstavená bližšie v kapitole 2.3.1). Súčasťou je analógový multiplexor, ktorý na prevodník môže prepojiť jeden z 10 kanálov. ADC poskytujú široké možnosti konfigurácie a to sa prejavilo aj na pomerne zložitej inicializácii. V našej implementácii používame nasledovnú inicializáciu s použitím STM32 standard lib:

```
// * * * * * AD converter init block * * * * *
RCC_APB2PeriphClockCmd(RCC_APB2Periph_ADC1, ENABLE); //pripojenie hodin na ADC1
RCC_ADCCLKConfig(RCC_PCLK2_Div6); //delicka hodin na ADC: 6
ADC_InitTypeDef ADCstruct;
ADCstruct.ADC_Mode = ADC_Mode_Independent; //konverzia jedinym AD prevodnikom
ADCstruct.ADC_ScanConvMode = DISABLE; //konverzia jednotlivych kanalov
ADCstruct.ADC_ContinuousConvMode = DISABLE; //jednorazova konverzia
ADCstruct.ADC_ExternalTrigConv = ADC_ExternalTrigConv_None; //neriadeny externym zdrojom
ADCstruct.ADC_DataAlign = ADC_DataAlign_Right; //zarovnanie 12-bitov doprava (zľava nuly)
ADCstruct.ADC_NbrOfChannel = 1; //pocet konvertovanych kanalov

ADC_Init(ADC1, &ADCstruct); //nastavenie konf. registrov strukturou

ADC_Cmd(ADC1, ENABLE); //povolenie ADC

ADC_ResetCalibration(ADC1);
while(ADC_GetResetCalibrationStatus(ADC1)); //cakanie na reset kalibracie
ADC_StartCalibration(ADC1);
while(ADC_GetCalibrationStatus(ADC1)); //cakanie na ukoncenie kalibracie
```

Prevodník funguje v nezávislom režime, bez spolupráce s ďalšími integrovanými ADC. Konvertuje vždy jediný kanál a len na vyžiadanie. Namerané údaje sa zarovnávajú doprava a zľava doplnia nulami. Na konci konfigurácia sa vykoná reset kalibračných nastavení prevodníka a spustí sa automatická rekalkulácia, ktorá zabezpečí korektné prevody pri rôznych teplotách. Následne sa v procese zberu údajov

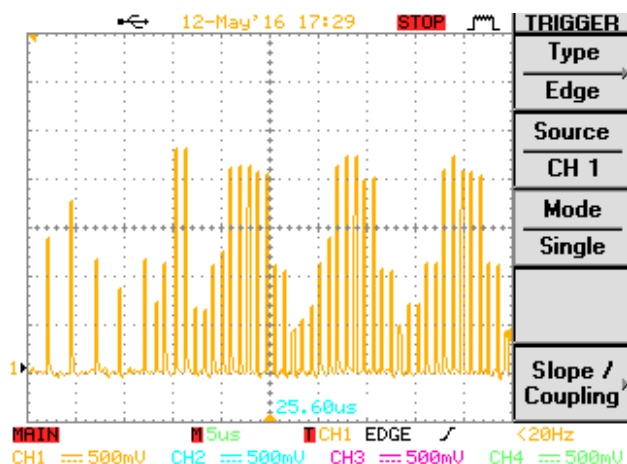
vyžiada konverzia postupne na všetkých štyroch kanáloch, tak ako je uvedené v príklade prevodu z jedného kanálu nižšie:

```
ADC-RegularChannelConfig(ADC1, ADC_Channel_0, 1, ADC_SampleTime_13Cycles5);  
//nastavenie konverzie  
ADC_SoftwareStartConvCmd(ADC1, ENABLE); //zaciatok konverzie  
while(ADC_GetFlagStatus(ADC1, ADC_FLAG_EOC) == RESET); //cakanie na ukoncenie prevodu  
ain0 = ADC_GetConversionValue(ADC1); //vystup ADC -> uint16_t
```

Príklad čítania nastaví parametre na konverziu z kanálu 0, prevodníku ADC1 po dobu 13,5 cyklu, čo stačí na úplný prevod v 12 bitoch.

5.3 SPI rozhranie a pamäťová karta

Zbernica SPI je veľmi často používaná na pripojenie zariadení požadujúcich vyššiu rýchlosť prenosu k mikropočítačom. Poskytuje plný duplex, teda zariadenia môžu paralelne komunikovať v oboch smeroch a vo frekvencii až do 18 MHz pri nami vybranom obvode. Hardvérovo je SPI implementované ako posuvný register, vybavený prerušeniami a možnosťou výberu aktívneho zariadenia zo skupiny paralelne pripojených (chipselect, konkrétne SS). Nakoľko STM32f103 neposkytuje SDIO rozhranie, ktoré dokáže komunikovať s kartou v paralelnom režime po 4bitovej zbernici, rozhodli sme sa práve pre SPI, ktoré je bežnou súčasťou radičov kariet SD, napriek nižšej dosiahnuteľnej rýchlosti [4].



Obr. 19.: Priebeh hodinového signálu na karte SD.

Pochopiteľne postupnosť údajov, ktoré zapíšeme na kartu nie je čitateľná pre počítač. Pretože v našich nefunkčných požiadavkách kladieme dôraz na jednoduchosť používania zariadenia, je dôležité pridať nad ovládač SPI a SD ďalšiu vrstvu, ktorá zabezpečí ukladanie údajov v štruktúre používanej počítačmi. Ďalšou vrstvou tak je súborový systém, najčastejšie používanými medzi mikropočítačmi je kvôli jednoduchosti FAT a FAT32.

Modul FatFS je jednou z najobľúbenejších implementácií súborového systému pre zariadenia s obmedzenými zdrojmi. Je platformovo nezávislý a programovaný v C. Úlohou vývojára je napísať nízkoúrovňové rutiny na komunikáciu s hardvérom. V našom prípade potrebujeme využívať zbernicu SPI a hodiny reálneho času.

Samotná karta sa zosúva do modulu, ktorý je pripojený k mikropočítaču. Pri prvých testoch sme použili kartu triedy Class 2, ktorá má maximálnu rýchlosť 2MBps. Vo výsledku zariadenie fungovalo rádovo jednotky sekúnd, pokiaľ funkcie FatFS nezačali vracaa chyby. Následne sme priebehy na SPI zbernici pripojili na osciloskop a ukázali sa, že sú značne deformované. Amplitúda hodinového signálu kolísala medzi 0,9V a 3,3V. Na obrázku 19 je vidieť priebeh tohto signálu, ktorý spôsoboval značnú nestabilitu prototypu. Rovnaká nespoľahlivosť sa ale prejavovala pri rôznych kartách rovnakej triedy, nepredpokladáme tak, že by sme mali chybný kus. Pri použití modernejšej karty s triedy Class 10, ktorá má 5 násobnú priepustnosť tento problém nenastával. Amplitúda signálu dosahovala vo všetkých vrcholoch 3,3 V a zariadenie bolo schopné pracovať bez chyby dlhodobo.

Samotná inicializácia SPI zbernice prebieha nasledovne:

```
GPIO_InitTypeDef gpio;

RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE); //hodiny na GPIO
RCC_APB2PeriphClockCmd(RCC_APB2Periph_SPI1, ENABLE); //hodiny na SPI1

gpio.GPIO_Pin = GPIO_Pin_5 | GPIO_Pin_6 | GPIO_Pin_7; //alternatívna
gpio.GPIO_Speed = GPIO_Speed_50MHz; //funkcia pinov
gpio.GPIO_Mode = GPIO_Mode_AF_PP; //5,6,7 ako MISO
GPIO_Init(GPIOA, &gpio); //MOSI a CLK

gpio.GPIO_Pin = GPIO_Pin_4; //sw riadený slave select
gpio.GPIO_Speed = GPIO_Speed_50MHz;
gpio.GPIO_Mode = GPIO_Mode_Out_PP;
GPIO_Init(GPIOA, &gpio);

SPI_InitTypeDef spi;
int prescaler = SPI_BaudRatePrescaler_128;

SPI_Cmd(SPI_SD, DISABLE);

spi.SPI_Direction = SPI_Direction_2Lines_FullDuplex; //obojsmerne
spi.SPI_Mode = SPI_Mode_Master; //SD je slave
spi.SPI_DataSize = SPI_DataSize_8b; //shiftuje sa 8 bitov
spi.SPI_CPOL = SPI_CPOL_Low; //hodiny su low v idle
spi.SPI_CPHA = SPI_CPHA_1Edge; //data su platne na prvej hrane

spi.SPI_NSS = SPI_NSS_Soft; //sw slave select
spi.SPI_BaudRatePrescaler = prescaler; //SPI_BaudRatePrescaler_128
spi.SPI_FirstBit = SPI_FirstBit_MSB; //data od MSB
spi.SPI_CRCPolynomial = 7; //hw vypocet CRC
SPI_Init(SPI_SD, &spi);

SPI_Cmd(SPI_SD, ENABLE);
```

Vo výsledku dostaneme plne inicializované komunikačné rozhranie SPI. Ihneď po pripojení hodín na VV piny a SPI1 modul nastavujeme štruktúru vstupno-výstupných pinov, kde pin 5, 6 a 7 je nastavený na alternatívnu funkciu. Podľa katalógového listu vyplýva, že práve tieto vodiče sú vnútorne pripojiteľné na SPI1 modul. Režim PP - push pull pripája dvomi tranzistormi výstup na zem alebo napájanie, nepotrebujeme tak externé rezistory na zbernici, terminátory. Pin 4 je nastavený na štandardný výstup. SS linku si tak budeme nastavovať v rutinách zbernice. Na záver štruktúru nastavíme na GPIOA modul.

