

UNIVERZITA KOMENSKÉHO V BRATISLAVE  
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

RIADIACI SYSTÉM PRESNÝCH KROKOVÝCH  
MOTOROV  
BAKALÁRSKA PRÁCA

2016  
TOMÁŠ WIEDERMANN

UNIVERZITA KOMENSKÉHO V BRATISLAVE  
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

RIADIACI SYSTÉM PRESNÝCH KROKOVÝCH  
MOTOROV  
BAKALÁRSKA PRÁCA

Študijný program: Informatika  
Študijný odbor: 2508 Informatika  
Školiace pracovisko: Katedra informatiky  
Školiteľ: doc. RNDr. Pavol Valko, PhD.

Bratislava, 2016  
Tomáš Wiedermann



Univerzita Komenského v Bratislave  
Fakulta matematiky, fyziky a informatiky

---

## ZADANIE ZÁVEREČNEJ PRÁCE

**Meno a priezvisko študenta:**

**Študijný program:** informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)

**Študijný odbor:** 9.2.1. informatika

**Typ záverečnej práce:** bakalárska

**Jazyk záverečnej práce:** slovenský

**Názov:**

**Cieľ:**

**Literatúra:**

**Kľúčové  
slová:**

**Vedúci:**

**Katedra:** FMFI.KI - Katedra informatiky

**Vedúci katedry:** doc. RNDr. Daniel Olejár, PhD.

**Dátum zadania:**

**Dátum schválenia:**

doc. RNDr. Daniel Olejár, PhD.  
garant študijného programu

.....  
študent

.....  
vedúci práce



## Abstrakt

Slovenský abstrakt v rozsahu 100-500 slov, jeden odstavec. Abstrakt stručne sumarizuje výsledky práce. Mal by byť pochopiteľný pre bežného informatika. Nemal by teda využívať skratky, termíny alebo označenie zavedené v práci, okrem tých, ktoré sú všeobecne známe.

**Kľúčové slová:** jedno, druhé, tretie (prípadne štvrté, piate)

## **Abstract**

Abstract in the English language (translation of the abstract in the Slovak language).

**Keywords:**

# Obsah

<b>Úvod</b>	<b>1</b>
<b>1 Príprava a úvodné nastavenia</b>	<b>2</b>
1.1 Hardware . . . . .	2
1.2 Operačný systém . . . . .	2
1.3 FTDI knižnica funkcií pre zariadenia typu D2XX . . . . .	3
1.3.1 Inštalácia . . . . .	3
1.3.2 Test funkčnosti . . . . .	4
1.3.3 Prehľad funkcií . . . . .	4
1.4 Požiadavky . . . . .	10
<b>2 API</b>	<b>11</b>
<b>3 Program</b>	<b>12</b>
3.1 Init . . . . .	12
3.2 prostredia . . . . .	12
3.2.1 cmd . . . . .	12
3.2.2 graficke . . . . .	12
3.2.3 webserver . . . . .	12
3.3 kompletizacia . . . . .	12
3.3.1 help . . . . .	12
3.3.2 man . . . . .	12
<b>Záver</b>	<b>13</b>

# Zoznam obrázkov



# Úvod

Cieľom mojej práce je naprogramovať program na ovládanie krokových motorov. Zameriavať sa budem na krokové motory vyrábané firmou Thorlabs. Dôvodov prečo som si zvolil práve túto firmu je viac, prvým je, že po dokončení práce, bude program využívaný profesormi na Slovenskej technickej Univerzite. Ďalším je možnosť testovania programu pri vývoji, keďže k zariadeniu mám prístup a navyše aj pre ich rozšírenosť a fakt, že firma poskytuje inštrukčnú sadu potrebnú k ovládaniu ich zariadení. Momentálne firma poskytuje ovládací program, ale iba pre operačné systémy Windows, ktoré nie sú na tento účel vhodné z dôvodu potreby licencie a náročnosti na výkon. Preto v mojej práci bude vývoj orientovaný na distribúcie operačného systému Linux.

# Kapitola 1

## Príprava a úvodné nastavenia

Táto kapitola sa priamo netýka vývoja programu, ale špecifikácie hardwaru a softwaru použitého pri vývoji. V kapitole sú uvedené aj nastavenia potrebné pre komunikáciu so zariadením. Špecifikácia napríklad ovplyvňuje možnosti využitia systémových volaní, preto na konci kapitoly sú uvedené aj požiadavky potrebné na inštaláciu a spustenie programu.

### 1.1 Hardware

Hardwarová špecifikácia zariadenia, na ktorom je program kompilovaný a spúšťaný. Najdôležitejším parametrom je procesor, jeho architektúra a inštrukčná sada, keďže priamo ovplyvňuje kompiláciu.

