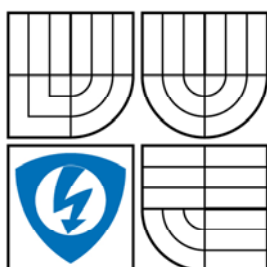


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

PROGRAMOVÉ VYBAVENÍ GAMA SPEKTROMETRU SOFTWARE FOR FIELD GAMASPECTROMETER

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. FILIP ADAMEC

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MIROSLAV ČEJKA, CSc.

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Adamec Filip, Bc.

Ročník: 2

ID: 89928

Akademický rok: 2007/08

NÁZEV TÉMATU:

Programové vybavení gamaspektrometru

POKyny PRO VYPRACOVÁNÍ:

Návrhněte SW vybavení pro polní spektrometr SatisGeo GS-512. SW má řešit uživatelské rozhraní a komunikaci s nadřazeným systémem

- 1/ Seznamte se s problematikou gama spektrometru a uveďte vlastnosti přístroje SatisGeo GS-512
- 2/ Navrhněte uživatelské rozhraní přístroje SatisGeo GS-512
- 3/ Realizujte uživatelské rozhraní, ověřte na reálném přístroji
- 4/ Analyzujte komunikační rozhraní s nadřazeným systémem, vyberte a ověřte vhodný komunikační protokol
- 5/ Zhodnoťte dosažené výsledky.


DOPORUČENÁ LITERATURA:

Firemní literatura GS-512

Termín zadání: 3.12.2007

Termín odevzdání: 26.5.2008

Vedoucí projektu: Ing. Miloslav Čejka, CSc.


prof. Ing. Pavel Jura, CSc.
předseda oborové rady



UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Vysoké učení technické v Brně

Fakulta Elektrotechniky a Komunikačních technologií

Ústav automatizace a měření

Programové vybavení gama spektrometru

Teze

Zaměření studie: Řízení a měření
Student: Bc. Filip Adamec
Vedoucí: Ing. Miroslav Čejka, CSc.

Abstrakt :

Tato práce se zabývá možnostmi řešení, implantací menu a komunikací z nadřazeným systémem (PC) gama spektrometru. V první kapitole autor pojednává o způsobu měření gama záření, o jeho zpracování a z toho vyplývajících požadavcích. V další kapitole je popsáno HW řešení procesorové desky spektrometru a jádro programu. Dále jsou zde rozebrány postupy jak zpracovávat vstup z klávesnice a výstup textové a grafické informace. Cílem třetí kapitoly je formulovat, jak má menu vypadat a jaké filozofické postupy budou v tomto menu použity, a představit čtenáři celkovou strukturu tohoto menu. Autor dále rozebírá, jaké budou potřeba menu, požadavky na vstup údajů od uživatelů, jakým způsobem se bude vstup těchto hodnot řešit a jejich omezení. Ve čtvrté kapitole autor uvádí možnosti komunikace s nadřazeným systémem, požadavky na tuto komunikaci a její řešení. Autor zde také popisuje, jaké požadavky musí být splněny z hlediska obou komunikujících platforem. V závěru autor shrnuje předchozí kapitoly a dosažené cíle.

Klíčová slova:

Gama spektrometr, Řízení menu, Komunikace s PC, USB, Vstup z klávesnice

Brno University of Technology

Faculty of Electrical Engineering and Communication

Department of Control, Measurement and Instrumentation

Software for field gamaspectrometer

Thesis

Specialisation of study: Control and Measurement

Student: Bc. Filip Adamec

Supervisor: Ing. Miroslav Čejka, CSc.

Abstract :

This work is about solution, implantation of menu and communication with a superior system (PC) of a gamaspectrometer. In first chapter, author describes how to measure gamma radiance, how to process the data and requirements for this. The next chapter describes HW solution of the processor board and the program core. In this chapter, possibilities are analyzed how to handle keyboard input, text and graphic output. Goals of the third chapter are to define how the menu should look like, basic principles of the menu control, and introduce the structure of the menu to the reader. Further, the author speaks of what sort of menus will be needed, of requirements on inputs from users and how this will be handled. In the fifth chapter, the author refers to possibilities of communication with superior system, requirements on this communication and how to handle this. The author also introduces further requirements from the point of view of two communication platforms. In the end, the author summarizes the previous chapters and reached goals.