Druhá polovica príkladu nastavuje štruktúru SPI, kde vyberáme bity tak, aby SPI fungovalo v plne duplexnom režime. Komunikácia tak môže prebiehať v jednom čase oboma smermi. V režime master generujeme hodinový signál našim zariadením, ktoré odosiela bloky v osmiciach bitov a začína vždy najvýznamnejším - MSB. Neaktívne hodiny ostávajú na úrovni logickej nuly a údaje sú platné už na prvej hrane hodín. Generovanie CRC je nastavené podľa špecifikácie komunikačného protokolu SD karty. Tak ako sme pred nastavením zakázali SPI rozhranie, musíme ho v závere, po aplikovaní štruktúry, povoliť.

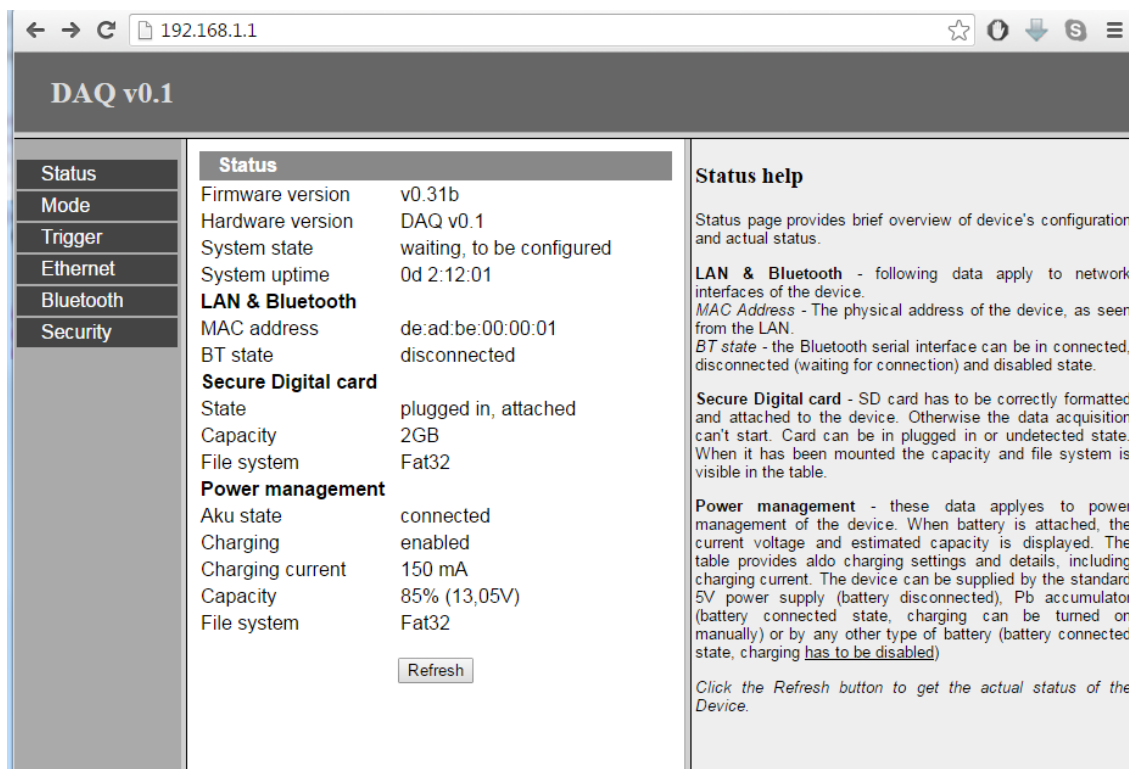
Na zápis a čítanie údajov sme využili FatFS modul, ktorý napriek minimálnym pamäťovým nárokom plne rieši problematiku súborového systému. Obslužné rutiny SD karty a SPI sme použili od Domena Puncera, ktorý ich uvádza na svojom blogu (<http://cba.si/>). V závere sme modifikovali rutiny na prácu s hodinami reálneho času. Vo výsledku vieme pristupovať k údajom na karte prostredníctvom štandardných funkcií `f_open` (otvorenie alebo vytvorenie súboru), `f_read` (čítanie súboru do zásobníka), `f_write` (zápis zásobníka do súboru), `f_close` (vyprázdnenie bufferu a zatvorenie súboru), `mount` (inicializácia karty), `umount` (deinicializácia karty).

5.4 SPI rozhranie a sieťová konektivita

Ako sme spomenuli v kapitole vyššie, SPI rozhranie je obľúbené na pripojenie periférií, ktoré vyžadujú vyššie prenosové rýchlosti (počítanie CRC, synchronizácia, duplex). Pochopiteľne sme sa rozhodli využiť túto zbernicu na pripojenie zariadenia k sieti ethernet. Konfigurácia je obdobná ako v príklade vyššie. Jediným rozdielom je nastavenie deličky tak, že výsledná rýchlosť je násobne vyššia.

Tak ako v príklade vyššie sme potrebovali súborový systém, pri ethernete potrebujeme protokolový zásobník tcp/ip. Veľmi bežnými knižnicami, ktoré umožnia

komunikovať po sieti sú uIP (microIP) a lwIP (lightweight TCP/IP). LwIP má plne implementovaný TCP protokol, DHCP, PPP. Na druhej strane jeho náročnosť na údajovú pamäť je približne dvojnásobkom celkovej kapacity STM32F103. Modul uIP je pamäťovo efektívnejší (to sa samozrejme prejavuje na nižšej výkonnosti), ale jeho implementácia je naviazaná na operačný systém, ktorý sme sa snažili hlavne kvôli obmedzeným zdrojom procesora nepoužiť. Ako alternatívu sme tak vybrali kód Patricka Leymana (<http://patrickleyman.be>), ktorý implementuje ovládač enc28j60 všeobecne pre stm32 architektúru a s pomerne nenáročnými úpravami je možné ho spustiť aj na našom, veľmi obmedzenom mikropočítači. Výraznou nevýhodou tohto riešenia je absencia implementácie TCP, ktorú potrebujeme pre HTTP protokol. Týmto protokolom chceme odosielať údaje do databázy (http klient) a aj sprístupniť webové rozhranie na konfiguráciu (http server). TCP sme sa rozhodli implementovať bez segmentovania paketov, čím značne ušetríme zdroje, aj keď budeme obmedzovaní veľkosťou stránky. Na obrázku 20 je snímok obrazovky konfiguračného rozhrania. Web sa skladá z rámcov, čím dokážeme stránku rozložiť na bloky menšie ako 1500 bajtov a vyhnúť sa tak segmentácii TCP paketov. Kaskádový štýl je odosielaný v jednom pakete, rovnako ako menu, hlavička, nápoveda a hlavný panel.



Obr. 20.: Návrh webovej stránky konfiguračného rozhrania.