- Názov: Raspebrry Pi 2 model B
- Procesor: Cortex A7 900MHz
- Architektúra: ARMv7-A
- Grafický čip: Broadcom Videocore IV 250MHz
- Operačná pamieť: 1GB

### 1.2 Operačný systém

Operačný systém a verzia jadra systému hovoria, ktoré systémové volania a prepínače môžu byť použité. Aj keď Linux-ové a Unix-ové systémy podporujú štandarty POSIX, existuje niekoľko rozdielov, ktorým bude vhodné sa pri vývoji vyhnúť.

- OS: GNU/Linux
- Distribúcia : Arch

- Veria jadra: 4.0.7

## 1.3 FTDI knižnica funkcií pre zariadenia typu D2XX

Pre ovládanie zariadenia je potrebné doplniť systém o knižnicu príkazov narábajúcich s chipom v zariadení. Zariadenie obsahuje chip vyrábaný firmou Future Technology Devices International, ktorá poskytuje ovládače a programátorský manuál pre volania funkcií.

### 1.3.1 Inštalácia

Všetky potrebné súbory sa dajú stiahnuť na adrese:

<http://www.ftdichip.com/Drivers/D2XX.htm>

Je potrebné vedieť architektúru operačného systému. Pri 32-bitových a 64-bitových operačných systémoch existuje len jedna možnosť, avšak pri architektúre ARM sú štyri, v závislosti od verzie architektúry a reprezentácie floating point (reálne čísla). Bežný užívateľ nemusí mať v týchto veciach prehľad a preto je vhodné vytvoriť spúsiteľný skript, ktorý z informácií získaných zo systému zvolí vhodný ovládač a užívateľa len informuje, poprípade mu dá možnosť voľbu zmeniť.

Uvediem len stručný popis inštalácie. Obsiahlejšia verzia v angličtine:

<http://www.ftdichip.com/Drivers/D2XX/Linux/ReadMe-linux.txt>

Pri inštalácii sú potrebné prístupové práva používateľa root!

1. `tar xfvz libftd2xx-<architektúra>-1.3.6.tgz`  
Rozbalenie archívu do aktuálnej zložky.
2. `su root`  
Prihlásiť ako používateľ root. Alternatívne môžeme použiť `sudo -s`
3. `cp build/libftd2xx.* /usr/local/lib`  
Skopíruje potrebné súbory do lokálnych knižníc systému.
4. `chmod 0755 /usr/local/lib/libftd2xx.so.1.3.6`  
Zmení prístupové práva tak, aby aj ostatní používatelia mohli k súboru prístupovať, ale bez možnosti zápisu.
5. `ln -sf /usr/local/lib/libftd2xx.so.1.3.6 /usr/local/lib/libftd2xx.so`  
Vytvorenie symbolického linku ku knižnici.
6. `exit`  
Odhlásenie z používateľa root.

### 1.3.2 Test funkčnosti

Pre test funkčnosti stačí skompilovať niektorý z príkladov. Po rozbalení archívu sa nachádzajú v zložke examples. Dobrým príkladom je čítanie z EEPROM. Makefile a zdrojový kód sú poskytnuté. Ak skompilovaný program po spustení zlyhá a zahlásí chybu pri otvorení portu najprv treba skontrolovať aktívne moduly systému. Pomocou príkazu lsmod získame aktuálne aktívne moduly. Ak sa medzi nimi nachádzajú moduly s názvami ftdi\_sio a usbserial, tak ich treba aj napriek intuícií vypnúť príkazom rmmod pri každom pripojení zariadenia alebo moduly úplne zakázať, čo ale môže spôsobiť problémy s inými zariadeniami. Príkaz rmmod vyžaduje prístupové práva používateľa root, preto je vhodné vytvoriť spustiteľný skript, ktorý bude mať zvýšené práva a moduly odstráni. Ak problém stále pretrváva, program nerozozná zariadenie a treba upraviť zdrojový kód o inštrukciu FT\_SetVIDPID s príslušnými parametrami. V našom prípade je VID = 0x0403 a PID = 0xFAF0. TODO kde vytiahnuť vid pid pre rôzne distribúcie, automatizovať v programe

### 1.3.3 Prehľad funkcií

V tejto časti sú uvedené knižničné funkcie špecifické pre komunikáciu s chipom, ktoré sú v programe využité. TODO - opísať WORD, doubleWORD...? , pridať použité funkcie a vymazať nepoužité

#### **FT\_SetVIDPID**

##### **Definícia**

FT\_STATUS FT\_SetVIDPID ( unsigned int VID, unsigned int PID )

##### **Popis**

Pridá kombináciu identifikátorov do tabuľky zariadení. Potrebné pre nahratie správneho ovládača do systému.