Key words:

Gamaspectrometer, menu control, communication with PC, USB, input form keyboard

Bibliografická citace

ADAMEC, F. Programové vybavení gama spektrometru. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 75 s. Vedoucí diplomové práce Ing. Miloslav Čejka, CSc.

P r o h l á š e n í

„Prohlašuji, že svou diplomovou práci na téma Programové vybavení gama spektrometru jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne :

Podpis:

P o d ě k o v á n í

Děkuji tímto Ing. Miloslavu Čejkovi, CSc., Ing. Antonínu Fousovi a RNDr. Janě Křížové za cenné připomínky a rady při vypracování diplomové práce.

V Brně dne :

Podpis:

OBSAH

1. GAMA SPEKTROMETR A POŽADAVKY.....	12
1.1 měření gama spektra	12
1.1.1 Detekce gama záření.....	12
1.1.2 Zpracování signálu ze scintilační jednotky	12
2. POPIS SPEKTROMETRU SATISGEO GS-512	14
2.1 popis SW jádra přístroje.....	15
2.2 Spracování stisku tlačítka	17
2.3 zobrazovací subsystém.....	21
2.3.1 Zobrazování textu	21
2.3.1.1 Znaková sada	21
2.3.1.2 Vykreslování znakové sady	22
2.3.1.3 Výpis formátovaného textu	23
2.3.2 Vykreslování grafiky	24
3. ROZBOR MENU	28
3.1 Popis ovládacích prvků přístroje.....	28
3.2 Popis menu.....	28
3.2.1 Úvodní menu	29
3.2.2 Hlavní menu	29
3.2.2.1 Měření v módu SINGLE	30
3.2.2.2 Měření v módu SCAN.....	31
3.2.2.3 Měření MONITOR.....	31
3.2.2.4 Nastavení linky, pozice a kroku	32
3.2.2.5 SEPUT menu	32
3.2.2.5.1 Nastavení ROI	33
3.2.2.5.2 Nastavení konstant.....	33
3.2.2.5.3 Menu CALIBRATION.....	34
3.2.2.5.3.1 SET PAD PARAMETERS menu.....	35
3.2.2.5.4 STABILISATION menu	35
3.2.2.6 DATA MEMORY menu	36
3.2.2.6.1 Prohlížení paměti	37

3.2.2.6.2	Prohlížení dat	37
3.2.2.6.3	Prohlížení spektra	38
3.3	Rozbor požadavků na menu	39
3.3.1	Společné vlastnosti prvních čtyř skupin	40
3.3.2	Zadávání celého čísla	40
3.3.3	Zadávání desetinného čísla	42
3.3.4	Zadávání času	43
3.3.5	Zadávání data	43
3.3.6	Řešení omezení hodnot čísel a času	44
3.3.7	Řešení omezení zadávání data	45
3.4	řešení řízení menu	48
3.4.1	Řízení zvýraznění položky menu	48
3.4.2	Řízení menu pro vstup více položek	50
3.4.3	Zobrazení uložených položek	52
3.4.4	Zobrazení naměřených dat	54
3.4.5	Zobrazení spektra	54
3.4.6	Zobrazení měření v módu SCAN	58
3.4.7	Zobrazení chybových hlášek, varování a stavů	59
3.4.8	Zobrazování časově proměnných hodnot	60
4.	KOMUNIKACE Z NADŘAZENÝM SYSTÉMEM	61
4.1	Komunkace přístroje s PC	62
4.2	formát odesílaných a přijímaných dat	63
4.3	Komunikační protokol	65
4.4	odesílání dat do PC	67
4.5	Příjem kalibračních konstant z PC	71
5.	ZÁVĚR	72
6.	LITERATURA	74

