

České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra řídicí techniky



Diplomová práce

**SW řídicích jednotek kamerového pozičního systému
určeného pro sledování cílů**

Bc. Luboš Koudelka

Vedoucí práce: doc. Ing. Jan Roháč, Ph.D.

Studijní program: Otevřená informatika

Studijní obor: Počítačové inženýrství

5. ledna 2015

Poděkování

Chtěl bych poděkovat vedoucímu práce, doc. Ing. Janu Roháčovi, Ph.D. za trpělivost, četné rady a připomínky. Dále bych chtěl poděkovat celému kolektivu laboratoře s151 za dva roky přínosné a poučné spolupráce. Také děkuji své rodině za péči a podporu.

Prohlášení

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

V Praze, dne 5. ledna 2015

.....

Podpis

Abstrakt

Cílem této diplomové práce je návrh a vytvoření firmwaru mikrokontroléru pro realizaci a zabezpečení komunikace mezi počítačem a periferiemi pozičního systému – systému používajícího kamery pro sledování cílů a určování jejich polohy. Obsah komunikace je navržen s ohledem na potřeby systému a přenosovou kapacitu dostupných komunikačních rozhraní. V práci je navrženo a prakticky ověřeno zabezpečení komunikace CRC kódem. Realizována je také komunikace s GPS modulem a zpracování dat z GPS pro měření vzdáleností.

The aim of this diploma thesis is to design and implement a microcontroller firmware for the realization of communication between computer and positioning system peripherals. Positioning system is a system, which is using cameras for small objects tracking and finding their position. Communication content is designed with respect to system requirements and transmission capacity of available communication interfaces. In the thesis is designed and verified securing of the communication using CRC code. There is also implementation of communication with GPS receiver and processing data from GPS for distance measurement.

Obsah

1. Úvod	1
2. Teoretický rozbor	2
2.1. Kamerový systém	2
2.2. Kamerová jednotka	4
2.3. Zastaničení.....	10
2.4. Softwarové zabezpečení komunikace	11
2.5. Mikrokontrolér STM32F407	13
2.6. Komunikační rozhraní RS232, RS422 a CAN bus	14
2.7. GPS.....	17
3. Návrh komunikačního protokolu.....	19
3.1. Příkazy centrálního počítače pro ovládání kamerové jednotky	23
4. Implementace CRC	27
4.1. Test generujících polynomů CRC	27
4.2. Výpočet CRC v mikrokontroléru	29
5. Realizace komunikačního protokolu v řídicí jednotce.....	33
5.1. Realizace komunikace centrální PC <-> ŘJ.....	36
5.2. Přeposílání příkazů do polohovacího zařízení	37
5.3. Přeposílání příkazů do objektivu	38
5.4. Kontrola činnosti objektivu	38
5.5. Reset ŘJ.....	38
5.6. Synchronizace kamery a polohovacího zařízení	39
6. Spolehlivost RS422 komunikace.....	41
7. Diferenční GPS.....	45
8. Závěr	51
Reference	52
9. Přílohy.....	55
9.1. Příloha A – batch skript pro zpracování GPS dat	55

9.2.	Příloha B – skripty pro přepočítání výstupu programů RTKLIB	57
9.3.	Příloha C – program pro výpočet CRC	59
9.4.	Příloha – hlavní část programu pro testování spolehlivosti CRC polynomů	61
9.5.	Příloha E – schéma řídicí jednotky.....	63
9.6.	Příloha F – seznam příkazů centrálního PC pro ovládání objektivu	64
9.7.	Příloha H – tabulka příkazů pro ovládání řídicí jednotky.....	66
	Obsah příloženého CD.....	67

1. Úvod

S rostoucí oblibou dronů a dalších malých, dálkově ovládaných bezpilotních prostředků, které jsou velmi levné a dostupné i v hračkářství, roste obava, co vše mohou drony ve špatných rukou způsobit. Incidentsy způsobené dronem se v poslední době rychle množí a často se i objevují v médiích – malý bezpilotní stroj způsobil nepokoje při sportovních událostech, ohrožuje civilní letadla a velmi závažně narušuje bezpečnost strategických objektů, např. jaderných elektráren. Například na obr. 1 je zobrazen dron upravený americkými nadšenci, který je vybaven kamerou a puškou na paintball [1]. To ukazuje jednoduchost vytvoření nebezpečné aplikace. Radarem jsou drony kvůli malé velikosti špatně detekovatelné, je tedy nutné hledat jiné způsoby ochrany.



Obr. 1: Dron vybavený kamerou a puškou na paintball [1]

Tato práce se zabývá návrhem a realizací komunikace nižších vrstev v projektu kamerového systému pro detekování malých cílů, např. dronů. V práci bude představena celá struktura i princip činnosti kamerového systému, blíže popsány budou přípravné režimy prováděné před samotným měřením. Budou představena různá komunikační rozhraní a popsány jejich základní vlastnosti. Komunikace mezi periferiemi kamerového systému a ovládacím PC bude realizována prostřednictvím vybraných komunikačních rozhraní za použití mikrokontroléru. Pro komunikaci bude představen obsah zpráv pro ovládání periferií a sběr dat v systému, taktéž bude představeno i softwarové zabezpečení proti chybám. Podrobně bude popsána implementace řídicích a komunikačních algoritmů do mikrokontroléru. Funkčnost a spolehlivost komunikační linky bude otestována praktickými testy i laboratorním měřením. V závěru práce bude představen možný způsob měření vzdálenosti pomocí GPS přijímačů.

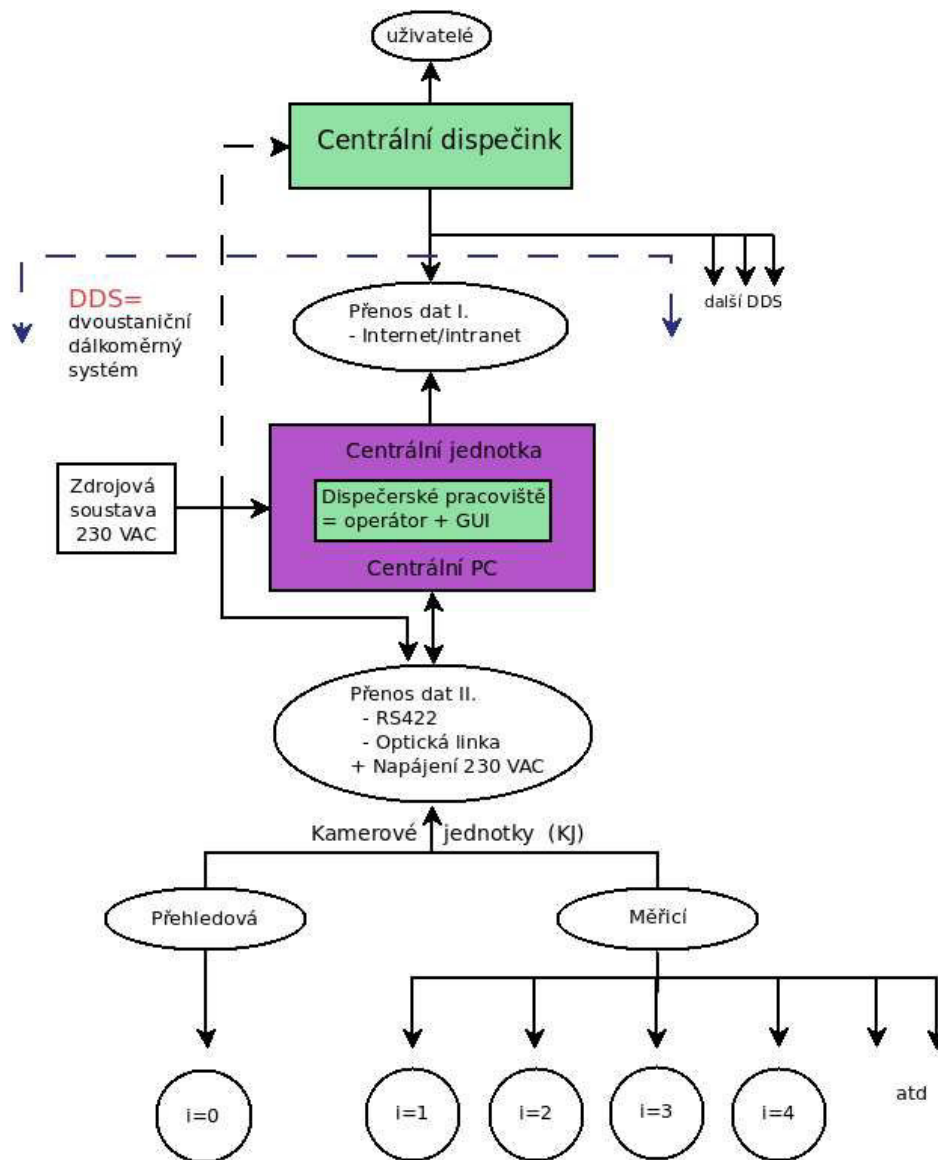
2. Teoretický rozbor

2.1. Kamerový systém

Problematika řešená v této diplomové práci je součástí rozsáhlého projektu, jehož cílem je vytvoření systému pro ochranu perimetru s využitím principu dvoustaničního dálkoměrného systému (DDS) s vodorovnou základnou. V systému se používá speciální pasivní senzorický systém využívající kamerových jednotek, což splněním podmínky sledování cíle alespoň dvěma kamerami současně umožní implementaci principu DDS. Systém umožňuje přesně měření polohy cíle (určení polohy při využití souřadných systémů UTM či WGS 84), určení vektoru rychlosti a extrapolaci dráhy objektu včetně přenosu dat uživatelům. Projekt je zaměřen na problematiku měření polohy pozemních a vzdušných cílů (nízkoletící bezpilotní prostředky, ultralighty, sportovní letadla atd.) do šikmé vzdálenosti cca 3000 m. Systém bude schopný poskytovat přesnou polohu cíle (odchylka < 1 m) pro činnost efektorů cca 30 sekund před přiletem cíle k objektu resp. perimetru.

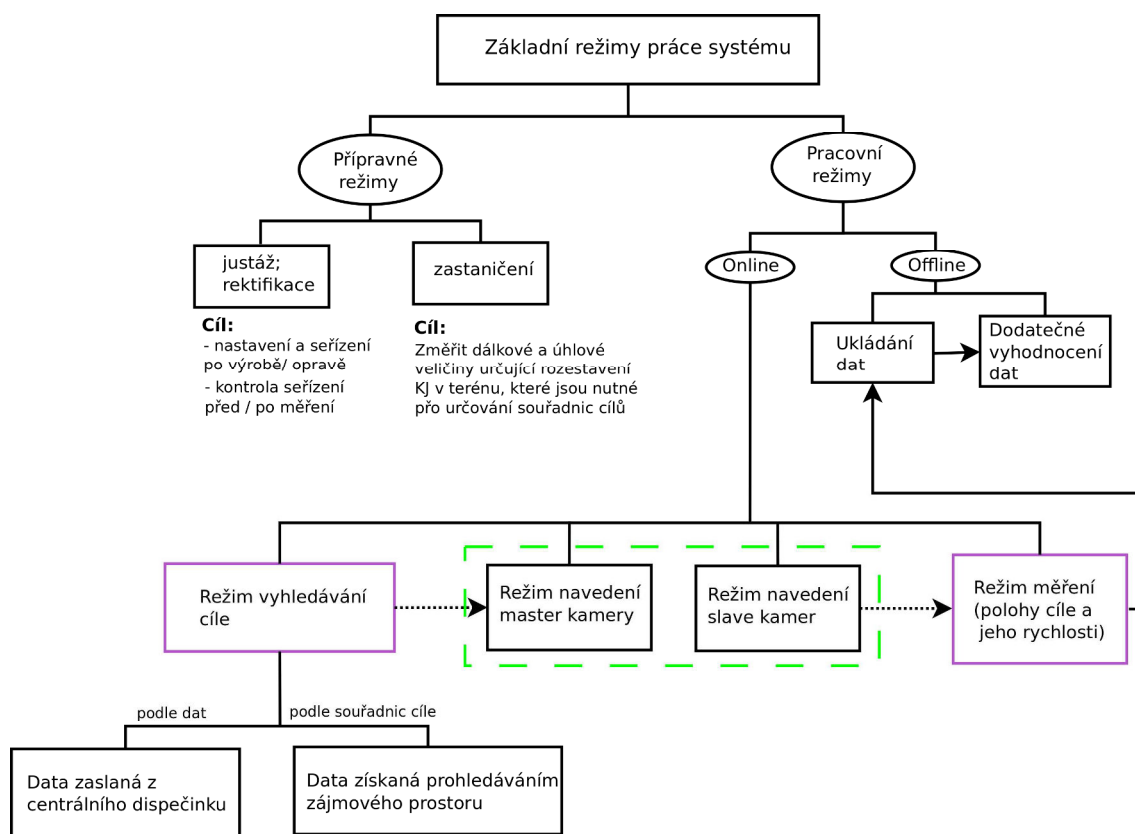
Princip DDS je založen na pozorování cíle dvěma kamerami, u kterých je známa přesná vzájemná poloha. Při pozorování cíle kamery určí úhel mezi pozorovaným objektem a spojnicí se druhou kamerou. Je tedy získán trojúhelník se známou přeponou a úhly mezi přeponou a oběma odvěsnami, z čehož lze vypočítat délky obou odvěsen a tím určit vzdálenost cíle. V systému, kde jsou známy zeměpisné koordináty obou kamer, lze určit přesnou polohu pozorovaného objektu v prostoru. Měření vzdáleností principem DDS klade vysoké nároky na přesnost určení vzdálenosti a vzájemného úhlu mezi měřicími kamerami, protože i malá odchylka způsobí velké chyby měření vzdálenosti pozorovaného objektu.

Struktura celého systému je naznačena na obr. 2. Zadání projektu počítá s vytvořením komplexní struktury ovládání systému. Hierarchicky nejnižší postavenou součástí je kamerová jednotka. Několik kamerových jednotek (minimálně 3) a centrální jednotka tvoří základ DDS. Centrální jednotka je PC umístěné v bezprostřední blízkosti kamerových jednotek, je vybavena softwarem pro ovládání kamerových jednotek a komunikaci s nadřazeným pracovištěm. Obsluha stanoviště je zajištěna operátorem. Úkolem centrální jednotky je v závislosti na aktuálním režimu systému (obr. 3) provádět a kontrolovat zastaničení, vybírat cíle pro sledování, vyhodnocovat výsledky a zpracovaná data posílat do centrálního dispečinku. Centrální dispečink je pracoviště, které sdružuje jeden nebo více DDS. Podle konceptu je centrální dispečink vzdálené pracoviště (desítky až stovky km), které monitoruje data z více DDS a poskytuje výstupní údaje koncovým uživatelům. Celý systém DDS je doplněn elektrocentrálou, spínanými zdroji a stabilizátory napětí pro napájení všech systémů. [2]



Obr. 2: Struktura kamerového systému [1]

Struktura činností systému je rozdělena na dva hlavní režimy – přípravné a měřicí. V přípravném režimu probíhá justáž a zastaničení. Přípravný režim slouží k určení vzájemných vzdáleností mezi jednotlivými kamerovými jednotkami a orientaci v prostoru. Přesnost provedení justáže a zastaničení má zásadní vliv na přesnost výsledných měření systému. Pracovní režimy umožňují měření a sledování objektů v okolí systému. Sledování objektů je možné provádět jak v reálném čase, tak je možné analyzovat i videosekvence uložené v předešlých měřeních. Přehled pracovních režimů celého systému je naznačen na obr. 3.

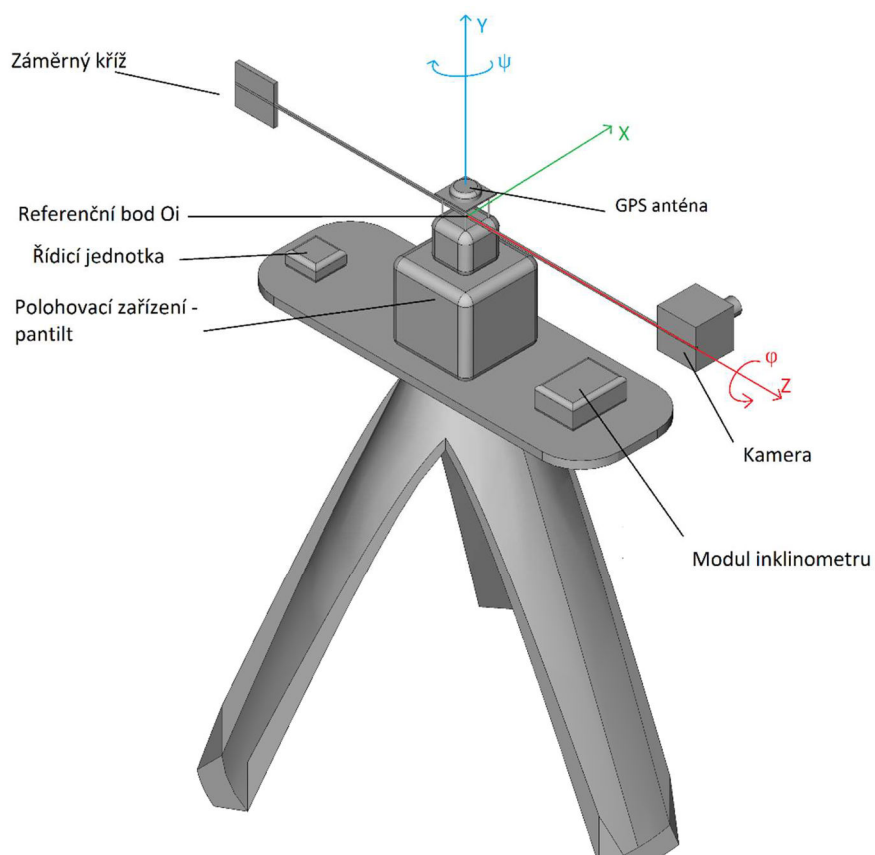


Obr. 3: Režimy práce kamerového systému [1]

2.2. Kamerová jednotka

Kamerová jednotka (obr. 4) je základem dvoustanovičního dálkoměrného systému. Hlavním prvkem kamerové jednotky je kamera umístěna na polohovacím zařízení. Další vybavení kamerové jednotky slouží pro komunikaci, nebo pro určení polohy a orientace v prostoru. Vybavení kamerových jednotek je umístěno na stativu pro snadné rozestavení v terénu. Ve schématu jsou kamerové jednotky děleny na dva druhy – v systému se nachází jedna jednotka přehledová a několik měřících. Přehledová kamerová jednotka je umístěna ve středu útvaru, do kterého jsou rozestaveny měřící kamerové jednotky. Slouží k pozorování celého okolí a hledání cílů, na které jsou pak zaměřeny měřící kamerové jednotky. Jako jediná kamerová jednotka je vybavena objektivem s možností elektronické kontroly zoomu, ostření a clony. Pro spojení kamerové jednotky s centrální jednotkou jsou zavedeny dvě komunikační cesty – optické vlákno (1000BaseFX-SC) pro komunikaci s kamerou a obousměrná linka RS422. Protože kamera podporuje pouze standardní ethernetové vedení, je přímo na kamerové jednotce přítomen převodník 1000BaseFX-SC ↔ 1000BaseTX-RJ45 (kroucený čtyřpár s konektorem RJ45). Řízení ostatních zařízení a sběr dat ze sensorů zajišťuje řídicí jednotka kamerového systému, která s centrální jednotkou komunikuje přes RS422. Kamerová jednotka je připojena k síťovému

napájení 230 V, součástí kamerové jednotky jsou tedy i spínané zdroje pro napájení jednotlivých systémů.



