Středoškolská technika 2012

Setkání a prezentace prací středoškolských studentů na ČVUT

DOMOVNÍ PŘÍSTUPOVÝ SYTÉM SE SNÍMAČEM OTISKŮ PRSTŮ

Martin Vitásek

Střední průmyslová škola elektrotechniky a informatiky, Ostrava, příspěvková organizace
Kratochvílova, 7/1490, Ostrava - Moravská Ostrava,702 00

Středoškolská odborná činnost

Domovní přístupový systém se snímačem otisků prstů

Martin Vitásek

Ostrava 2012

Středoškolská odborná činnost

Obor 10 – elektrotechnika, elektronika a telekomunikace

Domovní přístupový systém se snímačem otisků prstů

Autor:	Martin Vitásek
Škola:	Střední průmyslová škola elektrotechniky a informatiky Ostrava, příspěvková organizace, Kratochvílova 7
Obor a ročník:	Elektrotechnika, 3. ročník

© Martin Vitásek, 2012

Prohlášení

Prohlašuji, že jsem svou práci vypracoval samostatně, použil jsem pouze podklady (literaturu, SW atd.) uvedené v přiloženém seznamu a postup při zpracování a dalším nakládání s prací je v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V Ostravě dne:…… ……

podpis autora

© Martin Vitásek, 2012

Anotace

Tato práce popisuje kompletní návrh přístupového systému se snímačem otisků prstů, aplikovatelného především v menších objektech. V první části práce je popsán přístupový terminál, jak z pohledu hardwaru, tak i firmwaru. V druhé části je pak popsána aplikace pro správu dat v databázi uživatelů, celková funkce systému z pohledu běžného uživatele a návod k použití.

Klíčová slova:

Snímač otisků prstů, otisk prstu, přístupový terminál, Microchip, mikrokontrolér, databáze

 © Martin Vitásek, 2012

Obsah:

Prohlášení 3
Anotace 4
1. ÚVOD 6
2. TERMINÁL 7
2.1 Hardware 7
2.1.1 Blokové schéma terminálu 7
2.1.2 Obvodové schéma terminálu 8
2.1.3 Předloha desky plošného spoje terminálu 9
2.2 Komponenty terminálu 11
2.2.1 Mikrokontrolér 11
2.2.2 Modul čtečky otisku prstu 11
2.2.3 Ostatní obvody 13
2.3 Mechanická konstrukce terminálu 13
2.4 Firmware 14
2.4.1 Komunikace s modulem čtečky otisků prstů 14
2.4.2 Komunikace s pamětí EEPROM 18
2.4.3 Komunikace s obvodem reálného času 19
2.4.4 Komunikace s PC 20
2.4.5 Celkový pohled na firmware 22
2.5 Databáze uživatelů 24
3. APLIKACE PRO PRÁCI S DATABÁZI UŽIVATELŮ 25
4. POPIS FUNKCE SYSTÉMU 26
5. NÁVOD K POUŽITÍ 26
6. ZÁVĚR 27
7. LITERATURA 28
8. PŘÍLOHY 28

© Martin Vitásek, 2012

1. ÚVOD

Cílem tohoto projektu je navrhnout domovní přístupový systém se snímačem otisků prstů. Tento systém je vhodný spíše pro menší objekty, což je dáno hlavně kapacitou databáze uživatelů. Při požadavku na využití v rozsáhlejších aplikacích s velkým počtem přístupů a z toho vyplývajícím vysokým objemem zpracovávaných a archivovaných dat by musela být zavedena centrální databáze uživatelů. Dále by byla nutná úprava firmwaru, tak aby mohlo komunikovat více terminálů mezi sebou. Celý systém se skládá ze dvou základních částí. První část tvoří přístupový terminál zahrnující v sobě také funkci alarmu. Druhou část tvoří aplikace pro počítač, pomocí které se spravují data v databázi uživatelů. Aplikace pro správu dat slouží pouze k nastavení systému, z toho vyplývá, že po nastavení terminál pracuje již zcela samostatně. Propojení terminálu s počítačem je řešeno pomocí sériového rozhraní.