SEZNAM OBRÁZKŮ

Obrázek 1 – blokové schéma spektrometru	14
Obrázek 2 – vývojový diagram zpracování stisku tlačítka v přerušení	18
Obrázek 3 – vývojový diagram fungování funkce GetKey	19
Obrázek 4 – vývojový diagram detekce držení tlačítka	20
Obrázek 5 – vývojový diagram zobrazení grafu	25
Obrázek 7 – úvodní menu	29
Obrázek 8 – Hlavní menu	29
Obrázek 10 – měření v módu SINGLE	30
Obrázek 11 – nastavení ROI v SCAN	31
Obrázek 12 – nastavené měření MONITOR	31
Obrázek 13 – Nastavení linky, pozice a kroku	32
Obrázek 14 – SETUP menu	32
Obrázek 16 – nastavení ROIS	33
Obrázek 17 – nastavení konstant	33
Obrázek 18 – CALIBRATION menu	34
Obrázek 20 – SET PAD PARAMETERS menu	35
Obrázek 21 – STABILISATION menu	35
Obrázek 23 – DATA MEMORY menu	36
Obrázek 25 – prohlížení paměti	37
Obrázek 26 – Prohlížení dat obrazovka 1	37
Obrázek 27 – Prohlížení dat obrazovka 2	38
Obrázek 28 – prohlížení spektra	38
Obrázek 29 - vývojový diagram funkce pro získání celého čísla od uživatele	42
Obrázek 30 - vývojový diagram omezení maximální možné zadávané hodnoty čísla	44
Obrázek 31 - vývojový diagram omezení data	47
Obrázek 32 - vývojový diagram řízení označení položky	49
Obrázek 33 - vývojový diagram pro vstup vícenásobných položek	51
Obrázek 34 - vývojový diagram zobrazování uložených hodnot	53
Obrázek 35 – vývojový diagram zobrazení spektra	57

Obrázek 36 – řízení zobrazení spektra	58
Obrázek 37 – vývojový diagram zobrazení SCAN.....	59
Obrázek 38 – Stavový automat komunikace s přístrojem.....	66
Obrázek 39 - navázání komunikace	67
Obrázek 40 – vývojový diagram odesílání dat do PC.....	68
Obrázek 41 - odesílání dat do PC.....	69
Obrázek 42 – stavový diagram příjmu dat na straně PC.....	70
Obrázek 43 - příjem kalibračních konstant z PC	71

SEZNAM TABULEK

Tabulka 1 – základní funkce jádra	17
Tabulka 2 – příkazy činnosti komunikace	66

SEZNAM ZKRATEK

Zkratka/Symbol	Jednotka	Popis
UCHAR		1B neznaménkové číslo
WORD		2B neznaménkové číslo
DWORD		4B neznaménkové číslo
plong		Ukazatel na long se stránkováním
int přístroj		2B celé číslo
int PC		4B celé číslo
long přístroj		4B celé číslo
long PC		8B celé číslo
short PC		2B celé číslo
float		4B desetinné číslo
tm		Struktura pro uložení času viz time.h
ADC		Analogově digitální převodník
MCU		Micro controler unit
ROI	[-]	Request Of Interest

1. GAMA SPEKTROMETR A POŽADAVKY

Gama spektrometr GS-512 je přístroj pro měření energetických spekter gama záření vznikajících při rozpadu radionuklidů. Detekce gama záření probíhá pomocí scintilační sondy GSP-3, která slouží pro měření přírodních a umělých radionuklidů a jejich kvantitativní určení.

Polní spektrometr se dá využít jak uvádí [6] pro analýzu vzorků, geofyzikální mapování, monitorování přirozené radiace prostředí, ohodnocení radioaktivity stavebních materiálů a antropogenních objektů.

1.1 MĚŘENÍ GAMA SPEKTRA

1.1.1 Detekce gama záření

U gama spektrometru se většinou využívá scintilačních detektorů pro jejich výhodné spektroskopické vlastnosti. Samotná detekční látka, scintilator, může mít značné rozměry a téměř libovolný tvar. Hmotnost scintilačních látek je přitom dostatečně velká, takže lze dosáhnout velké detekční účinnosti, zejména pro záření gama. Výstupní signál ze scintilačního detektoru je tak velký, že nevyžaduje použití velmi citlivých zesilovačů. Scintilační detektor má také velmi rychlou odezvu, což je pro spektroskopii velmi důležité, protože se měří četnosti částic se stejnou energií.

Ve scintilatoru vzniká dopadem částice sekundární emise fotonů. Tuto emisi je třeba zesílit a převést na elektrický signál. Toto se provádí pomocí fotonásobiče, jehož výstupní signál se dále zpracovává elektronicky. Scintilační látka spolu s fotonásobičem se vyrábí jako kompaktní celek, což přináší určité výhody (lepší sběr fotonů, optimální výběr a vzájemné přizpůsobení scintilatoru a fotonásobiče). Tomuto celku říkáme scintilační jednotka.