Obr. 4: Vizualizace kamerové jednotky

Polohovací zařízení Flir D46-70

Polohovací zařízení (Pan-Tilt) se na kamerových jednotkách používá k nastavování polohy kamery a záměrného kříže ve dvou osách (podle obrázku Y a Z). Model D46-70 od firmy Flir je charakterizován těmito parametry

- nosnost až 4 kg
- rychlost otáčení až 60 ° za sekundu
- rozsah +- 180 ° v kurzové rovině, ψ na obr. 4
- rozsah 80 ° až +31 ° pro náklon, φ na obr. 4
- rozlišení jednoho kroku až 0.003 °

- komunikační rozhraní RS232
- vhodné pro venkovní použití – IP65
- velký rozsah napájecích napětí – 9 až 30 VDC

Více informací o zařízení lze nalézt na webových stránkách výrobce [3].



Obr. 5: Polohovací zařízení Flir D46-70

Kamera ProsilicaGT1290C

Pro snímání obrazu jsou v kamerovém systému použity přístroje od společnosti Allied Vision Technologies.

Základní charakteristika kamery:

- kvalitní CCD senzor Sony ICX445
- automatická clona
- závěrka typu global shutter
- maximální rozlišení 1280 x 960
- maximální snímková rychlost 33 snímků za vteřinu
- rozhraní IEE 802.3 1000baseT = Gigabitový ethernet
- implementovaný synchronizační protokol IEEE1588
- velké množství výstupních formátů
- GPIO piny pro ovládání a přerušení
- vhodné pro použití v náročných prostředích



Obr. 6: Kamera ProsilicaGT1290C

Objektiv kamery se liší podle použití – kamera přehledové kamerové jednotky je vybavena elektronicky ovladatelným objektivem Computar H6Z0812MSP, kamery měřicích jednotek používají objektivy Computar M5018-MP2. Kompletní popis k nalezení na webových stránkách výrobce [4].

Computar H6Z0812MSP

Elektricky nastavitelný objektiv přehledové kamery. Umožňuje elektronicky nastavovat ohniskovou vzdálenost v rozsahu 8 – 48 mm (šestinásobný zoom), clonu a zaostření. Je použit pro pozorovací kameru kamerové jednotky. V rámci bakalářské práce Andreje Čižmára [5] byla pro tento objektiv realizována řídicí deska, která je použita pro řízení objektivu kamery přehledové kamerové jednotky. Jednotka řízení objektivu používá komunikační rozhraní RS232.



Obr. 7: Objektiv Computar H6Z0812MSP

Computar M5018-MP2

Objektiv s pevnou ohniskovou vzdáleností 50 mm a manuálně nastavitelnou clonou a zaostřením. Využíván na měřicích kamerových jednotkách.



Obr. 8: Objektiv Computar M5018-MP2

Edimax ET-913SFP V2

Převodník ethernet na optickou linku. Podpora převodu ethernetu až kategorie 1000BASE-T na optické vlákno standardu 1000BASE-SX. Použit pro převod komunikace mezi centrálním PC a kamerou na kamerových jednotkách.

Modul inklinometru

Modul inklinometru se skládá z inklinometru RION HCA528T-30, převodní desky s ADS1282 a ovládací procesorové desky s STM32F407. Klíčové parametry inklinometru lze nalézt tab. 1, přesné informace jsou uvedeny na webu výrobce [6]. Tab. 2 obsahuje základní informace o A/D převodníku ADS1282, podrobnější informace lze opět nalézt na webu výrobce [7].

Inklinometr RION HCA528T-30

- Dvouosý, proudový výstup
- Rozsah $\pm 30^\circ$ v obou osách
- Rozlišení 0.001°
- Absolutní přesnost 0.01°
- Napájení $9 \sim 36V$ DC

Tab. 2: Parametry inklinometru RION HCA528T-30

A/D převodník ADS1282

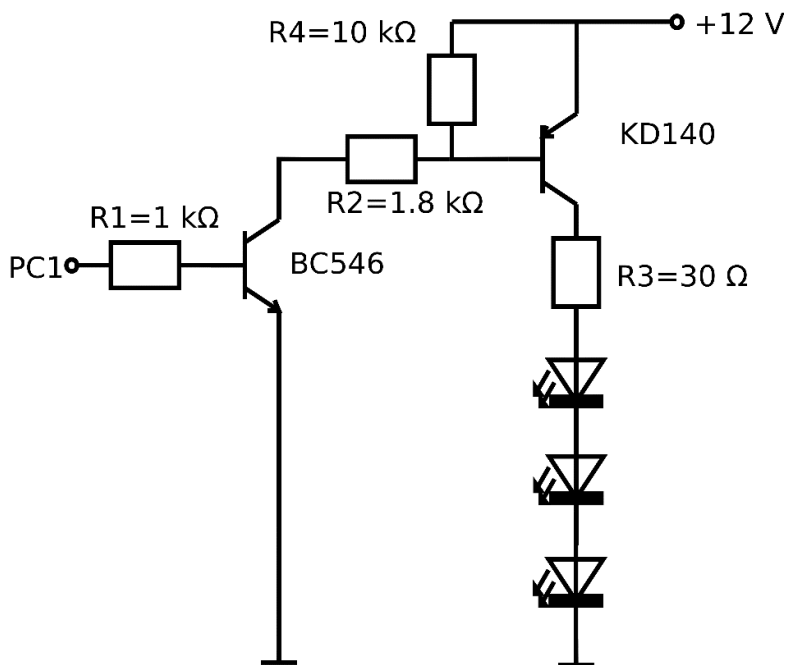
- Rozlišení 31 bitů
- Architektura sigma-delta
- Komunikační rozhraní SPI

Tab. 1: Parametry A/D převodníku ADS1282

Modul inklinometru není pevnou součástí kamerové jednotky – používá se pouze při zastavení k přesnému určení náklonu a sklonu podstavce, jehož poloha se již později během měření nemění. S řídicí jednotkou komunikuje použitím sběrnice CAN.

Záměrný kříž

Terč složený ze silných LED. Využití nachází při zastavení kamerových jednotek – pomocí záměrného kříže se na sebe zaměřují kamery, což slouží k určení vzájemných úhlu mezi kamerovými jednotkami. Rozsvícení a zhasínání záměrného kříže je ovládáno GPIO piny řídicí jednotky. Celé zapojení obvodu záměrného kříže je na obr. 9.



Obr. 9: Zapojení záměrného kříže

Řídicí jednotka

Řídicí jednotka (ŘJ) slouží jako ovládací prvek systému kamerové jednotky a zároveň koncentruje komunikaci kamerové jednotky – z centrálního stanoviště přichází komunikace po obousměrné lince RS422, pomocí které je ovládána celá kamerová jednotka kromě kamery ovládané přímo centrální jednotkou. ŘJ podle typů zpráv vykonává úkoly spojené s řízením kamerové jednotky nebo pouze přeposílá zprávy do dalších periférií. Zároveň také zajišťuje zabezpečení komunikace přidáním CRC kódu.

Jádrem řídicí jednotky je mikrokontrolér STM32F407, pro komunikaci jsou využity následující periférie:

- 3x RS232 (GPS modul, polohovací zařízení, řízení objektivu)
- 1x RS422 (centrální PC) – diferenciální vedení je vytvářeno HW budiči, z pohledu mikrokontroléru obsluhováno jako standardní rozhraní RS232
- 1x CAN bus (modul inklinometru)
- GPIO (záměrný kříž)

Protože mikrokontrolér STM32F407 vysílá a přijímá signály pro RS232, RS422 i CAN bus ve 3,3V TTL úrovních, musí být deska osazena také příslušnými HW budiči těchto komunikačních rozhraní.

Přímou součástí ŘJ je také GPS modul U-Blox NEO-6P. Jedná se o výkonný modul s technologií PPP (Precise Point Positioning) i příjmem korekčních zpráv SBAS. Bližší informace jsou dostupné v datasheetech na stránkách výrobce [8].

Převodník RS422 – USB

Průmyslové ani osobní počítače nebývají vybaveny komunikačním rozhraním RS422, u centrálního PC musí tedy být použit převodník. Použit je typ USB-COM422-PLUS4 od společnosti FTDI. Převodník umožňuje připojení až 4 linek RS422 k jedinému usb portu, každá linka se hostitelskému PC zobrazí jako virtuální sériový COM port. Více informací k tomuto produktu lze dohledat v katalogovém listu [9].



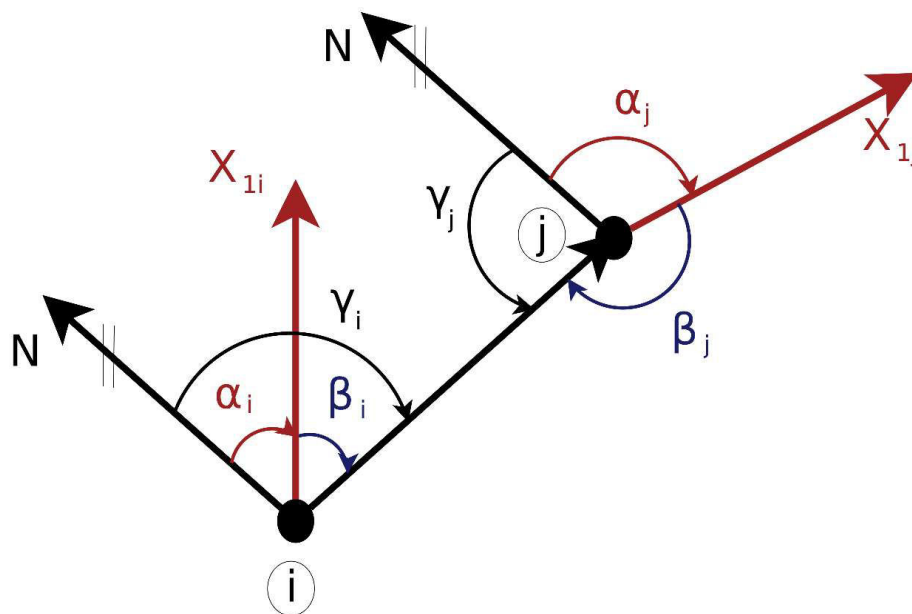
Obr. 10: Převodník RS422 - USB

2.3. Zastaničení

Je to souhrn úkonů vykonávaných k zaměření polohy všech kamerových jednotek dálkoměrného systému v prostoru. Cílem zastaničení je určení nejpřesnější možné polohy referenčního bodu O_i (reprezentuje střed kamerové jednotky) v prostoru pro každou kamerovou jednotku (viz obr. 4) a natočení kamerových jednotek v nulové pozici (X_{1i} a X_{1j} na obr. 11) vůči severu. Algoritmus zastaničení je navržen pro konfiguraci „3+1“ (3 měřicí kamerové jednotky, 1 přehledová), pro konfiguraci s více kamerami musí být program změněn. Výsledkem zastaničení jsou přesné hodnoty vzájemných vzdáleností mezi kamerovými jednotkami (přesnost v řádu cm při vzájemné vzdálenosti cca 100 m), úhly všech kamerových jednotek vůči severu a hodnoty náklonu a sklonu kamerových jednotek. Určení absolutní polohy DDS v prostoru počítá s chybou do 5 metrů.

Zastaničení je ovládáno programem centrálního počítače a probíhá v následujících krocích:

1. Rozestavení kamerových jednotek v terénu přibližně do tvaru trojúhelníku s přehledovou kamerou ve středu, zapojení kabeláže. Zapojení kamerových jednotek do zdroje elektrické energie a inicializace všech systémů kamerové jednotky. Spuštění centrálního PC s SW pro řízení zastaničení.
2. Inicializace GPS a polohovacích zařízení – nastavení rychlosti a velikosti kroku polohovacích zařízení pro všechny kamerové jednotky. Výsledkem je, že všechny polohovací zařízení jsou v poloze [0,0].
3. Zakázání pohybu polohovacích zařízení pro zajištění jednotné polohy kamerových jednotek během měření GPS. Ukládání přeposílaných dat z GPS modulů v centrálním PC. Během měření GPS dat připojí operátor DDS ke každé kamerové jednotce modul inklinometru pro zjištění náklonu a sklonu základen kamerových jednotek.
4. Výpočty absolutní polohy kamerové jednotky 1 a orientované vzdálenosti mezi kamerovými jednotkami číslo 1-0, 1-2, 1-3, 2-3 pomocí výpočetních funkcí z RTKlib, kde 0 představuje přehledovou kameru.
5. Načtení hodnot z inklinometru
6. Vzájemné zamíření kamer – pro dvojice 1-0, 1-2, 1-3 a 2-3 je pomocí kamery a záměrného kříže zjištěn úhel mezi nulovou pozicí polohovacích zařízení a spojnicí obou kamerových jednotek (viz obr. 11). Z diferenčního měření GPS jsou známy úhly γ_i a γ_j , vzájemným zamířením kamerových jednotek i a j ($i \neq j$) jsou získány hodnoty β_i a β_j . Z těchto hodnot se pak dopočítají hodnoty úhlu α_i a α_j – natočení kamerové jednotky vůči severu.



Obr. 11: Vzájemné zamíření kamer při zastaničení

7. Výpočty pro přepočet úhlů a vzdáleností vůči referenčnímu bodu O_i – použitím rotačních matic jsou na základě znalosti vzdáleností záměrného kříže a kamery od bodu O_i , parametrů kamer a objektivů, hodnot z inklinometru a natočení polohovacího zařízení při jednotlivých fázích měření výsledky zastaničení přepočteny do bodu O_i . Výsledné úhly kamerových jednotek vůči severu jsou zkontrolovány měřením kompasem.

2.4. Softwarové zabezpečení komunikace

U datových přenosů může docházet k chybám během přenosu dat od odesílatele k příjemci. Vzniku chyb není možné spolehlivě předcházet, existují ale metody pro zajištění spolehlivosti komunikace. Nejčastěji používanou metodou pro zajištění spolehlivosti je použití bezpečnostních kódů pro detekci chyb společně s potvrzováním příjmu zprávy od příjemce k odesílateli. Další možností je využití samoopravných kódů posílaných přímo se zprávou. Samoopravné kódy se ale kvůli velké režii a menší propustnosti v porovnání s opětovným posíláním používají minimálně, využití nacházejí např. pokud není možné opakovat vysílání zprávy. Informace o softwarovém zabezpečení komunikace jsou převzaty z [10].

Detekce chyb

Jako bezpečnostní kód se používá jeden nebo více bitů přidaných k původní odeslané zprávě. Hodnota zabezpečujících bitů se podle typu zabezpečení vypočítá z dat určených k odeslání, odeslány jsou poté původní data i se zabezpečením. Na straně příjemce je při příjmu vykonán stejný výpočet

bezpečnostního kódu z datové části přijaté zprávy, provede se porovnání přijatého a vypočítaného bezpečnostního kódu a při shodě kódů je přijatá zpráva považována za platnou.

Použitím bezpečnostní kódu se tedy do přenosu přidávají další bity, které nejsou pro přenos zprávy nutné. Při detekování chyby, stejně jako pro celé odesílání zprávy, není možné dosáhnout stoprocentní pravděpodobnosti bezchybného přenosu. Obecně platí, že s rostoucí délkou zabezpečující zprávy klesá pravděpodobnost nedetekované chyby. Zabezpečení komunikace je možné pro jednotlivé datové zprávy, ale i pro celé bloky dat.

Parita

Nejjednodušší a nejméně účinný bezpečnostní kód, je volitelnou součástí RS232 přenosů. Parita je jeden bit, kterým je zpráva doplněna na sudý, nebo lichý počet logických jedniček ve zprávě (sudá a lichá parita). Parita odhaluje lichý počet chybných bitů ve zprávě.

Pro zabezpečení blokových přenosů paritou se rozlišuje příčná parita (paritní bit na konci přenosu znaku) a podélná (po $x > 1$ znacích následuje paritní znak kontrolující počet jedničkových bitů ve stejnohlých bitových pozicích všech znaků). Při použití kombinace podélné a příčné parity je možné přesně detekovat a opravit chybný bit.

Kontrolní součet

Postupně odesílané zprávy se při odesílání také přičítají do bufferu o stejné velikosti, jako mají odesílané zprávy. Obsah bufferu je odeslán na konci vysílání bloku zpráv jako kontrolní součet. Kontrolní součet odhalí více chyb než parita.

CRC (Cyclic Redundancy Check)

Cyklický redundantní součet je obdobou hašovací funkce. Bezpečnostní kód je vypočítán jako zbytek po dělení datové posloupnosti generujícím polynomem s binárními koeficienty. Délka generujícího polynomu je uváděna jako číslo za písmeny CRC (např. CRC16 používá generující polynom s nejvyšší mocninou x^{16} , délka polynomu je 17 bitů). Při uvádění číselné hodnoty generujícího polynomu bývá úvodní bit často vynechán, protože jeho hodnota je vždy 1. Kontrolní součet je vždy o 1 bit menší než generující polynom → pro CRC16 odpovídá kontrolní součet 16 bitům.

CRC je nejspolehlivějším a nejpoužívanějším mechanismem pro detekci chyb komunikace. CRC dokáže odhalit všechny shluky chyb s lichým počtem bitů a všechny shluky chyb menší než velikost generujícího polynomu, větší shluky detekuje CRC s pravděpodobností 99.99999998%. Výhodou je také vysoká rychlost výpočtu CRC.

Potvrzování

Nejčastěji je spolehlivý přenos zajišťován potvrzovací zprávou, případně opakováním přenosu poškozených dat. Předpokladem pro potvrzování je, že příjemce vyhodnotí, zda byl datový blok přenesen bez chyb a následně vyšle příslušnou potvrzovací zprávu – kladné potvrzení znamenající bezchybný přenos, nebo záporné potvrzení znamenající chybu v přenosu. Při potvrzované komunikaci se musí počítat i se dvěma krajními možnostmi:

- při přenosu se ztratí celý blok a příjemce ani neví, že by měl něco potvrzovat.
- ztratí se vygenerované potvrzení

Z těchto důvodů je při komunikaci na straně odesílatele nutné zavést časovač, který po určité době bez potvrzení automaticky považuje odesílání za neúspěšné a přenos bloku se opakuje

V závislosti na umístění potvrzení do datového toku se rozlišují dva způsoby odesílání potvrzení - jednotlivé potvrzování – potvrzení je posíláno po každém datovém bloku a další datový blok se odesílá až po přijetí úspěšného potvrzení z předešlého bloku. Vhodné pouze pro systémy s nízkou latencí komunikace.

- kontinuální potvrzování – odesílání nečeká na úspěšné potvrzení předchozího bloku. Při přijetí záporného potvrzení se používá selektivní opakování (znova odeslán pouze blok s negativním potvrzením), nebo opakování s návratem (veškerá data přijatá po chybně přijaté zprávě jsou zahozena a odesílání se opakuje od chybně přijatého bloku dat).