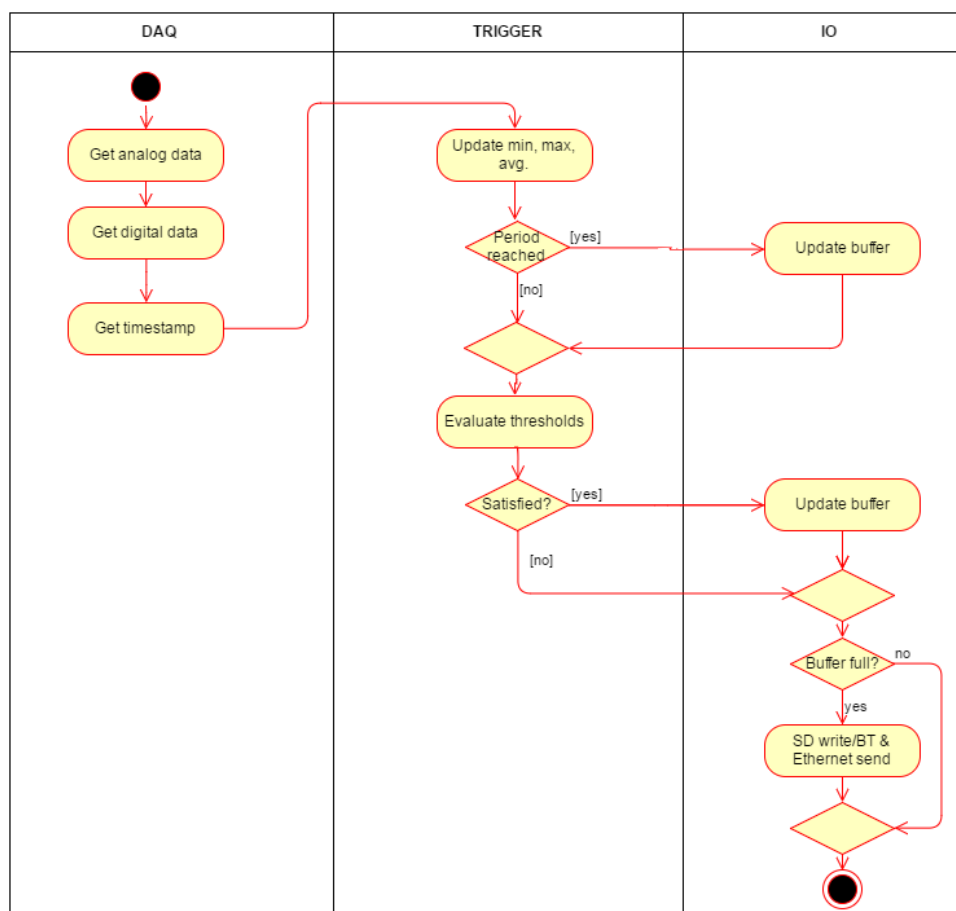
5.5 Spúšťací podsystem, trigger

Spúšťací systém zvyšuje relevantnosť výsledkov. Experimentátor podsystem nastaví tak, aby DAQ systém zaznamenával len údaje, z ktorých vyhodnotenia vychádzajú užitočné informácie. Vo výsledku tak ušetríme poplatky za prenos cez mobilnú sieť a zjednodušíme vyhodnocovanie (excel má obmedzenia v množstve riadkov, ktoré otvorí zo súboru, maximálny počet bodov zobraziteľných v grafe apod.). V kapitole 2.1.4 sme predstavili viacúrovňový trigger experimentu ATLAS. Ide o mimoriadne komplexný systém, ktorý je vyvinutý na konkrétnu aplikáciu. Bežne rozpoznávame nízkoúrovňový a vysokoúrovňový trigger. Nízkoúrovňový trigger je v prípade ATLASu Level-1, ktorý deteguje prítomnosť produktov zrážky. Neprebieha tak zber údajov v čase, keď v detektore nie sú žiadne častice. Obdobne môže nízkoúrovňový trigger čakať na zábezdnú hranu na jednom alebo kombinácii vstupov, prekročenie prahovej hodnoty apod. Vzhľadom na špecifikáciu všeobecného triggeru implementujeme hlavne nízkoúrovňový trigger s rozšíreniami v podobe priemerovania agregovaných údajov, zisťovania ich miním, maxím v nastaviteľnej množine. Spôsob akým sme špecifikovali trigger v kapitole 4, umožňuje efektívny zber údajov zo senzorov, ktoré sme si vybrali na experimenty v kapitolách 6.2 až 6.4, ktoré pokrývajú rôzne situácie z reálneho využitia DAQ systému.

Na obrázku 21 je diagram znázorňujúci zjednodušený pohľad na zber údajov, rozhodovanie triggeru a aktualizácia zásobníka a jeho vyprázdňovanie na SD kartu a komunikačné rozhrania. Implementácia tohto diagramu je vo funkcii, ktorá je volaná najčastejšie zo všetkých, ide o horúce miesto v kóde, ktoré má zmysel naozaj pochopiť a napísať maximálne optimálne. Tok funkcie začína načítaním vstupov AD prevodníka zo všetkých kanálov a načítanie hodnôt na digitálnych vstupoch. V ideálnom prípade sú vstupy načítané paralelne, čo ale na danej architektúre nie je možné. Reálne sa snažíme aspoň minimalizovať oneskorenia medzi čítaním rôznych kanálov.

K zaznamenaným hodnotám je doplnený čas z hodín reálneho času. Úlohu následne preberá rozhodovací podsystem, ktorý aktualizuje agregáčnne funkcie hľadania minima, maxima a priemeru. Pokiaľ bol dosiahnutý používateľom stanovený prah agregovania, sú hodnoty zapísané do zásobníka. Podobne používateľ rozhoduje o prahových hodnotách a kombinácii vstupov, na ktorých sa vyhodnocujú. Pokiaľ sú podmienky splnené, zachytené údaje sa presunú opäť do zásobníka, inak sú zahodené, pretože bolo rozhodnuté o ich neúčinnosti. Na konci rutiny sa v prípade naplnenia

buffer zapisuje na pamäťovú kartu a odosiela komunikačnými rozhraniami. Hlavne v prípade pamäťovej karty, nie je efektívne robiť zápis nenaplneného buffera, pretože sa zakaždým zapisuje celý blok o veľkosti 512 bajtov.



Obr. 21.: Spúšťač podsystem v kontexte zberu údajov a vstupno-výstupnom rozhraní.

Samotné rozhodovanie podmienovacie podsystemu prebieha nasledovne:

```

// * * * * * trigger * * * * *
if (trigger_edge_sensitivity != TRIGGER_DISABLED) {
    switch (trigger_src) {
        case ANALOG0: trigger_thres = ((ain0 < trigger_threshold_level) ? BELOW : ABOVE); break;
        case ANALOG1: trigger_thres = ((ain1 < trigger_threshold_level) ? BELOW : ABOVE); break;
        case ANALOG2: trigger_thres = ((ain2 < trigger_threshold_level) ? BELOW : ABOVE); break;
        case ANALOG3: trigger_thres = ((ain3 < trigger_threshold_level) ? BELOW : ABOVE); break;
        case DIGITAL0: trigger_thres = ((din0 == Bit_RESET) ? BELOW : ABOVE); break;
        case DIGITAL1: trigger_thres = ((din1 == Bit_RESET) ? BELOW : ABOVE); break;
        case DIGITAL2: trigger_thres = ((din2 == Bit_RESET) ? BELOW : ABOVE); break;
        case DIGITAL3: trigger_thres = ((din3 == Bit_RESET) ? BELOW : ABOVE); break;
        default: break;
    }

    switch (trigger_edge_sensitivity) {
        case RISING: trigger_result = (((trigger_thres_old != trigger_thres) && \
            (trigger_thres == ABOVE)) ? ENABLE : DISABLE); break;
        case FALLING: trigger_result = (((trigger_thres_old != trigger_thres) && \
            (trigger_thres == BELOW)) ? ENABLE : DISABLE); break;
        case ANY_EDGE: trigger_result = ((trigger_thres_old != trigger_thres) ? ENABLE : DISABLE); break;
        case BELOW_THR: trigger_result = (trigger_thres == BELOW) ? ENABLE : DISABLE; break;
        case ABOVE_THR: trigger_result = (trigger_thres == ABOVE) ? ENABLE : DISABLE; break;
        default: break;
    }

    trigger_thres_old = trigger_thres;
} else {
    trigger_result = ENABLE;
}

```

V prípade ak je trigger zakázaný, celé rozhodovanie je preskočené a výsledok rozhodovania je napevno nastavený na povolený. Ak je vybraný zdroj rozhodovania, podľa typu vstupného signálu sa vyhodnotí aktuálna pozícia amplitúdy vzhľadom na nastavený prah. Následne sa podľa pozície v aktuálnej a predchádzajúcej perióde vyhodnotí podmienka triggeru.

Obdobným spôsobom je implementované počítadlo. Agregáčné funkcie sú vyhodnocované nasledovne:

```
#define getmin(aggr_min, ain_min, ain)      if (aggr_min == ENABLE) \
ain_min = (ain_min > ain) ? ain : ain_min;

#define getmax(aggr_max, ain_max, ain)      if (aggr_max == ENABLE) \
ain_max = (ain_max < ain) ? ain : ain_max;

#define getavg(aggr_avg, ain_avg, ain)      if (aggr_avg == ENABLE) \
ain_avg += ain;
```

5.6 UART rozhranie a Bluetooth

Univerzálna sériová linka sa od SPI odlišuje v neexistujúcej synchronizácii a rozdielnom spôsobe výberu podruženého zariadenia na komunikáciu. V tomto prípade nevyužívame chip select linky, ale 9-bitový prenos, pri ktorom jeden z bitov odlišuje adresu od údajov. Všetky zariadenia počúvajú na prenášané adresy a práve jedno je následne vybrané a počúva aj prichádzajúce údaje.

Ako druhé komunikačné rozhranie sme sa rozhodli použiť bluetooth. Dôvodom je finančná nenáročnosť a dostupnosť množstva modulov na trhu. Od jednoduchých s dosahom rádovo v metroch až po priemyselné, ktoré so smerovými anténami komunikujú na jednotky kilometrov. Pridanou hodnotou je zabezpečenie rozhrania, ktorému sa venujeme v podkapitole nižšie.

```
GPIO_InitTypeDef gpio;
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE); //hodiny na pin
RCC_APB1PeriphClockCmd(RCC_APB1ENR_USART3EN); //hodiny na UART
gpio.GPIO_Pin = GPIO_Pin_10;
gpio.GPIO_Speed = GPIO_Speed_50MHz; //rychlost portu
gpio.GPIO_Mode = GPIO_Mode_AF_PP; //alternativna fun - TX
GPIO_Init(GPIOB, &gpio);

USART_InitTypeDef USART;
USART_StructInit(&USART);
USART_USART_BaudRate = 9600; //9600 baud
USART_USART_Mode = USART_Mode_Tx; //iba odosielanie
USART_USART_Parity = USART_Parity_No; //bez paritneho bitu
USART_USART_StopBits = USART_StopBits_1; //1 stop bit
USART_USART_WordLength = USART_WordLength_8b; //8b, bez adresy
USART_USART_HardwareFlowControl = USART_HardwareFlowControl_None;
USART_Cmd(USART3, ENABLE); //bez riadenia
```

6 Overenie riešenia

V tejto kapitole predstavujeme rôzne spôsoby overenia implementácie. Základom je teoretické overenie špecifikácie v laboratórnom prostredí, kedy na vstupno-výstupné piny nastavujeme hodnoty podľa spustených procesov alebo funkcií. Následne tieto hodnoty odčítujeme na osciloskope v reálnej rýchlosti toku programu a identifikuje dobu ich vykonávania. Podľa teorémov [6] takto vieme dokázať, že systém splní časové obmedzenia v rôznych počiatočných kombináciách na vstupe a vnútorných stavoch systému.