1.1.2 Zpracování signálu ze scintilační jednotky

Výstupní signál ze scintilační jednotky je nutné zesílit, popřípadě i tvarovat a následně pomocí špičkového detektoru měřit amplitudu impulzu, která odpovídá energii detekované částice. Následně je toto změřené špičkové napětí převedeno pomocí ADC. Při zpracovávání tohoto signálu je třeba detekovat signály větší, než je měřicí rozsah přístroje, jejichž počet se registruje. Tyto impulzy většinou odpovídají kosmickému záření z vysokou energií. Dále je také třeba generovat časové signály,

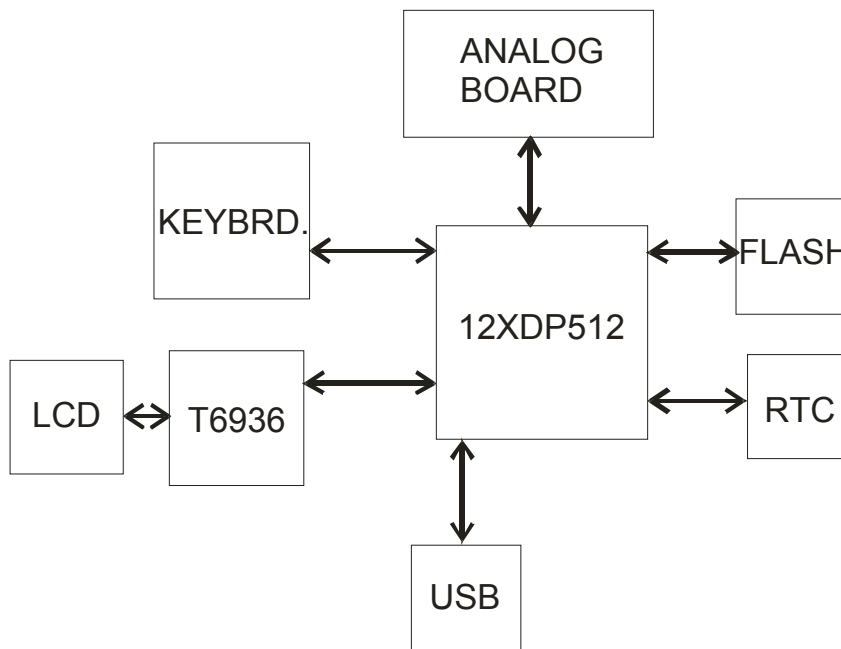
které odpovídají času strávenému převodem impulzu na číslo. To je tzv. mrtvá doba, kdy nelze detekovat další impulz/částici.

Následně je výstup z ADC zpracováván mikrokontrolerem, který vytváří, po určité době, pole četnosti impulzů ze stejnou energií. Protože parametry celého měřicího systému se s časem a teplotou mění, hlavně u scintilační jednotky, je třeba provádět stabilizaci. Pro stabilizaci se používá například cesiový izotop umístěný u scintilační jednotky. Následně se vyhodnotí jeho poloha ve spektru, které se opakovaně měří (většinou 20s) a ze známe polohy/energie tohoto izotopu se upraví zesílení měřicího řetězce.

Další úkoly CPU jsou zobrazování, ukládání, základní analýza naměřených spekter, vytvoření rozhraní mezi uživatelem a přístrojem, přenos dat do PC. Poslední dva úkoly budou popsány dále a jsou účelem této práce.

Vyčerpávající informace o jaderné elektronice a měření, ne sice nejnovější, ale zajímavé, se dají najít v [1,2].

2. POPIS SPEKTROMETRU SATISGEO GS-512



Obrázek 1 – blokové schéma spektrometru

CPU – z *Obrázek 1* vyplývá, je jako CPU použit mikrokontroler MC9S12XDP512MAL, který je v pouzdře LQFP 112. Tento typ nemá vyvedenou vnější sběrnici, proto jsou všechny periferie připojeny na univerzální porty a sběrnice je řešena programově. Dále obsahuje RISC koprocesor, který slouží pro vyčítání dat z ADC a pomoci něhož jsou vytvářena spektra. Podrobný popis viz [3].

