Зміст

1. Призначення й завдання переддипломної практики………………………..3
2. Характеристика ДП "Головний інформаційно обчислювальний центр Державної адміністрації залізничного транспорту України"..………….…4
3. Індивідуальне завдання, постановка задачі………………………………....6
   1. Аналіз поставленої задачі………………………………………………..6
   2. Огляд існуючих аналогів…………………………………………….......7
   3. Розробка концепції рішення…………………………………………..…9
   4. Опис програмного забезпечення та апаратних рішень……...………..12

Висновки…………………………………………………………………………16

Список використаної літератури……………………………………………….17

Додатки

Анотації

Технічне завдання

Зміст пояснювальної записки

Аналіз існуючих рішень та обґрунтування теми дипломного проекту

1. **Призначення і завдання переддипломної практики**

Практики студентів НТУУ "КПІ" проводяться у відповідності до навчальних планів спеціальностей, умов контрактів і договорів на підготовку спеціалістів, графіка навчального вищого навчального закладу, кваліфікаційних характеристик спеціаліста, освітньо-професійної програми підготовки фахівців, особливостей галузі, у якій буде працювати випускник після закінчення навчання, особливостей бази практики й повинні відповідати Положенню про проведення практики студентів вищих навчальних закладів України у терміни, передбачені навчальними планами.

Переддипломна практика є заключною ланкою практичної підготовки студентів, яка проводиться перед виконанням кваліфікаційної роботи. Робота студента в період переддипломної практики, в першу чергу, має бути узгоджена з темою дипломного проекту: опрацювання теми, вибір основних напрямків реалізації поставлених завдань, збір необхідної інформації та набуття практичних навичок у роботі з відповідним програмним середовищем та апаратними засобами.

Основною метою переддипломної практики є підготовка майбутніх бакалаврів до практичної роботи, раціонального поєднання теоретичних знань спеціалістів з умінням вирішувати практичні завдання за фахом.

Переддипломна практика покликана сформувати у майбутнього бакалавра-випускника ВНЗу професійні уміння і навички прийняття самостійних рішень в реальних робочих умовах.

Під час проходження практики студент повинен одержати навички самостійної роботи. При цьому в окремих випадках допускається робота студентів на штатних посадах (програміста, інженера та ін.). Для розширення світогляду і знань під час практики планується проведення екскурсій на базу практики, а також на інші підприємства та організації. Найважливішим завданням переддипломної практики є збирання матеріалів для виконання дипломної роботи (проекту).

**2. Характеристика ДП "Головний інформаційно обчислювальний центр Державної адміністрації залізничного транспорту України"**

ДП "ГІОЦ Укрзалізниці" займається впровадження та експлуатацією автоматизованих систем, наданням інформаційних послуг підприємствам залізничного транспорту та клієнтам залізниць України, а також підтримкою та розвитком мережі передачі данних Укрзалізниці. Серед найважливіших автоматизованих систем та програмно-аппаратних комплексів особливу роль відіграють наступні:

* АСК ПП УЗ

Автоматизована система керування пасажирськими перевезеннями(АСК ПП УЗ) забезпечує обслуговування пасажирів щодо перевізних документів. АСК ПП УЗ забезпечує бронювання місць та продаж квитків на пасажирські поїзди. АСК ПП УЗ прийшла на зміну автоматизованих систем Експрес 2 та Експрес УЗм. Автоматизована система керування пасажирськими перевезеннями працює на потужному серверному комплексі IBM Z9. Це дає змогу обслуговувати сотні запитів за секунду з мінімальною затримкою.

* АСК ВП УЗ-Є

Єдина автоматизована система керування вантажними перевезеннями (АСК ВП УЗ-Є) була введена в експлуатацію 07.07.2012 замість систем АСОУП та АСК ВП УЗ, які функціонували на рівні залізниць України. Нова система архітектурно побудована для централізованого керування процесом вантажних перевезень.

* АС Клієнт

Система АС Клієнт УЗ надає сучасний сервіс клієнтам залізничного транспорту України по оформленню електронної накладної.