Ďalšie testy spočívajú v overení prevádzky pri reálnych experimentoch. Pred testom špecifikujeme požiadavky na výstupy testu z pohľadu používateľa. Testy považujeme za vhodné aj na vyhodnotenie nefunkcionálnych požiadaviek, nakoľko budú vykonané v cieľových doménach použitia zariadenia.

6.1 Overenie v laboratórnom prostredí

Na overenie hlavne funkcionálnych požiadaviek, špeciálne časových obmedzení RT systému, používame testovacie prostredie opisované Chrisom Tacke a Lawrencom Ricci v [15], ktorého zovšeobecnenie sme publikovali v článku na konferencii IIT.SRC 2015 [17] [Príloha C]. Hlavná výhoda spočíva v testovaní zariadenia v reálnej, prevádzkovej frekvencii, reálnymi perifériami, vstupnými údajmi a minimálnou pridanou réžiou.

V článku sa zaoberáme RT vlastnosťami rôznych RT operačných systémov. Vyhodnocujeme s metrikou RheaStone index kľúčové parametre, akými sú napr. čas preplánovania, čas preempcie a reakčný čas na vybavenie aperiodického stimulu, ktorý je spracovávaný vysoko-prioritnou úlohou. V doméne DAQ systému sa zameriavame na overenie plánovacích teorémov [6]. Ako premenné tak vystupujú doby vykonávania úloh, ich periodicita a špecifikované časové ohraničenia. Neznámymi sú v takomto prípade doby vykonávania, ktoré získavame experimentálne. Každá úloha nastavuje kombináciu ladiacich výstupov, ktoré sú zaznamenávané digitálnym osciloskopom. Nastavením zotrvačnosti displeja vieme identifikovať aj extrémny v dobe trvania a celkové rozloženie trvaní vykonávania v čase. Napriek vysokej užitočnosti výstupov sa k meraniu využíva len základné laboratórne vybavenie a výsledok je ovplyvnený minimálne. Pridaná réžia spočíva v nastavovaní výstupných pinov a je veľmi jednoducho a presne odhadnuteľná, čo umožní jej zohľadnenie vo výstupoch merania.

6.1.1 Experimentálne získané hodnoty

Ako sme opísali v kapitole vyššie, pri meraní nás zaujíma primárne doba vykonávania úloh. Úlohy sú v kóde obalované nasledovne:

```
#define DEBUG_PIN_TSK1 GPIO_Pin_0
#define DEBUG_PORT GPIOB
#define DEBUG 1

#if defined DEBUG 1
GPIO_WriteBit(DEBUG_PORT, DEBUG_PIN_TSK1, Bit_SET);
#endif
//task body
#if defined DEBUG 1
GPIO_WriteBit(DEBUG_PORT, DEBUG_PIN_TSK1, Bit_RESET);
#endif
```

V prípade ak je definovaná konštanta preprocesora DEBUG, na porte GPIOB sa budú objavovať nábežné hrany úloh, ktoré sa začínajú vykonávať a dobežné hrany úloh, ktoré boli ukončené.

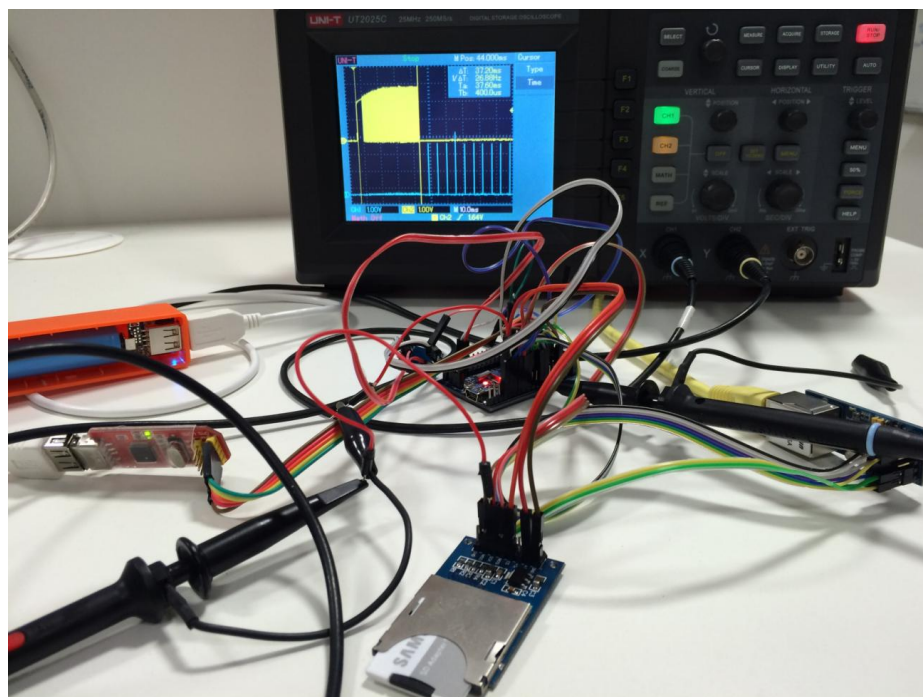
Na osciloskope nastavíme zotrvačnosť displeja a trigger na nábežnú hranu. Vo výsledku sa začiatok úlohy stabilizuje v jednom bode (trigger) a dobežná hrana, ktorá vyjadruje ukončenie úlohy bude rozložená v čase podľa ukončenia jednotlivých volaní funkcie.

V tabuľke 3 sú uvedené experimentálne zaznamenané doby trvania jednotlivých úloh. Úlohy označené hviezdičkou sa vykonávajú len raz pri inicializácii zariadenia. Úloha označená dvomi hviezdičkami je súčasťou úlohy práce so zásobníkom z dôvodu rozloženia odosielaných údajov v čase po pomerne pomalej linke.

Tabuľka 3.: Experimentálne namerané doby trvania úloh

Úloha	C _{min}	C _{max}	C _{rozptyl}	T
Zber údajov	28,6 μs	28,6 μs	0	520 μs
Trigger	2,8 μs	2,8 us	0	520 μs
Agregačné funkcie	3,1 μs	39,2 μs	36,1 μs	520 μs
Počítadlo impulzov	2,6 μs	2,6 μs	0	520 μs
Ukladanie na kartu	5,8 ms	5,8 ms	0	11,84 ms
Práca so zásobníkom	7,4 us	50 us	42,6 μs	520 μs
Inicializácia karty *	112	112	0	0
Inicializácia systému *	153 ms	153 ms	0	0

Odosielanie cez UART **	0 μ s	0 μ s	0 μ s	11,1 ms
Monitorovanie systému	1,2 μ s	1,8 μ s	0,6 μ s	1 s



Obr. 22.: Meranie doby trvania úlohy a rýchlosť inicializácie pamäťovej karty.

Na obrázku 22 je na osciloskope viditeľná inicializácia SD karty a začiatok periodického zberu údajov.

6.1.2 Teorém o hranici využitia CPU

Teória plánovateľnosti sa vo všeobecnosti zaoberá analýzou množiny závislých alebo nezávislých procesov so známymi časmi vykonávania (v našom prípade získane experimentálne). Teória ďalej predpokladá preemptívne plánovanie na základe priorít procesov. V takýchto podmienkach platí prvý teorém o hranici využitia procesorového času: Množina n nezávislých, periodických úloh, ktoré sú plánované rovnomerne a s rovnakou prioritou, je naplánovateľná, pri akejkoľvek kombinácii počiatkových parametrov, ak platí nasledovné:

$$\sum_{i=1}^n \frac{C_i}{T_i} \leq n \left(2^{\frac{1}{n}} - 1 \right) = U(n) \quad [1]$$

Pričom C_i je doba vykonávania i -tej úlohy, T_i je perióda, v ktorej sa i -tá úloha pripravená na vykonávanie. Rovnica teorému o hranici využitia CPU predpokladá periodické úlohy, v prípade existencie aperiodických úloh v systéme uvažujeme vždy o

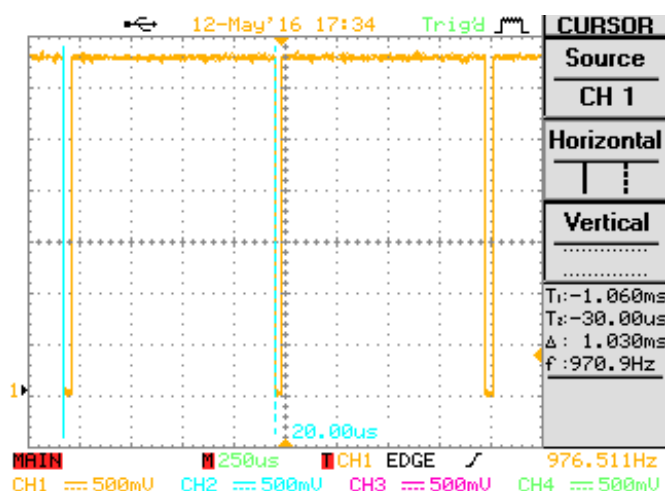
najhoršom možnom scenári, teda úlohy sú pripravené vždy pri dosiahnutí ich minimálnej periódy. Podobne pri rozptyle vykonávania úlohy uvažujeme vždy o najdlhšom možnom, teda C_{\max} .

Je zrejmé, že pri jedinej úlohe v systéme je maximálna užitočná využiteľnosť strojového času 100%. S rastúcim počtom úloh klesá maximálne využiteľné vytťaženie CPU k 69%.