- **Klávesnice** – stisk klávesy je vyhodnocován obvodem MM74C923, výstup tohoto obvodu je připojen na programovou sběrnici. Stisk klávesy je procesoru oznamován tímto obvodem na vstup vnějšího přerušení. Dokumentace viz [7].
- **T6963 (řadič displeje)** – tento řadič je připojen na grafický LCD displej s rozlišením 128x240 bodů a na programovou sběrnici. Podrobné informace viz [4].
- **USB** – jedná se o často užívaný obvod firmy FTDI FT245BM dokumentace viz [9]. Tento obvod je připojen na samostatnou programovou sběrnici sběrnici.
- **RTC** – jedná se o obvod reálného času a může generovat 1 a 2Hz signály, dle nastavení. Vývod, na kterém jsou tyto signály generovány, je připojen na vstup vnějšího přerušení CPU. Dokumentace viz [5].

- **FLASH** – jedná se o sériovou OR FLASH paměť připojenou na SPI modul CPU s kapacitou 16Mb.
- **ANALOG BOARD** – jedná se o analogovou desku, která slouží pro předzpracování signálu ze scintilační jednotky a obsahuje signály pro její řízení a sběr dat.

2.1 POPIS SW JÁDRA PŘÍSTROJE

Jádro programu je řešeno systémem zpráv podobně jako ve Windows. Tělo hlavního programu se vykonává ve funkci

```
void MessageLoop(WORD message, DWORD param1, DWORD param2)
```

kde první parametr je číslo zprávy a další dvě hodnoty jsou parametry zprávy. Jádro tedy zajišťuje pasivní čekání na události (stisk tlačítka, přerušení od USB, stav baterie aj.). Jednotlivé události jsou zařazovány do kruhové FIFO fronty, kde čekají na zpracování, což umožňuje zpracovat zprávy vzniklé při zpracovávání zprávy jiné.

Jádro také poskytuje ms časovače pro generování časových intervalů pro různé účely, funkce pro přístup k HW, zobrazování grafiky, přístup k uloženým datům aj. V následující tabulce budou popsány funkce jádra, které budou využity pro vytvoření a řízení systému menu.

Název Funkce	Popis funkce
Funkce jádra	
char SendMessage(WORD message, DWORD param1, DWORD param2)	Funkce odešle zprávu do aplikace
Funkce pro práci s časovači a časem	
void SetSekundTimer();	Zapnutí sekundového čítače
void KillSekundTimer();	Vypnutí sekundového čítače
char SetTimer(DWORD msperiod, UCHAR id)	Nastavení ms časovače
void KillTimer(UCHAR id);	Vypnutí ms časovače
void DisableOffTimer()	Vypnutí časovače pro automatické vypnutí
void SetOffTimer(DWORD speriod);	Nastavení časovače pro automatické vypnutí
void SetAlarm(tm* alarm);	Nastavení alarmu/budíku

<code>void CloseAlarm();</code>	Vypnutí alarmu/budíku
<code>void GetCoreTime(tm* ctime);</code>	Načtení aktuálního času
<code>void SetCoreTime(tm* ctime);</code>	Nastavení aktuálního času
Funkce pro čtení a zápis dat z USB	
<code>void ReadUSB(UCHAR* data, int length);</code>	Čtení z USB
<code>void WriteUSB(UCHAR* data, int length);</code>	Zápis do USB
Funkce pro povolení podsvícení a pípání	
<code>void EnableBI();</code>	Zapnutí/vypnutí podsvícení
<code>void DisableBI();</code>	
<code>void DisableBip();</code>	Zapnutí/vypnutí pípání
<code>void EnableBip();</code>	
Funkce datového rozhraní	
<code>int OpenFile(DWORD file_id);</code>	Funkce vytvoří novou hlavičku souboru pro zápis
<code>int WriteFile(PUCHAR data, WORD length);</code>	Zápis dat
<code>int ReadFile(PCHAR data, WORD length);</code>	Čtení dat
<code>int CloseFile();</code>	Zapíše data do FLASH
<code>int ScanFiles(BOOL first);</code>	Postupným voláním načítá uložené hodnoty od nejnovější k nejstarší
<code>void AddScanItem(FILE_TABLE* item);</code>	Přidá aktuální položku od, které se bude postupovat dále
<code>int ReadFileHead(FILE_TABLE* tabulka);</code>	Načte datovou hlavičku
Grafické a textové funkce	
<code>void Line(int ox, int oy,int dx, int dy);</code>	Funkce vykreslující čaru
<code>void ZobrazBuffer(plong buffer, long length, long max,int zesil);</code>	Funkce vykreslující naměřené spektrum
<code>void Graf(plong buffer, int length,char krok,long max);</code>	Funkce vykreslující graf při měření SCAN