* АС Месплан

Автоматизована система документообігу замовлень на перевезення вантажів та формування планів(АС Месплан) створенна для керування вагонним парком Укрзалізниці та можливості більш зручного способу замовлення вагонів під навантаження клієнтами залізничного транспорту України.

* Акредитиваний центр сертифікації ключів(АЦСК)
* Сервіс онлайн резервування та покупки квитків
* АС СКЕДО(система електронного документообігу)

Автоматизована система керування електронним документообігом Укрзалізниці(АС СКЕДО) створено для переходу на безпаперовий обмін документами між Адміністрацією залізничного транспорту, підприємствами залізничного транспорту та залізницями України. Система АС СКЕДО розроблена в середовищи Lotus Domino та функціонує як складова частина документообігу Укрзалізниці. Працює з цифровими підписами акредитованого центру сертифікації ключів та використовує алгорітми підпису згідно ДСТУ 4145-2002, що дає юридичну силу електронним документам.

* АСУ Кадри(система управління кадрами)

Автоматизована система керування кадрами Укрзалізниці(АСК Кадри) створено для ефективного управління кадрами Адміністрацією залізничного транспорту, підприємствами залізничного транспорту та залізницями України.

* Корпоративна поштова система Lotus Domino

Зараз на підприємстві працює майже 400 фахівців. Багато зі спеціалістів працюють на підприємстві з самого його народження, але є вже багато таких які прийшли на підприємство молодими фахівцями і вже тут стали спеціалістами у своїй галузі.

ДП "Головний інформаційно обчислювальний центр Державної адміністрації залізничного транспорту України" було створено у 1993 році. **3. Індивідуальне завдання, постановка задачі**

* 1. **Аналіз поставленої задачі**

Завданням даної практики є розробка програмного забезпечення мікроконтролера STM32F103RET6 на мові C++ з використанням операційної системи реального часу FreeRTOS та технології Ethernet (на базі ІС Microchip ENC28J60) для комунікації між мікроконтролерами та розробка протоколу взаємодії модулів на їх базі.

FreeRTOS – це багатозадачна операційна система реального часу для вбудованих систем. Адаптована для декількох мікропроцесорних архітектур. Розповсюджується за умовами модифікованої ліцензії GPL з виключенням, що дозволяє розробнику привласнити модифікований код операційної системи. Більша частина цієї операційної системи написана мовою C, за винятками асемблерних вставок мінімального розміру, де неможливо застосувати C через специфіку конкретної апаратної платформи.

Планувальник FreeRTOS підтримує три типи багатозадачності:

1. витісняльну;
2. кооперативну;
3. гібридну.

Розмір ядра FreeRTOS складає всього 4-9 Кб, залежно від типу платформи і налаштувань ядра. Ця ОС підтримує задачі та співпрограми. Співпрограми спеціально створені для мікроконтролерів з малим об’ємом ОЗП. FreeRTOS підтримує черги, двійкові семафори, рекурсивні семафори, м’ютекси (з успадкуванням пріоритету). У ній відсутні обмеження на кількість задач та кількість пріоритетів. FreeRTOS працює на архітектурах з розрядністю 8, 16 та 32 біти.

Мікроконтролери серії STM32F103 містять у собі ARM ядро Cortex-M3, яке здатне працювати на частоті 72 МГц. Мікроконтролер STM32F103RET6 запакований в корпус LQFP-64, містить в собі 512 Кб флеш-пам’яті для зберігання програмного коду та статичних даних з латентністю доступу

2 такти при частоті 72 МГц, 64 Кб пам’яті статичного ОЗП, вісім 16-бітних таймерів загального призначення, 16 каналів АЦП та 2 канали ЦАП розрядністю 12 біт, годинник реального часу, 51 порт вводу/виводу. Мікроконтролер підтримує 3 канали SPI, 2 канали I2S, 2 канали I2C, 3 канали USART, 2 канали UART, SDIO, USB, CAN.