Do výrazu 1 dosadíme namerané hodnoty periodických a aperiodických z tabuľky 3 nasledovne:

$$\frac{28,6}{500} + \frac{2,8}{500} + \frac{39,2}{500} + \frac{2,6}{500} + \frac{5800}{11100} + \frac{50}{500} + \frac{1,8}{10^6} \leq 7 \left(2^{\frac{1}{7}} - 1 \right)$$

Vo výsledku vidíme, že využiteľnosť CPU pri najhoršom možnom scenári je 72,68 %, čo je menej ako $7 \cdot (2^{\frac{1}{7}} - 1) = 72,8$ %, úlohy sú tak naplánovateľné pri akomkoľvek počiatocnom stave a kombinácii podmienok. Meranie parametrov prebiehalo pri frekvencii zberu údajov 1920 Hz, ktorá je tak maximálna možná rýchlosť zariadenia. Pripomíname, že hodiny procesora pracujú pri tretinovej rýchlosti 24 Mhz z dôvodu potreby nízkeho odberu hlavného experimentu vypustenia stratosférického balóna.



Obr. 23.:Priebeh naplánovania úlohy zberu údajov.

Tento teorém sa nezaobera možnými časovými ohraneničiami. Vyjadruje len schopnosť systému úlohy naplánovať. V doméne zberu údajov je kritický hlavne rozptyl začiatku vykonávania záznamu (údaje musia byť zachytené v pravidelných intervaloch). Tento parameter vieme odmerať obdobne ako v kapitole 6.1.1, s rozdielom, ktorý spočíva v neobalení celej funkcie zmenami logickej úrovne na ladiacom výstupe, ale v generovaní čo najkratšieho impulzu od momentu vygenerovania prerušenia. Následne nastavíme na osciloskope zotrvačnosť displeja a nastavíme správne trigger osciloskopu

na nábežnú hranu. Výsledkom je rozloženie spustení zberu údajov v čase, ktorého priebeh je na obrázku 23.

6.2 Overenie v cieľovej doméne - kvalita spánku

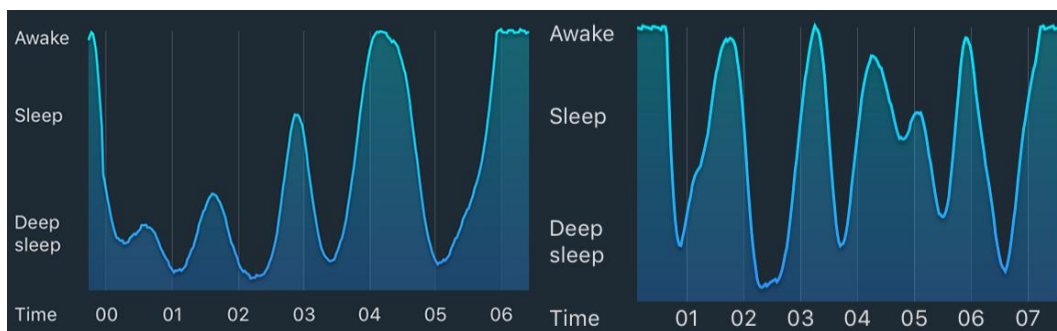
V tomto experimente použijeme implementovaný DAQ systém na získanie informácií o kvalite spánku. Cieľom je namerať dĺžky jednotlivých spacích fáz a ich priebeh. Podobným spôsobom fungujú mobilné aplikácie, ktoré monitorujú spánok používateľa a odhadujú ideálny čas na prebudenie, tj. moment, kedy je človek v najplytšom spánku.

Tabuľka 4.: Testované aspekty systému, očakávané výstupy a nastavenie zberu údajov

Testované aspekty systému	kvalita zberu analógových údajov; funkčnosť a praktická použiteľnosť analógového triggeru
Očakávané výstupy	csv súbor, spracovateľný napr. v Exceli
Dĺžka testu	9 hodín
Nastavenie DAQ	3x analógový vstup frekvencia zberu 1kHz analógový trigger nastavený na prekročenie úrovne 1,7V (reprezentuje zrýchlenie $0,45\text{ms}^{-2}$)
Napájanie	akumulátor bez dobíjania počas experimentu

Spánok človeka začína ľahkým spánok, prechádza do hlbokého a následne do fázy snívania, označovanej aj ako REM (rapid eyes movement). Na obrázku 24 vidíme vizualizáciu programu Sleep Cycle (<http://www.sleepcycle.com/>), ktorá využíva mikrofón mobilného telefónu na identifikovanie aktivity tela a teda aj konkrétnej fázy. Na ľavej strane obrázku je graf zdravého spánku, fázy pravidelne a plynulo prechádzajú v cykle, pričom ľahký spánok je čoraz aktívnejší, až dôjde k prirodzenému zobudeniu sa. Na pravej časti obrázku je graf nepokojného spánku. Fázy sú nepravidelné a ľahký spánok je veľmi nepokojný, čo sa prejavilo aj v čase cca 3:15, kedy došlo ku krátkemu prebudeniu.

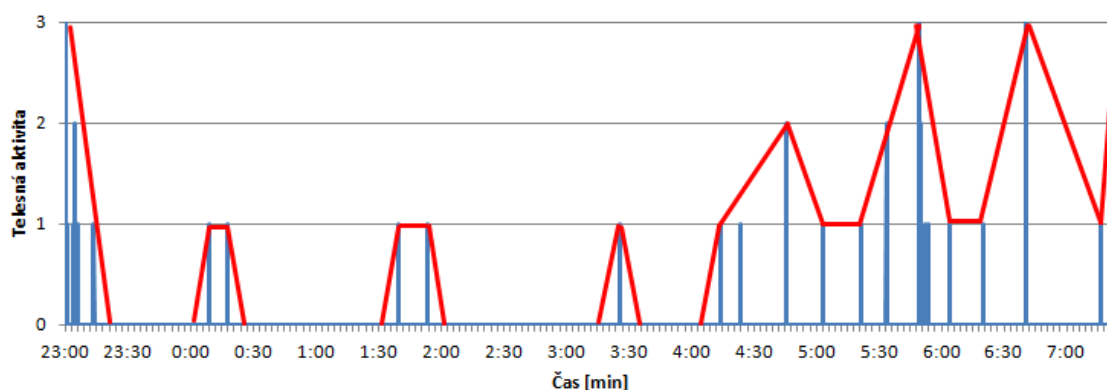
Na snímanie aktivity tela využijeme senzor zrýchlenia, ktorý bude položený s DAQ systémom na posteli (pohyb tela sa bude prenášať na zariadenie a k nemu pripojený akcelerometer).



Obr. 24.: Pravidelné a nepravidelné striedanie spánkových fáz.

Rozhodli sme sa pre konkrétny senzor s čipom MMA7361L, ktorý má analógový výstup a nastaviteľnú citlivosť v rozsahu $\pm 1,5$ alebo $\pm 6G$. Spotreba obvodu je na úrovni $400\mu A$ a umožňuje režim spánku s odberom $3\mu A$. Odozva akcelerometra je na úrovni 400 Hz na osi x a y, v prípade vertikálnej osi je odozva 300 Hz. Nakoľko má senzor analógový výstup, je veľmi zásadná kvalita zberu údajov, hlavne precíznosť AD prevodníka, ktorý je v našom systéme 12 bitov, z čoho vyplýva 4096 kvantovacích úrovní a rozlíšenie po $7,32 \cdot 10^{-4} G$. Oba parametre, tj. frekvencia a rozlíšenie, považujeme v tomto experimente za dostatočné. Na prvý pohľad sa zdá využitie mikrofónu u mobilných aplikácií na telefónoch vhodnejšie a citlivejšie, ale akcelerometer nie je toľko šumom prostredia (v tomto prípade ruch okolia).

Detekcia spánkových fáz



Obr. 25.: Zaznamenaná aktivita tela počas spánku.

Na obrázku 25 je znázornená nameraná aktivita tela počas spánku. Namerané hodnoty zo senzoru zrýchlenia sme očistili o šum a jednotlivé hodnoty zaradili do 3 intervalov. Na vodorovnej osi je priebeh času a na zvislej závislá hodnota aktivity tela. Vyšší interval značí vyššiu aktivitu tela. Z grafu sme odstránili obdobia pred zaspátním a po zobudení. Červenou čiarou sme prepojili jednotlivé nenulové intervaly tak, aby bol graf lepšie čitateľný. Je zrejmé, že k zaspatiu došlo v čase približne 23:20, kedy aktivita tela významne poklesla. Nasleduje fáza hlbokého spánku bez pohybu a v čase po

polnoci aktivita opäť narastá vo vrchole okolo 0:15, kedy je spánok najľahší. Celkovo sa takto zopakujú prvé tri fázy s priemernou periódou 90 minút. V čase o 3:30 už začína aktivita tela narastať a aj najhlbší spánok je pomerne rušný. V čase 7:30 nastáva prebudenie. Výsledok merania považujeme za správny, nakoľko sa ponáša na priebeh z obrázku 23. Fázy sa prechádzajú pravidelne s postupne zvyšujúcou sa aktivitou. Na druhej strane hodnotíme kvalitu snímania akcelerometrom za problematickú, nakoľko po odstránení šumu je aktivita pozorovateľná veľmi nahrubo. Na zrýchlení sa síce neprejavuje šum okolia, ale samotný akcelerometer dáva pomerne nejasný výstup obsahujúci vlastné rušenie.

