

Pracownia dyplomowa 1

Jakub Postpski

27 stycznia 2017

1 Wstp

1.1 Zarys pracy

Wynikiem tej pracy inzynierskiej ma by zrealizowanie oprogramowania, przy pomocy którego komunikowa bd si podzespy robota mobilnego Elektron.

1.2 Opis robota

Instytut Automatyki i Informatyki Stosowanej posiada trzy roboty Elektron, które su jako pomoc dydaktyczna. Konstrukcja ta jest platform o napdzie rónicowym, relalizowanym przez dwa silniki elektryczne. Kady z nich sprzony jest z trzema koami, lewymi bd prawymi. Ich prac bezporednio nadzoruje sterownik silników z mikrokontrolerem dsPIC33FJ32MC302, posiadajcym port komunikacyjny RS-485. Gównym moduem decyzyjnym jest umieszczony w obudowie robota komputer klasy PC, z procesorem Intel Atom, dyskiem SSD oraz systemem Ubuntu z zainstalowanym ROsem. Po odpowiedniej konfiguracji robot moe by podczony do sieci wi-fi. Mona te do niego podczy róne peryferia takie jak Microsoft Kinect. Robot zasilany jest z zestawu baterii, adowanych przy pomocy zasilacza 24V.

Robot wyposaony jest w gówny sterownik, który posiada cza komunikacyjne z rónymi czciami robota. W szczególności to z tym moduem komunikuje si komputer centralny oraz sterownik silników, lecz moe on te sterowa prac innych peryferiów. Dodatkowo sterownik ten zarzdza zasilaniem robota, ma wbudowany wywietlacz LCD, zestaw przekaników mocy oraz cztery przyciski monostabilne. Sterownik ten posiada mikrokontroler STM32F107VCT.

1.3 Motywacja

Wykonanie niezawodnego i odpowiednio szybkiego oprogramowania moe spowodowa wzrost uyteczności robotów Elektron, a wic uatwi prac innym studentom. Dodatkow zacht dla autora jest moliwo nauki systemów robotycznych. Jest to te dobra okazja, aby zdoby dowiadczenie w programowaniu mikrokontrolerów firmy ST oraz pracy ze sprztem wykorzystywanym w robotyce mobilnej.

2 Wymagania stawiane pracy

Gównym celem pracy ma by wytworzenie oraz opis rozwizania, które pozwoli na szybki, deterministyczny i niezawodny komunikacj pomidzy komputerem centralnym oraz gównym sterownikiem z mikrokontrolerem STM32F107. Konsekwencj takiego dziaania ma by sprawny nadzór podzespoów Elektronu, z których najwaniejsza jest transmisja sterowania silników. Dziaania w pocztkowej fazie maj by prowadzone na robocie Elektron 3. W przypadku pomylnego ich ukoczenia, istnieje szansa, e wszystkie trzy roboty Elektron zostan zmodyfikowane w ten sam sposób. Wykonywane dziaania maj w praktyce obj napisanie programu dla mikrokontrolera STM32F107, oraz odpowiedniej biblioteki dla systemu Ubuntu. Biblioteka ma zarzdza komunikacj midzy komputerem centralnym i gównym sterownikiem oraz silnikami robota. Program sterownika silników jest ju napisany, a jego kod i sposób komunikacji s dostpne.

2.1 Wymagania funkcjonalne wysokopoziomowe

1. Komunikacja ma by deterministyczna w dziedzinie czasu.
2. Komunikacja ma si odbywa z czstotliwoci 100Hz.
3. Komunikacja ma by odporna na rónego rodzaju bdy.
4. Ma by dostpna biblioteka, dla komputera centralnego, która bdzie realizowa okrelone funkcje.
5. Zakada si brak moliwoci zmiany i rozbudowy istniejcego sprztu.
6. Zakada si brak moliwoci zmiany sytemu operacyjnego Ubuntu 14.04 LTS.