Microchip ENC28J60 – контролер, що використовується для взаємодії з мережами Ethernet. Цей контролер містить у собі реалізації двох рівнів еталонної моделі (фізичний рівень – PHY, канальний рівень - MAC). PHY-рівень підтримує стандарт 10Base-T(10 Мбіт/с, повний дуплекс) з автоматичним визначенням полярності сигналу та типу кабелю. На MAC рівні підтримується визначення колізій, повторна відправка кадрів при виникненні колізій, автоматичне генерування та перевірка контрольних сум кадрів, буферизація кадрів (об’єм буфера 8 Кб), інтелектуальна фільтрація кадрів. У контролері використовується технологія прямого доступу до пам’яті для швидкого переміщення даних. Обмін даними з мікроконтролером відбувається через послідовний інтерфейс SPI.

uIP - це відкритий TCP/IP-стек, розроблений для мікроконтролерів з 8, 16 та 32 бітною архітектурою. На відміну від lwIP uIP оптимізований з точки зору ресурсів пам'яті. Стек uIP веде обмін мережевими даними через спеціальний статичний буфер. Внаслідок такої оптимізації uIP не підтримує деякі можливості TCP/IP стека, наприклад збірку фрагментованих IP пакетів, алгоритм Нейгла, відновлення правильної послідовності пакетів, кілька пакетів на один ACK. Проблема резервного зберігання прийомною стороною непідтверджених даних винесена з uIP і покладена на користувача стека.

* 1. **Огляд існуючих аналогів**

Wiren Board Smart Home - контролер для домашньої автоматизації. Основа контролеру – ARM9 процесор частотою 454 МГц та ОЗП об’ємом

64 Мб. Контролер працює під керуванням операційної системи GNU Linux 3.13. Для комунікації використовуються інтерфейси Ethernet та WiFi.

Цей контролер містить у собі:

1. 2 порти USB;
2. 2 інтерфейси RS-485 (Modbus);
3. інтерфейс CAN;
4. GSM/GPRS модуль;
5. радіомодуль NRF24;
6. модуль пакетного радіо HopeRF RFM69H 433 МГц;
7. інтерфейс Wiegand;
8. 3 інтерфейси 1-Wire;
9. інтерфейси для резистивних датчиків;
10. 4 входи АЦП;
11. 2 реле 220В 5А.

Пристрій має можливість роботи від зовнішнього акумулятора та керується за допомогою ПЗ Ninja Blocks.

Ab-log MegaD-328 – багатофункціональний контролер для домашньої автоматизації. Побудований на базі контролера Atmel ATMega328. Для комунікації використовується інтерфейс Ethernet реалізований на мікросхемі Microchip ENC28J60. Для керування використовується децентралізована схема. Контролер містить вбудований веб-сервер. Конструкція пристрою – модульна. Базовий модуль містить 7 цифрових входів, 7 аналогових силових виходів .

KernelChip Laurent – мережевий модуль з різноманітною периферією для керування реле, моніторингу та замірів різних параметрів по мережі. Цей контролер призначений для узгодження зовнішніх цифрових і аналогових пристроїв, датчиків і механізмів з комп’ютером (або мережею комп’ютерів) через інтерфейс Ethernet. Laurent містить у собі потужні електромагнітні реле, вхідні та вихідні дискретні лінії, датчик температури, лічильник імпульсів, вихід з широтно-імпульсною модуляцією, порт АЦП та порт RS-232. Модулем можна керувати через вбудований веб-сервер або набором текстових команд керування по TCP/IP протоколу.

Модуль містить:

1. інтерфейс Ethernet;
2. 4 електромагнітні реле;
3. 6 вхідних оптоізольованих дискретних ліній;
4. інтерфейс TCP-2-COM;
5. 12 вихідних дискретних ліній керування (до 50 В, 0.5 А на кожну лінію);
6. інтерфейс RS-232.

UniPing v3 – пристрій для постійного моніторингу температури, вологості, доступу в приміщення, наявності затоплення через інтерфейс Ethernet. Дані можуть збиратися через E-mail, по протоколу SNMP або через syslog. UniPing містить вбудований веб-сервер.

До модуля можливе підключення:

1. одного датчика вологості;
2. до 16 датчиків затоплення;
3. до 16 датчиків розбиття скла;
4. до 8 датчиків удару;
5. до 4 термодатчиків (резистивних);
6. датчик диму;
7. до 16 датчиків руху;
8. до 16 датчиків наявності напруги в мережі;
9. пристрою через інтерфейс RS-232 або RS-485;
10. модуля ІЧ управління.
    1. **Розробка концепції рішення**

Microsoft Visual Studio — серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight.