Obr.1 fotka hotového terminálu

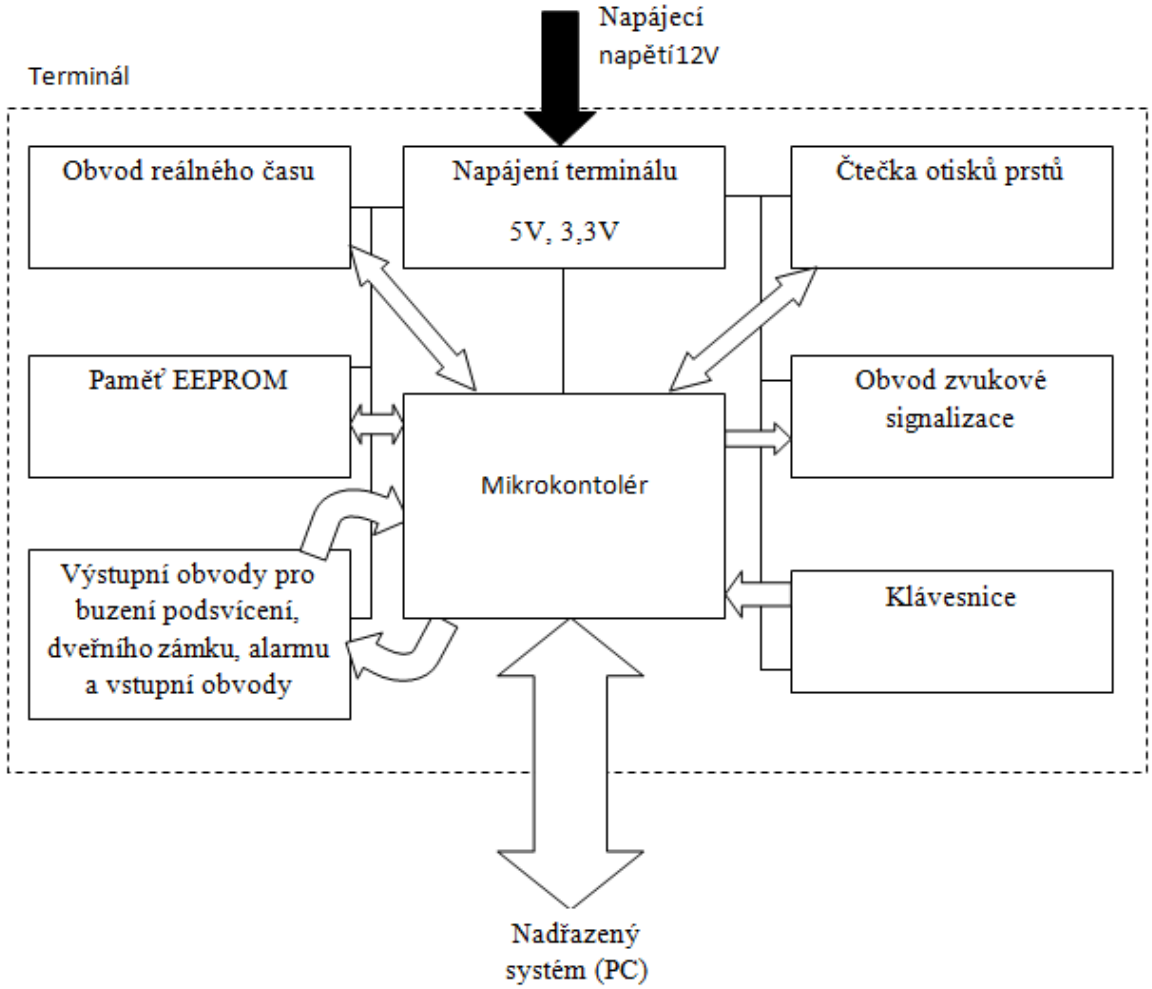
© Martin Vitásek, 2012

2. TERMINÁL

2.1. Hardware

2.1.1. Blokové schéma terminálu

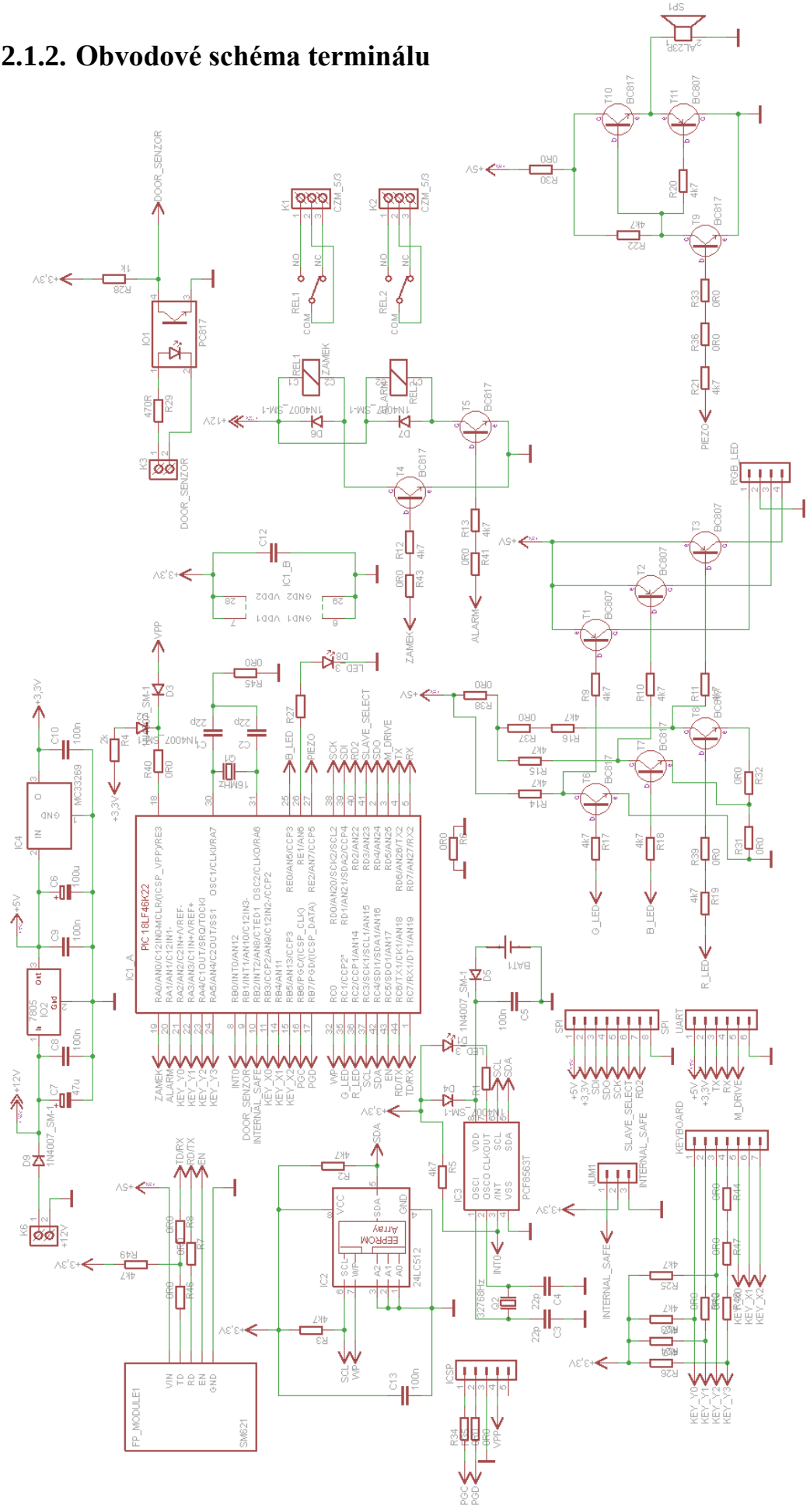
Na obr. 2 je znázorněno blokové schéma uspořádání jednotlivých součástí terminálu. Černými čárami je znázorněno napájení jednotlivých komponent. Prázdné šipky pak představují tok dat v systému. Jedním z nejdůležitějších prvků systému je mikrokontrolér, který má za úkol zprostředkovat komunikaci se všemi součástmi systému, vyhodnocovat data a řídit celý systém. Dalším nezbytně důležitým prvkem systému je modul čtečky otisků prstů, se kterým mikrokontrolér neustále komunikuje a vyhodnocuje data z tohoto modulu. Třetí nejdůležitější součástí systému je paměť EEPROM, ve které je uložena databáze uživatelů. Systém také obsahuje obvod reálného času, ve kterém se udržuje systémový čas, obvod pro zvukovou signalizaci, klávesnici a výstupní obvody pro buzení podsvícení, spínání elektromagnetického zámku a alarmu. Napájecí zdroj pak dodává jednotlivým komponentům potřebné napětí a proud.