6.3 Overenie v cieľovej doméne - radiačné pozadie

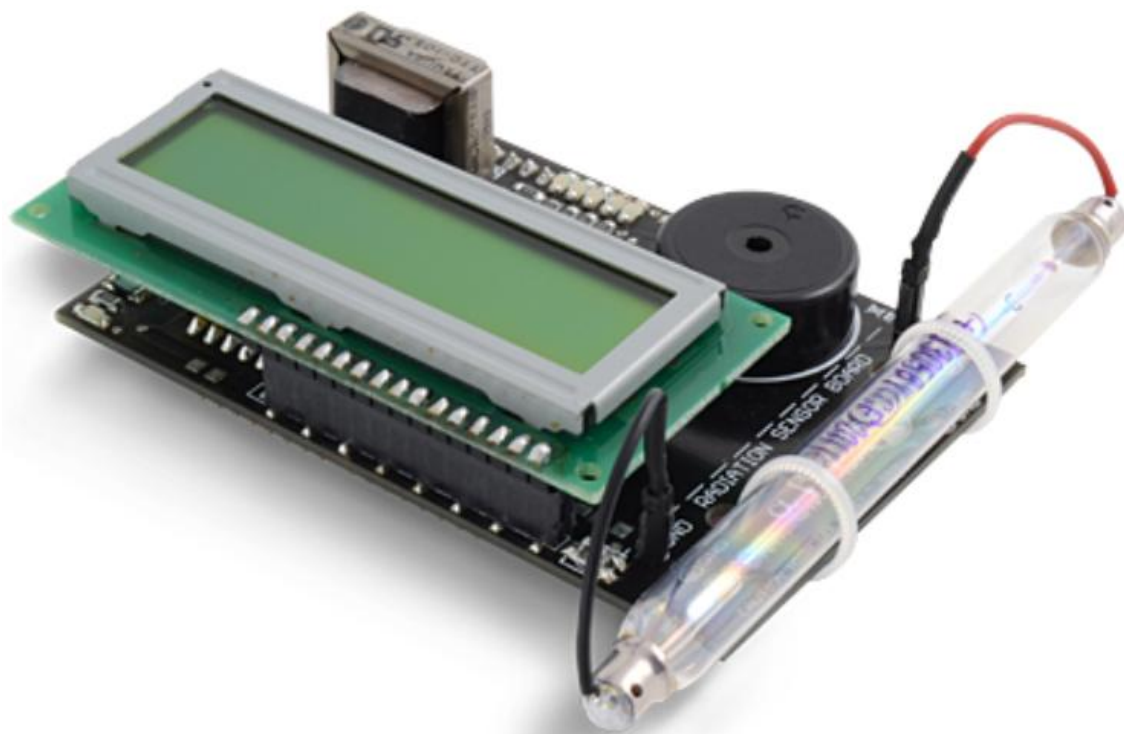
V tomto experimente realizujeme prípad použitia zberu údajov v nízkej frekvencii, bežne využívanom v prípade meteorologických staníc. Táto diplomová práca bola vypracovaná na Ústave počítačových systémov a sietí FIIT STU, ktorý sídli v Bratislave. Bratislava sa rozprestiera na úpätí Malých Karpát, ktorých zloženie spôsobuje výraznejší výskyt radónu a jeho produktov v oblasti, preto považujeme meranie rádioaktivity v Bratislave za vhodný a zaujímavý príklad použitia na otestovanie nášho systému.

Tabuľka 5.: Testované aspekty systému, očakávané výstupy a nastavenie zberu údajov

Testované aspekty	kvalita zberu digitálnych údajov kvalita a použiteľnosť digitálneho triggeru, konkrétne časti na záznam početnosti impulzov
Očakávané výstupy	csv súbor, spracovateľný napr. v Exceli
Dĺžka testu	4 hodiny
Nastavenie DAQ	1x analógový vstup s vyhodnocovaním početností zostupných hrán (radiácia) frekvencia zberu 2kHz (zachytenie krátkotrvajúcich impulzov) analógový trigger nastavený na záznam početnosti impulzov v perióde 10s
Napájanie	akumulátor s dobíjaním riadeným DAQ systémom

Na meranie úrovne radiačného pozadia v experimente používame dosku s geiger-mullerovým počítačom, ktorý bol navrhnutý na jednoduché použitie s Arduinom alebo počítačom Raspberry Pi (Obr. 26).

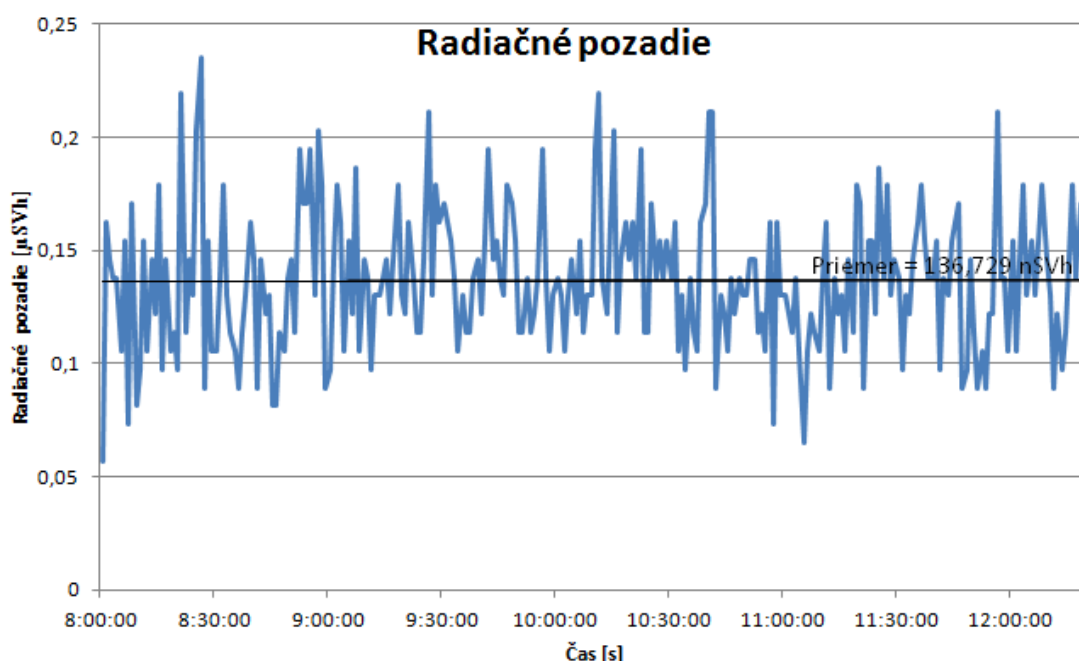
Použitý Geiger-Mullerov počítač vznikol z iniciatívy po nehode vo Fukušime v marci 2011. Tvorcovia chceli poskytnúť jednoduché a cenovo dostupné zariadenie pre obyvateľov, ktorí si tak mohli svojpomocne merať lokálnu úroveň radiačného pozadia v dotknutej oblasti (prvé vyrobené kusy boli odoslané bezplatne na Tokyo Hackerspace a do rôznych iných skupín). Zariadenie má kompatibilný výstup s veľmi obľúbenými mikropočítačmi, ktorými si môžu zdatnejší používatelia výstupy zo zariadenia rôzne spracovávať.



Obr. 26.: Radiačný senzor s trubicou J305β

Doska sa skladá z dvoch hlavných častí, napájacej a signálovej. Úlohou napájacej časti je hlavne vytvoriť potrebné vysoké napätie na trubicu. Ak má trubica privedené pracovné napätie, je možné signál z nej zosilniť a filtrovať v signálovej časti a následne spracovávať buď v mikropočítači, ktorý je súčasťou dosky alebo iným zariadením, ktoré dokáže spracovať danú výstupnú logiku. Spravidla sa pri spracovaní zisťuje početnosť impulzov v 10 sekundách, táto hodnota je prenasobená šiestimi, čím získavame počet impulzov za minútu (cpm - clicks per minute). Trubice používané v zariadení majú v katalógovom liste uvedenú konštantu, ktorou prenasobením získame už výslednú hodnotu v jednotke $\mu\text{SV/h}$. Z univerzálnej povahy nami navrhnutého

systemu budeme ukladať údaje z prvého kroku a teda početnosť impulzov za 10 sekúnd. Následné výpočty robí používateľ alebo skript z výsledného súboru csv alebo z databázy.



Obr. 27.: Priebeh nameraného radiačného pozadia.

Prvé meranie neprinieslo žiadne výsledky. Systém nedetegoval častice napriek tomu, že ich prítomnosť bola akusticky hlásená detektorom. Dôvodom bol nekompatibilný výstup, ktorý z hodnoty 5 V klesal len na úroveň približne 1,1 V. Použitý mikropočítač pracuje s 3,3 V logikou (niektoré piny tolerujú 5V logiku) a logická nula na vstupe je zaznamenaná až pri prekročení prahu 0,8 V. Riešením bolo pripojenie na analógový vstup cez odporový delič a nastavenie prahovej úrovne, pri ktorej analógový trigger rozpoznáva hranu impulzov. Výsledok je vyobrazený na obrázku 27 v podobe grafu priebehu zmeny radiačného pozadia počas 4 hodín. Hodnoty z csv súboru sme nakoniec z dôvodu výraznej nerovnomernosti rozloženia detegovaných častíc v čase zoskupili do šestic. Šestice sme už iba prenásobili koeficientom 0,00812037, ktorý sa nachádza v katalógovom liste k použitej trubici J305β. Priemerná úroveň radiačného pozadia počas testu bola 136, 729 nSVh.