2.2 Wymagania funkcjonalne

2.2.1 Funkcje dostępne w bibliotece

1. Odczyt enkoderów (odometria).
2. Zadawanie prdkoci silników.
3. Zwracanie informacji o napięciu baterii zasilających.
4. Zaczanie i stan przekaników.
5. Informacja o wcinitych przyciskach.
6. Moliwo wyczenia zasilania robota, z opónieniem czasowym, pozwalajcym na bezpieczne wyczenie systemu operacyjnego komputera centralnego.
7. Sterowanie wbudowanym w sterownik gówny goniczkim.
8. Pomiar czasu transmisji w milisekundach.

2.2.2 Funkcje oprogramowania mikrokontrolera

1. Wywietlacz LCD informujcy o stanie poczenia z komputerem centralnym, stanie baterii, stanie przekaników.
2. Sygnalizacja dwikowa faktu rozadowania baterii.
3. Automatyczne wyczenie robota, w przypadku krytycznie niskiego napięcia.
4. Kontrola sterownika silników.
5. Realizacja komunikacji z komputerem centralnym, umoliwiajca wykorzystanie zdefiniowanych powyzej funkcji biblioteki.

3 Dobór rozwizania

3.1 Moliwoci sprztowe

Podstawowym ograniczeniem, przy wyborze sposobu realizacji zadania, s moliwoci sprztowe. Pyta gówna komputera centralnego, jak atwo sobie wyobrazi, wyposaona jest w standardowy zestaw portów komunikacyjnych, a wic zczy Ethernet, zczy USB 2.0 oraz zczy DB-9 z protokoem RS232. Mikrokontroler STM32F107 posiada wbudowany kontroler Ethernetu wraz z wasnym DMA, szyny komunikacyjnej CAN, portu USB oraz dwa ukady USART i trzy ukady UART. Wszystkie wymienione kontrolery maj wyprowadzenia na pytce gównego sterownika, przy czym niektóre wyprowadzenia urzdu UART i USART dziki ukadom MAX3485 konwertowane s do standartu RS-485, a inne dziki ukadom MAX3232 konwertowane s do standartu RS-232.

3.2 Dostpne rozwizania komunikacji

Aby wybra odpowiedni architektura rozwizania autor wykona rozpoznanie dostpnych rodków z uwzglndnieniem ogranicze sprztu. Poniej zamieszczono krótki opis niektórych z technologii.

1. **Ethercat** - jest to standard wykorzystujcy sie Ethernet do szybkiej wymiany informacji w czasie rzeczywistym. W sieci Ethercat musi by jeden wze okrelany jako master, oraz praktycznie dowolna ilo wzów slave. Mimo, e jest to potencjalnie najlepsze rozwizanie, ze wzgldu na determinizm oraz wytkowo mae opónienia, nie mogo zosta zastosowane, przez ograniczenia sprztowe. O ile weze master moe uywa normalnego kontrolera Ethernetu, standard wymaga, aby wzy slave posiaday specjalnie przystosowane kontrolery Ethernetu.
2. **Pakiety TCP/IP oraz UDP/IP** - midzy komputerem centralnym, a mikrokontrolerem gównego sterownika miaayby by wysyane standardowe pakiety danych, przez zczy Ethernet. Dziki temu, e w robocie mona zapewni bezporednie poczenie kabla Ethernetowego midzy urzdzzeniami, nie wystpowayby kolizje pakietów w warstwie cza danych. Dziki temu mona potencjalnie liczy na efekty zblione do tych uzyskiwanych w sieciach czasu rzeczywistego.
3. **Magistrala CAN** - sie szeregow, zapewniajca transmisj rzdu 1Mb/s. Rozwizanie to, mimo e popularne i wzgldnie tanie, zostao odrzucone, przez brak interfejsu po stronie komputera.
4. **USB** - Popularny interfejs, który zosta odrzucony przez brak gwarancji determinizmu czasowego.
5. **RS-232** - Jedno z prostszych czy szeregowych, co jest jednoczenie wad i zalet. Nie zapewnia zbyt szybkich transferów, lecz atwo je uruchomi i powinno by deterministyczne czasowo.