Obr. 2 blokové schéma uspořádaní terminálu

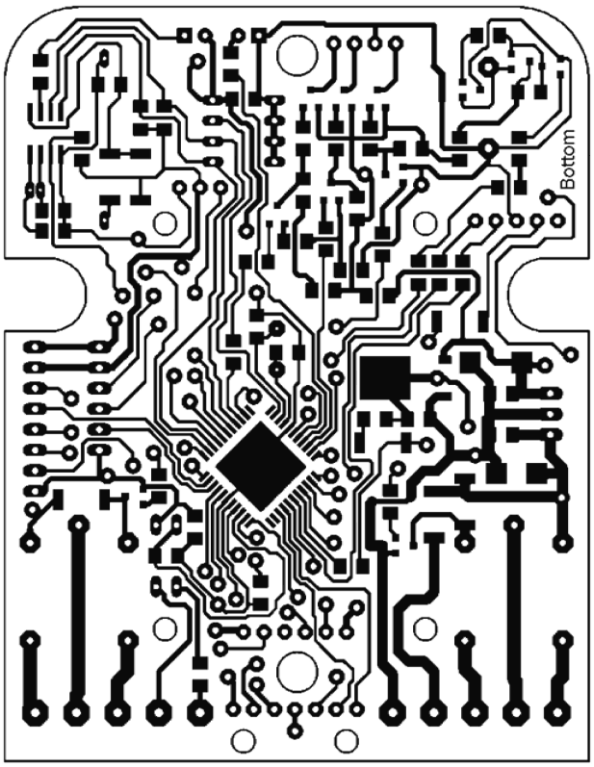
© Martin Vitásek, 2012

2.1.2. Obvodové schéma terminálu

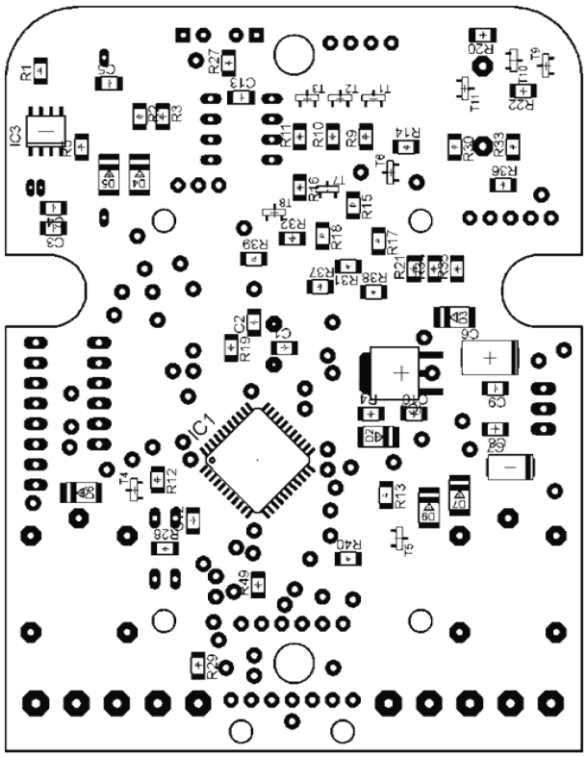


© Martin Vitásek, 2012

2.1.3. Předloha desky plošného spoje terminálu

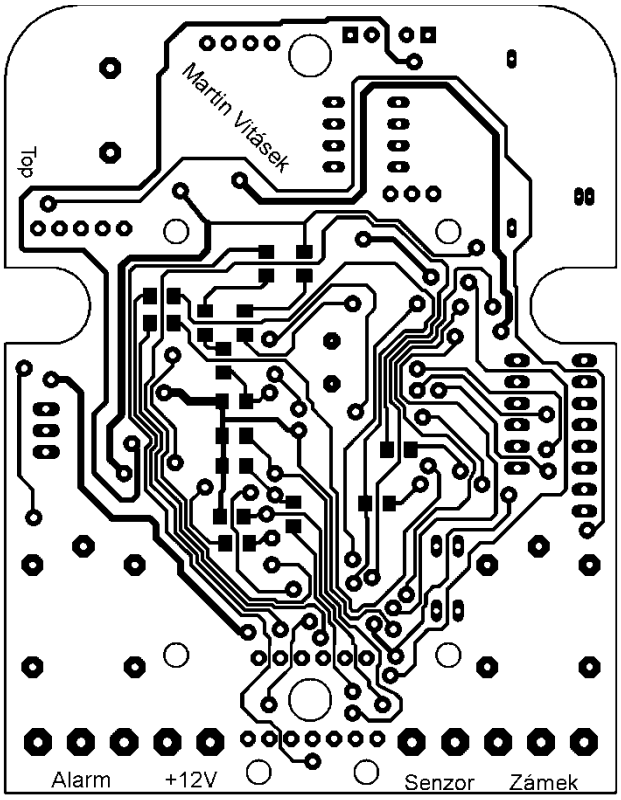


Obr. 3 předloha spodní strany plošného spoje

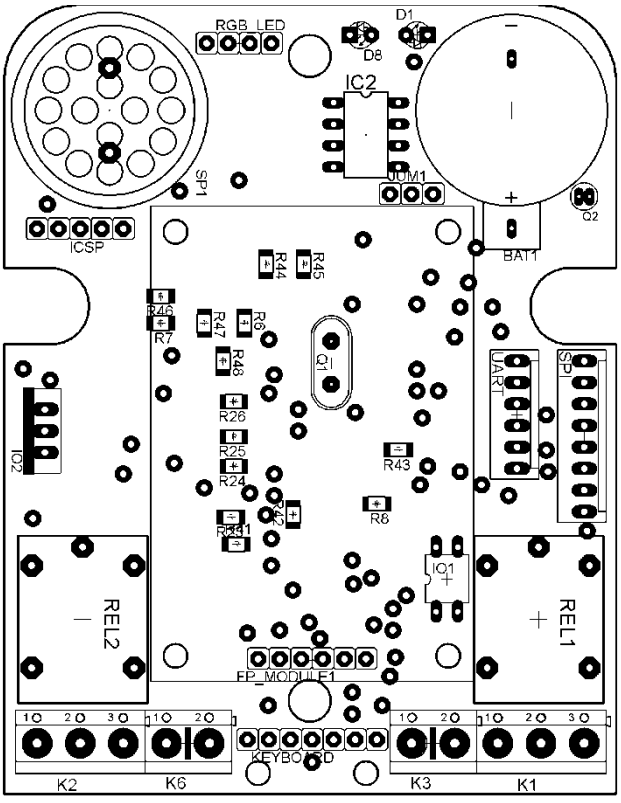


Obr. 4 osazovací plán spodní strany plošného spoje

© Martin Vitásek, 2012



Obr. 5 předloha vrchní strany plošného spoje



Obr. 6 osazovací plán vrchní strany plošného spoje

© Martin Vitásek, 2012

2.2. Komponenty terminálu

2.2.1. Mikrokontrolér

Po dlouhé rozvaze byl pro řízení celého systému vybrán osmibitový mikrokontrolér firmy Microchip PIC18LF46K22, který patří mezi nejnovější osmibitové mikrokontroléry firmy Microchip na trhu. Důvodem jeho volby je poměr cena/výkon. Také má dobré hardwarové vybavení vyhovující této aplikaci a jeho architektura je optimalizovaná pro programování v jazyce C. Základní vlastnosti tohoto mikrokontroléru jsou uvedeny v následující tabulce.

Device	Program Memory		Data Memory		I/O(1)	10-bit A/D Channels(2)	CCP	ECCP (Full-Bridge)	ECCP (Half-Bridge)	MSSP		EUSART	Comparator	CTMU	BOR/LVD	SR Latch	8-bit Timer	16-bit Timer
	Flash (Bytes)	# Single-Word Instructions	SRAM (Bytes)	EEPROM (Bytes)						SPI	I²C™							
PIC18(L)F46K22	64k	32768	3896	1024	36	30	2	2	1	2	2	2	2	Y	Y	Y	3	4

Cena ke dni 10.3.2012 v internetovém obchodě Farnell 78,9Kč (bez DPH) při odběru 1 – 9 ks

Tab. 1 Základní parametry mikrokontroléru PIC18LF46K22

2.2.2. Modul čtečky otisku prstu

Modul čtečky otisků prstů je jednou z nejdůležitějších součástí celého systému. Vzhledem k tomu, že čtečka otisků prstů obsahuje specifické součástky (jako je například optický hranol ve snímači), které nejsou běžně k sehnání a také z důvodu složitých algoritmů pro porovnávání otisků prstů, jenž jsou většinou know-how výrobců, nemá význam se doma pokoušet o vlastnoruční výrobu podobného modulu. Pro tento systém byl vybrán modul čtečky otisků prstů SM-621 výrobce Miaxis biometrics, jež je k sehnání u firmy Flajzar s.r.o. Modul se skládá ze dvou částí navzájem propojených plochým kabelem. První část obsahuje optický hranol a vlastní snímač otisků prstů. Druhou část tvoří deska plošného spoje, na níž je umístěna flash paměť o kapacitě 4Mbit pro uchování šablon otisků prstů, napájecí obvody a digitální signálový procesor TMS320 výrobce Texas Instruments. Čtečka komunikuje s nadřazeným systémem pomocí rozhraní UART a je vybavena signálem EN, pomocí kterého ji lze přepnout do pohotovostního režimu. Samotná komunikace s modulem probíhá v paketech, ale komunikační protokol bude popsán až v kapitole 2.4.1.

© Martin Vitásek, 2012

No.	Item	Parameter	Test Condition
1	System Power Supply	3.6V-7V	
2	Working Current	< 100mA	
3	Peak Current	< 120mA	5V
4	Fingerprint Image Enrollment Time	< 250ms	5V
5	1:1 Verification Time	< 600ms	Minutiae extraction + Fingerprint Verification
6	1:240 searching Time	< 2s	
7	Fingerprint Storage Capacity*	240/752/1776	
8	FAR	< 0. 001 %	
9	FRR	< 1.5 %	
10	External Interface	UART**	
11	Module Dimension	56.0×38.5×8.5mm	
12	Sensor Dimension	31×21×4.5mm	
13	Fingerprint Template Size	256 bytes	

*Template Storage Capacity is classified as three levels: 240, 752 & 1776.

**Baud rate 57600bps

Tab. 2 Základní technické parametry Modulu čtečky otisků prstů SM-621

Obr. 7 Modul čtečky otisků prstů SM621 firmy Miaxis Biometrics

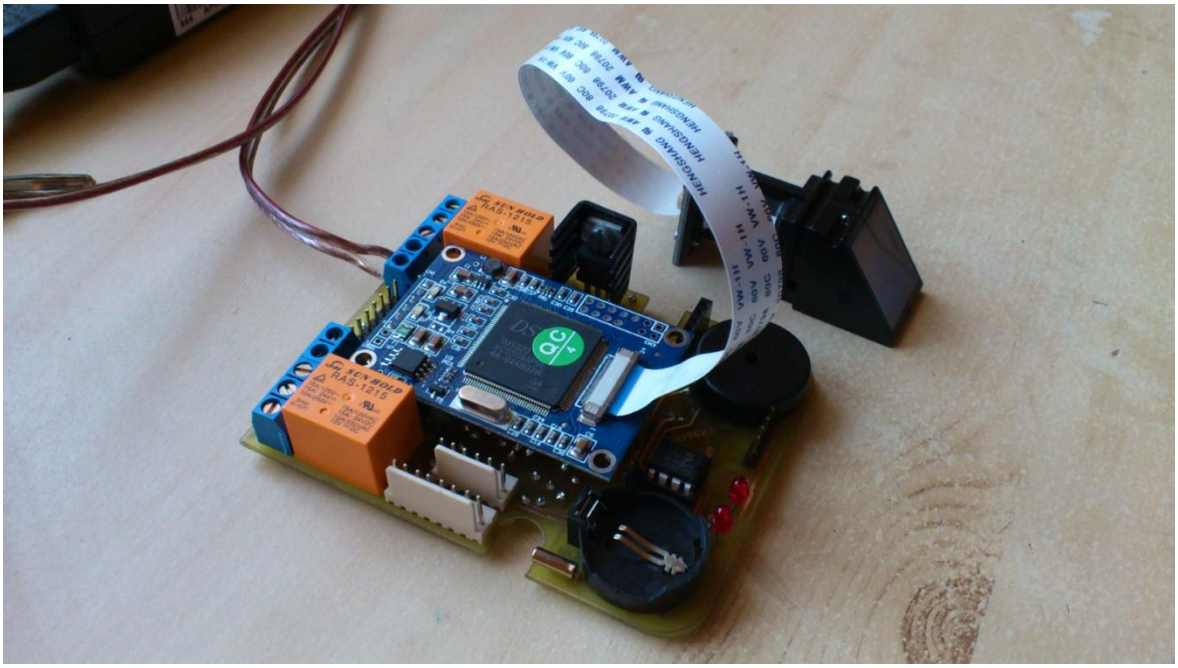
© Martin Vitásek, 2012

2.2.3. Ostatní obvody

Mezi ostatní obvody patří paměť EEPROM 24LC512, do níž se ukládá databáze uživatelů a systémová data. Další součástí je obvod reálného času PCF8563, ve kterém se udržuje systémový čas. Mezi důležité součásti patří výstupní obvody pro spínání elektromagnetického zámku a alarmu, které jsou tvořeny relé spínanými přes tranzistory. Nezbytné jsou i obvody signalizující stav systému, kde patří budící obvod pro podsvícení (podsvícení zde má funkci signalizace stavu systému) a piezoměnič. Terminál je také vybaven galvanicky odděleným vstupem pro připojení vnějšího snímače. Na desce jsou vyvedeny konektory pro připojení klávesnice, počítače (nutný převodník UART⇔RS232 nebo UART⇔USB) a Konektor, na němž je vyvedena sběrnice SPI pro možné rozšíření systému. Nechybí ani konektor ICSP pro pohodlné programování a odlaďování firmwaru přímo v aplikaci.