##### **Parametre**

VID - identifikátor výrobcu zariadenia

PID - identifikátor produktu zariadenia

##### **Návratová hodnota**

FT\_OK alebo chybový kód v prípade zlyhania.

#### **FT\_CreateDeviceInfoList**

##### **Definícia**

FT\_STATUS FT\_CreateDeviceInfoList ( unsigned int\* NumDevs )

**Popis**

Vytvorí zoznam zariadení typu D2XX a do smerníka v uloží počet zariadení pripojených do systému. Následne je možné alokovať potrebný priestor pre získanie kompletných informácií o zariadeniach.

**Parametre**

NumDevs - smerník na premennú do ktorej bude uložený počet zariadení

**Návratová hodnota**

FT\_OK alebo chybový kód v prípade zlyhania.

**FT\_GetDeviceInfoList****Definícia**

```
FT_STATUS FT_GetDeviceInfoList ( FT_DEVICE_LIST_INFO_NODE *pdest,  
                                unsigned int* NumDevs )
```

**Popis**

Uloží zoznam informácií o zariadeniach do poskytnutého pola, ktoré musí mať alokovanú dostatočnú veľkosť.

**Parametre**

pdest - smerník na pole pre uloženie zoznamu

NumDevs - smerník na počet zariadení

**Návratová hodnota**

FT\_OK alebo chybový kód v prípade zlyhania.

**FT\_GetDeviceInfoDetail****Definícia**

```
FT_STATUS FT_GetDeviceInfoDetail( unsigned int Index, unsigned int* Flags,  
                                unsigned int* Type, unsigned int* ID, unsigned int* LocID, char* SerialNumber,  
                                char* Description, FT_HANDLE *handle )
```

**Popis**

Z listu informácií o zariadeniach vyberie určené a informácie uloží do poskytnutých premenných.

**Parametre**

Index - index na vybratie zariadenia

Flags - smerník pre uloženie informácie

Type - smerník pre uloženie typu zariadenia

ID - smerník pre uloženie identifikátoru zariadenia

LocID - smerník pre uloženie lokácie, v Linuxe nepodporované

SerialNumber - smerník na začiatok pola znakov pre uloženie sériového čísla, ukončené ako bežný cstring

Description - smerník na začiatok pola znakov pre uloženie popisu, ukončené ako bežný cstring

\*ft\_handle - smerník na uloženie handle pre ďalšie narábanie so zariadením

#### **Návratová hodnota**

FT\_OK alebo chybový kód v prípade zlyhania.

## **FT\_Open**

#### **Definícia**

```
FT_STATUS FT_SetVIDPID ( int iDevice, FT_HANDLE *handle )
```

#### **Popis**

Pripojí zariadenie a uloží handle.

#### **Parametre**

iDevice - index zariadenia

\*handle - smerník na uloženie handle

#### **Návratová hodnota**

FT\_OK alebo chybový kód v prípade zlyhania.

## **FT\_OpenEx**

#### **Definícia**

```
FT_STATUS FT_OpenEx ( void* arg1, unsigned int Flags, FT_HANDLE *handle )
```

#### **Popis**

Pripojí bližšie špecifikované zariadenie a uloží handle. Špecifikovať zariadenie možno cez sériové číslo alebo deskriptor.

#### **Parametre**

arg1 - smerník na argument určujúci zariadenie

Flags - určuje typ poskytnutého argumentu

\*handle - smerník na uloženie handle

#### **Návratová hodnota**

FT\_OK alebo chybový kód v prípade zlyhania.

## **FT\_Close**

### **Definícia**

FT\_STATUS FT\_Close ( FT\_HANDLE handle)

### **Popis**

Odpojí zariadenie.

### **Parametre**

handle - handle zariadenia na odpojenie

### **Návratová hodnota**

FT\_OK alebo chybový kód v prípade zlyhania.

## **FT\_Read**

### **Definícia**

FT\_STATUS FT\_Read ( FT\_HANDLE handle, void\* buffer, unsigned int BytesToWrite, unsigned int\* BytesWritten )

### **Popis**

Prečíta dáta poslané zariadením. Čítanie dát sa ukončí, keď sa dosiahne BytesToRead alebo vyprší čas nastavený funkciou `TODO(link) FT_SetTimeout`. Pri vypršaní času sa uložia aj čiastočné dáta a funkcia vráti FT\_OK.

### **Parametre**

handle - handle zariadenia

buffer - smerník na buffer pre dáta

BytesToRead - počet bajtov na prečítanie

BytesReturned - smerník na počet prečítaných bajtov

### **Návratová hodnota**

FT\_OK alebo chybový kód v prípade zlyhania.