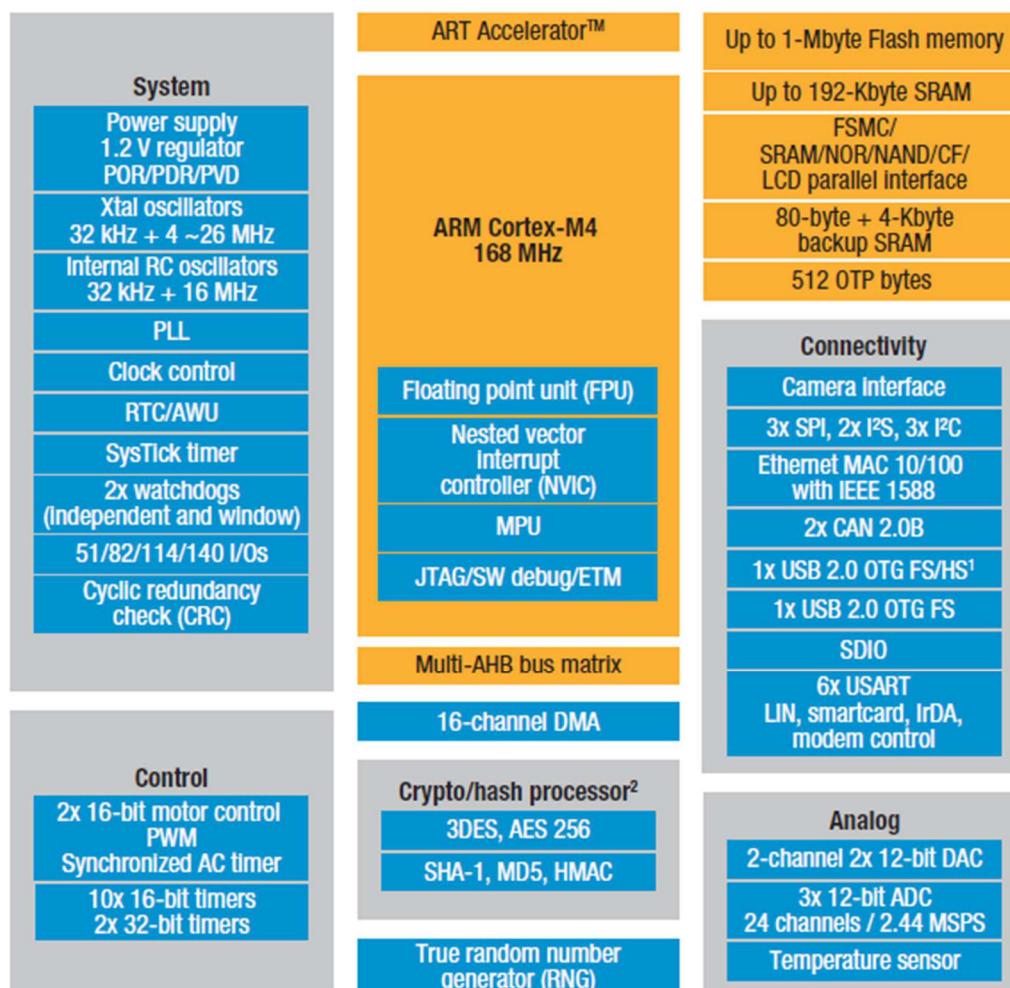
2.5. Mikrokontrolér STM32F407

Jako mikrokontrolér je označován kompletní výpočetní systém na čipu, který obsahuje procesor i komunikační periferie. V angličtině bývá mikrokontrolér označován zkratkou SOC – Systém On Chip. Tyto systémy se vyznačují nízkým výkonem a spotřebou. Mikrokontroléry se používají pro řízení vestavných systémů.

Mikrokontrolér řady STM32F4 od společnosti STMicroelectronics založený na jádře ARM Cortex M4 se výkonem řadí do vyšší řady mikrokontrolérů od ST. Díky možnosti užívání MAC (multiple and accumulate – sčítání násobků do akumulátorové proměnné v jednom hodinovém cyklu procesoru) funkcí je tento mikrokontrolér označován jako DSP (digitální signálový procesor).

STM32F407 poskytuje všechna potřebná komunikační rozhraní pro splnění funkce řídicí jednotky, komplexní přehled možností mikrokontroléru je uveden na obr. 12. Spotřebu mikrokontroléru je

možné snížit na úkor výpočetního výkonu snížením taktovací frekvence (základní frekvence 168 MHz). Popis registrů a další informace pro práci s STM32F407 jsou k nalezení v referenční příručce [11].



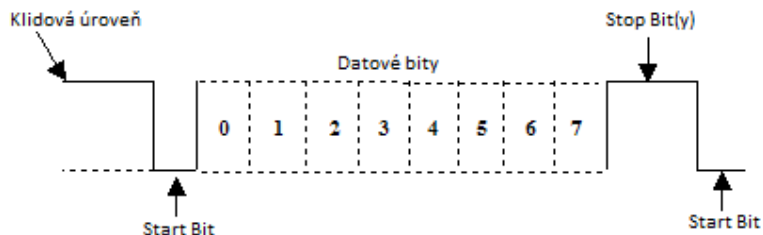
Obr. 12: Schéma procesoru STM32F4xx [12]

2.6. Komunikační rozhraní RS232, RS422 a CAN bus

RS232

RS232, známý jako sériový port nebo sériová linka, je sériové digitální komunikační rozhraní, které je pro svou jednoduchost používáno ve vestavných zařízeních, z osobních počítačů bylo toto rozhraní vytlačeno modernějšími rozhraními, např. USB. Standard definovaný roku 1969 popisuje mimo jiné napěťové úrovně, časování signálů a formát přenosu. Pro přenos je ve verzi bez hardwarového řízení (Request to Send, Clear to Send) zapotřebí jen tři vodičů – vysílací TxD, přijímací RxD a zemní GND. Přenos je vždy zahájen start bitem (log. 0), následuje 7 až 8 datových bitů, a celý přenos je ukončen

stop bitem velikosti 1, 1.5, nebo 2 bity (log. 1). Na konci přenosu je volitelně vysílána sudá či lichá parita pro kontrolu přenosu. Pořadí datových bitů začíná nejvíce významným bitem. Bitová rychlost dosahuje až 1 Mbps a při nízkých rychlostech je přípustná délka spojení až 100 m. Napěťové úrovně se pohybují v rozsahu -3 V až -25 V pro logickou 1, 3 V až 25 V pro logickou 0 [13].

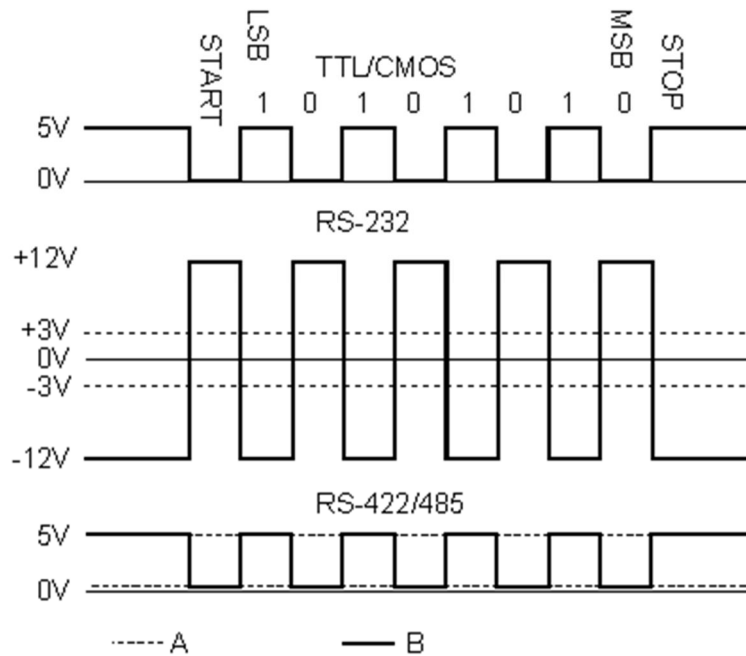


Obr. 13: Přenos zprávy rozhraním RS232 [13]

RS422

RS422 je sběrníkový systém používající pro přenos dat v jednom směru dvojici vodičů, pro identifikaci hodnoty je na přijímací straně vyhodnocena polarita difference napětí mezi oběma vodiči. Použitím diferenciálního vedení kroucenými kabely se vylepšuje šumová odolnost komunikační linky a je možné použít RS422 na delší vzdálenosti a s větší bitovou rychlostí (1200 m na 100 kbps) než RS232. V praxi je RS422 používán mimo jiné také na zvětšení dosahu a odolnosti sériové linky použitím konvertoru RS232 ↔ RS422 [14].

ASCII "U" = 85 Decimal = 55 Hexidecimal = 01010101 Binary



Obr. 14: Porovnání napěťových úrovní při komunikaci pro TTL/CMOS logiku, RS232 a RS422 [14]

Sběrnice CAN

CAN je komunikační protokol, jehož vývoj začal roku 1983 ve firmě Bosch. Protokol byl navrhován pro použití v automobilovém průmyslu, nyní je ale rozsah použití mnohem širší a CAN můžeme nalézt i v jiných průmyslových aplikacích. Fyzická vrstva je založena na tzv. Drátovém součinu – na sběrnici existují 2 logické úrovně, recesivní a dominantní. Pokud vysílají všechny připojené jednotky na sběrnici recesivní úroveň, je na sběrnici recesivní úroveň. Pro dominantní úroveň na sběrnici stačí, aby jediná jednotka vysílala dominantní úroveň. V praxi se pro buzení logických úrovní používají budiče typu otevřený kolektor – sběrnice je přes pull up rezistor držena na úrovni high (recesivní úroveň), tranzistorem v budiči je sběrnice uzemňována do dominantní úrovně. Jako vedení se používá symetrický kroucený pár s charakteristickou impedancí 120 Ω . Signály sběrnice se nazývají CAN_H a CAN_L. Výsledná logická hodnota je určena jako rozdíl napětí mezi vodiči – diferenční vedení. Při maximální přenosové rychlosti 1 Mb/s je délka sběrnice omezena na 40 m. S omezenou rychlostí je možné použít délku sběrnice až 5 km (při 10 kb/s). Spojová vrstva sběrnice zajišťuje přístup ke sdílené sběrnici metodou CSMA/CR (Carrier Sense Multiple Access with Collision Resolution). Tato metoda dává prioritu odesílání zprávy s nejnižším ID. Uzly na sběrnici nemají adresy, každý uzel na sběrnici přijímá celou komunikaci ze sběrnice a podle ID zprávy, které je v systému jedinečné, se uzel rozhoduje,

zda zprávu zpracuje. Protokol zajišťuje spolehlivost komunikace pomocí bit stuffing (po 5 bitech stejné polarity je automaticky vložen bit opačné logické hodnoty) a CRC [15].

2.7. GPS

GPS (Global Positioning System) je globální polohovací systém vyvíjený od roku 1973 Ministerstvem obrany Spojených států. Plně funkční a dostupný po celém světě je systém od začátku 90. let. Systém byl vyvinut primárně pro vojenské účely a až do roku 2000 byla v některých místech záměrně zhoršována přesnost určení polohy civilních zařízení [16].

Struktura systému je rozdělena na 3 části:

- Kosmický segment - Ve výšce 20350 km nad povrchem Země obíhá 32 družic (minimální počet pro plnou funkčnost systému je 24). Družice obíhají v 6 kruhových drahách se sklonem 55 °, dráhy jsou vzájemně posunuty o 60 °. Družice se pohybují rychlostí 3,8 km/s, doba oběhu družice okolo Země je 11 hodin a 58 minut. Každá družice je vybavena trojicí až čtveřicí přesných atomových hodin (přesnost 10^{-13} s), anténami pro vysílání radiových kódů, komunikaci s řídicím segmentem i vzájemnou komunikaci s dalšími GPS satelity. Obsahují také sensory pro detekci startů balistických a jaderných raket.
- Řídicí segment – sada pozemních stanic pro monitorování a servis družic, vypočítávají přesnou dráhu družic a korigují hodiny družice.
- Uživatelský segment – pasivní uživatelské stanice, které přijímají signály z jednotlivých satelitů k získání informace o své poloze a čase.

Princip určení polohy pomocí GPS

Každá družice vysílá signál obsahující informaci o své poloze a přesném čase. Z těchto údajů počítá GPS přijímač tzv. pseudovzdálenosti, což jsou vzdálenosti mezi přijímačem a viditelnými družicemi (nad obzorem). Výpočet pseudovzdáleností vychází ze znalosti rychlosti šíření družicového signálu a rozdílu času mezi vysláním a přijetím signálu. K tomu jsou ještě přidány doplňující výpočty pro zpřesnění výsledné polohy.

Pro dosažení stejné přesnosti hodin v GPS přijímači je z přijímaného signálu vypočítáván tzv. Clock offset, který v kombinaci s velmi přesnými časovými značkami vysílanými z družic umožňuje přijímači určit čas s chybou menší než 1 mikrosekunda. Poloha je GPS přijímačem vypočítávána na základě srovnání vzdáleností od několika družic (minimální počet jsou tři družice pro tzv. 2D fix, s vyšším počtem viditelných družic se zpřesňuje určená poloha) [16].

Diferenční GPS

Při běžné měření pozice pomocí systému GPS s jednou anténou je poloha určena přijetím signálu z minimálně 4 satelitů, přičemž každý z těchto signálů je zatížen chybou způsobenou průchodem signálu různým prostředím od satelitu až k anténě GPS přijímače. Principem diferenčního měření GPS je použití více přijímačů – vzhledem k velké vzdálenosti satelitů od Země jsou dva přijímače vzdálené i stovky kilometrů relativně blízko u sebe a lze u nich předpokládat podobné chyby přijatých dat. Toho lze využít dvěma způsoby:

- Přijímač s velmi přesně určenou polohou vyhodnocuje přijímaná data z GPS. Místo vyhodnocení pozice z přijatých dat používá známou polohu pro vyhodnocení chyby příchozího signálu. Rozdíl mezi přijatým signálem a určením, jaký signál by měl být přijat, pak tento přijímač poskytuje dalším přijímačům jako korekce. Pro přenos korekcí musí být vytvořena komunikační cesta.
- Pro dva blízké přijímače se očekává stejná chyba v příchozím signálu pro oba přijímače, z čehož lze vypočítat vzájemnou orientovanou vzdálenost s přesností na jednotky centimetrů [17].

RTKLIB

RTKLIB je sada aplikací pro standardní i precizní určování pozic za použití GNSS (globální navigační satelitní systémy). RTKLIB podporuje zpracování pozice pro všechny GNSS, nabízí různé módy pro zpracování v reálné čase i postprocessing. Programy z RTKLIB podporují velké množství komunikačních zpráv a přijímačů od různých výrobců. Programy z RTKLIB lze spustit na operačních systémech Windows, Linux i Android. Obsaženy jsou aplikace s grafickým rozhraním (GUI), konzolové aplikace (CUI) [18].

3. Návrh komunikačního protokolu

Tato kapitola se zabývá popisem komunikace mezi centrální jednotkou a kamerovými jednotkami, ale i popisem komunikace mezi jednotlivými periferiemi kamerových jednotek. Jsou zde popsány zprávy posílané mezi jednotlivými periferiemi a parametry použitých komunikačních rozhraní a případné zabezpečení komunikace proti chybám. Komunikace byla vytvářena a testována s konkrétním SW pro centrální PC, který bude sloužit pro zastaničení DDS. Cílem práce je takový návrh a realizace komunikačního protokolu ŘJ, které umožní ovládání kamerových jednotek i z jiných PC aplikací.

Komunikace centrálního PC s kamerovými jednotkami

Až na kameru, která komunikuje přímo s centrálním počítačem, prochází veškerý komunikační tok řídicí jednotkou. Z důvodu velké vzdálenosti mezi kamerovými jednotkami a centrálním počítačem je použito komunikační rozhraní RS422. Linka RS422 je plně duplexní, z hlediska ŘJ se ovládá jako RS232 rozhraní, diferenciací vedení zajišťuje hardwarový budič. Komunikace mezi řídicí jednotkou a centrálním PC má pevně definované zprávy, jejichž délka je neměnná. Zprávy jsou rozlišovány podle ID bytu na začátku každé zprávy. Podle ID se zprávy dělí na zprávy určené k přeposlání do objektivu nebo polohovacího zařízení, a na zprávy pro řízení ŘJ. Při příjmu zprávy tedy řídicí jednotka podle prvního přijatého bytu zjistí, kterou zprávu přijímá a kolik ještě zbývá přijmout bytů do konce zprávy. Teprve po přijetí celé zprávy ŘJ provede akci určenou podle ID zprávy. ŘJ podporuje zabezpečení komunikace CRC8 kódem, řízení komunikace se ale očekává od centrálního počítače. Příkazy jsou posílány z centrálního PC, a pokud ŘJ vrátí negativní potvrzení či nevrátí potvrzení vůbec, je na programu centrálního PC, aby vysílání opakoval. Kamery na kamerových jednotkách jsou k centrálnímu PC připojeny přímo pomocí optického kabelu. Obsah komunikace s kamerou je tedy čistě v režii SW pro ovládání centrálního PC.

Komunikace ŘJ a GPS modulu

GPS modul Ublox NEO-6P je přímo součástí řídicí jednotky. K procesoru řídicí jednotky je připojen přes RS232. GPS je používána pouze během zastaničení k určení přesné orientované vzdálenosti mezi kamerovými jednotkami. Ublox moduly typu NEO-6P umožňují nastavení, ve kterém se posílají pouze zprávy RXM-RAW a RXM-SFRB, maximální nastavitelná frekvence při vysílání těchto dvou zpráv je 10 Hz. Zpráva RXM-RAW obsahuje všechny potřebné informace pro konverzi do univerzálního formátu pro předávání GPS zpráv RINEX (Receiver Independent Exchange Format). Ze zpráv RXM-SFRB jsou získávány SBAS zprávy pro korekci ionosférických vlivů.

K GPS modulu je připojena zálohovací baterie, modul si tedy pamatuje své poslední nastavení. Z hlediska inicializace je to spíše komplikace – nově připojený modul může mít libovolné nastavení a

postupným testováním je nutné zjistit, na jaké rychlosti aktuálně modul komunikuje, teprve poté je možné nastavit požadované parametry Ublox modulu:

1. nastavení rychlosti sériové linky GPS modulu i ŘJ na 11520 Bd/s (datový tok závislý na počtu satelitů ve výhledu GPS antény, při experimentálních měřeních dosahoval hodnot maximálně 3,5 kB/s)
2. povolení posílání zpráv RXM-RAW a RXM-SFRB
3. zakázání posílání NMEA zpráv
4. nastavení frekvence posílání zpráv na 10 Hz

Po nastavení vysílá GPS modul zprávy vhodné pro diferenční měření a výpočty vzdáleností mezi jednotlivými kamerovými jednotkami. Samotné zpracování neprobíhá v ŘJ, ale v centrálním počítači programy z RTKLIB. Do centrálním PC je tedy přeposílán stejný obsah, který ŘJ přijímá z GPS modulu. Přeposílání GPS logu je řízeno příkazy z centrálního počítače – je rozlišována zpráva pro spuštění přeposílání a zpráva pro zastavení přeposílání. Pokud je přeposílání zapnuto, posílají se do centrálního počítače zprávy s GPS logem skládající se z 8 bitové hlavičky 64 bytů dat. Po ukončení měření GPS je další zprávou přeposílání dat přijímaných z GPS modulu zastaveno. Zabezpečení dat pomocí CRC není u rámců s GPS daty zavedeno – příjem GPS zpráv má charakteristiku isochronního přenosu. Pokud dojde k chybě v komunikaci, zpracování v RTKLIB pokračuje ve výpočtech dalším vzorkem bez ztráty důležité informace. Případné zajištění spolehlivosti by bylo náročné na režii přenosu a nepřineslo by statisticky významné zlepšení vyhodnocených výsledků.

Komunikace ŘJ s modulem objektivu

S objektivem komunikuje pouze přehledová kamerová jednotka, měřicí jednotky jsou vybaveny pouze manuálně nastavitelnými objektivy. Ovládací deska objektivu komunikuje s řídicí jednotkou přes RS232. Komunikace je rozdělena na dva typy zpráv – zprávy pro nastavení parametrů objektivu a odpovědi na tyto příkazy a kontrolní komunikaci ŘJ ↔ objektiv. Ovládací příkazy posílané z centrálního PC i odpovědi ovládací jednotky objektivu nevyžadují od ŘJ žádnou úpravu a jsou pouze přeposílány. Komunikace je zabezpečena pomocí CRC8, obsahem zpráv se ale po identifikaci hlavičky řídicí jednotka již nezabývá, pouze přijme počet bytů odpovídající celé zprávě a celou zprávu následně přepoše do ovládací jednotky objektivu. Ověření CRC probíhá až v ovládací desce objektivu. Stejně jsou přeposílány i odpovědi ovládací jednotky objektivu do centrálního PC.