2.3. Mechanická konstrukce terminálu

Převážná většina použitých součástek je v provedení SMD. Díky použití SMD součástek bylo dosaženo optimální velikosti celého terminálu, což by při použití pouze obyčejných vývodových součástek nebylo možné. Celý terminál se skládá ze dvou desek plošných spojů, které jsou mezi sebou propojeny konektorem a připevněny na distančních sloupcích. Jednu desku tvoří zakoupený modul čtečky otisků prstů a druhou je deska obsahující „mozek“ celého systému a další podpůrné obvody. Zařízení je pak celé umístěno do plastové krabičky s plechovým čelním panelem.

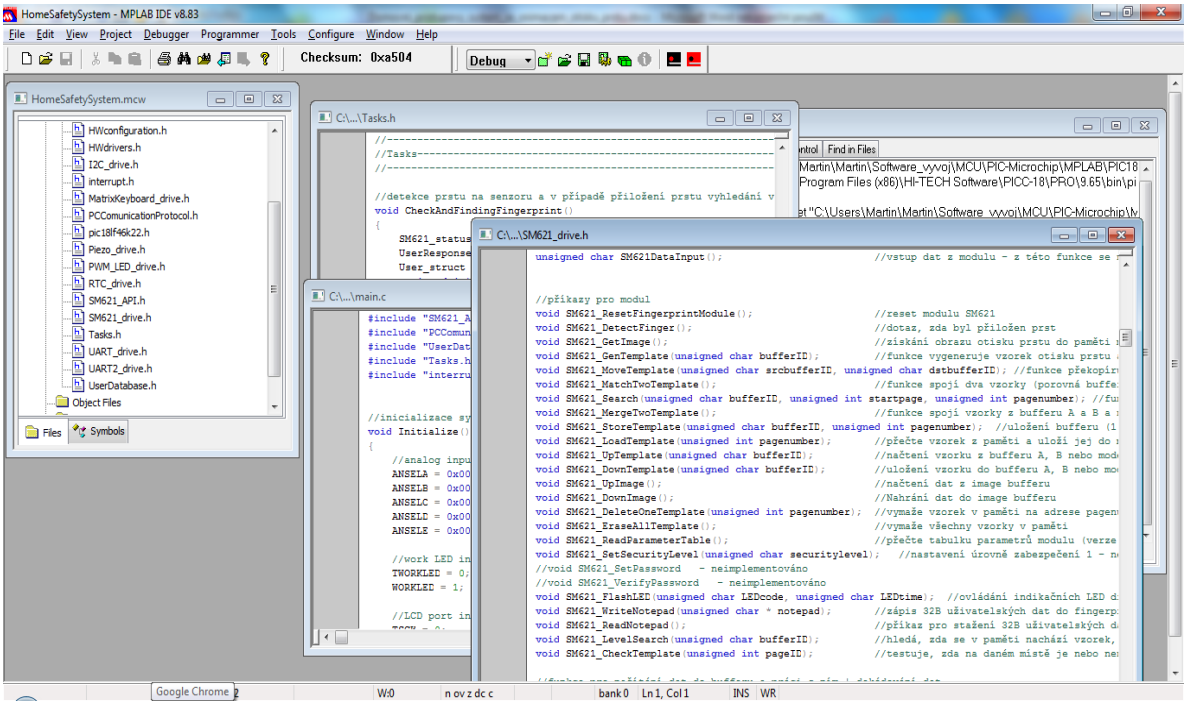


Obr. 8 „vnitřnosti“ terminálu

© Martin Vitásek, 2012

2.4. Firmware

Firmware pro mikrokontrolér PIC18LF46K22 byl napsán ve vývojovém prostředí MPLAB v jazyku C. Jelikož je celý firmware poměrně rozsáhlý, budou v následujících podkapitolách popsány pouze nejdůležitější částí.



Obr. 9 Vývojové prostředí MPLAB

2.4.1. Komunikace s modulem čtečky otisků prstů

Klíčovým bodem projektu bylo vytvoření knihovny pro komunikaci s modulem čtečky otisků prstů. Modul komunikuje s nadřazeným systémem pomocí UARTu rychlostí 56700baud. Přenos informací probíhá v paketech. Systém rozlišuje tři druhy paketů a to příkazový paket, datový paket a koncový datový paket. Příkazové pakety slouží k zasílání příkazů do modulu a datové pakety obsahují odpověď modulu. Obsah datových paketů pak záleží na konkrétním zaslaném příkazu.

1 byte	4 bytes	2 bytes	1 byte	… …				2 bytes
Packet Flag01	Equipment Address Code	Packet Length	Command	Parameter 1	Parameter 2	…	Parameter n	Check Sum

Obr. 10 Příkazový paket modulu SM-621

© Martin Vitásek, 2012

1 byte	4 bytes	2 bytes	128 bytes…	2 bytes
Packet Flag02	Equipment Address Code	Packet Length	Data	Check Sum

Obr. 11 Datový paket modulu SM-621

1 byte	4 bytes	2 bytes	N bytes…	2 bytes
Packet Flag08	Equipment Address Code	Packet Length	Data	Check Sum

Obr. 12 Koncový datový paket modulu SM-621

Všechny pakety jsou ohraničeny na začátku a konci značkou (0xC0), čímž je jednoznačně určen začátek a konec paketu. Díky použití značky označující začátek a konec paketu musí být kódováním zajištěno, že se uvnitř paketu nikdy tato hodnota neobjeví.

Kódování probíhá podle následujících pravidel:

Pokud paket obsahuje znak 0xC0, je nahrazen dvěma znaky – 0xDB a 0xDC
Pokud paket obsahuje znak 0xDB, přidá se za něj navíc znak 0xDD

Složení paketu:

PacketFlag – označuje typ paketu
AdressCode – čtečka tento parametr zatím ignoruje, posílají se 4 znaky 0x00
PaketLength – součet délek částí paketu Command a Parameters (data)
Command – kód příkazu
Parameters – parametry předávané příkazu, závisí na konkrétním příkazu
CheckSum – kontrolní součet počítaný přes celý paket

Modul čtečky otisků prstů SM-621 podporuje příkazy, které jsou uvedeny v tab. 3. Ve vytvořené knihovně pro zadávání příkazů a přijímání odpovědí jsou implementovány všechny příkazy modulu až na příkaz SetPwd a VfyPwd. Tato knihovna je napsána tak, že aplikace využívající tuto knihovnu volá funkce pro odeslání příkazu do modulu. Následně je z přerušení od UARTu při příjmu dat z modulu volána funkce pro dekódování dat a ukládání do vstupního bufferu. Při přijetí celého paketu s odpovědí je nastaven příslušný příznak, který signalizuje, že mohou být vyzvednuta data z bufferu. Jakmile je nastaven příznak přijetí celého paketu s odpovědí od modulu, obslužná aplikace zavolá funkci pro kontrolu paketu, která zkontroluje, zda se při přenosu nevyskytla nějaká chyba a pokud je příchozí paket v pořádku, je z něj přečtena hodnota confirmcode, nesoucí informaci o výsledku provedeného příkazu. Seznam možných confirmcode od modulu je uveden v tab. 4.