Visual Studio Express – набір середовищ розробки, що представляють собою обмежену версію Visual Studio. На відміну від повної версії, кожне таке середовище призначене для якоїсь однієї платформи. Воно включає в себе невеликий набір інструментів, і на відміну від повних версій, в ній: відсутній дизайнер класів і багато інших інструментів, а також підтримка плагінів і віддалених баз даних в дизайнера даних. Компілятори в 64-бітний код також недоступні в Express редакціях до версій 2012 року (хоча компілятор безкоштовно розповсюджується з Windows SDK і його можна використовувати, компілювати автоматично з IDE не можна). Microsoft позиціонує цю лінійку IDE для студентів і аматорів.

VisualGDB – плагін для Microsoft Visual Studio, який дозволяє інтегрувати набір інструментів GNU (GCC та GDB) в середовище розробки. Також плагін дозволяє розробку та налагодження програмного забезпечення, в якому використовуються інструменти GNU – ПЗ для мікроконтролерів та для Linux-систем.

Sourcery CodeBench Lite – набір інструментів для розробки програмного забезпечення призначеного для виконання на системах з архітектурами ARM, MIPS, LSI Axxia, Nios II, Qualcomm Hexagon, IA32, IA64. Набір включає інструменти для компіляції ассемблерного, C та C++ коду, лінкувальник, та налагоджувач (GDB).

User Datagram Protocol - один із протоколів в стеку TCP/IP. Від протоколу TCP він відрізняється тим, що працює без встановлення з'єднання. UDP — це один з найпростіших протоколів транспортного рівня моделі OSI, котрий виконує обмін дейтаграмами без підтвердження та гарантії доставки. При використанні протоколу UDP обробка помилок і повторна передача даних має виконуватися протоколом вищого рівня. Але, незважаючи на всі недоліки, протокол UDP є ефективним для серверів, що надсилають невеликі відповіді великій кількості клієнтів.

Протокол UDP використовують такі сервіси та протоколи вищого рівня:

1. TFTP (англ. Trivial File Transfer Protocol, найпростіший протокол передачі файлів);
2. SNMP (англ. Simple Network Management Protocol, простий протокол управління мережею);
3. DHCP (англ. Dynamic Host Configuration Protocol, протокол динамічної конфігурації вузла);
4. DNS (англ. Domain Name System, служба доменних імен).

Заголовок UDP конверту складається з 4 полів, з яких 2 є опціональними. «Порт відправника» та «порт одержувача» — це 16-бітні поля, котрі ідентифікують відправляючий та одержуючий процеси. «Порт відправника» є необов'язковим, оскільки UDP працює без встановлення з'єднання та відправник може не потребувати відповіді. В такій ситуації «порт відправника» повинен дорівнюватися нулю. Поле «Розмір» є обов'язковим, воно визначає довжину усієї UDP дейтаграми в байтах, з полем «Дані» включно. Мінімальне значення цього поля дорівнює 8 байт.

Останнє поле заголовка довжиною 16 біт містить у собі контрольну суму заголовка і поля даних. «Контрольна сума» теж є необов'язковим полем, але на практиці воно майже завжди використовується.

Програми, що використовують UDP в якості транспортного протоколу, мають бути готові до помилок, втрати деяких конвертів та повторної передачі даних. Деякі програми, такі як TFTP, можуть використовувати додаткові програмні механізми для підвищення надійності передачі. Але у більшості випадків, для таких програм, надійність не є необхідною і може навіть завадити уповільненням зв'язку. Потокове відео, ігри реального часу та VoIP(голос поверх IP) є прикладами програм, що дуже часто використовують UDP. Якщо ж програма потребує високого рівня надійності, то може використовуватися такий протокол як TCP або надлишковість коду, за допомогою якої можна знаходити помилки при передачі даних.