Kontrolní komunikace mezi ŘJ a ovládací deskou objektivu je realizována po vzoru watchdog kontroly – periodickým odesíláním zprávy z ŘJ do ovládací desky objektivu. Ovládací deska objektivu na tyto zprávy odpovídá definovanou zprávou. Pokud ŘJ neobdrží odpověď na tři po sobě následující kontrolní zprávy, je ovládací jednotka objektivu resetována a celá událost je nahlášena na centrální stanoviště.

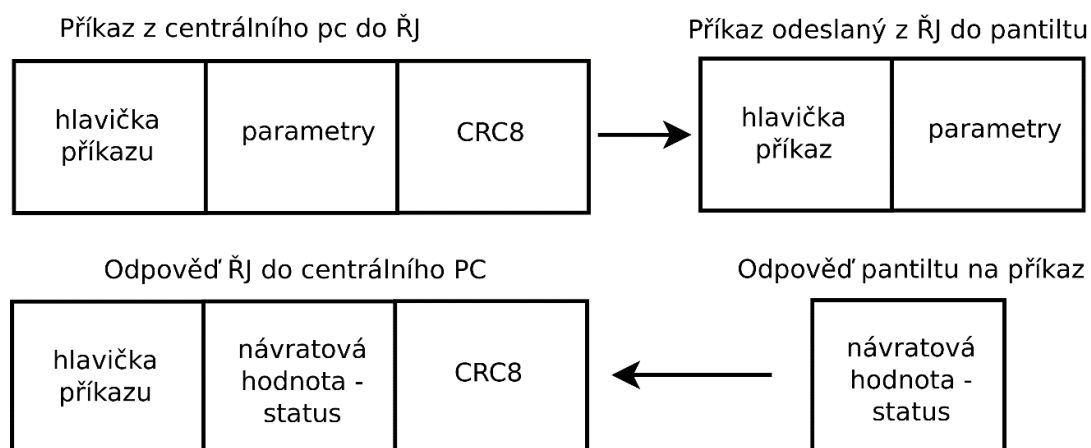
Reset ovládací jednotky objektivu se provádí změnou logické úrovně GPIO pinu napojeného na NRST pin mikrokontroléru ovládací desky objektivu.

Komunikace ŘJ a polohovacího zařízení

Tyto dvě jednotky jsou propojeny sériovou linkou. Polohovací zařízení nabízí dva formáty komunikace – binární a ascii. Binární komunikace, která je v systému primárně použita, je mnohem efektivnější – potřebuje méně bytů pro odeslání stejných příkazů než ascii komunikace. Ascii komunikace je zase přehlednější pro uživatele komunikujícího přímo přes textový terminál a pro některé ascii příkazy není v polohovacím zařízení naprogramován ekvivalentní příkaz v binárním formátu.

Podobně jako u komunikace s objektivem, je většina příkazů pro polohovací zařízení zasílaných z centrálního stanoviště pouze přeposílána. Zpět do centrálního počítače jsou řídicí jednotkou také přeposílány odpovědi polohovacího zařízení centrálnímu počítači na tyto příkazy. Výrazným rozdílem v komunikaci oproti komunikaci s objektivem je zajištění spolehlivosti komunikace. Spolehlivost komunikace je zajištěna pouze na komunikační cestě mezi centrální jednotkou a řídicí jednotkou. Komunikační rozhraní polohovacího zařízení neumožňuje libovolné zabezpečení komunikace, je tedy žádoucí, aby komunikační cesta mezi ŘJ a polohovacím zařízením byla nejkratší možná a kvalitně stíněná.

Binární formát příkazů má pro každý příkaz definovanou osmibitovou hlavičku, po které následuje podle typu zprávy až čtyřbytový argument příkazu. K celému příkazu je přidána detekce chyb v podobě CRC8. Odpověď polohovacího zařízení na zpracovaný příkaz je zpět do centrálního PC odeslána připojena za osmibitovou hlavičku o stejné hodnotě jako příkaz, na který bylo odpovídáno. K této zprávě je nakonec připojen kontrolní součet CRC8. Názorně je obsah zpráv při přeposílání zobrazen na obr. 15.



Obr. 15: Schéma přeposílání zpráv z centrálního PC do polohovacího zařízení

Pro inicializaci polohovacího zařízení na nastavení požadované testované aplikace v centrálním PC je potřeba použít také ASCII příkaz pro nastavení jemnosti kroku, který nemá binární ekvivalent. Tento příkaz je společně s příkazy na nastavení rychlosti otáčení polohovacího zařízení a vypnutí limitů otáčení integrován do jednoho příkazu z pohledu centrálního PC. V praxi to znamená, že při příjmu dvoubytové zprávy pošle řídicí jednotka 35 bytů dlouhou konfigurační zprávu do polohovacího zařízení. Pokud se příkazy úspěšně provedou a polohovací zařízení vrátí řídicí jednotce správně 73bytovou odpověď, ŘJ odešle do centrálního PC informaci, že byla sada příkazů vykonána bezchybně. V opačném případě posílá ŘJ negativní potvrzení a celá operace se musí opakovat.

Komunikace ŘJ ↔ Modul inklinometru

Modul inklinometru je k ŘJ připojen použitím sběrnice CAN. Inklinometr je v systému použit pouze jednorázově – slouží k přesnému změření náklonu a sklonu základní platformy kamerové jednotky. Protože se platforma kamerové jednotky po zastavení nesmí již hýbat, stačí pro celý dálkoměrný systém jediný modul inklinometru, který se mezi kamerovými jednotkami přenáší. Po připojení modulu k ŘJ, ze které je modul i napájen, se modul spustí a čeká 10 vteřin na dokončení operátorovy manipulace s kabeláží, aby bylo dosaženo přesného měření. Poté je naměřena 1 sekunda dat při vzorkovací rychlosti 100 Hz. Z naměřených dat jsou aritmetickým průměrem vypočítány hodnoty náklonu a sklonu, které jsou ve dvou zprávách s rozdílnými ID odeslány přes CAN do řídicí jednotky. Po odeslání hodnot je úkol modulu inklinometru splněn a je přepnut do režimu spánku pro úsporu energie. Stav činnosti modulu jsou pro operátora signalizovány pomocí LED. Celá procedura je řízena pouze programem v modulu inklinometru, ŘJ z CAN sběrnice data pouze přijímá.

Naměřené hodnoty jsou uloženy ve volatilní paměti ŘJ. Po resetu nebo odpojení napájení jsou tedy tyto hodnoty ztraceny. V případě odpojení napájení je tato funkcionality žádoucí – po startu je hodnota náklonu a sklonu inicializována na hodnotu NaN a je možné snadno rozlišit, že má ŘJ aktuální data. V systému je ale také příkaz na softwarové resetování ŘJ při uviznutí komunikace, a při volání příkazu pro resetování ŘJ je hodnota stále platná a může být požadována pro vyčtení. Je tedy použita nevolatilní paměť flash, do které jsou hodnoty z inklinometru ukládány těsně před softwarovým resetováním mikrokontroléru. Při každém startu ŘJ jsou hodnoty náklonu a sklonu z flash paměti načteny do paměti a následně jsou hodnoty ve flash paměti přepsány na hodnotu NaN. Tím je zajištěno uchování dat pouze po softwarovém resetu. Hodnoty náklonu a sklonu platformy kamerové jednotky jsou na požadavek posílány do centrálního počítače.

Synchronizace kamery a polohovacího zařízení

V systému je také požadováno, aby centrální počítač pro každý snímek z kamery znal aktuální pozici polohovacího zařízení. K realizaci jsou použity GPIO piny kamery ProsilicaGT1290C a mikrokontroléru

ŘJ – při expozici každého snímku kamery je v ŘJ vyvoláno přerušení, při kterém jsou do polohovacího zařízení vyslány požadavky na aktuální pozici. Odpovědi z polohovacího zařízení obsahující hodnoty pozice nejsou do centrálního PC posílány přímo, ale jsou odesílány zabaleny do společné zprávy s čítačem zpráv a CRC8. Čítač zpráv je osmibitová proměnná, která se inkrementuje při každém přerušení. Jejím účelem je kontrola počtu obdržených snímků oproti počtu přijatých pozic polohovacího zařízení a jejich vzájemné spárování. Pokud dojde k desynchronizaci, může centrální PC příkazem čítač zpráv vynulovat.

Reset řídicí jednotky

V případě chyby komunikace, špatné délky příchozí komunikace nebo přijetí nedefinované hodnoty hlavičky existuje možnost, že se komunikace ŘJ zastaví nebo vysílá nesmyslné hodnoty a špatná potvrzení. Pokud centrální PC identifikuje komunikaci jako uvízlou v nedefinovaném stavu, má možnost resetovat celou ŘJ. Reset ŘJ je proveden, kdykoliv ŘJ přijme z centrálního PC šest bytů hodnoty 0x64 v řadě, nezávisle na hlavičkách a pořadí bytů ostatních příkazů.

3.1. Příkazy centrálního počítače pro ovládání kamerové jednotky

V této kapitole je vypsána syntaxe a popis příkazů, kterými program centrálního PC ovládá kamerovou jednotku.

Použity jsou následující číselné datové typy:

- u8 bezznaménkové číslo typu integer o velikosti 8 bitů
- u16 bezznaménkové číslo typu integer o velikosti 16 bitů
- int16 znaménkové číslo typu integer o velikosti 16 bitů
- u32 bezznaménkové číslo typu integer o velikosti 32 bitů
- float číslo s plovoucí řádovou čárkou podle IEEE754, velikost 32 bitů

Příkazy pro řízení objektivu

Příkazy pro řízení objektivu byly definovány při vývoji ovládací jednotky objektivu. Všechny zprávy mají pevnou délku a strukturu, význam jednotlivých polí je uveden v příloze v tabulce příkazů. Obecná struktura těchto příkazů:

u8 hlavička 4x u8 parametry u8 CRC8

Obecná struktura odpovědí řídicí desky objektivu na příkazy:

u8 hlavička u8 parametr u8 CRC8

Hlavička příkazu na ovládání objektivu nejvýše 0x06. S objektivem také souvisí zpráva řídicí jednotky do centrálního PC pro oznámení chyby během watchdog kontroly ovládací jednotky objektivu.

u8 hlavička = 0x10 u8 CRC8

Příkazy pro řízení polohovacího zařízení

Struktura přeposílání zpráv pro polohovací zařízení byla již demonstrována na obr. 15. Hlavičky těchto zpráv mají hodnotu 0x81 a výše, seznam těchto příkazů je uveden na CD.

Inicializační zpráva je implementována pro účely SW centrálního počítače určeného pro zastaničení kamerových jednotek DDS:

u8 hlavička = 0x80 u8 CRC8

Pří této zprávě posílá ŘJ do polohovacího zařízení zprávu v ascii formátu, protože příkaz nastavení kroku nemá binární ekvivalent. Posílaná zpráva je „a wtq a wpq a ps2000 a ts2000 a ld a “, význam jednotlivých znaků je:

a – čekání na vykonání předchozího příkazu

wtq – nastavení velikosti kroku pohybu v ose náklonu na 1/8

wpq – nastavení velikosti kroku pohybu v ose sklonu na 1/8

ps2000 – nastavení rychlosti otáčení v pan ose na 2000 pozic/s

ts2000 - nastavení rychlosti otáčení v tilt ose na 2000 pozic/s

ld – vypnutí limitů pohybu, čímž se zvýší pohybový rozsah polohovacího zařízení. Absolutní hranice pohybu v obou osách musí být ošetřeny v příkazech centrálního PC – nesmí přijít příkaz, který by se pokoušel nastavit polohu mimo povolené hranice. Při pohybu přes dovolené hranice může dojít k poškození polohovacího zařízení [3].

Následně ŘJ porovná odpověď polohovacího zařízení s očekávanou odpovědí a odešle potvrzení do centrálního PC, ve kterém status nabývá hodnoty 0 při úspěšné inicializaci, jiné hodnoty značí chybu inicializace příkazů.

u8 hlavička = 0x80 u8 status u8 CRC8

Příkazy ovládní řídicí jednotky

Zapnutí a vypnutí přeposílání logovaných dat z GPS modulu do centrálního PC řídí následující zpráva. Podle pole příkaz se řídí funkce příkazu: pro hodnotu 1 se přeposílání zapne, libovolná hodnota rozdílá 1 zařídí vypnutí přeposílání. ŘJ odpovídá zprávou ve stejném tvaru, hodnota v poli příkaz signalizuje nastavenou funkcionalitu (1=zapnuto, !1 = vypnuto).

u8 hlavička = 0x20 u8 příkaz u8 CRC8

Přeposílané zprávy s daty přijatými z GPS modulu mají následující tvar.

u8 hlavička = 0x21 64x u8 data

Pro ověření správného nastavení GPS modulu slouží zpráva

u8 hlavička = 0x22 u8 CRC8

Odpověď ŘJ obsahuje výsledek inicializace, hodnota čísla výsledek je 0 pro úspěšnou inicializaci, jakákoliv jiná hodnota značí neúspěšnou inicializaci.

u8 hlavička = 0x22 u8 výsledek u8 CRC8

Inicializaci GPS modulu lze opakovat následujícím příkazem. Odpověď ŘJ na opakování inicializace je stejná jako pro při dotazu na výsledek inicializaci – zpráva s hlavičkou 0x22.

u8 hlavička = 0x23 u8 CRC8

Zpráva synchronizace kamery s polohovacím zařízením má hlavičku 0x30. Pozice polohovacího zařízení jsou vráceny v počtu kroků, které ještě musí být přepočítány na hodnotu úhlu natočení.

u8 hlavička = 0x30 int16 pan_position int16 tilt_position u8 msg_counter u8 CRC8

Pro nulování msg_counteru použitého pro synchronizaci kamery s polohovacím zařízením slouží následující příkaz, stejnou zprávou odpovídá také ŘJ na provedení příkazu.

u8 hlavička = 0x31 u8 CRC8

Požadavek na odeslání hodnot naměřených modulem inklinometru je určen příkaz

u8 hlavička = 0x40 u8 CRC8

Odpověď ŘJ je zpráva

u8 hlavička = 0x40 float roll float pitch u8 CRC8

Příkaz pro rozsvícení a zhasínání záměrného kříže je závislý na parametru příkaz – 1 pro rozsvícení, ostatní hodnoty zajistí zhasnutí. Stejnou zprávou odpovídá i ŘJ, v poli příkaz je zobrazen aktuální stav záměrného kříže na kamerové jednotce.

u8 hlavička = 0x41 u8 příkaz u8 CRC8

Posledním příkazem je reset řídicí jednotky.

0x64 0x64 0x64 0x64 0x64 0x64 0x64

4. Implementace CRC

Při implementaci CRC pro zajištění komunikace je žádoucí vybrat vhodnou délku a hodnotu generujícího polynomu, druhým požadavkem je rychlost výpočtů CRC pro přijatou či odesílanou zprávu.

Délka generujícího polynomu je určována podle délky zpráv, které mají být zajištěny. Vzhledem k použití RS422 s 8 datovými bity v jednom rámci je logické použití polynomu délky násobku osmi. Nejdelší zpráva v systému má délku 9 bytů, nejčastěji posílané zprávy obsahují maximálně 6 bytů, takže použití CRC s generujícím polynomem delším než 8 bitů by bylo plýtvání → v systému je komunikace zabezpečena pomocí CRC8.

K výběru vhodného generujícího polynomu byla použita data z článku [19]. Správnou volbou generujícího polynomu se minimalizuje pravděpodobnost zamaskování chyby přenosu, kdy pro zprávu přijatou s chybou je výsledek výpočtu CRC stejný jako pro původní zprávu. Vlastnosti různých polynomů pro generování CRC jsou posuzovány podle Hammingovi vzdálenosti (Hamming Distance) – minimálního počtu bitových inverzí ve zprávě, pro které CRC nedokáže chybu detekovat. Například pokud má polynom $HD = 4$ znamená, že neexistuje kombinace chyb v 1, 2 nebo 3 bitech, která by nebyla detekována v CRC. Ale existuje alespoň jedna kombinace čtyř chyb ve zprávě, která bude v CRC zamaskována. Hodnota HD pro polynom je omezena velikostí datové zprávy, na které jsou prováděny výpočty. Jako optimální hodnota osmibitového polynomu je pro zprávy do délky 2048 bitů uváděna hodnota 0x97, pro kterou je v literatuře uváděna $HD = 4$ pro zprávy kratší 120 bitů. V systému je tedy pro výpočet CRC8 zabezpečení používán polynom 0x97.

4.1. Test generujících polynomů CRC

Pro ověření parametrů generujícího polynomu byl napsán testovací program v jazyce C v IDE Code::Block. Testovací program vytváří náhodnou zprávu délky $N=2$ až 9 bytů, spočítá CRC vygenerované zprávy a provede $x = 1$ až 8 bitových inverzí na náhodné pozici ve zprávě. Poté je vypočítáno CRC změněné zprávy a zjišťuje se, zda byla chyba detekována kontrolou CRC. Při prvním testu byla měřena pravděpodobnost chybné detekce polynomu 0x97, každé měření probíhalo na 10000000 vzorcích. Výsledky v tab. 3 ukazují procentuální pravděpodobnost selhání detekce chyby.

		velikost zprávy [byte]							
		2	3	4	5	6	7	8	9
počet chyb ve zprávě	1	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	2	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	3	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	4	0,66	0,77	0,76	0,78	0,79	0,80	0,80	0,80
	5	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	6	0,81	0,79	0,79	0,79	0,78	0,78	0,78	0,78
	7	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	8	0,79	0,78	0,78	0,78	0,78	0,78	0,78	0,78

Tab. 3: Pravděpodobnost (%) nedetekované chyby CRC8 s polynomem 0x97

Pro porovnání bylo stejné měření provedeno s další polynomy používanými ve vestavných systémech – 0xCB a 0xD5. Pro tyto polynomy byla provedena stejná měření jako pro polynom 0x97, výsledky zpracované v tab. 4 a tab. 5 naznačují rozdíl pravděpodobnosti selhání kontroly CRC v porovnání s polynomem 0x97. Modře zbarvené hodnoty značí menší chybovost, červeně je značena větší chybovost CRC polynomu 0xCB/0xD5 oproti 0x97.

		velikost zprávy [byte]							
		2	3	4	5	6	7	8	9
počet chyb ve zprávě	1	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
	2	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
	3	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
	4	-0,109	-0,111	0,012	0,019	0,002	-0,008	-0,012	-0,001
	5	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
	6	0,064	0,045	-0,004	-0,007	-0,002	0,004	0,001	0,001
	7	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
	8	-0,033	-0,017	0,001	0,003	0,002	0,000	-0,002	0,000

Tab. 4: Porovnání pravděpodobnosti (%) nedetekované chyby CRC8 polynomu 0xCB a 0x97

		velikost zprávy [byte]							
		2	3	4	5	6	7	8	9
počet chyb ve zprávě	1	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
	2	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
	3	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	4	0,000	-0,028	0,009	0,001	0,010	0,017	0,020	0,027
	5	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	6	0,053	-0,001	-0,004	-0,003	-0,001	-0,002	-0,001	-0,002
	7	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	8	-0,093	0,005	0,002	0,000	0,000	-0,001	-0,001	0,000

Tab. 5: Porovnání pravděpodobnosti (%) nedetekované chyby CRC8 polynomu 0xD5 a 0x97

Demonstrace nevhodné volby generujícího polynomu je ukázána v tab. 6. Polynom 0xE8 má HD=4 pouze pro zprávy kratší 3 bytů, nedetekuje spolehlivě lichý počet chyb větší než 1 a průměrná chybovost pro sudé počty chyb je vyšší.

		velikost zprávy [byte]							
		2	3	4	5	6	7	8	9
počet chyb ve zprávě	1	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
	2	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
	3	0,000	0,298	0,686	0,079	0,815	0,839	0,830	0,822
	4	0,219	0,000	-0,016	-0,028	-0,014	-0,009	-0,010	0,001
	5	0,937	0,918	0,809	0,786	0,778	0,774	0,776	0,778
	6	-0,110	-0,010	0,007	0,006	0,003	0,004	0,000	0,002
	7	0,841	0,747	0,774	0,780	0,780	0,781	0,781	0,782
	8	0,028	0,005	-0,004	-0,001	-0,001	-0,001	-0,001	0,000

Tab. 6: Porovnání pravděpodobnosti (%) nedetekované chyby polynomu 0x8E a 0x97

Provedená měření ukazují, že pravděpodobnost nedetekované chyby ve zprávě je menší než 1 % a volba nejlepšího generujícího polynomu není jednoznačná. Naopak při volbě nevhodného generujícího polynomu je zhoršení spolehlivosti chyby CRC velmi výrazné.