© Martin Vitásek, 2012

NO.	Command & Code		Function
1	DetectFinger	01H	Detect finger on sensor
2	GetImage	02H	Get image from sensor
3	GenTemplet	03H	Generate fingerprint minutiae from images
4	MoveTemplet	20H	Move characteristic file among three buffers
5	MatchTwoTemplet	04H	Accurate verification for characteristic files in CharBufferA and CharBufferB
6	Search	05H	Search part of or the whole fingerprint database based on minutiae files in CharBufferA or CharBufferB
7	MergeTwoTemplet	06H	Merge the files in CharBufferA and CharBufferB, generate template in ModelBuffer
8	StoreTemplet	07H	Store files in ModelBuffer to flash fingerprint database
9	LoadTemplet	08H	Read a template to ModelBuffer from flash fingerprint database
10	UpTemplet	09H	Upload files in characteristic buffer to HOST
11	DownTemplet	0AH	Download a characteristic file from HOST to characteristic buffer
12	UpImage	0BH	Upload images
13	DownImage	0CH	Download images
14	DeletOneTemplet	0DH	Delete a characteristic file in flash fingerprint database
15	EraseAllTemplet	0EH	Empty flash fingerprint database
16	ReadParTablel	0FH	Read Parameter List
17	SetSecurLecel	12H	*Set secure levels*
18	SetPwd	13H	Set handshaking password
19	VfyPwd	14H	*Verify handshaking password*
20	Reset	15H	System reset and go to initial status when power on
21	FlashLED	16H	*Light flash indication*
22	WriteNotepad	23H	Write notepad
23	ReadNotepad	24H	*Read notepad*
24	LevelSearch	25H	Conduct level search
25	CheckTemplet	28H	*Search fingerprint database*

Tab. 3 Seznam příkazů podporovaných modulem čtečky otisků prstů SM-621

© Martin Vitásek, 2012

No.	Confirm Code	Definition
1	00h	Command execution finished or OK;
2	01h	Packet receive error;
3	02h	No finger on sensor;
4	03h	Fingerprint image enrollment failure;
5	04h	Finger too dry or fingerprint too light and minutiae fail to be generated;
6	05h	Finger too wet or fingerprint unclear and minutiae fail to be generated;
7	06h	Fingerprint in great disorder and minutiae fail to be generated;
8	07h	Fingerprint image is normal, but minutiae are too few to be generated;
9	15H	There is no valid original image files in Buffer to generate fingerprint image;
10	08h	Fingerprints do not match;
11	09h	No fingerprint found;
12	0ah	Minutiae merge failure;
13	0bh	When store templates into database, address ID number is out of fingerprint database range;
14	0ch	Read templates from fingerprint database error;
15	0dh	Minutiae upload failure;
16	0eh	Module fails to receive subsequent data packets;
17	0fh	Image upload failure;
18	10h	Template deletion failure;
19	11h	Empty fingerprint database failure;
20	12h	Fail to sleep;
21	13h	Incorrect Password;
22	14h	System reset failure;
23	15h	Invalid fingerprint data in the current image buffer;
24	16h	Online upgrade failure;
25	17h	There is fingerprint remaining on sensor or pressing finger on sensor for too long;
26	18h	Operate FLASH error
27	19h	No valid templates in the designated position
26	1ah—ffh	Reserved.

Tab. 4 seznam možných confirmcode v příchozím paketu

Jako příklad je uveden příkaz pro detekci prstu na senzoru. Mikrokontrolér odešle do modulu paket, který je uveden na obr. 13. Modul po zpracování příkazu odešle paket s odpovědi, jenž je uveden na obr. 14. Pro více informací ohledně komunikace s modulem čtečky otisků prstů vás odkazuji do dokumentace tohoto modulu, neboť popis všech příkazů je velice rozsáhlý.

1 byte	4 bytes	2 bytes	1 byte	2 bytes
Packet Flag	Module Address	Packet Length	Command Code	Check Sum
01H	00H x 4	0001H	01H	0003H

Obr. 13 Paket příkazu detekuj prst na senzoru

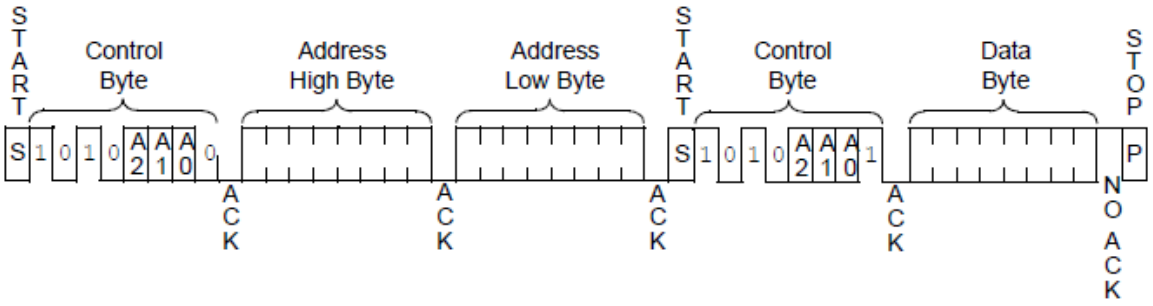
© Martin Vitásek, 2012

1 byte	4 bytes	2 bytes	1 byte	2 bytes
Packet Flag	Resvd	Packet Length	Confirm Code	Check Sum
07H	00H x 4	0001H	xxH	sum

Obr. 14 Paket odpovědi na příkaz detekuj prst na senzoru

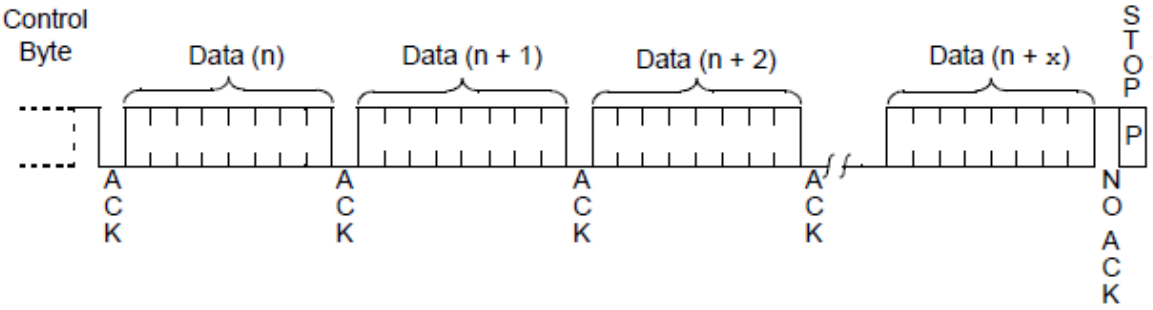
2.4.2. Komunikace s pamětí EEPROM

Mikrokontrolér komunikuje s pamětí EEPROM přes sběrnici I2C. V základní verzi této sběrnice je frekvence určena na 100kHz. Zde je využita vylepšená verze sběrnice I2C, jejíž frekvence je nastavena na 400kHz díky podpoře této komunikační rychlosti všemi komponenty připojenými na společnou I2C sběrnici (EEPROM a RTC). Paměť EEPROM umožňuje čtení dat jak po jednom bajtu, tak i po datových blocích. Zápis dat může probíhat také po jednotlivých bajtech, nebo po blocích bajtů. Při zápisu bloku více bajtů je nutno brát v potaz velikost vstupního bufferu paměti. Při zápisu bloku dat totiž nesmí přesáhnout adresa násobků velikosti tohoto bufferu, protože data by se nezačala zapisovat dále do paměti, jak by se mohlo zdát, ale zpět od začátku dané stránky paměti. Z toho vyplývá, že při zápisu do jiné stránky paměti je nutno s pamětí EEPROM zahájit komunikaci znovu. Samotná komunikace začíná vygenerováním startbitu na I2C sběrnici, následně je odeslána sedmibitová adresa součástky, se kterou se bude komunikovat a R/W bit, který určuje směr přenosu dat. Následně se odešle adresa buňky v paměti, se kterou se bude pracovat. Poté pokud je bitem R/W určeno, že se bude do paměti zapisovat, odešlou se data pro zapsání do paměti. Jestliže bylo R/W bitem oznámeno, že se bude z paměti číst, je znova vygenerován startbit, je odeslána adresa součástky a následně jsou přečtena data z adresy, která byla nastavena. Celá komunikace končí odesláním stopbitu na sběrnici.