## **FT\_Write**

### **Definícia**

FT\_STATUS FT\_Write ( FT\_HANDLE handle, void\* buffer, unsigned int BytesToRead, unsigned int\* BytesReturned)

### **Popis**

Pošle dáta zariadeniu.

### **Parametre**

handle - handle zariadenia

buffer - smerník na buffer pre zapisované dáta

BytesToWrite - počet bajtov na zapísanie

BytesReturned - smerník na počet zapísaných bajtov

**Návratová hodnota**

FT\_OK alebo chybový kód v prípade zlyhania.

**FT\_SetBaudRate****Definícia**

FT\_STATUS FT\_SetBaudRate ( FT\_HANDLE handle, unsigned int BaudRate)

**Popis**

Nastaví rýchlosť komunikácie pripojeného zariadenia.

**Parametre**

handle - handle zariadenia

BaudRate - rýchlosť v počte bitov za sekundu

**Návratová hodnota**

FT\_OK alebo chybový kód v prípade zlyhania.

**FT\_SetDataCharacteristics****Definícia**

FT\_STATUS FT\_Write ( FT\_HANDLE handle, unsigned char WordLength, unsigned char StopBits, unsigned char Parity)

**Popis**

Nastaví formu dát pre komunikáciu.

**Parametre**

handle - handle zariadenia

WordLength - počet bitov v slove, možné nastavenia sú 7 alebo 8

StopBits - počet stop bitov, možné nastavenia sú 1 alebo 2

Parity - nastavenie pre kontrolný bit

**Návratová hodnota**

FT\_OK alebo chybový kód v prípade zlyhania.

**FT\_SetTimeouts**



**Definícia**

FT\_STATUS FT\_SetTimeouts ( FT\_HANDLE handle, unsigned int ReadTimeout, unsigned int WriteTimeout )

**Popis**

Nastaví čas, ktorý sa má čakať pri zápise a čítaní dát, ak sa nedosiahne požadovaný počet.

**Parametre**

handle - handle zariadenia

ReadTimeout - čas v milisekundách

WriteTimeout - čas v milisekundách

**Návratová hodnota**

FT\_OK alebo chybový kód v prípade zlyhania.

**FT\_Purge****Definícia**

FT\_STATUS FT\_Purge ( FT\_HANDLE handle)

**Popis**

Vymaže dáta pre čítania a výpis v zariadení.

**Parametre**

handle - handle zariadenia

mask - kombinácia pre vstupný a výstupný buffer

**Návratová hodnota**

FT\_OK alebo chybový kód v prípade zlyhania.

**FT\_ResetDevice****Definícia**

FT\_STATUS FT\_ResetDevice( FT\_HANDLE handle )

**Popis**

Reštartuje zariadenie.

**Parametre**

handle - handle zariadenia

**Návratová hodnota**

FT\_OK alebo chybový kód v prípade zlyhania.

## **1.4 Požiadavky**

TODO - na konci otestovať a doplniť(verzia kernelu, programy - gcc)

# Kapitola 2

## API

TODO - vytvorit api cez FT Write a messages

# Kapitola 3

## Program

### 3.1 Init

nastavenia

### 3.2 prostredia

#### 3.2.1 cmd

#### 3.2.2 graficke

#### 3.2.3 webserver

### 3.3 kompletizacia

#### 3.3.1 help

#### 3.3.2 man

# Záver

Na záver už len odporúčania k samotnej kapitole Záver v bakalárskej práci podľa smernice [?]: „V závere je potrebné v stručnosti zhrnúť dosiahnuté výsledky vo vzťahu k stanoveným cieľom. Rozsah záveru je minimálne dve strany. Záver ako kapitola sa nečísluje.“

Všimnite si správne písanie slovenských úvodzoviek okolo predchádzajúceho citátu, ktoré sme dosiahli príkazmi `\glqq` a `\grqq`.

# Literatúra

- [1] Kernighan Brian W. and Ritchie Dennis M. *The C Programming Language*. Prentice Hall Software Series, 1988.
- [2] Frank B. Brokken. *C++ Annotations*. University of Groningen, 2015.
- [3] Future Technology Devices International (FTDI). D2xx programmer's guide, 2012. Manuál k chipom firmy FTDI, ktoré komunikujú cez port USB 2.0.
- [4] Michael Kerrisk. *Linux Programming Interface*. No Starch Press, 2010.
- [5] *FCNTL(2) Linux Programmer's Manual*, 2015. fcntl(2) manuál k fcntl.
- [6] Linux user's manual, 2015. Manuálové stránky poskytované systémom. Dostupné v každej distribúcii.
- [7] Thorlabs. Thorlabs host-controller communication's protocol, 2015. Komunikačný protokol poskytovaný výrobcom.