4.2. Výpočet CRC v mikrokotroléru

Na obr. 16 je na příkladu demonstrován výpočet CRC8 na zprávě 0x0D01 (binárně 0000110100000001) za použití polynomu 0x97 (binárně 110010111).

$$\begin{array}{r}
 0000110100000001 \mid 00000000 \\
 110010111 \\
 \hline
 110111001 \mid 00000000 \\
 110010111 \\
 \hline
 101110 \mid 00000000 \\
 110010 \mid 111 \\
 \hline
 11100 \mid 11100000 \\
 11001 \mid 0111 \\
 \hline
 101 \mid 10010000 \\
 110 \mid 010111 \\
 \hline
 11 \mid 11001100 \\
 11 \mid 0010111 \\
 \hline
 \mid 11100010 = 0xE2
 \end{array}$$

Obr. 16: Demostrace výpočtu CRC8

Přímočarý algoritmus pro výpočet CRC8 je následující (viz obr. 16):

1. K binární podobě zprávy je přidáno 8 nul, nebo 8 jedniček (nemá vliv na funkci nebo spolehlivost výpočtu, závisí pouze na autorovi/použité hodnotě v systému, musí být ale stejné pro obě strany komunikace), které reprezentují zbytek po dělení a na konci výpočtu představují výslednou hodnotu.
2. Postupně se prochází každý bit zprávy od nejvýznamnějšího bitu. Pokud je hodnota aktuálně označeného bitu 1, provede se operace bitového XOR mezi zprávou a generujícím polynomem. To zajistí vynulování aktuálně nejvyššího bitu zprávy.

3. Opakováním postupu z bodu 2 se vynuluje celá původní zpráva, výsledná hodnota se nachází v přidaných osmi bitech.

Alternativní algoritmus pro výpočet CRC8 vhodnější pro programovou implementaci je založen na rozložení zprávy na byty a postupné přičítání (XOR) jednotlivých bytů zprávy do sumy. Po každém přidaném bytu je celá suma procházena bit po bitu (8 bitů = 8 iterací), a pokud je na nejvyšší pozici jednička, je proveden bitový posun doleva o jednu pozici a je proveden bitový XOR s generujícím polynomem. Pokud při iteraci je nejvyšší bit 0, dojde pouze k bitovému posunu doleva o 1 pozici. Po zpracování celé zprávy je výsledná suma hledanou hodnotou CRC. Dobře viditelný je tento postup na ukázce kódu v C (obr. 17). Při výpočtu není v polynomu uveden nejvyšší bit, jeho vynulování je zajištěno bitovým posunem [20].

Výsledek tohoto výpočtu je stejný, jako při výpočtech přímočarým algoritmem.

```
#define POLYNOMIAL 0x97
unsigned char crc8(unsigned char message[], int nBytes)
{
    unsigned char remainder = 0x0;
    unsigned short byte;
    unsigned char bit;

    for (byte = 0; byte < nBytes; ++byte)
    {
        remainder ^= (message[byte]);
        for (bit = 8; bit > 0; --bit)
        {
            if (remainder & 0x80)
            {
                remainder = (remainder << 1) ^ POLYNOMIAL;
            }
            else
            {
                remainder = (remainder << 1);
            }
        }
    }
    return remainder;
}
```

Obr. 17: Ukázka výpočtu CRC8 v C [20]

Velkou nevýhodou tohoto postupu je časová náročnost výpočtu, kdy se prochází každý bit zprávy. Algoritmus je možné zefektivnit využitím vlastnosti výpočtu – pro daný vstup velikosti jeden byte a generující polynom je výsledek vždy stejný. Je tedy možné si připravit tabulku výsledků pro všechny možné vstupy a při výpočtu vynechávat procházení všech bitů a počítat pouze s celými byty. Přiložená

funkce (obr. 18) slouží k vypočtení tabulky. Zde je ale potřeba myslet na úskalí při programování mikrokontroléru – pokud by se funkce prováděla přímo v mikroprocesoru, bude hodnota uložena v paměti RAM, která je velmi malá – STM32F407 má velikost RAM 192 kB. Přestože paměťová náročnost na mikrokontrolér není v tomto projektu příliš vysoká, je správným řešením celou tabulku připravit předem na PC, tabulku hodnot přepsat do programu s klíčovým slovem Const. Tím se zajistí, že překlad uloží tabulku do paměti programu ve flash paměti o velikosti 1 MB.

```
#define POLYNOMIAL 0x97
unsigned char crcTable[256];

void crclnit(void)
{
    unsigned char remainder;
    unsigned short dividend;
    unsigned char bit;

    for (dividend = 0; dividend < 256; ++dividend)
    {
        remainder = dividend;
        for (bit = 8; bit > 0; --bit)
        {
            if (remainder & 0x80)
            {
                remainder = (remainder << 1) ^ POLYNOMIAL;
            }
            else
            {
                remainder = (remainder << 1);
            }
        }
        crcTable[dividend] = remainder;
    }
}
```

Obr. 18: Funkce pro přípravu tabulky CRC8 hodnot [20]

S připravenou tabulkou se pak pro výpočet CRC8 používá funkce `crc8_fast` (obr. 19).

```

#define POLYNOMIAL 0x97
const unsigned char crcTable[256];

unsigned char crc8_fast(unsigned char message[], int nBytes)
{
    unsigned char c=0x00;
    unsigned char i;
    for (i = 0; i < nBytes; ++i){
        c = crcTable[(c ^ message[i]);
    }
    return c;
}

```

Obr. 19: Funkce CRC8_fast [20]

Bylo provedeno měření časové náročnosti obou typů výpočtů na mikrokontroléru STM32F407 taktovaném na 168 MHz. Výsledky měření jsou uvedeny v tab. 7. Měření záměrně končí na délce zprávy 9 bytů, protože to je délka nejdelší zprávy v systému.

počet bytů	crc8_fast [us]	crc8 [us]	ΔT [μs]	zrychlení [%]
1	0,35	0,99	0,64	284,48
2	0,43	1,70	1,27	395,35
3	0,50	2,40	1,90	480,00
4	0,58	3,12	2,54	537,93
5	0,64	3,84	3,20	600,00
6	0,75	4,56	3,81	608,00
7	0,84	5,30	4,46	630,95
8	0,92	6,00	5,08	652,17
9	1,01	6,60	5,59	653,47

Tab. 7: Zrychlení výpočtu CRC8 použitím CRC8_fast

Naměřené hodnoty ukazují výrazné zrychlení použitím metody výpočty s uloženou tabulkou hodnot. Zrychlení proti přímé metodě výpočtů je nejméně 2,8násobné a roste se zvětšující se velikostí zabezpečované zprávy.

5. Realizace komunikačního protokolu v řídicí jednotce

Tato část popisuje návrh a realizaci firmwaru řídicí jednotky. Aplikace běžící na mikrokontroléru STM32F407 jsou napsány v Keil IDE podporující uživatelsky pohodlné rozhraní ST-LINK SWD, které umožňuje programovat a ladit programy v mikrokontrolérech řad STM8 a STM32 [21].

Pro realizaci komunikace v mikroprocesoru řídicí jednotky je důležité zajistit přístup všech periférií komunikujících s centrálním PC ke sdílenému médiu – komunikační rozhraní sériové linky. Je nutné zajistit, aby zprávy od všech modulů na kamerové jednotce přicházely do centrálního počítače i odcházely z ŘJ bez narušení pořadí bytů nebo nedetekované změny hodnoty. Zároveň musí ŘJ zajistit přeposlání komunikace z centrálního PC do cílových periférií (polohovací zařízení, objektiv).

Přístup ke sdílenému komunikačnímu médiu je řešen pomocí front – pouze hodnoty z fronty mohou být odeslány, jinde v programu nemůže být funkce odeslání volána. Odeslání zprávy se tedy realizuje přidáním zprávy do fronty a povolením odesílání dat z fronty. Hodnoty jsou do odesílacích front přidávány pouze v podobě kompletních zpráv. Zaručením, že při přidávání prvků do výstupní fronty nemůže být činnost přerušena jiným přerušením, je zajištěno, že každá zpráva je odeslána kompletní a se správným pořadím bytů.

Každá fronta je v ŘJ realizována jako kruhových buffer osmibitových hodnoty typu char, doplněná dvěma proměnnými s hodnotami pozice začátku a konce fronty a funkcí pro vložení prvku na začátek fronty. Pro odstranění dat z fronty stačí pouze posunutí ukazatele začátku fronty na další pozici, hodnotu v bufferu není třeba přepisovat – před dalším použitím je hodnota na pozici v bufferu inicializována. Jedinou výjimkou je fronta hodnot značící očekávané množství bytů z polohovacího zařízení.

```

#define BUFFER_SIZE 64
u8 buffer[BUFFER_SIZE];
u8 buffer_head=0;
u8 buffer_tail=0;

//vlození prvku na konec fronty
void buffer_enqueue(u8 msg){
    buffer[buffer_tail]=msg;
    buffer_tail++;
    buffer_tail=buffer_tail % BUFFER_SIZE;
}
//čtení první hodnoty z fronty
u8 temp= buffer [buffer_head];
//posun fronty na další prvek
buffer_head++;
buffer_head=buffer_head % BUFFER_SIZE;

```

Obr. 20: Ukázka práce s frontou

Celá komunikace je v ŘJ realizována pomocí přerušení. Využívají se dva typy přerušení – přijetí zprávy ze sériové linky nebo sběrnice CAN a signalizace klidové úrovně vysílání na sériové lince – možnost zahájení přenosu dat.

Jednodušší, a pro všechny sériové linky stejná, je rutina odesílání dat z výstupních bufferů při přerušení (obr. 21). Při odesílání se funkci pro poslání jednoho bytu předávají hodnoty ze začátku odesílací fronty, následuje posunutí fronty a po dokončení odeslání bytu se celý proces opakuje. Odesílání dat je ukončeno ve chvíli, kdy se hodnoty pozic začátku a konce fronty rovnají – všechny hodnoty ve frontě byly odeslány.


```

#define SEND_BUFFER_SIZE 64
u8 send_buffer[SEND_BUFFER_SIZE];
u8 send_buffer_head=0;
u8 send_buffer_tail=0;

void USARTx_IRQHandler(void)
{
    if( USART_GetITStatus( USARTx, USART_IT_TXE) != RESET){
        USART_SendData( USARTx, send_buffer[send_buffer_head]);
        send_buffer_head++;
        send_buffer_head=send_buffer_head % SEND_BUFFER_SIZE;

        if(send_buffer_head==send_buffer_tail){
            USART_ITConfig( USARTx, USART_IT_TXE, DISABLE);
        }
    }
}

```

Obr. 21: Odesílání dat po sériové lince

Před odesláním dat je nutné naplnit odesílací frontu. Příklad z obr. 22 demonstruje odeslání zprávy 0x0D01 přes USARTx. Poslední příkaz „*USART_ITConfig(USARTx, USART_IT_TXE, ENABLE);*“ slouží k povolení odesílání dat v přerušení pro USARTx.

```

send_buffer_enqueue(0x0D);
send_buffer_enqueue(0x01);
USART_ITConfig( USARTx, USART_IT_TXE, ENABLE);

```

Obr. 22: Vložení prvku do odesílací fronty a povolení odesílání

Realizace obsluhy přerušení pro příjem dat z centrálního počítače a okolních periférií kamerové jednotky je mnohem složitější než odesílání dat a implementace odesílání v ŘJ není pro všechny periférie stejná. Společný je pouze základ v podobě vyčtení bytu přijatého po sériové lince v přerušení (obr. 23), další zpracování přijaté zprávy se liší v závislosti na připojené periférii.

```

void USARTx_IRQHandler(void)
{
    if (USART_GetITStatus(USARTx, USART_IT_RXNE) == SET) {
        usartx_recieved=USART_ReceiveData(USARTx) & 0xFF;
        //operace s přijatým bytem....

        //potvrzení obslužení přerušení
        USART_ClearITPendingBit(USARTx, USART_IT_RXNE);
    }
}

```

Obr. 23: Přijetí bytů po sériové lince v přerušení

Přehled všech komunikačních rozhraní ŘJ včetně parametrů je uveden v tab. 8.

Komunikující strany		Typ rozhraní	Název periferie procesoru ŘJ	parametry	
Centrální PC	↔	Řídicí jednotka	RS422	USART6	115200 Bd, 8 datových bitů, 1 stop bit, bez parity, bez HW řízení
Řídicí jednotka	↔	Polohovací zařízení	RS232	USART2	9600 Bd, 8 datových bitů, 1 stop bit, bez parity, bez HW řízení
Řídicí jednotka	↔	GPS modul	RS232	USART1	115200 Bd, 8 datových bitů, 1 stop bit, bez parity, bez HW řízení
Řídicí jednotka	↔	Objektiv	RS232	USART3	9600 Bd, 8 datových bitů, 1 stop bit, bez parity, bez HW řízení
Řídicí jednotka	←	Modul inklinometru	CAN bus	CAN1	1Mb/s

Tab. 8: Přehled komunikačních rozhraní ŘJ

5.1. Realizace komunikace centrální PC <-> ŘJ

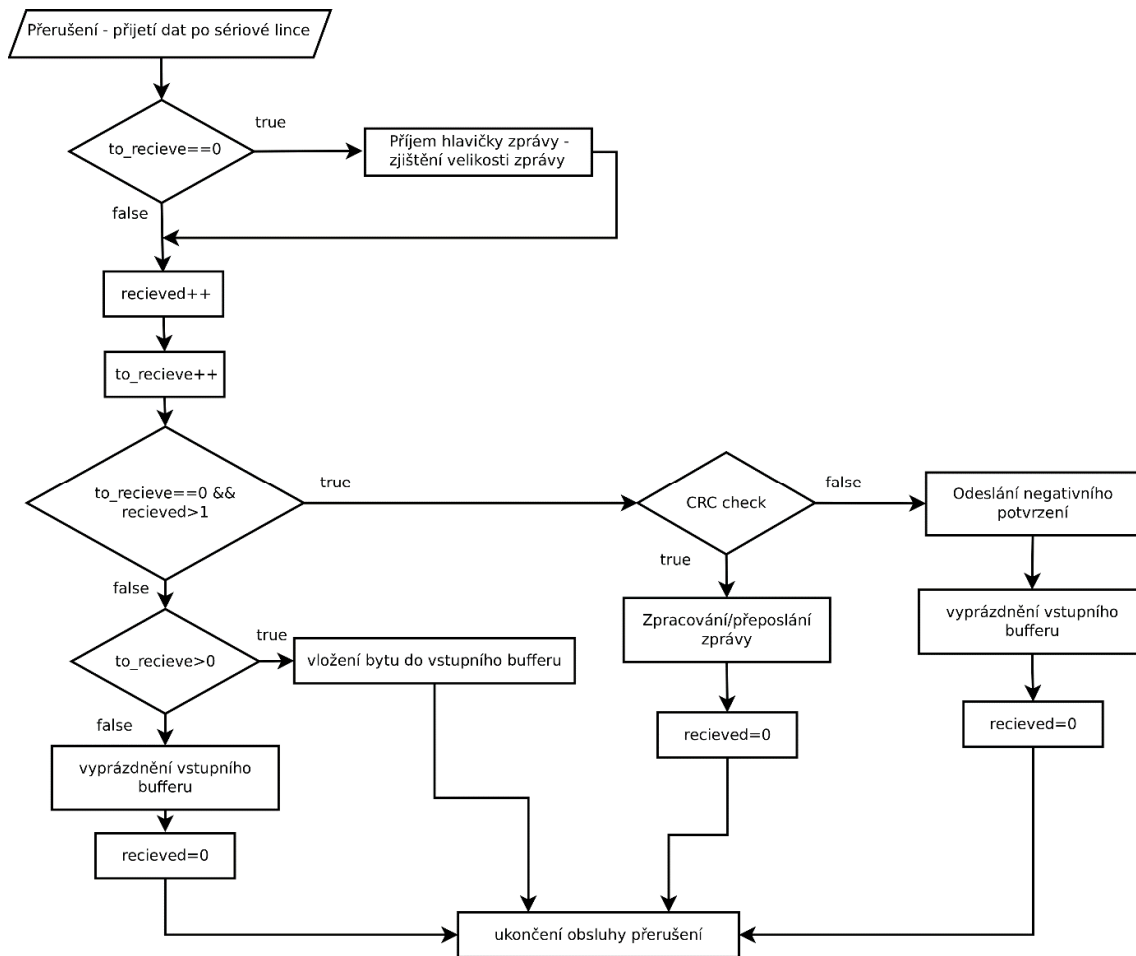
Řízení komunikace je založeno na znalosti délky příchozích zpráv – příjmem prvního bytu je určen jak význam zprávy a případný adresát pro přeposlání, ale také zbývající počet bytů. Každá zpráva definovaná v systému je rozlišena unikátní hodnotou 8bitové hlavičky, délka každé zprávy je pevně daná. Přijetím hlavičky se zjistí, kolik bytů chybí do přijetí celé zprávy.

Operace s přijatou zprávou se provádí až po jejím kompletním přijetí. U zprávy je zkontrolováno CRC, a v případě detekované chyby přenosu je odesláno negativní potvrzení složené z hlavičky příkazu a hodnoty 0xFF, chybně přijatá zpráva je následně zahozena. Pokud je přijetí zprávy vyhodnoceno jako bezchybné, je přijatá zpráva vyhodnocena.