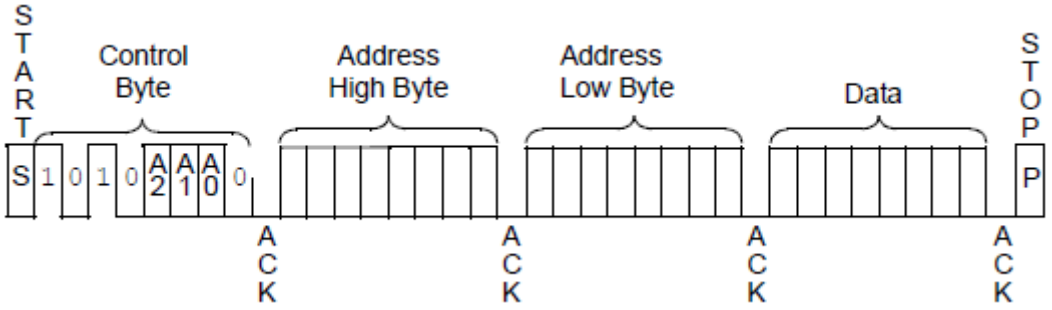


Obr. 15 čtení jednoho bajtu z paměti EEPROM

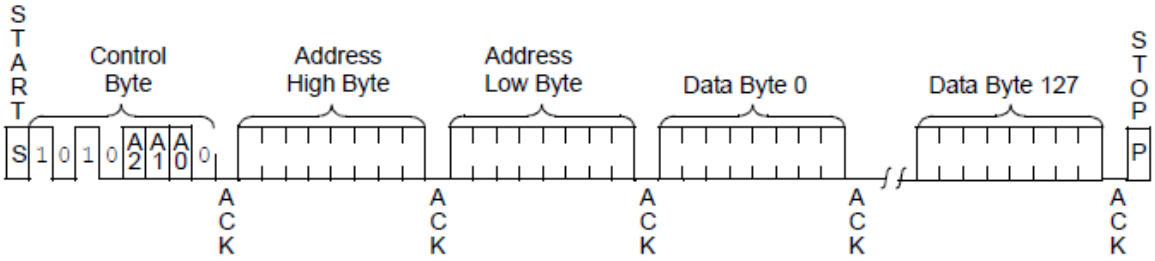
© Martin Vitásek, 2012



Obr. 16 čtení bloku dat z paměti EEPROM



Obr. 17 zápis jednoho bajtu do paměti EEPROM



Obr. 18 zápis bloku dat z paměti EEPROM

2.4.3. Komunikace s obvodem reálného času

Obvod reálného času je připojen taktéž na sběrnici I2C. Komunikace s tímto obvodem probíhá obdobně jako komunikace s pamětí EEPROM. Současná verze firmwaru využívá obvod reálného času pouze k odměřování doby zapnutí systému. V následujících verzích firmwaru se ale počítá s jeho lepším uplatněním, a to takovým, že se bude do databáze zaznamenávat čas přístupu jednotlivých osob.

© Martin Vitásek, 2012

2.4.4. Komunikace s PC

Pro komunikaci s aplikací pro PC, pomocí níž se spravují data v databázi uživatelů, byl vytvořen komunikační protokol podobající se komunikačnímu protokolu modulu čtečky otisků prstů. Přenos probíhá v paketech. Každý paket je ohraničen počáteční/koncovou značkou stejně jako u komunikačního protokolu modulu čtečky otisků prstů. Kódování dat uvnitř paketu je taktéž stejné. Velikost paketu není pevně dána a je omezena pouze velikostí přijímacích bufferů jak na straně terminálu tak PC. Samotnou velikost paketu si pak přijímací strana sama vypočítá. Za datový blok je přidán osmibitový cyklický kontrolní součet. Na obr. 20 je znázorněna základní struktura paketu.

1B	nB	1B	1B
Border mark (0xC0)	Data (n – bytes)	CRC – 8	Border mark (0xC0)

Obr. 20 struktura paketu komunikace s PC

Při komunikaci jsou rozeznávány dva typy paketů, příkazový a paket odpovědi. Na obr. 21 a 22 je zobrazena struktura datové části v jednotlivých typech paketů.

1B	1B	nB
Typ paketu (0x01 – příkaz)	Kód příkazu	Parametry příkazu

Obr. 21 struktura datové části v příkazovém paketu

1B	1B	nB
Typ paketu (0x02 – odpověď na příkaz)	Kód příkazu, ke kterému se vztahuje odpověď	Odpověď

Obr. 22 struktura datové části v paketu odpovědi

Příkazové pakety odesílá aplikace běžící na PC, kterými určuje terminálu, co má provést. Pakety odpovědi pak odesílá terminál do PC. Struktura odpovědi závisí na daném příkazu. V tab. 5 jsou uvedeny příkazy a odpovědi na ně podporované aktuální verzi firmwaru v terminálu.

 © Martin Vitásek, 2012

číslo	příkaz	kód	funkce	parametry	odpověď
1	Get and store fingerprint	0x01	Sejme otisk prstu ze senzoru při přiložení prstu a uloží do paměti	adresa otisku prstu v paměti modulu (1B)	úspěšnost operace (1B), chybový kód (1B)
2	Cancel get and store fingerprint	0x02	Zruší příkaz pro získání otisku prstu a uložení do paměti	žádné	úspěšnost operace (1B)
3	Erase all fingerprints and users	0x03	Smaže všechny otisky prstů v paměti a všechny uživatele	žádné	úspěšnost operace (1B), chybový kód modulu(1B), chybový kód eeprom (1B)
4	Get adress of free position in database	0x04	Zadá příkaz o zaslání odpovědi s adresou volné pozice v databázi	žádné	úspěšnost operace(1B), adresa volného místa v databázi (1B)
5	Get number of users in database and their adress	0x05	Zadá příkaz o zaslání odpovědi a počtem uživatelů v paměti a s jejich adresami	žádné	úspěšnost operace(1B), počet uživatelů v databázi (1B), adresy všech uživatelů (n*1B)
6	Get user info	0x06	Zadá příkaz o zaslání všech informací o uživateli uloženého v paměti na dané adrese	adresa uživatele(1B)	úspěšnost operace(1B), adresa uživatele v paměti(1B), datová struktura jednoho uživatele (n*1B) (struktura uživatele 1B - příznaky, 32B - jméno)
7	Create new user	0x07	Vytvoří nového uživatele na dané adrese	adresa uživatele (1B), Jméno uživatele (max. 32B)	úspěšnost operace(1B), kód chyby (1B)
8	Delete One User	0x08	Vymaže jednoho uživatele z paměti	adresa uživatele(1B)	úspěšnost operace (1B), chybový kód modulu(1B), chybový kód eeprom (1B)
9	Delete One Fingerprint	0x09	smaže jeden otisk prstu z paměti modulu	adresa otisku prstu v paměti modulu(1B)	úspěšnost operace(1B), chybový kód (1B)
10	Erase all user's fingerprints	0x0a	smaže všechny otisky prstu jednoho uživatele	adresa uživatele(1B)	úspěšnost operace(1B)
11	Check One user's fingerprints	0x0b	Zkontroluje, zda jsou u uživatele na daných pozicích uloženy otisky prstů	adresa uživatele (1B)	úspěšnost operace(1B), adresa uživatele (1B), otisk1 je/není (1B), otisk2 je/není (1B), otisk3 je/není (1B), otisk4 je/není (1B)

Tab. 5 Seznam příkazů podporovaných aktuální verzí firmwaru a struktura odpovědí na ně