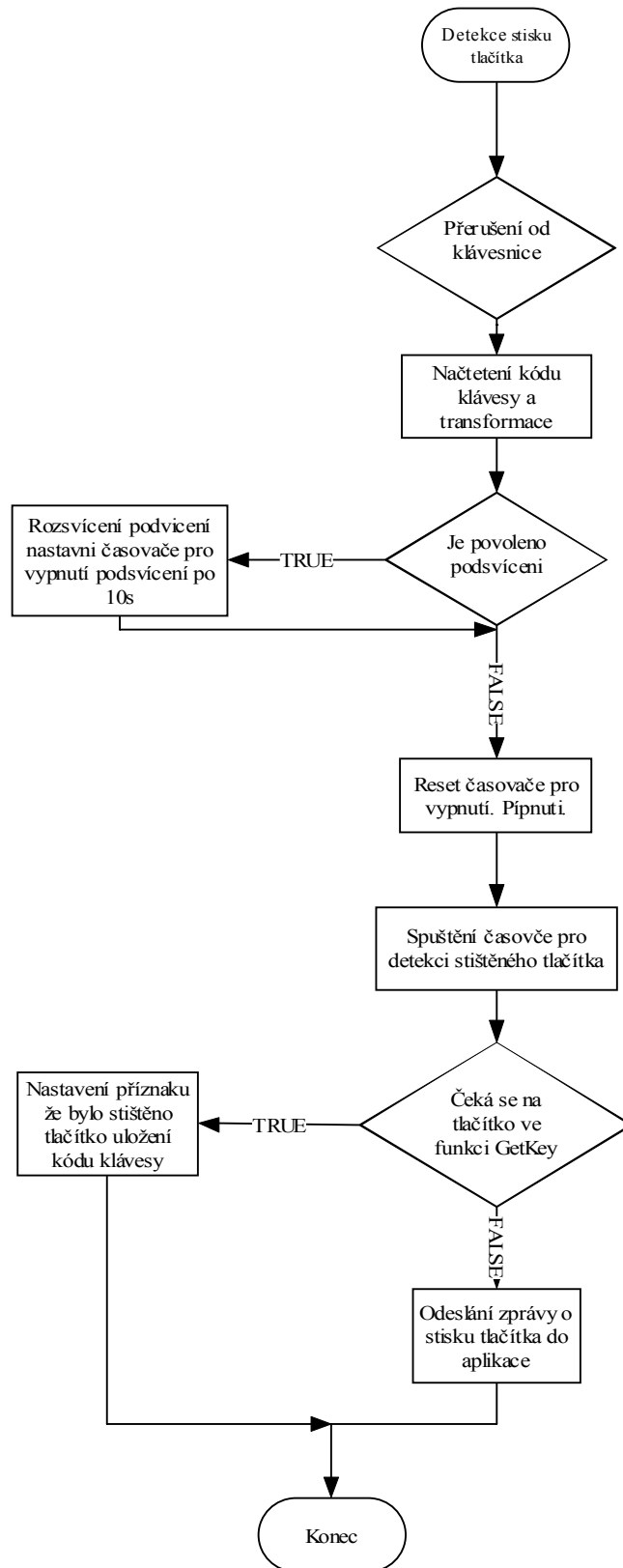
<code>void SetPixelCursor(char line, int pixel);</code>	Nastavení kurzoru
<code>int printf (const char * format, ...);</code>	Funkce pro formatovaný výstup
<code>void ClearDisplay();</code>	Funkce vymaže displej
<code>void WriteLine(int line);</code>	Funkce zapíše vybraný řádek do displej
<code>void DisplayWrite();</code>	Funkce zapíše veškerá data do displeje
Funkce klávesnice	
<code>char GetKey();</code>	Funkce pro pasivní čekání na stisk tlačítka v programu
<code>char GetLastKey();</code>	Funkce vrátí kód naposledy stisknutého tlačítka