Proměnné v programu ŘJ pro realizaci komunikace:

u8 recieved – vzestupný čítač počtu přijatých bytů, po přijetí a zpracování kompletní zprávy vynulován

u8 to_recieve – sestupný čítač bytů zbývajících do přijetí celé zprávy, hodnota této proměnné je nastavena při příjmu hlavičky zprávy



Obr. 24: Diagram příjmu zpráv z centrálního PC

5.2. Přeposílání příkazů do polohovacího zařízení

Jak bylo již naznačeno na obr. 15, komunikační rozhraní polohovacího zařízení nepodporuje CRC zabezpečení komunikace, CRC obsahují tedy pouze zprávy mezi centrálním PC a ŘJ. ŘJ navíc zajišťuje odeslání hlavičky příkazu při odesílání výsledku operace z polohovacího zařízení do centrálního PC. Celé je to realizované pomocí přidanych front:

panilt_send_buffer – fronta pro odesílání dat do polohovacího zařízení, hodnoty jsou do této fronty přidávány při úspěšném přijetí příkazu určeného pro polohovací zařízení z centrálního PC.

panilt_rec_size_buffer – při zadávání hodnot do odesílacího bufferu polohovacího zařízení je do této fronty přidána hodnota očekávané délky odpovědi polohovacího zařízení na poslední přidany příkaz v *panilt_send_bufferu*. Tato fronta je při posouvání na další prvek nulována, čímž se zajistí nepřijímání nežádoucích asynchronních zpráv polohovacího zařízení (oznámení o dojezdu do krajní polohy jsou polohovacím zařízením posílána, přestože jsou vypnuty limity a krajní hodnoty natočení jsou hlídány programem centrálního PC).

pantilt_rec_type_buffer – v této frontě je uložena hlavička posledního příkazu odeslaného do polohovacího zařízení. Podle hodnoty hlavičky z této fronty je zpracována odpověď z polohovacího zařízení. Při odpovědi na příkaz z centrálního PC je hlavička přidána na začátek zprávy obsahující odpověď polohovacího zařízení posílanou zpět na centrální stanoviště (obr. 15). Další využití má tato fronta pro synchronizaci kamery s polohovacího zařízení.

pantilt_recieved_buffer – fronta přijatých bytů z polohovacího zařízení

u8 pantilt_recieved_count – vzestupný čítač počtu přijatých bytů, po přijetí a zpracování kompletní zprávy vynulován

5.3. Přeposílání příkazů do objektivu

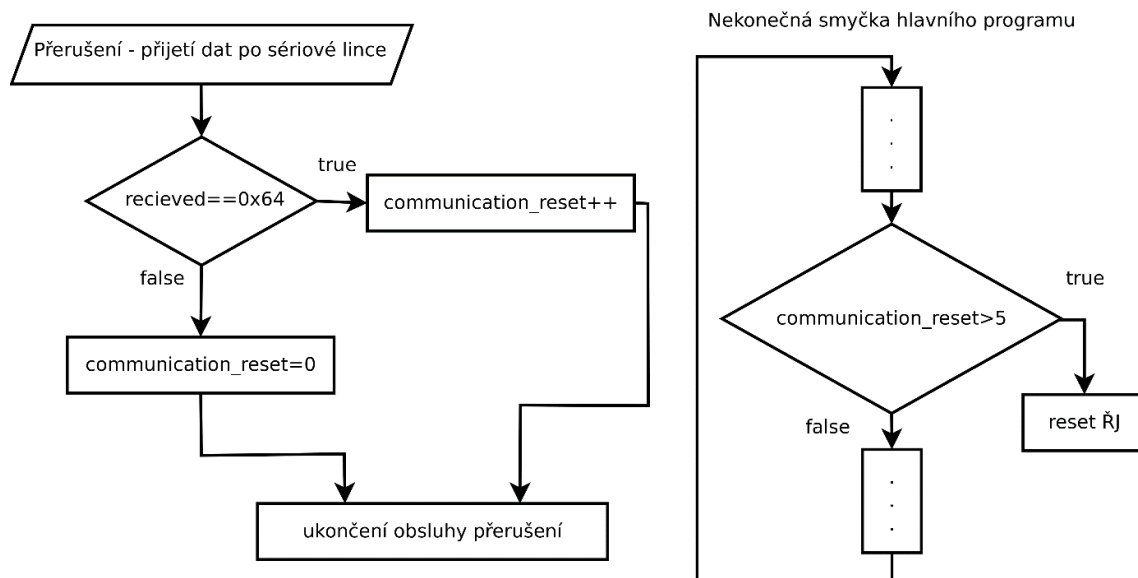
Kontrolní jednotka pro ovládání objektivu je uzpůsobena pro komunikaci s centrálním PC i se zabezpečením CRC kódem. Na straně ŘJ stačí pouze hlídat přijetí kompletní zprávy od centrálního PC a přeposlat do objektivu. Stejně se zachází i s odpověďmi objektivu, které jsou bez úprav přeposílány do centrálního PC.

5.4. Kontrola činnosti objektivu

Řídicí jednotka posílá s použitím časovače TIM3 s periodou 10 Hz dotazy do ovládací jednotky objektivu kontrolní zprávu typu watchdog – definovanou odpovědí potvrdí ovládací jednotka objektivu správnou funkčnost. Při odeslání každé kontrolní zprávy do objektivu je inkrementován čítač počtu dotazů, který je nulován při obdržení odpovědi na kontrolní zprávu od ovládací jednotky objektivu. V přerušení časovače TIM3 je také kontrolován stav odpovědi na kontrolní zprávy – pokud ŘJ nedostane odpověď na tři po sobě následující kontrolní zprávy, je ovládací jednotka objektivu resetována změnou úrovně GPIO pinu ŘJ, který je připojen na pin NRST na mikrokontroléru jednotky objektivu. Celá událost je ohlášena centrálnímu PC.

5.5. Reset ŘJ

Reset ŘJ pomocí komunikace z centrálního PC je implementován z důvodu možnosti uváznutí komunikace v důsledku chyby. Provedení resetu tedy nemůže být realizováno pomocí standardních příkazů vyhodnocených podle hlavičky vyžadujících synchronizaci, ale je provedeno bez použití čítačů přijatých bytů a zbývajících bytů. Jak je ilustrováno na obr. 25, při přijetí každého bytu s hodnotou 0x64 z centrálního PC je inkrementována hodnota čítače pro reset ŘJ, při přijetí bytu jiné hodnoty nebo resetování celé jednotky je vynulován čítač pro reset ŘJ. Ve smyčce hlavního programu je kontrolována hodnota čítače resetu ŘJ. Pokud hodnota čítače překročí hodnotu 5, je ŘJ resetována.

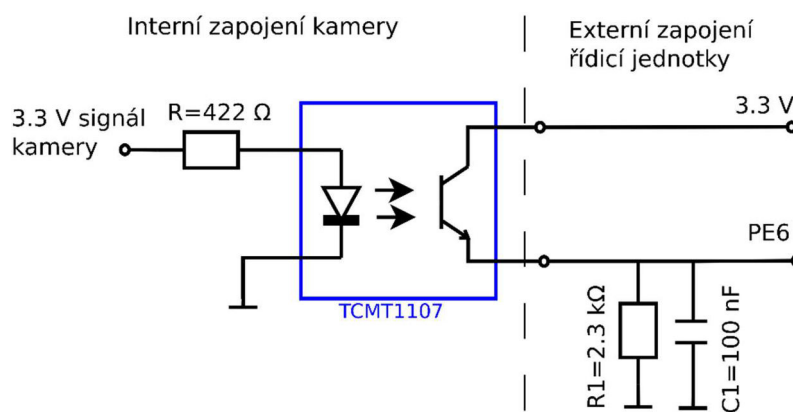


Obr. 25: Diagram obsluhy resetu ŘJ

Samotné resetování ŘJ je složené z více kroků – do polohovacího zařízení je odeslán příkaz obsahující oddělovač – podle [3] slouží k vyhodnocení přijatých hodnot. Při resetu existuje možnost, že byly do polohovacího zařízení odeslány nedefinované hodnoty, které brání dalšímu vyhodnocení. Oddělovačem jsou nedefinované hodnoty v řízení polohovacího zařízení vyhodnoceny jako neplatné a jsou zahozeny, a po resetu je možné znovu posílat příkazy. Druhou operací před samotným resetem je uložení aktuálních hodnot náklonu a sklonu z inklinometru do paměti flash.

5.6. Synchronizace kamery a polohovacího zařízení

Pro synchronizaci nastaví program centrálního PC kameru tak, aby při expozici každého snímku aktivovala GPIO pin, kterým je vyvoláno přerušení ŘJ. Použit je opticky oddělený pin kamery. Celé zapojení obvodu pro propojení GPIO pinu kamery a řídicí jednotky je znázorněno na obr. 26.



Obr. 26: Propojení GPIO pinů kamery a ŘJ

V obsluze přerušení pinu PE6 ŘJ vloží do *pantilt_send_bufferu* dotazy na aktuální pozici polohovacího zařízení v obou osách. Využije se při tom vlastností zpracování popsaných v části pro přeposílání zpráv z centrálního PC do polohovacího zařízení a do fronty *pantilt_rec_type_buffer* jsou pro příkazy vloženy jiné hodnoty hlaviček, než mají dotazy na polohu určené k přeposlání do centrálního PC. Tím se zajistí odchycení odpovědí polohovacího zařízení na oba dotazy, hodnoty polohy jsou poté odeslány do centrálního PC zabalené do zprávy společně s čítačem synchronizačních zpráv. Podoba zprávy je popsána v kapitole 3.1 - Příkazy centrálního počítače pro ovládání kamerové jednotky.

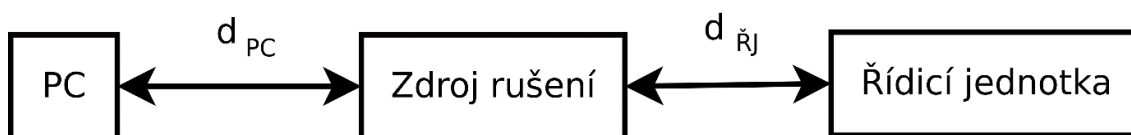
6. Spolehlivost RS422 komunikace

Pro praktické ověření spolehlivosti komunikace byly provedeny testy podle normy ČSN EN 61000-4-4 (ekvivalent mezinárodní normy řady IEC 1000-4-4). Tato norma definuje podmínky a nastavení systému při měření, během kterého je vodič komunikační linky uzavřen v kapacitní vazební svorkovnici délky 1 metr (obr. 27), ve které je generátorem vytvářeno rušení. Podle normy má rušení charakter rychlých shluků impulsů (burstů) s malou energií – shluk impulsů trvá 15 ms, frekvence impulsu ve shluku je 5 kHz a shluky rušivých impulsů se opakují s frekvencí 3 Hz. Podle velikosti napětí impulsů se posuzují parametry měřených zařízení a je provedeno zařazení do úrovně odolnosti.



Obr. 27: Kapacitní vazební svorkovnice Haefely [22]

Na obr. 28 je uvedeno blokové schéma zapojení při měření spolehlivosti komunikace. Pro spojení ŘJ a PC byl použit kabel FTP cat.5e o délce 100 m, dvě dvojice kroucených párů jsou použity pro datové linky, zbylé dva páry jsou uzemněny. Na výsledky měření má velký vliv umístění vazební svorkovnice (d_{PC} a $d_{ŘJ}$). To je zapříčiněno velkou délkou kabelu, který tlumí rušení a při délce cca 98 m od přijímače ke zdroji rušení je již vliv rušení přijímanou komunikací nepatrný. Výsledky měření jsou tedy rozděleny na případ, kdy je u vazební svorkovnice umístěna blízko ŘJ ($d_{ŘJ} \sim 1\text{m}$), při druhém experimentu je vazební svorkovnice v blízkosti PC ($d_{PC} \sim 1\text{m}$). V rámci těchto měření byl i testován vliv galvanického oddělení komunikačního rozhraní v ŘJ – k dispozici pro měření byl jak běžný modul bez galvanického oddělení, tak i modul s pokročilejším návrhem obsahující galvanické oddělení.



Obr. 28: Schéma umístění zdroje rušení komunikace při měření spolehlivosti komunikace

Pro měření je upraven jak firmware ŘJ, tak je i v Microsoft Visual Studiu vytvořen v jazyce C# obsluhující program na PC se zobrazením výsledků. Při testování je opakovaně posílána stejná zpráva ve tvaru

```
u8 hlavička = 0x22  int32 data1= 12345  int32 data2= -6789  u8 CRC8=0xAB
```

Komunikace je stejně jako v původním programu ŘJ synchronizována na hlavičku zprávy. Testovací program PC a ŘJ funguje ve dvou režimech:

- V prvním režimu je odesílatelem dat PC, ŘJ je příjemcem. Při této konfiguraci odešle PC na začátku experimentu do ŘJ zprávu o začátku měření, poté je periodicky posílána testovací zpráva z PC do ŘJ. Na konci měření pošle PC do ŘJ zprávu o ukončení měření, na kterou z ŘJ je odeslána statistika měření do PC, který statistiku zobrazuje.
- Ve druhém režimu je odesílatelem testovací zprávy ŘJ. PC určuje začátek i konec vysílání, na konci vysílání je z ŘJ odeslána informace o počtu přijatých zpráv a celá statistika měření je zobrazena.

Při měření jsou pozorovány následující statistiky – počet odeslaných zpráv, celkový počet přijatých zpráv, počet přijatých zpráv bez chyby přenosu, počet zpráv s chybným CRC, počet chyb komunikace detekovaných CRC zabezpečením a počet chybných zpráv nedetekovaných CRC kódem.

Rušení komunikace u ŘJ

Umístění kapacitní vazební svorkovnice s rušivým signálem do blízkosti ŘJ ($d_{\text{ŘJ}} \sim 1\text{m}$) mělo za následek, že při překročení určité hranice velikosti rušivého signálu docházelo k pravidelnému resetování ŘJ. Při pokusech se nepotvrdil teoretický předpoklad zlepšení použitím galvanického oddělení komunikačního modulu ŘJ, naopak pro tento modul byly naměřené výsledky horší, což může být způsobeno špatným hardwarovým návrhem. Resety ŘJ nastávaly při napětí rušivých impulsů 0,9 kV, při použití galvanicky oddělených modulů bylo naměřené napětí pouze 0,7 kV.

Byla provedena měření, při kterých se sledovaly statistiky chybovosti přenosu zpráv pro nejvyšší úroveň rušení, pro kterou nedochází k resetům ŘJ. V tab. 9 jsou uvedeny výsledky měření s běžným modulem při úrovni rušení 0,8 kV, v tab. 10 jsou naměřené výsledky pro galvanicky oddělený modul při rušení o velikosti 0,6 kV.

Odeslané	Přijaté celkem	Bezchybně přijaté zprávy	Chybné CRC	Detekovaná chyba	Nedetekovaná chyba
130718	130714	130711	3	3	0
132944	132941	132989	2	2	0
133183	133182	133181	1	0	0

Tab. 9: Statistika přenosu zpráv z PC do ŘJ, modul bez galvanického oddělení, rušení 0,8 kV

Odeslané	Přijaté celkem	Bezchybně přijaté zprávy	Chybné CRC	Detekovaná chyba	Nedetekovaná chyba
141881	141877	141873	4	3	0
132251	132250	132249	1	1	0
130709	130707	130705	2	2	0

Tab. 10: Statistika přenosu zpráv z PC do ŘJ, modul s galvanickým oddělením, rušení 0,8 kV

Ze statistik měření v tab. 9 a 10 lze vypožorovat, že při přenosu dochází ke ztrátě dat (odeslané > přijaté). To je způsobeno chybou příjmu hlavičky zprávy, na kterou se komunikace synchronizuje a následným zahazování bytů do další hlavičky. Během sledované komunikace nedošlo k žádné chybě, kterou by CRC nedokázalo detekovat. V tab. 9 je ve třetí řádce detekovaná jedna zpráva s chybným CRC, ale detekované i nedetekované chyby ukazují nulové hodnoty. To označuje situaci, kdy byly správně přeneseny všechny datové bity zprávy až na kontrolní součet CRC, ve kterém došlo k chybě přenosu.

Statistiky měření komunikace ve směru od ŘJ do PC jsou uvedeny v tab. 11 a 12.

Odeslané	Přijaté celkem	Bezchybně přijaté zprávy	Chybné CRC	Detekovaná chyba	Nedetekovaná chyba
150032	150032	150032	0	0	0
114422	114422	114422	0	0	0
128316	128316	128316	0	0	0

Tab. 11: Statistika přenosu zpráv z ŘJ do PC, modul bez galvanického oddělení, rušení 0,8 kV

Odeslané	Přijaté celkem	Bezchybně přijaté zprávy	Chybné CRC	Detekovaná chyba	Nedetekovaná chyba
124913	124913	124913	0	0	0
127989	127989	127989	0	0	0
125614	125614	125614	0	0	0

Tab. 12: Statistika přenosu zpráv z ŘJ do PC, modul s galvanickým oddělením, rušení 0,6 kV

Z hodnot v tab. 11 a 12 je pozorovatelné, že rušení je na dlouhém úseku kabeláže vyfiltrováno, všechny zprávy byly přeneseny bezchybně.

Rušení komunikace u PC

Při umístění kapacitní vazební svorkovnice s rušivým signálem do blízkosti PC ($d_{PC} \sim 1m$) přestalo kvůli velké vzdálenosti ŘJ od rušení ($d_{RJ} \sim 98m$) docházet k resetování ŘJ. Tato konfigurace je měřena pro zjištění odolnosti komunikace na straně PC, tedy odolnosti převodníku USB-COM422-PLUS4. Při testování byla určena hranice 1,5 kV, do které systém fungoval spolehlivě, při vyšších hodnotách docházelo k chybě převodníku, který přestal komunikovat s PC. Při rušení o velikosti 2,5 kV došlo dokonce k zamrznutí PC, které vyžadovalo restart.

Odeslané	Přijaté celkem	Bezchybně přijaté zprávy	Chybné CRC	Detekovaná chyba	Nedetekovaná chyba
131950	131950	131950	0	0	0
130196	130196	130196	0	0	0
130558	130558	130558	0	0	0

Tab. 13: Statistika přenosu zpráv z PC do ŘJ, modul bez galvanického oddělení, rušení 1,5 kV

Odeslané	Přijaté celkem	Bezchybně přijaté zprávy	Chybné CRC	Detekovaná chyba	Nedetekovaná chyba
133338	133338	133338	0	0	0
130728	130728	130728	0	0	0
131206	131206	131206	0	0	0

Tab. 14: Statistika přenosu zpráv z ŘJ do PC, modul bez galvanického oddělení, rušení 1,5 kV

Výsledky měření (tab. 13 a 14) ukazují očekávanou bezchybnou komunikaci ve směru od PC do ŘJ. Nulová chybovost komunikace v opačném směru značí kvalitnější HW provedení převodníku v porovnání s HW řídicí jednotky.

Zhodnocení měření odolnosti komunikace

System je v aktuální HW podobě vhodný pouze do laboratorních podmínek, pro použití i např. v průmyslovém prostředí je potřeba vylepšeného návrhu komunikačního rozhraní RS422 na straně řídicí jednotky, zvláštní pozornost by měla být věnována kvalitnímu galvanickému oddělení. Zajištění komunikace CRC zabezpečovacím kódem spolehlivě detekuje případné chyby komunikace.

7. Diferenční GPS

Při zastaničení kamerových jednotek DDS se měřením GPS a zpracováním v RTKLIB zjišťuje orientovaná vzdálenost mezi kamerovými jednotkami. Postup zjištění polohy je následující:

- Nastavení parametrů modulu
- Na příkaz centrálního PC přeposílá ŘJ zprávy z GPS modulu do centrálního PC, které ukládá přijaté zprávy do souboru
- Po ukončení měření je spuštěn batch skript (skript pro vykonání v příkazové řádce systémů MS Windows) pro volání funkcí RTKLIB, kterými jsou zpracovávána naměřená data. Výstupem skriptu jsou soubory s absolutní pozicí kamerové jednotky jedna a diferencemi mezi jednotlivými kamerovými jednotkami.
- Výstupní data z RTKLIB obsahují vypočítanou pozici/vzdálenost pro každý vzorek měření. Pro získání jediné hodnoty jsou tato data matematicky zpracována měřením směrodatné odchylky a průměrováním.