© Martin Vitásek, 2012

2.4.5. Celkový pohled na firmware

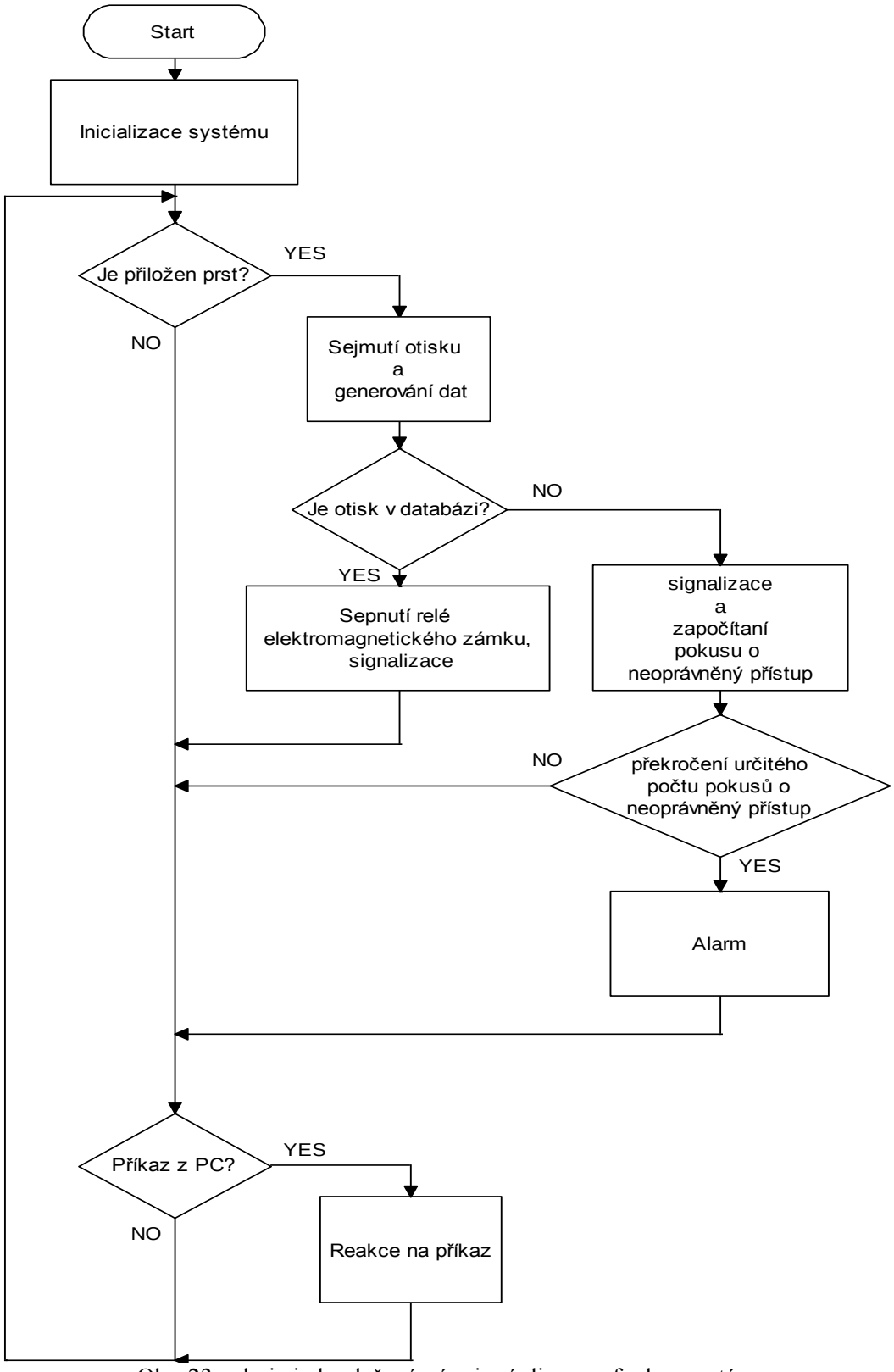
Firmware je ještě pořád ve vývoji. Současná verze firmwaru nepodporuje všechny hardwarové prostředky, které systém nabízí. Aktuální verze podporuje komunikaci s aplikací pro správu databáze uživatelů, zvukovou a světelnou signalizaci stavů systému, kontrolu otisků prstů a porovnávání s databází uživatelů, a následné reakce na správný, či nesprávný otisk prstu. Zahrnuje také funkci alarmu, který je spuštěn po několika pokusech o neoprávněný přístup za jednotku času. Na obr. 23 je zobrazen velmi zjednodušený vývojový diagram funkce celého systému.

Souhrn všech hlavních modulů firmwaru:

main.c	hlavní smyčka programu a inicializace systému
HWconfiguration.h	hardwarová konfigurace systému (definice portů)
HWdrivers.h	soubory s "drivery" pro hardware
I2C_drive.h	„driver" hardwarového modulu sběrnice I2C
interrupt.h	nastavení a obsluha přerušení
MatrixKeyboard_drive.h	skenování klávesnice
PCComunicationProtocol.h	komunikační protokol PC <=> terminál
Piezo_drive.h	zvuková signalizace
PWM_LED_drive.h	ovládání podsvícení
RTC_drive.h	„driver“ pro obvod reálného času
SM621_API.h	aplikační rozhraní pro práci s modulem čtečky otisků prstů
SM621_drive.h	„driver“ modulu čtečky otisků prstů
Tasks.h	jednotlivé akce systému
UART_drive.h	„driver“ hardwarové jednotky UART1
UART2_drive.h	„driver“ hardwarové jednotky UART2
UserDatabase.h	práce s databází uživatelů

Tab. 5 seznam všech souborů firmware

© Martin Vitásek, 2012



Obr. 23 velmi zjednodušený vývojový diagram funkce systému

 © Martin Vitásek, 2012

2.4.6. Databáze uživatelů

Databáze uživatelů je rozdělená na dvě základní části. První část tvoří vzorky otisků prstů, které jsou uloženy v modulu čtečky otisků prstů. Další část tvoří data o uživatelích (v současné verzi firmwaru pouze jméno uživatele). Data o uživatelích se ukládají do paměti EEPROM. Každému uživateli v databázi je přidělena paměť pro čtyři otisky prstů. Na obr. 24 je vyobrazena struktura ukládání dat do paměti a její adresování. Maximální počet uživatelů v databázi je aktuálním systémem omezen na 60. Po menších změnách v systému je možné maximální počet uživatelů v databázi zněkolikanásobit.

uživatel	adresa uživatele	adresa otisku prstu	struktura dat v paměti EEPROM
uživatel 1	0	0	systémové příznaky (1B)
		1	jméno uživatele (32B)
		2	
		3	
uživatel 2	1	4	systémové příznaky (1B)
		5	jméno uživatele (32B)
		6	
		7	
uživatel 3	2	8	systémové příznaky (1B)
		9	jméno uživatele (32B)
		10	
		11	
uživatel 4	3	12	systémové příznaky (1B)
		13	jméno uživatele (32B)
		14	
		15	
…	…	…	…
uživatel n	n-1	4*(n-1)	systémové příznaky (1B)
		4*(n-1)+1	jméno uživatele (32B)
		4*(n-1)+2	
		4*(n-1)+3	

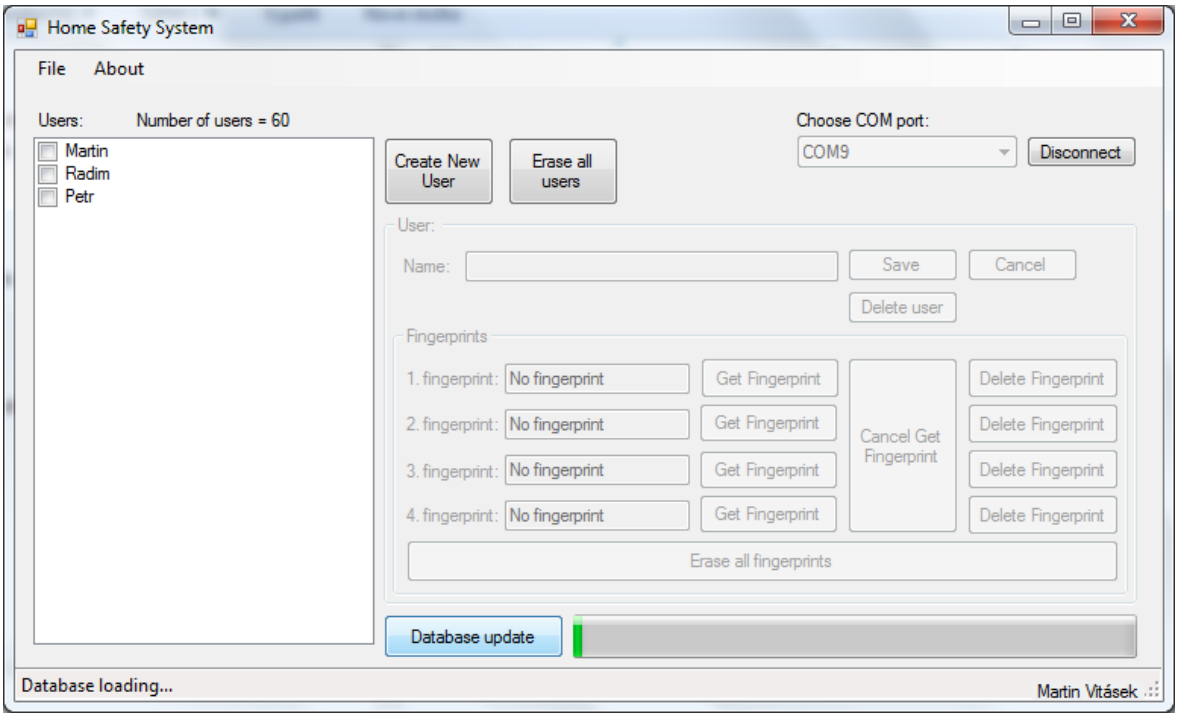
Obr. 24 struktura ukládání dat o uživatelích do paměti a její adresování

 © Martin Vitásek, 2012

3. APLIKACE PRO PRÁCI S DATABÁZI UŽIVATELŮ

Aplikace pro správu dat v databázi uživatelů byla napsána v jazyku C# ve vývojovém prostředí Microsoft Visual Studio 2010. Aplikace je založena na platformě NET. framework. Tato platforma byla navržena firmou Microsoft tak, aby každá aplikace používající tento framework využila maximální potenciál počítače, na kterém poběží, a byla optimalizována pro hardware daného počítače. Aplikace komunikuje s terminálem pomocí protokolu, který byl popsán v kapitole 2.4.4.