Tabulka 1 – základní funkce jádra

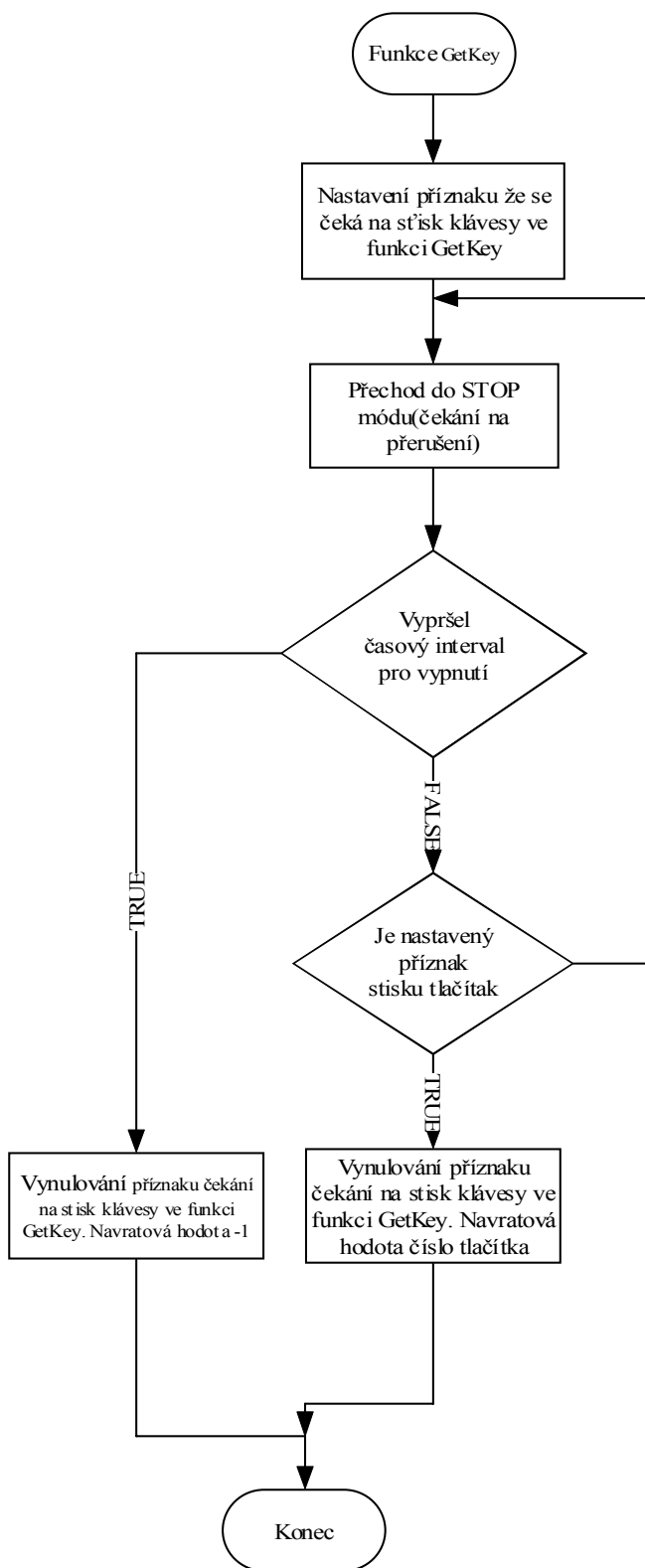
2.2 SPRACOVÁNÍ STISKU TLAČÍTKA

Jak již jsem se zmínil v kapitole 2 je stisk tlačítka vyhodnocován obvodem MM74C923. Tento obvod indikuje na vývodu DATA AVAILABLE, že byla stisknuta klávesa tento vývod je připojen na vnější přerušení do MCU, kde se jako odezva na toto přerušení přečte číslo stisknuté klávesy, v mém případě se ještě převede pomocí tabulky na jinou hodnotu. To z toho důvodu aby číselné hodnoty na klávesnici odpovídaly kódu klávesy. Následně se spustí časovač pro opakované testování stisku klávesy s intervalem 1s a následně 0,5s a pomocí systému zpráv se tato událost pošle ke zpracování aplikací. V případě že se čeká na stisk tlačítka ve funkci GetKey, se nastaví příznak, že bylo tlačítko stisknuto. Dále se musí při obsluze přerušení od klávesy resetovat časovač pro vypnutí přístroje, pokud je povoleno podsvícení tak ho zapnout a provést zvukovou signalizaci stisknutí tlačítka.