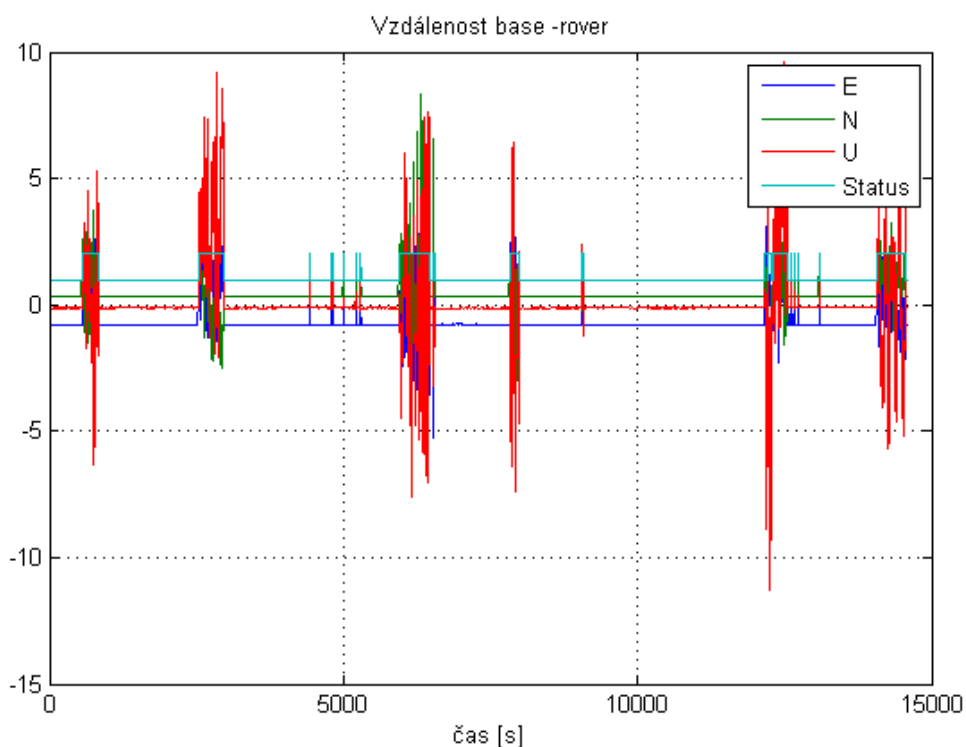
Ve skriptu pro zpracování GPS dat jsou použity dvě funkce z RTKLIB:

- `convbin` – funkce pro převod dat z formátu UBX do formátu RINEX. RINEX (Receiver Independent Exchange Format) je univerzální formát pro uchovávání dat z GNSS (globální navigační satelitní systém). Použitými výstupy z funkce `convbin` jsou soubory s koncovkami `obs` (observation data – časové údaje a parametry signálu z družice), `nav` (navigation data – časové korekce) a `sbs` (sbas korekce).
- `rnx2rtkp` – funkce pro analýzu a vyhodnocení pozic z naměřených dat ve formátu RINEX (postprocessing). Byl použit režim `moving-base`, kterým je zjišťována orientovaná vzdálenost mezi dvěma stanicemi – anténami GPS přijímačů, které se během měření mohou i pohybovat. Zpracování dat je závislé na mnoha parametrech, při experimentech byly měněny následující parametry:
 - Elevation mask (10°, 15°, 20°)
 - integer ambiguity (Fix and Hold, Continuous) – vyhodnocení nejednoznačnosti fázového měření
 - min ratio to fix ambiguity (2.0 až 5.0)

Funkce pro zpracování GPS dat z RTKLIB jsou náročné na výpočetní výkon, každá spuštěná instance dokáže využít pouze jedno výpočetní jádro procesoru PC, na kterém jsou výpočty spuštěny. Trvání výpočtů lze výrazně zrychlit paralelním spouštěním programů pro zpracování. Paralelně je spouštěno

zpracování dat do formátu RINEX aplikací convbin pro data ze všech 4 kamerových jednotek. Po dokončení zpracování posledního jsou také paralelně spuštěny aplikace rnx2rtkp pro požadované dvojice kamerových jednotek. Paralelní volání příkazů a čekání na dokončení jednotlivých příkazů je popsáno v [23].

Příklad výsledku zpracování aplikaci rnx2rtkp je uveden na obr. 29, kde jsou v metrech znázorněny rozdíly pozice GPS antén v orientaci na sever (N), orientaci na východ (E) a nadmořské výšce (U). Status je ukazatel z RTKLIB označující určení celočíselné nejednoznačnosti – Ambiguity Resolution (1=FIX, 2=FLOAT).

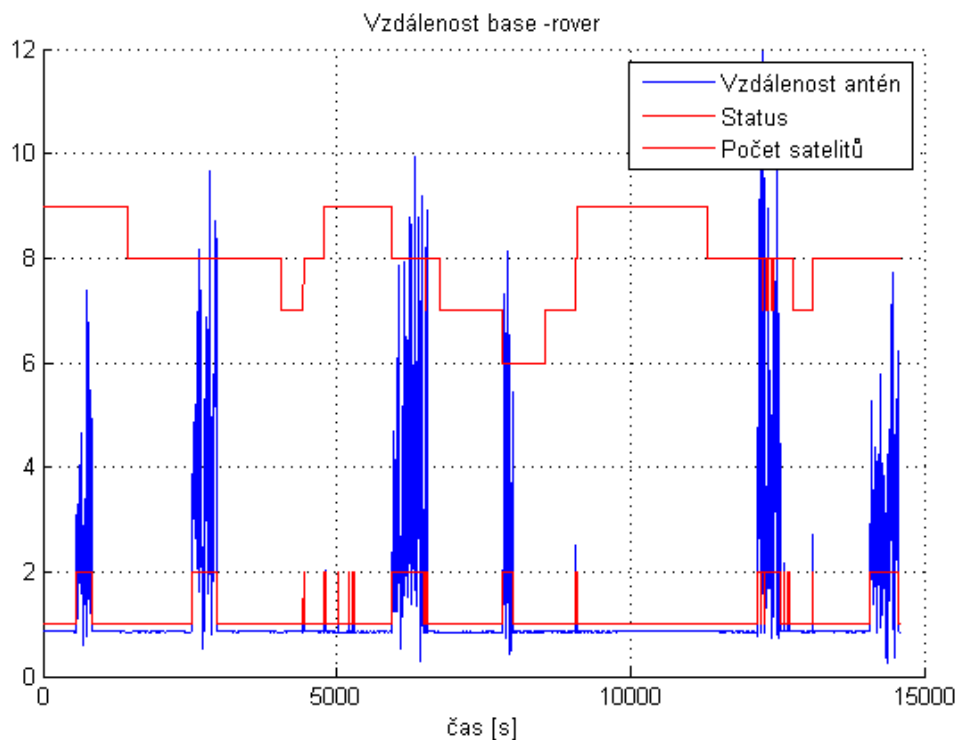


Obr. 29: Vizualizace vzdáleností dvou GPS antén po přepočtu programy RTKLIB

Šikmá vzdálenost mezi anténami je vypočítána jako velikost vektoru

$$vzdálenost = \sqrt{E^2 + N^2 + U^2} \text{ [m]}$$

Vzdálenost mezi anténami pro každý čas měření z obr. 29 je znázorněna na obr. 30.



Obr. 30: Vizualizace vypočtené vzdálenosti GPS antén z diferenciálního měření

Pro další zpracování potřebujeme z měření získat jedinou hodnotu N, E a U. Z obrázku je zřejmé, že naměřená data jsou platná, pouze pokud je status roven 1. Při zpracování jsou tedy nejprve zahazována všechna data, pro která je status různý od 1, poté je pro každý čas, který je od konce měření vzdálen více než 10 sekund, vypočítána směrodatná odchylka pro data délky 10 sekund. Výsledkem je aritmetický průměr všech hodnot, u kterých byla směrodatná odchylka nižší než prahová hodnota 0.002. Tento postup je proveden pro N, E i U. Výpočty je snížena vyhodnocená doba měření o 10 vteřin.

Směrodatná odchylka ze sto vzorků pro j-tý prvek je definována následovně

$$s_j = \sqrt{\frac{1}{100} \left(\sum_{i=j}^{j+100} x_i^2 + N \bar{x}_j^2 \right)} \quad , j \in \langle 1; N - 100 \rangle$$

, kde N je celkový počet naměřených vzorků indexovaný od jedné do N. Pro tyto účely musel být upraven i výpočet střední hodnoty, který je prováděn pouze pro 100 následujících hodnot

$$\bar{x}_j = \frac{1}{N} \sum_{i=j}^{j+100} x_i$$

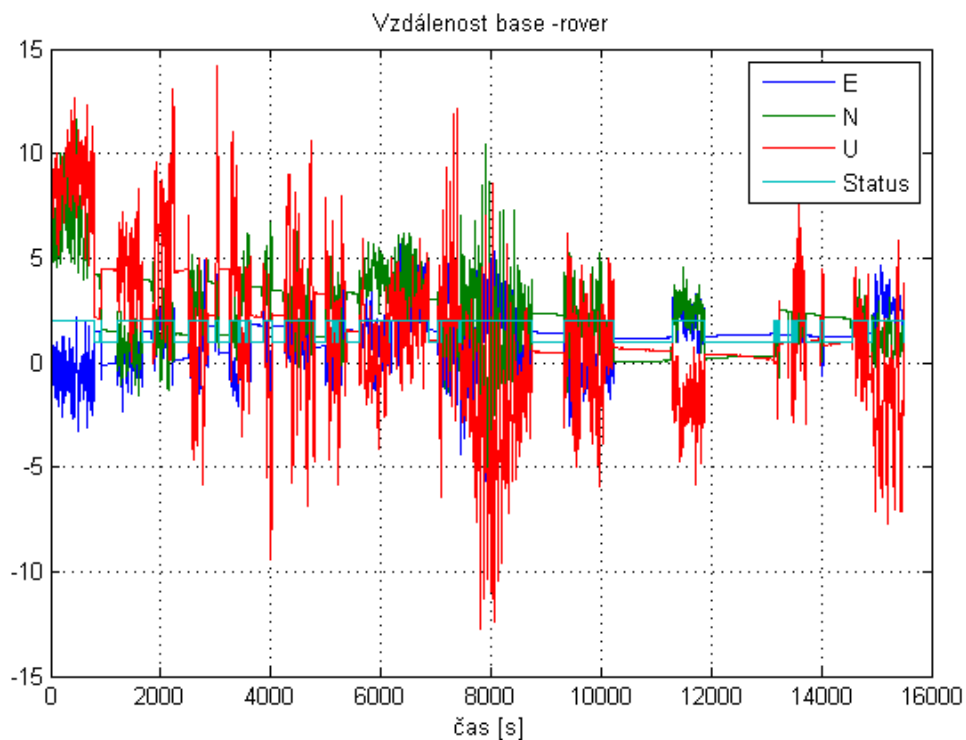
Úhel mezi spojnicí měřených GPS antén a zeměpisným severem (naznačeno na obr. 11) je vypočten vzorcem

$$\alpha = \text{atan2}(N, E) \text{ [rad]}$$

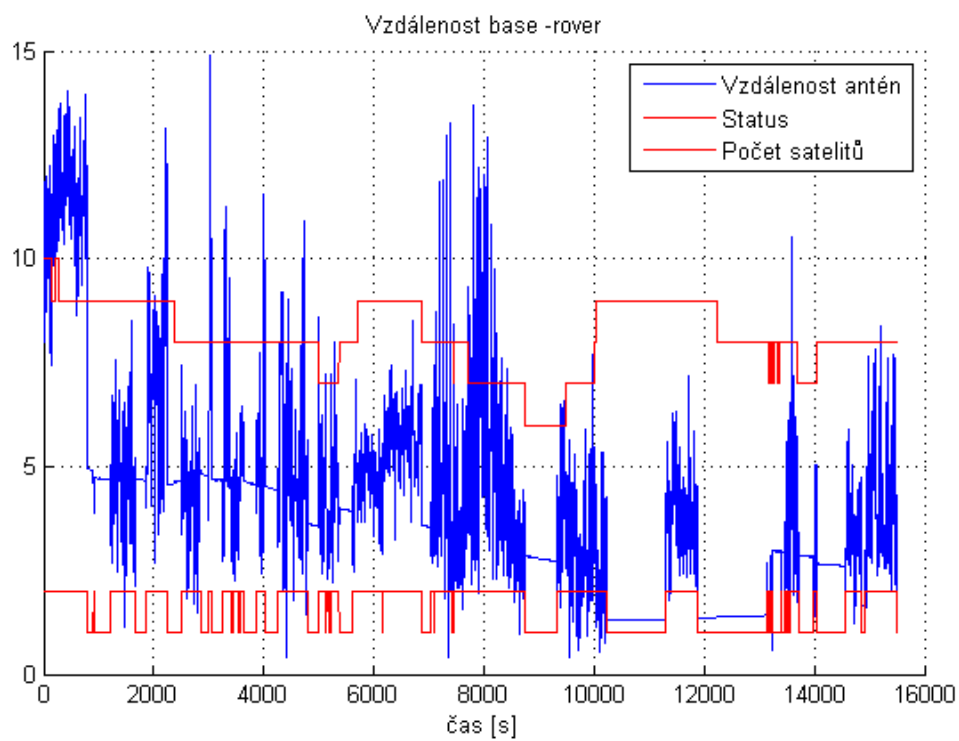
, funkce atan2 vypočítává inverzní tangens s ohledem na kvadrát. Je implementován ve vyšších programovacích jazycích (obsažena v Matlabu). Funkce atan2 je definována následujícím vzorcem

$$\text{atan2}(x, y) = \begin{cases} \arctan \frac{x}{y} & y > 0 \\ \arctan \frac{x}{y} + \pi & x \geq 0, y < 0 \\ \arctan \frac{x}{y} - \pi & x < 0, y < 0 \\ +\frac{\pi}{2} & x < 0, y = 0 \\ -\frac{\pi}{2} & x > 0, y = 0 \\ \text{nedefinované} & x = 0, y = 0 \end{cases}$$

Velkým problémem této metody je ale nestabilita řešení – data z obr. 29 a 31 pochází z jiných časových úseků stejného měření. A zatímco z dat na obr. 29 lze spolehlivě zjistit vzájemnou vzdálenost antén s chybou maximálně 2 cm, data z obr. 31 vykazují fix pouze v 41 % měřeného času (oproti 84 % z dat na obr. 28), navíc se v různých časech hodnota E, N i U fixuje na rozdílných hodnotách, které „plavou“ v řádu desítek centimetrů. Přepočtené vzdálenosti dat z obr. 30 jsou zobrazeny na obr. 32.



Obr. 31: Vizualizace vzdáleností dvou GPS antén po přepočtu programy RTKLIB



Obr. 32: Vizualizace vypočtené vzdálenosti GPS antén z diferenciálního měření

Při měření se nepodařilo najít příčinu této nestability výsledků. Možné důvody mohou být nedostatečně kvalitní antény, chybná metoda zpracování, či negativní vlivy na měření GPS dat (měření v zarušeném prostředí, vliv počasí, nízký počet viditelných satelitů v různé fázi dne apod.).

Pro použitelnost této metody je potřeba spolehlivé určení či garanci, že jsou naměřená data platná. Dokud toto nebude zajištěno, nelze popsáním měřením a zpracováním diferencíální GPS měřit vzdálenost dvou bodů.

8. Závěr

Diplomová práce se skládá ze šesti částí. V první části je teoreticky představen kamerový systém, druhá část se zabývá návrhem komunikace v kamerovém systému, ve třetí části bylo rozebráno použití CRC zabezpečení komunikace v mikrokontrolérech, čtvrtá část popisuje realizaci firmwaru řídicí jednotky. V páté části je popsáno měření spolehlivosti komunikace, šestá část popisuje použití systému GPS pro přesné měření orientovaných vzdáleností.

Teoretická část se zabývá popisem struktury, hardwarových komponent a základních pracovních kamerového systému pro sledování cílů, je popsán mikrokontrolér STM32F407 a komunikační rozhraní použitá v systému. Závěr této části je věnován základnímu popisu systému GPS.

Ve druhé části jsou rozebrány komunikační nároky všech periférií kamerových jednotek, je popsán obsah komunikace a jsou navrženy a popsány příkazy pro ovládání systému včetně zabezpečení komunikace.

Třetí část se zabývá volbou a testy generujícího polynomu pro CRC, v této části je naznačena realizace výpočtů CRC v mikrokontroléru a možnosti zrychlení výpočtů. Snížení časových nároků výpočtů je prakticky ověřeno.

Čtvrtá část detailně popisuje softwarovou realizaci řídicí jednotky kamerového systému. V této části je realizován návrh komunikační části z druhé kapitoly. V páté části práce je prakticky otestována spolehlivost komunikační linky RS422, je popsána metodika měření, prezentovány a vyhodnoceny naměřené výsledky.

Šestá část se zabývá metodou přesného měření orientované vzdálenosti dvou bodů použitím systému GPS s využitím knihovny RTKLIB s chybou měření < 1 cm. Jsou představeny používané programy a metody zautomatizování výpočtů.

V práci byl navržen SW mikrokontroléru poskytující komunikační rozhraní pro ovládání periférií kamerového pozičního systému z PC. Komunikace mezi mikrokontrolérem a PC používá navrženou strukturu zpráv, zabezpečení komunikace proti chybám bylo optimalizováno z hlediska časové náročnosti i snižování možnosti výskytu chyby. Funkčnost a spolehlivost celého systému byla ověřena praktickými testy při ladění SW pro zastavení kamerového systému, které je v práci teoreticky popsáno. Je také představena a realizovaná metoda pro přesné zjištění orientované vzdálenosti kamerových jednotek v systému pomocí GPS. Touto metodou je možné měřit vzdálenosti v řádu kilometrů s přesností na centimetry. Praktické ověření této metody ale ukázalo nestabilitu výsledků a pro praktické použití musí být metoda vylepšena.