3.3 Rozwizanie

Pocztkowo autor chca wykorzysta zwyke poczenie Ethernetu, aby za pomoc pakietów UDP, przesya dane. Rozwizanie to jest do szybkie, jak na postawione wymagania, potencjalnie bezawaryjne oraz praktycznie bezkosztowe. Niestety okazao si, e autor z pewnych przyczyn, opisanych szczegóowo w dalszej pracy nie jest w stanie uruchomi kontrolera Ethernetu mikrokontrolera. W zwizku z tym, ostatecznie zdecydowano sina realizacj w oparciu o RS-232. Miao to swoje negatywne konsekwencje, takie jak konieczno samodzielnego opakowania danych w ramk zawierajc sum CRC oraz spowolnienie transmisji.

Zdecydowano te, e oba wzy, to jest komputer centralny i gówny sterownik, bd sikomunikowa w architekturze zapytanie i odpowied, a wzem nadrzdnym ma by komputer centralny. Jest to najprostsza z moliwych architektur, lecz w zupenoci wystarcza do wymiany danych midzy dwoma urzdzzeniami.

Z uwagi na fakt, e mikrokontroler gównego sterownika, ma wykonywa kilka zada jednoczenie i niektóre z nich maj narzucony reim czasowy, zdecydowano si, na zastosowanie systemu czasu rzeczywistego. Przez wsparcie ze strony producenta mikrokontrolera, wybrany zosta system FreeRTOS.

4 czno pomidzy komputerem centralnym a gównym sterownikiem

Po wstpnyim poznaniu sprztu zostay wyspecyfikowane dodatkowe zaoenia pracy. Transmisja danych odbywa sibdzie midzy UART4 mikrokontrolera, a portem RS-232 komputera centralnego, przy pomocy konwertera napi MAX3232. Prdko transmisji to 115200bit/s. Transmisja jest omiobitowa, bez bitów parzystoci. Aby zapewni poprawno danych wysyane struktury opakowywane s w ramki, zgodnie ze standartem High-Level Data Link Control, zatwierdzonym w jednej z norm ISO. Komputer centralny wysya do gównego sterownika struktur TxFrame inicjujc komunikacj. W odpowiedzi wysane zostaj dane zgodne ze struktur RxFrame. Struktury maj pola opisane poniej.

```
typedef struct
{
    uint32_t timestamp;
    int16_t left_speed;
    int16_t right_speed;
    uint8_t relays;
    uint8_t sound;
    uint8_t shutdown;
}TxFrame;

typedef struct
{
    uint32_t timestamp;
    uint8_t buttons;
    int32_t left_position;
    int32_t right_position;
    uint16_t battery;
    uint8_t error;
}RxFrame;
```

Dziki dodaniu pola timestamp istnieje moliwo atwego pomiaru czasu caej transmisji, wraz z przetwarzaniem danych przez mikrokontroler.

5 Opis wykonanych czynnoci

5.1 Uruchomienie mikrokontrolera STM32F107

Autor pracy nigdy wczeniej nie programowa mikrokontrolerów tej firmy. Pewien czas zosta wic spoytkowany na wdroenie siw dokumentacjmikrokontrolera, oraz wybór rodowiska programistycznego. W trakcie rozpoznania zosta skonfigurowany debbuger wraz z programatorem ST-Linkv2. Skonfigurowano te system FreeRTOS w wersji ósmej. Na tym etapie duym uatwieniem byo skorzystanie z oprogramowania ze strony producenta, czyli aplikacji CubeMX oraz SystemWorkbench. Uruchomiono podstawowe urzdzzenia peryferyjne mikrokontrolera z wykorzystaniem biblioteki HAL.

5.2 Uruchomienie wywietlacza LCD

Wywietlacz zamontowany w robocie Elektron ma sterownik KS0108. Ma rozdzielczo 128x64 pikseli i jest monochromatyczny.