6.4 Overenie v cieľovej doméne - stratosférický balón

Hlavným experimentom, ktorým prakticky testujeme navrhnutý systém je vypustenie stratosférického balóna. Tento experiment je rozhodne najnáročnejším, okrem samotných nárokov na spoľahlivosť systému je potrebné v maximálnej miere

zabezpečiť filtrovanie nepotrebných údajov, maximalizovať tak ich relevantnosť a v konečnom dôsledku uľahčiť ich spracovávanie. Vhodným praktickým testom je zber údajov v stratosférickom balóne. Zaznamenávame údaje z rôznych senzorov, každý z nich má inú odozvu v ktorej je zaujímavé čítať údaje, resp. rôzne veličiny nás zaujímajú v rôznych podmienkach a rôznych frekvenciách.

Tabuľka 6.: Testované aspekty systému, očakávané výstupy a nastavenie zberu údajov

Testované aspekty	kvalita zberu analógových údajov schopnosť pracovať bez prístupu operátora
Očakávané výstupy	csv súbor, spracovateľný napr. v Exceli
Dĺžka testu	4 hodiny
Nastavenie DAQ	4x analógový vstup (termistor, UV senzor, os Z akcelerometra, detektor ozónu) frekvencia zberu 1kHz rôzna frekvencia záznamu z rôznych vstupov
Napájanie	akumulátor bez dobíjania počas experimentu

V gondole stratosférického balóna budeme mať umiestnené senzory na meranie teploty, UV žiarenia, zrýchlenia a náklonu v čase. Na meranie UV žiarenia použijeme senzor s detektorom GUVA-S12D. Na meranie zrýchlenia použijeme senzor z testu kvality spánku (kapitola 6.2). Vyhodnocovanie údajov akcelerometra bude v tomto prípade výrazne zložitejšie, nakoľko nám nestačí sledovať relatívne zmeny v zrýchlení (detegovanie pohybu a pokoja). Vyhodnocovať budeme absolútne hodnoty zrýchlenia zaznamenávané so známou periódou. Pokiaľ poznáme počiatočné podmienky (nulová prejdená dráha, nulová rýchlosť na zemi pred štartom), vieme veľmi jednoducho vypočítať priebeh rýchlosti a zmeny výšky a maximálnu dosiahnutú výšku balónu. Využívame vzťahy medzi zrýchlením, rýchlosťou (prvá derivácia rýchlosti podľa času je zrýchlenie) a dráhou (druhá derivácia dráhy podľa času je zrýchlenie).

Podľa National Oceanic and Atmospheric Administration by minimálne teploty v troposfére a stratosfére (vrstvy atmosféry s dostatočnou hustotou vzduchu na udržanie meteorologického balónu) nemali klesnúť pod $-51\text{ }^{\circ}\text{C}$. Bežne používané senzory na meranie teploty s digitálnym výstupom tak neposkytujú dostatočný rozsah a preto sme sa rozhodli použiť termistor, konkrétne KTY-122 s rozsahom od -55 do $+150\text{ }^{\circ}\text{C}$.

Minimálna perióda merania pri ktorej je zaujímavé sledovať meniacu sa veličinu je závislá od teplotnej zotrvačnosti materiálu termistoru a ďalších parametrov. V prípade KTY-122 je katalógová hodnota τ vo vzduchu rovná 30 sekundám, čo znamená, že na dosiahnutie 63,2% teplotného gradientu je potrebných práve 30 sekúnd. Ide o veľmi peknú ukážku nezmyselnosti záznamu výstupov vo vyšších frekvenciách z niektorých senzorov a nutnosti použiť podmieňovací podsystem.

Detektor ozónu bol zapožičaný zo Slovenskej organizácie pre vesmírne aktivity a ku dňu odovzdania diplomového projektu nemáme jeho špecifikáciu. Senzor poskytuje výstup údajov analógovo, takže nepredpokladáme žiadne komplikácie.

Vypustenie stratosférického balóna je finančne a technologicky pomerne náročný projekt. Projekt je financovaný nadáciou Tatrabanky a samotný proces vypustenia je realizovaný Slovenskou organizáciou pre vesmírne aktivity - SOSA, ktorá poskytne potrebné skúsenosti zo série predtým realizovaných letov. V týchto mesiacoch SOSA vypúšťa prvú slovenskú družicu a od termínu jej vypustenia sa odvíja aj termín vypustenia nášho balóna. V čase posledných úprav tejto práce je zrejmé, že vypustenie bude najskôr koncom mája, a tak nie je možné výsledky testu uviesť na nasledovných stranách (veríme, že výsledky budeme prezentovať počas obhajoby diplomovej práce). Rozhodli sme sa teda o doplnenie zjednodušeného testu (kapitola nižšie) pri ktorom overíme vyhodnocovanie údajov z akcelerometra.

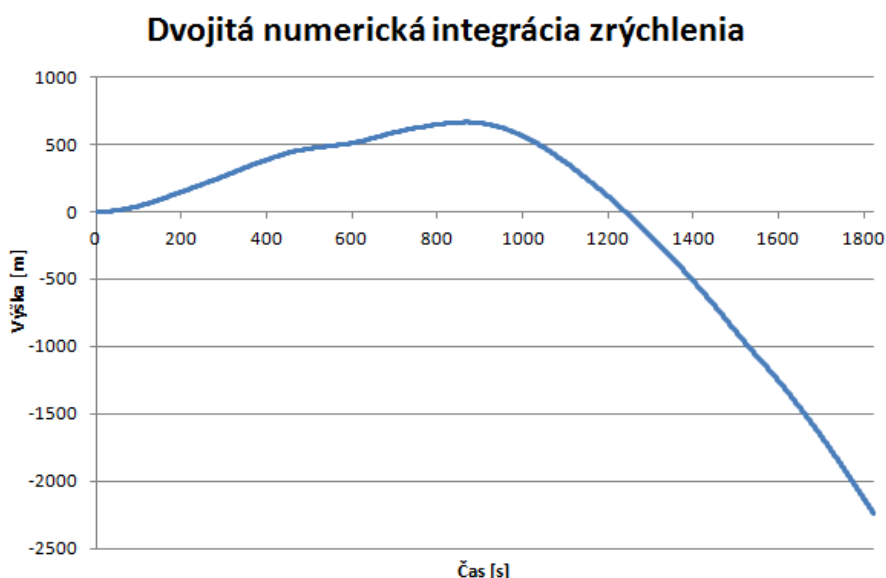
6.5 Overenie v cieľovej doméne - let lietadlom

Posledný test pred samotným vypustením balóna overil kvalitu zbieraných údajov z akcelerometra a možnosť ich spracovania. Zariadenie bolo umiestnené na palube lietadla, ktoré vynášalo parašutistov do výšky 3 kilometrov.

Tabuľka 7.: Parametre modifikovaného testu

Testované aspekty	zber údajov z akcelerometra
Očakávané výstupy	csv súbor, z ktorého sa získa priebeh zmeny výšky
Dĺžka testu	30 minút
Nastavenie DAQ	1x analógový vstup (os Z akcelerometra) frekvencia zberu 1kHz
Napájanie	akumulátor bez dobíjania počas experimentu

Lietadlo štartovalo z letiska v Senici. Vzhľadom na minimálnu oblačnosť a veternosť, veľmi rovnomerne vystúpalo do výšky 3 km a následne rovnomerne klesalo na rovnaké letisko. Celková dĺžka letu bola 30 minút. Na obrázku 28 je graf zmeny nameranej výšky. Je zrejmé, že meranie neprebehlo korektne, pretože konečná výška je približne 2200 metrov pod úrovňou počiatočnej. To bolo spôsobené naklonením lietadla pri stúpaní, kedy os Z akcelerometra bola odklonená od zvislice lietadla. Nameraná hodnota vertikálneho zrýchlenia tak bola menšia ako skutočná. Pri klesaní podlaha lietadla bola takmer v rovine, čo sa prejavilo na pomerne presnej hodnote celkovej zostupnej výšky.



Obr. 28.: Zrýchlenie po dvojitej integrácii - zmena výšky.

Chyba nastala práve v nezohľadnení korekcie podľa náklonu osi Z akcelerometra. Podobný problém by nastal aj pri lete stratosférického balóna, kedy je gondola vystavovaná nárazom vetra. Riešenie spočíva v pripojení ďalšieho senzoru, gyroskopu, ktorý vracia hodnotu zmeny náklonu za časovú jednotku. Výstup z gyroskopu sa numericky integruje v oboch horizontálnych osiach, čím získame priebeh zmeny náklonov na osi X a Y a z nich výslednicu, ktorá je náklonom gondoly. Výslednica je teda odklon osi Z akcelerometra od zvislice a aj korekciou hodnoty zvislej zložky zrýchlenia. Zrýchlenie je po zohľadnení korekcie dvakrát integrované ako v experimente letu lietadlom.

Záver

V práci analyzujeme existujúce riešenia systémov na zber údajov naprieč doménou. Pokrývame oblasť od jednoduchých dedikovaných zariadení, cez všeobecné systémy spracovávajúce analógové i digitálne signály až po jeden z najkomplexnejších systémov, ktorý bol vytvorený človekom - DAQ systém detektora ATLAS. Naším cieľom bolo vytvorenie univerzálneho systému, ktorý by bol vhodným nástrojom na základné merania v laboratórnom i externom prostredí. Snažili sme sa o vytvorenie zariadenia, ktoré by bolo cenovo dostupné a plne autonómne, čo by podporilo experimenty vykonávané paralelne.