Popis uživatelského rozhraní aplikace:
Komunikace s terminálem probíhá přes sériový port, proto je po spuštění aplikace nutné vybrat port, ke kterému je terminál připojen. Po připojení k portu je z terminálu načtena databáze uživatelů. Průběh načítání databáze zobrazuje progres bar. V levé části se nachází seznam uživatelů, kteří jsou uloženi v databázi systému. Pro změnu informací u daného uživatele stačí kliknout na položku v seznamu uživatelů. Informace označeného uživatele jsou načteny do pravé části okna, kde lze tyto informace modifikovat. Uživateli lze změnit jméno, vymazat jej, načíst nebo vymazat jeho otisky prstů. Dále aplikace obsahuje tlačítko pro vytvoření nového uživatele a smazáni celé databáze. O aktuálním stavu, úspěchu či neúspěchu provedení jednotlivých operací aplikace informuje ve stavovém řádku.



Obr. 25 Okno aplikace při načítání databáze uživatelů

© Martin Vitásek, 2012

4. POPIS FUNKCE SYSTÉMU

Po spuštění se systém nachází v režimu kontroly otisků prstů, což je signalizováno modrou barvou podsvícení. Po přiložení prstu na senzor je sejmut otisk prstu, je vygenerován vzorek, který je porovnán se vzorky v databázi. Je-li vzorek nalezen v databázi, je barva podsvícení změněna na zelenou, spustí se zvuková signalizace posloupnosti vyšších tónů a sepne se relé elektromagnetického zámku. Doba sepnutí relé elektromagnetického zámku je tři sekundy. Po třech sekundách se zvukovou signalizací oznámí uplynutí doby sepnutí elektromagnetického zámku, barva podsvícení je změněna zpět na modrou a je vypnuto relé elektromagnetického zámku. Jestliže však není vzorek otisku prstu v databázi nalezen, změní se barva podsvícení na červenou a spustí se zvuková signalizace posloupnosti nižších tónů. Za jednu sekundu je barva podsvícení opět změněna na modrou. Systém také započítává pokusy o neoprávněný přístup. Pokud je během jedné minuty zaznamenáno pět pokusů o neoprávněný přístup, na dobu třicet sekund se sepne relé alarmu. Tento stav je signalizován červenou barvou podsvícení a zvukovou signalizací. Po tuto dobu je systém zablokován. Pokud ještě nejsou v databázi žádná data, je potřeba přidat uživatele, nebo upravit data uživatelů, musí být terminál připojen k počítači. Pomocí aplikace pro správu databáze uživatelů je možno databázi upravit. Při vytváření uživatelů je nutno získat otisky prstů do databáze. To se provede stiskem tlačítka pro získání otisku prstu. Stisknutím tohoto tlačítka přejde systém do režimu pro načtení a následné uložení otisku prstu do databáze. Tento stav je signalizován problikáváním červené a modré barvy podsvícení. Jakmile se přiloží prst na senzor, je sejmut otisk prstu, vygeneruje se vzorek a ten je uložen do paměti. Po uložení vzorku se systém přepne zpět do výchozího režimu. Jestliže bylo stisknuto tlačítko pro získání otisku prstu a systém je v režimu čekání na načtení otisku prstu do paměti, ale nechceme otisk prstu načíst, jednoduše stiskneme tlačítko pro zrušení čekání na sejmutí a uložení otisku prstu do paměti, čímž systém přepneme zpět do výchozího režimu. Dojde-li během činnosti systému ke kritické chybě, například selže-li komunikace s modulem otisků prstů je spuštěna přerušovaná zvuková signalizace.

5. NÁVOD K POUŽITÍ

Nejprve je nutné vytvořit databázi uživatelů, kteří budou mít povolen přístup do chráněného objektu.

A) Vytvoření a správa databáze uživatelů

1) Připojte terminál k počítači a zapněte jej.

2) Na počítači spusťte aplikaci pro správu databáze systému.

3) Po spuštění aplikace vyberte port, ke kterému je terminál připojen.

4) Pokud je vše bez problémů, aplikace si stáhne z terminálu databázi uživatelů.

5) Následně postupujte dle návodu pro vytvoření nového uživatele, nebo pro úpravu dat daného uživatele.

 © Martin Vitásek, 2012

B) Vytvoření nového uživatele

1) Pro vytvoření nového uživatele stiskněte příslušné tlačítko.

2) Zadejte jméno uživatele a uložte jej.

3) Nyní můžete k danému uživateli načíst až čtyři otisky prstů.

4) Stiskněte tlačítko pro získání otisku prstu.

5) Přiložte prst k senzoru otisků prstů na terminálu, po zaznění krátkého pípnutí je otisk prstu uložen.

6) Pokud jste tlačítko pro získání otisku prstu stiskli omylem, lze terminál přepnout zpět do normálního režimu stiskem tlačítka pro zrušení získání otisku prstu.

7) Otisk prstu lze z dané pozice také smazat, a to stisknutím příslušného tlačítka.

8) Všechny otisky prstů lze smazat stisknutím tlačítka pro smazání všech otisků prstů daného uživatele.

C) Úprava dat daného uživatele

1) Vyberte příslušného uživatele ze seznamu uživatelů v databázi.

2) Nyní můžete u daného uživatele změnit jméno, smazat nebo načíst nové otisky prstů.

D) Běžný pracovní režim terminálu (povolení/nepovolení přístupu do chráněného objektu)

1) Přiložte prst na senzor.

2) Pokud je váš otisk prstu nalezen v databázi uživatelů, ozve se příslušná zvuková signalizace, rozsvítí se zelené podsvícení a sepne se elektromagnetický zámek na dobu tří sekund. Po uplynutí doby tří sekund jste zvukovou signalizací upozorněni na vypnutí elektromagnetického zámku a barva podsvícení se změní na modrou.

3) Pokud váš otisk prsu nebyl nalezen v databázi uživatelů, ozve se příslušná signalizace a na jednu sekundu se barva podsvícení přepne na červenou.

4) Jestliže je během jedné minuty zaznamenáno pět po sobě následujících otisků prstů, které nejsou v databázi, je na dobu 30 sekund sepnut alarm. Po dobu alarmu je barva podsvícení červená a systém je zablokován.

** Při práci s aplikací pro správu databáze sledujte stavový řádek, ve kterém aplikace oznamuje svůj aktuální stav, případně probíhající akce a informace o úspěchu, či neúspěchu provedení jednotlivých operací.*

 © Martin Vitásek, 2012

6. ZÁVĚR

Součástí celého projektu byl návrh a sestavení hardwaru přístupového terminálu, vývoj firmwaru pro terminál a vývoj aplikace pro správu databáze uživatelů v terminálu. Všechny základní cíle, které jsem si položil při vypracovávání tohoto projektu, byly splněny. I když aktuální verze firmwaru pro mikrokontrolér nepodporuje všechny funkce, které hardware nabízí, jsem se současným stavem zařízení i odvedenou prací spokojen. Jelikož se systém neustále vyvíjí, nemusejí být některé zde uvedené informace aktuální. Další verze firmwaru již budou podporovat také zadávání kódu pomocí klávesnice, ukládání času přístupu jednotlivých osob do databáze a další funkce. Také bude vylepšena aplikace pro správu databáze.

7. LITERATURA

MIAXIS BIOMETRICS. Datasheet SM-621 Fingerprint Verification Module User Manual, [online], dostupné na: http://www.flajzar.cz/data/files/235-FPM1.pdf

MICROCHIP. Datasheet PIC18LF46K22, [online], dostupné na: http://ww1.microchip.com/downloads/en/DeviceDoc/41412E.pdf

MICROCHIP. Datasheet 24LC512, [online], dostupné na: http://ww1.microchip.com/downloads/en/devicedoc/21754e.pdf

NXP. Datasheet PCF8563, [online], dostupné na: http://www.nxp.com/documents/data_sheet/PCF8563.pdf

 © Martin Vitásek, 2012