Оскільки у протоколі UDP відсутній будь-який контрольний механізм запобігання перевантаженням, мережні механізми повинні мати засоби для зменшення ефекту потенційних перевантажень від великого, неконтрольованого потоку UDP-трафіку. Кажучи інакше, оскільки UDP-відправники не спроможні виявляти перевантаженість, мережні елементи, такі як роутери, що використовують «черги конвертів» залишаються єдиним інструментом для призупинення надмірного UDP-трафіку. DCCP(англ. Datagram Congestion Control Protocol, протокол контролю навантаженості дейтаграм) був створений як часткове рішення цієї проблеми. Він контролює навантаження на кінцевих вузлах високошвидкісних потоків UDP-трафіку, наприклад, потокового відео.

Хоча кількість UDP трафіку в типовій мережі сягає лишень кількох відсотків, проте багато важливих програм використовують UDP. Серед них: DNS (Domain Name System, служба доменних імен), SNMP (Simple Network Management Protocol, простий протокол управління мережею), DHCP (англ. Dynamic Host Configuration Protocol, протокол динамічної конфігурації вузла), RIP (Routing Information Protocol, протокол маршрутизації інформації) та багато інших.

* 1. **Опис програмного забезпечення та апаратних рішень**

Обмін інформацією між контролерами відбувається через мережу Ethernet. Був реалізований протокол обміну з використанням протоколу UDP (його реалізація в стеку протоколів uIP). Підтримка TCP протоколу була вимкнена для економії ресурсів контролера, оскільки надійна реалізація витрачає чималу кількість ресурсів, а недостатньо надійна реалізація TCP не має ніяких переваг перед UDP в межах одного сегменту локальної мережі.

Для максимального використання можливостей мікросхеми доступу до Ethernet ENC28J60 довелося реалізовувати механізми буферизації в контролері, оскільки наявний в мікросхемі механізм не дозволяє ефективно передавати дані.

Різні логічно відокремлені частини алгоритму виконуються переважно в окремих потоках для можливості їх розподілення по пріоритетах та для мінімізації їх взаємного впливу. Потоки даних між частинами алгоритмів передаються за допомогою вбудованих в операційну систему (FreeRTOS) черг.

Частина програмного коду, що описує основні структури протоколу взяємодії через мережу:

#ifndef PACKETTYPE\_H\_

#define PACKETTYPE\_H\_

//#include "stm32f10x.h"

#include <stdint.h>

#ifdef \_\_cplusplus

extern "C" {

#endif

#define PACKET\_DATA\_SIZE 123

typedef uint16\_t target\_t;

typedef uint8\_t udp\_data\_type\_t;

typedef uint8\_t climate\_info\_type\_t;

typedef uint8\_t info\_data\_type\_t;

typedef uint8\_t climate\_info\_type\_t;

typedef uint8\_t alarm\_info\_type\_t;

typedef uint8\_t light\_info\_type\_t;

typedef uint8\_t jalousie\_info\_type\_t;

typedef uint8\_t ventilation\_info\_type\_t;

typedef uint8\_t watering\_info\_type\_t;

typedef uint8\_t heater\_info\_type\_t;

typedef uint8\_t lock\_info\_type\_t;

typedef uint8\_t watersupply\_info\_type\_t;

struct app\_udp\_packet;

struct app\_info\_hdr;

struct app\_climate\_hdr;

struct app\_climate\_data;

typedef struct {

target\_t destination\_id;

target\_t source\_id;

udp\_data\_type\_t data\_type;

uint8\_t data[PACKET\_DATA\_SIZE];

} app\_udp\_packet;

/\* In app\_udp\_packet : udp\_data\_type is UDP\_DATA\_TYPE\_INFO \*/

typedef struct {

target\_t destination\_id;

target\_t source\_id;

udp\_data\_type\_t data\_type; // is UDP\_DATA\_TYPE\_INFO

info\_data\_type\_t info\_data\_type;

uint8\_t data[PACKET\_DATA\_SIZE - sizeof(info\_data\_type\_t)];

Частина програмного коду, яка взаємодіє з мікросхемою ENC28J60 через SPI:

#include "stm32f10x.h"

#include <stdint.h>

#include "enc28j60.h"

#include "stm32f10x\_spi.h"