Systém je riadený mikropočítačom s jadrom Cortex-M3, konkrétne STM32F103C8T8, ktorý v cene rádovo jednotiek eur poskytuje dostatočný výkon na implementáciu relatívne komplexných vyhodnocovacích podmienok. S výhodou sme využili integrovaný AD prevodník a ďalšie periférie, čím sa docielilo maximálne zjednodušenie výsledného zapojenia.

Výstupom tejto diplomovej práce je fungujúci prototyp systému na zber údajov. Zariadenie dokáže pripojiť 4 senzory s analógovým a 4 senzory s digitálnym výstupom. Zbierané údaje sú vyhodnocované v podmieňovacom podsystéme, ktorý je nastaviteľný používateľom, a len vybrané relevantné výsledky sú ukladané na Secure Digital kartu, prípadne preposielané cez ethernet alebo bluetooth. Údaje sú rovnako aj agregované s výstupom miním, maxím, priemerov, prípadne početností. Táto konfigurácia umožnila pripojiť široké spektrum senzorov, s ktorými sme systém aj prakticky testovali a výsledky uvádzame v tejto práci. Údaje odosielané prostredníctvom komunikačných rozhraní je možné šifrovať algoritmom. Popri praktických testoch sme zariadenie testovali laboratórne metódou, ktorú sme odvodili od metódy z práce Chrisa Tackeho a Lawrenca Ricciho a prezentovali na študentskej konferencii IIT.SRC v roku 2015 (príloha C).

Naše úsilie počas práce na diplomovom projekte vyvrcholí koncom mája vypustením stratosférického balóna, ktorý počas trojhodinového letu dosiahne výšku 35 km a bude zaznamenávať vertikálnu zložku zrýchlenia, zmeny náklonu v čase oboch vertikálnych osí, teploty vnútri a mimo gondoly, okolité UV žiarenie a množstvo ozónu v okolí. Výsledky experimentu budú prezentované na obhajobe diplomového projektu v júni.

Vyhotovený prototyp spĺňa všetky náležitosti vyplývajúce zo zadania diplomovej práce.

Ďalšia práca na projekte sa môže týkať vytvorenia portu na mikropočítači vyššej triedy v rámci rodiny STM32. Pri implementácii tejto práce sme narazili na výkonnostné limity pripojenia SD karty rozhraním SPI bez SDIO, výrazné limitácie vyplývajúce z malej kapacity SRAM a nakoniec aj programovej pamäte flash. Druhá oblasť rozšírenia projektu spočíva v nadviazaní spolupráce so študentmi prírodných vied, ktorí by mohli zariadenie využiť na vlastné experimenty a výskum. Širšie portfólio scenárov by prinieslo rozšírenie požiadaviek na systém a jeho implementáciu.

Slovník termínov a skratiek

ADC - prevodník analógových hodnôt na digitálne [13].

ASIC - zákaznícky čip [9].

Data logger – zariadenie, ktoré spravidla dlhodobo zaznamenáva údaje v nízkych frekvenciách. Príkladom je meteostanica, ktorá robí záznamy rádovo v minútach, hodinách [19].

DAQ – zber údajov [19].

FPGA - programovateľné hradlové pole [7].

Mikropočítač - počítač integrovaný do jedného púzdra [6].

Mikroprocesor - procesor integrovaný do jedného púzdra [6].

RT – reálny čas, fyzikálne vyjadrenie času [13][22].

RT systém – systém reálneho času, SRČ [7].

RT operačný systém – je operačný systém, ktorý je špeciálne navrhovaný pre RT zariadenia. Jeho hlavnými vlastnosťami sú predpovedateľnosť časových charakteristík a determinizmus [13].

RTSAD – štruktúrovaná analýza a návrh v systémoch reálneho času[6].

Systém pre zber údajov – DAQ systém – systém, ktorý slúži na zber údajov, reprezentovaných fyzikálnymi veličinami ako napätie, prúd, teplota, tlak apod. prostredníctvom počítača [19].

SD karta - Secure Digital karta.

Trigger – je podmieňovací systém, ktorý čaká na splnenie jednoduchých kritérií, kedy iniciuje zber údajov. Takto sa zníži množstvo zachytených dát a zvýši sa ich relevantnosť [8].

Zoznam použitej literatúry

- [1] Airapetian, A. et al.: ATLAS Detector and Physics Performance Technical Design Report. 1999. CERN-LHCC-99-15, ATLAS-TDR-15
- [2] ATmega32 datasheet. [Online] <http://www.atmel.com/images/doc2503.pdf>
- [3] Berger, A. S.: Embedded Systems Design: An Introduction to Processes, Tools, and Techniques. CMP Books, 2002. 237 s. ISBN 1578200733
- [4] Brown, G.: Discovering the STM32 Microcontroller. Indiana University, 2016. 244 s.
- [5] Buttazzo, G. C.: Hard Real-Time Computing Systems. Springer Science, 2011. 521 s. ISBN 978-1-4614-0675-4
- [6] Gomaa, H.: Software Design Methods for Concurrent and Real-Time Systems. Addison-Wesley publishing company, 1993. 447 s. ISBN 0-201-52577-1
- [7] Krajčovič, T.: Základné pojmy v angličtine a ich preklad do slovenčiny. 1995. Osobná komunikácia.
- [8] Lindenstruth, V, Kisel, I.: Overview of trigger systems. Nucl. Instr. and Meth. A 535 (2004) 48-56. doi:10.1016/j.nima.2004.07.267, <http://www.sciencedirect.com/science/article/pii/S0168900204015748>
- [9] Moore, A.: FPGAs for Dummies. 2014. JohnWiley & Sons. 52 s. ISBN 978-1-1118-94220-8
- [10] Radon Risk in Urban Environment. [online] <http://www.sazp.sk/slovak/periodika/sprava/psr/urban/state/radon/radon.html>
- [11] Secure socket layer (SSL) for STM32F217xx microcontroller. Application Note. [online] http://www2.st.com/content/ccc/resource/technical/document/application_note/b1/ab/e3/5b/80/8e/4d/4f/DM00024859.pdf/files/DM00024859.pdf/jcr:content/translations/en.DM00024859.pdf
- [12] Staines, A. S.: A Comparison of Software Analysis and Design Methods for Real Time Systems. International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:1, No:3, 2007 523-527. ISSN: 1307-6892
- [13] STM32F103x8 data sheet. [online] <http://www2.st.com/content/ccc/resource/technical/document/datasheet/33/d4/6f/1d/df/0b/4c/6d/CD00161566.pdf/files/CD00161566.pdf/jcr:content/translations/en.CD00161566.pdf>
- [14] Strnadel, J.: Real-time operační systémy, FIT VUT v Brně, Brno, 2006, 120 s.

- [15] Tacke, Ch., Ricci, L.: Benchmarking Real-time Determinism in Microsoft Windows CE [online] , <http://msdn.microsoft.com/en-us/library/ms836535.aspx>
- [16] Tichý M.: Elektronika, skripta. <http://physics.mff.cuni.cz/kfpp/skripta/elektronika/>
- [17] Valiček, M.: Real-Time Performance Comparison of FreeRTOS and User-written Scheduler, In: M. Bieliková, ed., Proceedings in Informatics and Information Technologies, Student Research Conference, Vol. 2, (2015), Vydavatel'stvo STU, pp. 339 – 344
- [18] White, E.: Making Embedded Systems. O'Reilly, 2011. 311 s. ISBN 978-1-449-30214-6
- [19] White paper: Data Acquisition Systems Review. [online] http://www.engineersedge.com/instrumentation/data_acquisition_systems.htm
- [20] White paper: Fundamentals of Sampled Data Systems: Analog-Digital Conversion. [online] <http://www.analog.com/library/analogDialogue/archives/39-06/Chapter%20%20Sampled%20Data%20Systems%20F.pdf>
- [21] White paper: Which solar charge controller: PWM or MPPT?. Victron Energy 2014. [online] <http://www.victronenergy.com/upload/documents/White-paper-Which-solar-charge-controller-PWM-or-MPPT.pdf>
- [22] Williams, R.: Real-Time Systems Development. Butterworth-Heinemann, 2008. 455 s. ISBN 978-0-7506-6471-4
- [23] Xilinx Spartan-3E FPGA Family Data Sheet. [online] http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf

Prílohy

Príloha A: Používateľská príručka

Príloha B: Inštalačná príručka

Príloha C: Valiček, M.: Real-Time Performance Comparison of FreeRTOS and User-written Scheduler. In Student Research Conference 2015. Vol. 2. 11th Student Research Conference in Informatics and Information Technologies Bratislava, April 23, 2010 : Proceedings in Informatics and Information Technologies. Bratislava: Nakladateľstvo STU, 2015, s. 339-344. ISBN 978-80-227-4342-6.

Príloha D: Valiček, M.: Software Implementation of Advanced Server Watchdog. In Student Research Conference 2010. Vol. 2. 6th Student Research Conference in Informatics and Information Technologies Bratislava, April 21, 2010 : Proceedings in Informatics and Information Technologies. Bratislava: Nakladateľstvo STU, 2010, s. 396--403. ISBN 978-80-227-3267-3.

Príloha E: Obsah elektronického média

Príloha F: Elektronické médium