Reference

- [1] Paintball dron, [Online]. Available: <http://www.instructables.com/id/Paintball-Drone-Gunship-a-DIY-Combat-UAV-from-Ga/>
- [2] V. Čech, Dvoustaniční dálkoměrný systém - zastaničení, Oprox a.s., 2014
- [3] Pan-Tilt Unit-D46-70, Flir, [Online]. Available: <http://www.flir.com/mcs/view/?id=53712&collectionid=582&col=53715>
- [4] Prosilica GT1290, Allied Vision, [Online]. Available: <http://www.alliedvisiontec.com/us/products/cameras/gigabit-ethernet/prosilica-gt/gt1290.html>
- [5] A. Čížmár, Regulátor objektivu - bakalářská práce, 2014
- [6] HCA528T - dual axis inclinometer, RION, [Online]. Available: http://en.rion-tech.net/products_detail/&productId=47.html
- [7] ADS1282, Texas Instruments, [Online]. Available: <http://www.ti.com/product/ads1282>
- [8] NEO6, U-blox, [Online]. Available: http://www.u-blox.com/images/downloads/Product_Docs/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf
- [9] Datasheet převodníku RS422 - USB, FTDI, [Online]. Available: http://www.ftdichip.com/Support/Documents/DataSheets/Modules/DS_USB-COM422-PLUS4.pdf
- [10] J. Peterka, Zabezpečení dat při datových přenosech, 2011. [Online]. Available: <http://www.earchiv.cz/a91/a141c110.php3>
- [11] STM32F407 reference manual, STMicroelectronics, [Online]. Available: http://www.st.com/web/en/resource/technical/document/reference_manual/DM00031020.pdf

- [12] Schéma procesoru STM32F4, STMicroelectronics, [Online]. Available:
<http://uk.mouser.com/new/stmicroelectronics/stm32f4/>
- [13] Introduction To RS232 Serial Communication, WCSC, [Online]. Available:
<http://wscsnet.com/tutorials/introduction-to-rs232-serial-communication/>
- [14] Polarities for Differential Pair Signals RS422, B&B electronics, [Online]. Available:
<http://www.bb-elec.com/Learning-Center/All-White-Papers/Serial/%E2%80%A2-Polarities-for-Differential-Pair-Signals-%28RS-422.aspx>
- [15] J. Novák, materiály k předmětu A4M38KRP
- [16] J. Chod, *GPS - materiály k předmětu A2M32MKS*
- [17] How Differential GPS works, Trimble, [Online]. Available:
http://www.trimble.com/gps_tutorial/dgps-how.aspx
- [18] T. Takasu, RTKLIB, [Online]. Available: <http://www.rtklib.com/>
- [19] P. Koopman a T. Chakravarty, Cyclic Redundancy Code (CRC) Polynomial Selection For Embedded Networks, 2004. [Online]. Available:
http://users.ece.cmu.edu/~koopman/roses/dsn04/koopman04_crc_poly_embedded.pdf
- [20] CRC Implementation Code in C, BARR group, [Online]. Available:
<http://www.barrgroup.com/Embedded-Systems/How-To/CRC-Calculation-C-Code>
- [21] ST-LINK SWD, STMicroelectronics, [Online]. Available: http://www.st.com/st-web-ui/static/active/jp/resource/technical/document/user_manual/DM00026748.pdf
- [22] Kapacitní vazební svorkovnice Haefely, [Online]. Available:
http://www.tmgtestequipment.com.au/products/index.php?app=bkd0&ns=prodshow&ref=HAE_IP_4A
- [23] Paralelní volání příkazů batch skriptů, [Online]. Available:
<http://stackoverflow.com/questions/12577442/waiting-for-parallel-batch-scripts>

9. Přílohy

9.1. Příloha A – batch skript pro zpracování GPS dat

```
@echo off
set sourcedir=%1
title GPS evaluation

set filename="GPS_%date:~-4%%date:~6,2%%date:~3,2%%time:~0,2%%time:~3,2%%time:~6,2%"
md %filename%
set rtklib_folder=D:\rtklib_2.4.2_bin\rtklib_2.4.2\bin
set sourcedir=D:\gps_rtk\gps_test

set "lock=%temp%\wait%random%.lock"
set "lock2=%temp%\w8ait%random%.lock"

copy %sourcedir%\cam0 %filename%
copy %sourcedir%\cam1 %filename%
copy %sourcedir%\cam2 %filename%
copy %sourcedir%\cam3 %filename%

start "" 9>"%lock%1" %rtklib_folder%\convbin %filename%\cam0 -r ubx -d %filename%
start "" 9>"%lock%2" %rtklib_folder%\convbin %filename%\cam1 -r ubx -d %filename%
start "" 9>"%lock%3" %rtklib_folder%\convbin %filename%\cam2 -r ubx -d %filename%
start "" 9>"%lock%4" %rtklib_folder%\convbin %filename%\cam3 -r ubx -d %filename%

:Wait for all processes to finish (wait until lock files are no longer locked)
1>nul 2>nul ping /n 2 ::1
for %%F in ("%lock%*") do (
    (call) 9>"%%F" || goto :Wait
) 2>nul

start "" 9>"%lock%5" %rtklib_folder%\rnx2rtkp %filename%\cam0.obs %filename%\cam1.obs
filename%\cam0.nav -p 4 -f 1 -x 0 -y 0 -m 10 -h -v 4.0 -c -a -o %filename%\c1c0.pos
start "" 9>"%lock%6" %rtklib_folder%\rnx2rtkp %filename%\cam2.obs %filename%\cam1.obs
filename%\cam2.nav -p 4 -f 1 -x 0 -y 0 -m 10 -h -v 4.0 -c -a -o %filename%\c1c2.pos
start "" 9>"%lock%7" %rtklib_folder%\rnx2rtkp %filename%\cam3.obs %filename%\cam1.obs
filename%\cam3.nav -p 4 -f 1 -x 0 -y 0 -m 10 -h -v 4.0 -c -a -o %filename%\c1c3.pos
start "" 9>"%lock%8" %rtklib_folder%\rnx2rtkp %filename%\cam3.obs %filename%\cam2.obs
filename%\cam3.nav -p 4 -f 1 -x 0 -y 0 -m 10 -h -v 4.0 -c -a -o %filename%\c2c3.pos
start "" 9>"%lock%9" %rtklib_folder%\rnx2rtkp %filename%\cam1.obs %filename%\cam1.nav -p 0 -
1 -x 0 -y 0 -m 10 -v 4 -h -c -o %filename%\c1.pos

:Waitrnx for all processes to finish (wait until lock files are no longer locked)
1>nul 2>nul ping /n 2 ::1
for %%F in ("%lock%*") do (
    (call) 9>"%%F" || goto :Waitrnx
) 2>nul

del "%lock%*
```

```
del "%lock2%*
```

```
DEL %filename%\cam0.nav  
DEL %filename%\cam0.obs  
DEL %filename%\cam0.hnav  
DEL %filename%\cam0.sbs  
DEL %filename%\cam1.nav  
DEL %filename%\cam1.obs  
DEL %filename%\cam1.hnav  
DEL %filename%\cam1.sbs  
DEL %filename%\cam2.nav  
DEL %filename%\cam2.obs  
DEL %filename%\cam2.hnav  
DEL %filename%\cam2.sbs  
DEL %filename%\cam3.nav  
DEL %filename%\cam3.obs  
DEL %filename%\cam3.hnav  
DEL %filename%\cam3.sbs
```

```
exit
```

9.2. Příloha B – skripty pro přepoččet výstupu programů RTKLIB

```
clear all;
close all;
sourcefile='D:\gps_hb\gps_diff\GPS_dataset_AA\';

file='b_r1_v30_h_m10.pos';
fname=fullfile(sourcefile,file);
[ distance,heading ] = GPS_evaluation( fname,'Vzdálenost base-rover' );
fprintf('measurement br1_v30_h_m10\n');
fprintf('distance = %4.4f m\n',distance);
fprintf('heading = %4.4f °\n',heading);
fprintf('\n');

function [ distance,heading ] = GPS_evaluation(filename,dataname)
%GPS_evaluation Funkce pro vypocet a zobrazeni vzdalenosti a uhlu ze
%souboru s prepoctenymi daty z RTKLIB

    %nacteni dat ze souboru
    br1_import=importdata(filename, '');
    br1=br1_import.data;
    data=br1;
    %zobrazeni dat
    figure;
    plot(data(1:10:end,1:4));
    title(dataname); grid on;
    legend('E', 'N', 'U', 'Status');
    xlabel('čas [s]');

    %vypocet a zobrazeni vzdalenosti v kazdem bode mereni
    distanceplot=sqrt(data(:,1).^2+data(:,2).^2+data(:,3).^2);
    figure;
    hold on;
    plot(distanceplot(1:10:end,:));
    xlabel('čas [s]');
    plot(data(1:10:end,4:5),'r');
    title(dataname); grid on;
    legend('Vzdálenost antén','Status','Počet satelitů');
    hold off;
    %zahozeni neplatnych dat, zpracovani vysledku
    data(data(:,4)==2,:)=[];
    data(data(:,4)==5,:)=[];
    e=mean_filter(data(:,1));
    n=mean_filter(data(:,2));
    h=mean_filter(data(:,3));
    distance=sqrt(e^2+n^2+h^2);
    heading = rad2deg(atan2(n,e));
    if heading<0
        heading=180.0+heading;
    end
    distance=0;
    heading=0;
```

```
end
```

```
function a=mean_filter(data)
```

```
%mean_filter Funkce pro filtrovani vysledku GPS dat podle smerodatne  
%odchylky. Vystupem je stredni hodnota vseh dat, jejichz smerodatna  
%odchylka pro nasledujicich 100 prvku je mensi nez 0.002
```

```
data2=zeros(size(data));
```

```
for i=1:1:(size(data,1)-101)
```

```
    data2(i,1)=std(data(i:(i+100),1));
```

```
end;
```

```
data(data2(:,1)>0.002,:)=[];
```

```
a=mean(data);
```

```
end
```


9.3. Příloha C – program pro výpočet CRC

```
#include "crc.h"
#define POLYNOMIAL          0x97

unsigned char crcTable[256];

void crclnit(void)
{
    unsigned char remainder;
    unsigned short dividend;
    unsigned char bit;

    for (dividend = 0; dividend < 256; ++dividend)
    {
        remainder = dividend ;
        for (bit = 8; bit > 0; --bit)
        {
            if (remainder & 0x80)
            {
                remainder = (remainder << 1) ^ POLYNOMIAL;
            }
            else
            {
                remainder = (remainder << 1);
            }
        }
        crcTable[dividend] = remainder;
    }
}

unsigned char crc8(unsigned char message[], int nBytes)
{
    unsigned char c=0x00;
    unsigned char i;
    for (i = 0; i < nBytes; ++i){
        c = crcTable[(c) ^ message[i]];
    }
    return c;
}

unsigned char crc8_slow(unsigned char message[], int nBytes)
{
    unsigned char remainder = 0x00;
    unsigned short byte;
    unsigned char bit;

    for (byte = 0; byte < nBytes; ++byte)
    {
```

```
remainder ^= (message[byte]);
for (bit = 8; bit > 0; --bit)
{
    if (remainder & 0x80)
    {
        remainder = (remainder << 1) ^ POLYNOMIAL;
    }
    else
    {
        remainder = (remainder << 1);
    }
}
}
return remainder;
}
```

9.4. Příloha – hlavní část programu pro testování spolehlivosti CRC polynomů

```
#include <stdio.h>
#include <string.h>
#include <time.h>
#include "crc.h"
#define ARRAY_RANGE 6
#define NUM_OF_ERRORS 3

int main(void)
{
    unsigned char msg[ARRAY_RANGE];
    unsigned char msg2[ARRAY_RANGE];
    int error_position[NUM_OF_ERRORS];
    unsigned char crc_sender,crc_reciever;
    int i,j,k,l,temp,byte_number,position;
    unsigned long number_of_msg=0,crc_error=0,counter;
    crclnit();

    for(counter=0;counter<100000000;counter++){
        for(i=0;i<ARRAY_RANGE;i++){
            msg[i]=rand() % 255;
            msg2[i]=msg[i];
        }
        crc_sender=crc8(msg, ARRAY_RANGE);

        for(k=0;k<NUM_OF_ERRORS;k++){
            error_position[k]=rand() % (ARRAY_RANGE*8);
            if(k>0){
                for(l=(k-1);l>=0;l--){
                    if(error_position[k]==error_position[l]){
                        error_position[k]=rand() % (ARRAY_RANGE*8);
                        l=k;
                    }
                }
            }
        }

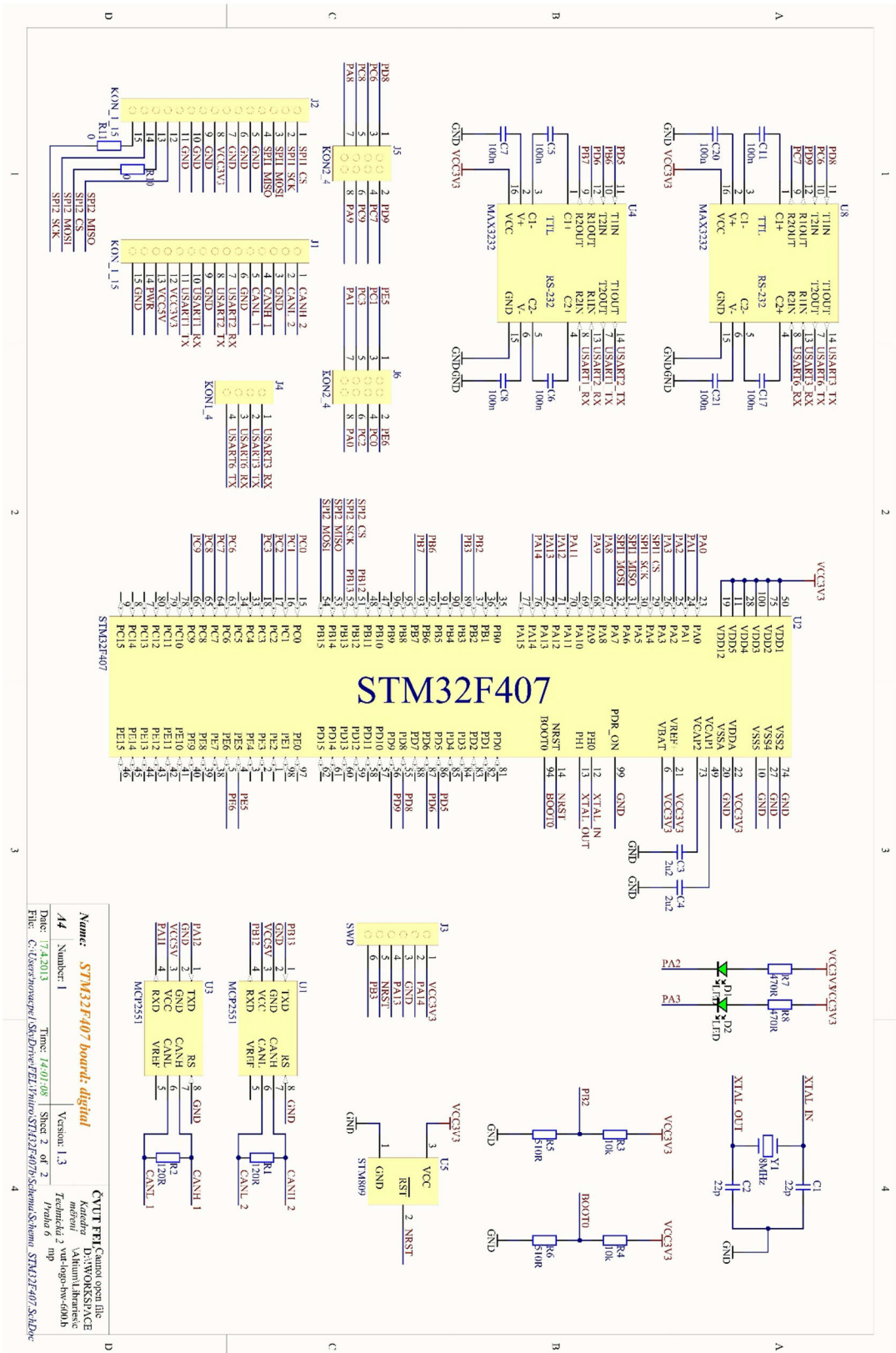
        for(k=0;k<NUM_OF_ERRORS;k++){
            byte_number=error_position[k]/8;
            position=error_position[k] %8;
            msg[byte_number]^=(1<<position);
        }

        crc_reciever=crc8(msg, ARRAY_RANGE);

        number_of_msg++;
        if(crc_sender==crc_reciever){
            crc_error++;
            for(j=0;j<ARRAY_RANGE;j++){
```

```
    printf("%x",msg[j]);
}
printf("\n");
for(j=0;j<ARRAY_RANGE;j++){
    printf("%x",msg2[j]);
}
printf("\n");
printf("\n");
}
}
printf("Celkovy pocet zprav je %u\n",number_of_msg);
printf("Celkovy chyb CRC je %u\n",crc_error);
return 0;
}
```

9.5. Příloha E – schéma řídicí jednotky



9.6. Příloha F – seznam příkazů centrálního PC pro ovládání objektivu

Ovládací příkazy						
Hlavička (1 byte)	param. 1	param. 2	param. 3	param. 4	param. 5	Popis
0x01	poloha	rychlost	0	0	CRC 8bit	Změna ohniskové vzdálenosti objektivu ZOOM: Parametr <i>poloha</i> má rozsah <0,255> a určuje hodnotu ohniskové vzdálenosti, která je v rozsahu 8 - 48 mm. Parametr <i>rychlost</i> udává maximální rychlost pojezdu mechanismu ZOOM. Plná rychlost = 0x01, poloviční rychlost = 0x02.
0x02	poloha	0	0	0	CRC 8bit	Změna zaostření FOCUS: Parametr <i>poloha</i> má rozsah <0,255> a určuje, na jakou vzdálenost má objektiv zaostřit, <1,2m, inf.)
0x03	doba	směr	0	0	CRC 8bit	Změna nastavení clony IRIS 1: Parametr <i>doba</i> určuje, jakou dobu se má IRIS pohybovat. Rozsah má <0,255>. Pro plné zatížení regulátoru platí: 0x01 ≈ 130 ms. Parametr <i>směr</i> udává, jakým směrem se má mechanismus otáčet. 0xF0 = otevírání, 0x0F = zavírání.
0x04	rychlost	směr	0	0	CRC 8bit	Změna nastavení clony IRIS 2: Parametr <i>rychlost</i> určuje rychlost pojezdu IRIS. Rozsah <1,255> odpovídá střídě řídicí PWM modulace <59%,100%>. 0x00 = zastavení. Parametr <i>směr</i> udává směrem otáčení. 0xF0 = otevírání, 0x0F = zavírání.
0x05	0	0	0	0	CRC 8bit	Reset tří mechanismů v objektivu. Nastavení mechanismů do základní polohy. ZOOM = 0x00 = 8 mm ohnisková vzdálenost. FOCUS = 0x00 = 1,2 m ostření. IRIS = plně otevřeno.
0x06	mechan.	0	0	0	CRC 8bit	Reset jednotlivých mechanismů v objektivu. Parametr <i>mechan.</i> nabývá hodnot: 1 = ZOOM, 2 = FOCUS, 3 = IRIS.
0x07	0	0	0	0	CRC 8bit	Ověření funkčnosti komunikace.

Příznakové zprávy

Hlavička (1 byte)	param. 1	Popis
0x01	příkaz	Příznak úspěšného zpracování příkazu. Parametr <i>příkaz</i> je v rozsahu <0x01, 0x07>.
0x02	mechan.	Příznak dojezdu mechanismů do krajních poloh. Parametr mechan. nabývá hodnot: 1 = ZOOM, 2 = FOCUS, 3 = IRIS.
0x04	příkaz	Chybový příznak CRC. Parametr příkaz je v rozsahu <0x01, 0x07> a označuje, u kterého typu ovládacího příkazu došlo k chybě.
0x07	0	Příznak funkční komunikace. Synchronní příznak, který je vyvoláván ovládacím příkazem 0x07.

9.7. Příloha H – tabulka příkazů pro ovládání řídicí jednotky

havička	číslo bytu									popis	
	1	2	3	4	5	6	7	8	9		
<0x7											zprávy přeposílané do objektivu
0x10	CRC8										Chyba komunikace s objektivem
0x20	1 = on, !1 = off	CRC8									Přeposílání GPS logu do PC
0x20	1 = on, !1 = off	CRC8									odpověď ŘJ
0x21	64 bytů										Zpráva s GPS daty
0x22	CRC8										požadavek na gps status
0x22	0 = ok, !1 = error	CRC8									odpověď ŘJ
0x23	CRC8										reinit gps, ret_gps_status(0x22)
0x30	u16 pan position		u16 tilt position		u8 msg counter	CRC8					synchronizace kamery a pan-tilt
0x31	CRC8										nulování msg counteru
0x31	CRC8										odpověď ŘJ
0x40	CRC8										požadavek na data z inklinometru
0x40		float roll (32b)				float pitch (32b)				CRC8	odpověď ŘJ
0x41	1 = on, !1 = off	CRC8									ovládání záměrného kříže
0x41	1 = on, !1 = off	CRC8									odpověď indikující aktuální stav
0x64	0x64	0x64	0x64	0x64	0x64						reset komunikace
0x80	CRC8										inicializace pan-tilt
0x80	0 = ok, !1 = error	CRC8									odpověď ŘJ
>0x80											příkazy pro pantilt

Obsah příloženého CD

- DP_koudelub.pdf – Diplomová práce ve formátu PDF
- Crc_polynom_test – složka s programem pro testování spolehlivosti CRC polynomů
- Ridici_jednotka – složka s programem řídicí jednotky
- SW_mereni_spolehlivosti_komunikace – složka s programy pro PC i ŘJ pro měření spolehlivosti komunikace RS422 linky
- Vyhodnoceni_GPS – složka se skripty pro vyhodnocení GPS dat, obsahuje i vzorová data
- Prikazy_RJ.xlsx – tabulka všech příkazů pro ŘJ, obsahuje i příkazy pro ovládání polohovacího zařízení
- Ridici_jednotka_objektiv.pdf – popis řídicí jednotky objektivu
- STM32F407board.pdf – schéma desky řídicí jednotky