5.3 Uruchomienie portu Ethernetu

Niestety ta czynno nie zostaa zakoczona sukcesem. Mimo usilnych stara, oraz zastosowania sido zalece producenta mikrokontroler stale zawiesza sina konfiguracji DMA Ethernetu. Pewn poszlak w diagnozie zaistniaej sytuacji moe by lektura erraty opisujcej mikrokontroler, znajdujca si na stronie producenta. Wynika z niej, e mikrokontroler dokadnie w tej rewizji w której by uywany, moe faktycznie mie bd w konfiguracji DMA Ethernetu. Jednak wedug tej erraty mikrokontroler powinien sizawiesza, tylko jeli wczeniej by usypiany, a fakt ten nie mia miejsca. Inne moliwe przyczyny niedziaania to bd w projekcie sterownika, powodujcy zakócenia elektromagnetyczne, albo zwyczajny bd programisty. Wykluczone zostay jednak ewentualne uszkodzenia fizyczne, poprzez programowanie takiego samego sterownika w robocie Elektron 2. Autor pracy z ciekawoci uruchomi port Ethernetu na płytce Nucleo z mikrokontrolerem STM32F207, z tym samym co w STM32F107 rdzeniem Coretex M3, bez wikszych problemów. Brak moliwoci komunikacji w ten sposób spowodowa zmian zaoe pracy.

5.4 Uruchomienie silników

Wykorzystujc bibliotekNFv2dostarczondo sterownika silników udao si zarówno zadawa prdko jak i uzyska odczyt danych z enkoderów. Poczzenie ze sterownikiem silników odbywa si poprzez USART2 mikrokontrolera STM32F107, z prdkoci 5600bit/s. Wykorzystywany jest do tego standard RS485 w trybie half-duplex. Dokonano te niewielkich modyfikacji w kodzie sterownika silników, co w przyszoci moe owocowa stabilniejsz prac sterownika.

5.5 Przyciski i przekaniki

Przekaniki s sterowane, przez porty GPIO mikrokontrolera, za pomoc tranzystorów. Take przyciski s obsugiwane w ten sposób. Do odczytu stanu przycisków zastosowano algorytm cyfrowej eliminacji drga.

5.6 Pomiar napicia baterii

Realizowany jest poprzez przetwornik ADC mikrokontrolera.

5.7 Prace serwisowe

W trakcie uytkowania wykonano przegld okablowania, wymieniono na nowy dysk twardy, oraz niedziaajcy ukad MAX3485, umieszczony w sterowniku silników.

5.8 Implementacja cznoci

Zostaa wykonana, poprzez napisane biblioteki dla systemu Linux, udostpniajcej API zgodne z wczeniejszymi zaoeniami. Drug czynnociwykonan w ramach zadania byo napisanie odpowiedniego programu dla głównego sterownika, współpracujcego z t bibliotek. W ramach tego zadania uruchomiono port RS-232 na obu urzdzeniach.

6 Program głównego sterownika

Program pisany by, przy wykorzystaniu biblioteki HAL, oraz systemu czasu rzeczywistego FreeRTOS. Skada sion z kilku zada, z których najwianiejsze odpowiadaj za komunikacj ze sterownikiem silników i komputerem centralnym. Implikacj tego faktu jest zastosowanie rónych priorytetów dla poszczególnych z zada.

7 Opis planowanych dziaa

Autor pracy uwaa, e ogólny stan prac jest dobry. Zostaa wykonana najwianiejsza cz pracy, czyli funkcjonujca biblioteka dla systemu Ubuntu, wraz z odpowiednim programem dla głównego sterownika robota. Pozostao jednak jeszcze kilka zada.

1. Integracja biblioteki z ROSem.
2. Implementacja prostej aplikacji demonstrujcej dziaanie cznoci.
3. Testy przedstawionego rozwizania.
4. Zainstalowanie nowego oprogramowania i ewentualna adaptacja sprztu w robotach Elektron 1 i Elektron 2.

Literatura

- [1] “Stm32f107vc.” https://my.st.com/content/my_st_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32f1-series/stm32f105-107/stm32f107vc.html.
- [2] “Freertos.” <http://www.freertos.org/>.
- [3] “Information technology telecommunications and information exchange between systems high-level data link control (hdlc) procedures.” <https://www.iso.org/obp/ui/\#iso:std:iso-iec:13239:ed-3:v1:en>.
- [4] “Ethercat.” <https://en.wikipedia.org/wiki/EtherCAT>.
- [5] “Can bus.” https://en.wikipedia.org/wiki/CAN_bus.
- [6] “Rs232.” <https://en.wikipedia.org/wiki/RS-232>.
- [7] W.R.Stevens, *Programowanie zastosowa sieciowych w systemie Unix*. 1995.