/\*

\* SPI

\*/

volatile uint8\_t enc28j60\_current\_bank = 0;

volatile uint16\_t enc28j60\_rxrdpt = 0;

#define enc28j60\_select() GPIO\_ResetBits(GPIOB, GPIO\_Pin\_12)

#define enc28j60\_release() GPIO\_SetBits(GPIOB, GPIO\_Pin\_12)

uint8\_t enc28j60\_rxtx(uint8\_t data)

{

while(SPI\_I2S\_GetFlagStatus(SPI2, SPI\_I2S\_FLAG\_TXE)==RESET);

SPI\_I2S\_SendData(SPI2,data);

while(SPI\_I2S\_GetFlagStatus(SPI2, SPI\_I2S\_FLAG\_RXNE)==RESET);

return SPI\_I2S\_ReceiveData(SPI2);

}

#define enc28j60\_rx() enc28j60\_rxtx(0xff)

#define enc28j60\_tx(data) enc28j60\_rxtx(data)

// Generic SPI read command

uint8\_t enc28j60\_read\_op(uint8\_t cmd, uint8\_t adr)

{

uint8\_t data;

enc28j60\_select();

enc28j60\_tx(cmd | (adr & ENC28J60\_ADDR\_MASK));

if(adr & 0x80) // throw out dummy byte

enc28j60\_rx(); // when reading MII/MAC register

data = enc28j60\_rx();

enc28j60\_release();

return data;

}

// Generic SPI write command

void enc28j60\_write\_op(uint8\_t cmd, uint8\_t adr, uint8\_t data)

{

enc28j60\_select();

enc28j60\_tx(cmd | (adr & ENC28J60\_ADDR\_MASK));

enc28j60\_tx(data);

enc28j60\_release();

}

// Initiate software reset

void enc28j60\_soft\_reset()

{

enc28j60\_select();

enc28j60\_tx(ENC28J60\_SPI\_SC);

enc28j60\_release();

enc28j60\_current\_bank = 0;

volatile uint32\_t i;

for (i=0; i<72000; i++) {

}

//\_delay\_ms(1); // Wait until device initializes

}

/\*

\* Memory access

\*/

// Set register bank

void enc28j60\_set\_bank(uint8\_t adr)

{

uint8\_t bank;

if( (adr & ENC28J60\_ADDR\_MASK) < ENC28J60\_COMMON\_CR )

{

bank = (adr >> 5) & 0x03; //BSEL1|BSEL0=0x03

if(bank != enc28j60\_current\_bank)

{

enc28j60\_write\_op(ENC28J60\_SPI\_BFC, ECON1, 0x03);

enc28j60\_write\_op(ENC28J60\_SPI\_BFS, ECON1, bank);

enc28j60\_current\_bank = bank;

}

}

}

**Висновки**

Вирішення конкретної задачі допомогло розширити технічний кругозір, вивчити нову інформацію та застосувати на практиці знання, отримані в університеті. Було покращено рівень знань та умінь, які необхідні для конкурентоспроможності на реальному ринку праці та полегшення входу до нього після закінчення вузу.

Вирішення поставленого індивідуального завдання допомогло розширити знання мови C++ та її підмножини для систем реального часу. Отримані навички проектування протоколів обміну інформацією в комп’ютерних мережах. Для вирішення завдання було задіяно багато різних периферійних блоків мікроконтролера, тому після проходження практики з’явилося краще розуміння принципів їх функціонування.

**Список використаної літератури**

1. The JSF air vehicle C++ coding standards [Електроний ресурс] – Режим доступу: http://www.stroustrup.com/JSF-AV-rules.pdf
2. MISRA C++2008 [Електроний ресурс] – Режим доступу: http://frey.notk.org/books/MISRA-Cpp-2008.pdf
3. STM32F103 - STMicroelectronics [Електроний ресурс] – Режим доступу: http://www.st.com/web/catalog/mmc/FM141/SC1169/SS1031/LN1565
4. ENC28J60 Stand-alone Ethernet Controller [Електроний ресурс] – Режим доступу: http://ww1.microchip.com/downloads/en/devicedoc/39662a.pdf