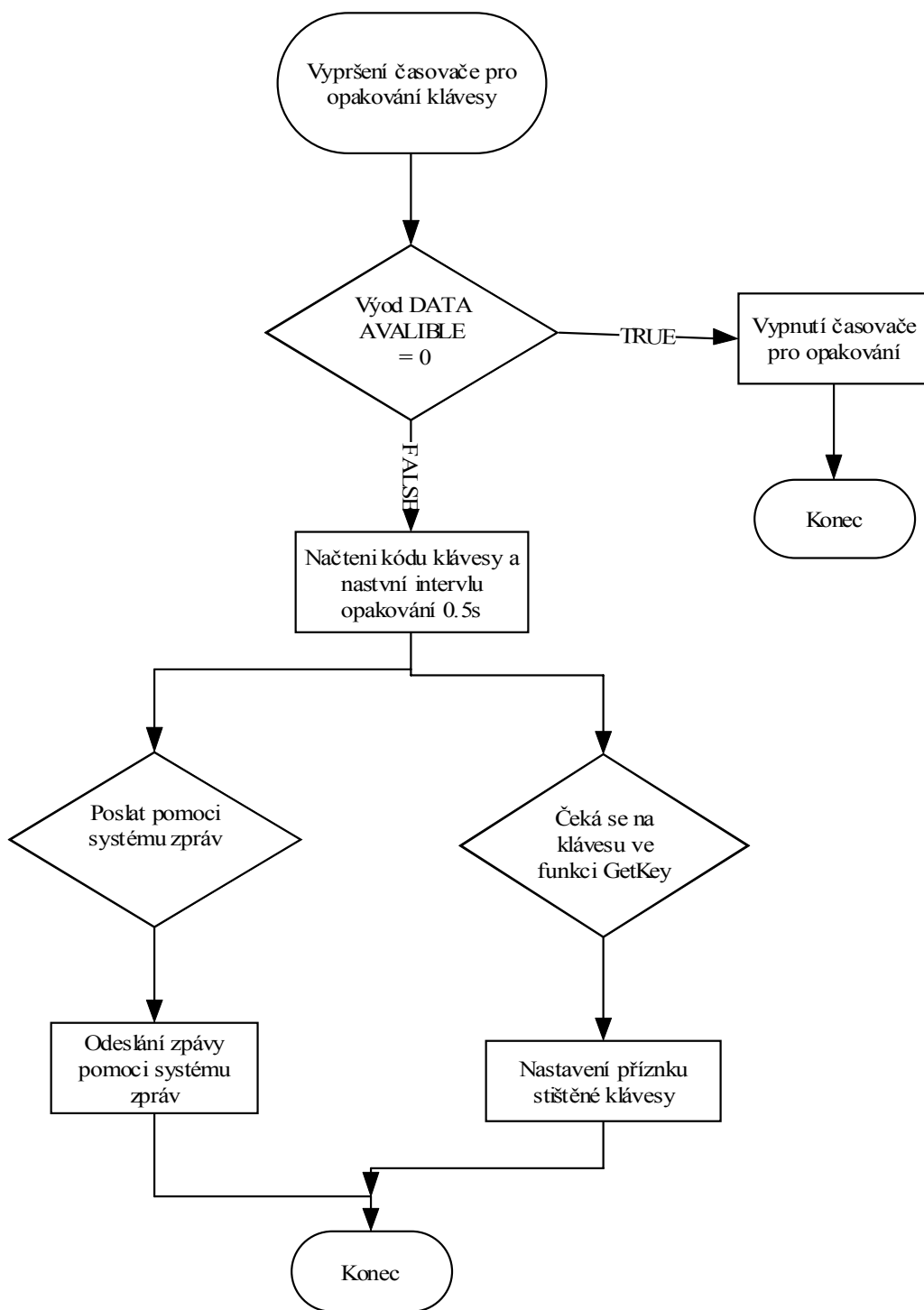
Funkce GetLastKey vrací kód naposledy stisknutého tlačítka. Jde o hodnotu, která se ukládá vždy při přerušení od klávesnice a používá se i pro předávání kódu klávesy s přerušení do funkce GetKey. Detekce stisku tlačítka a jeho zpracování viz následující vývojové diagramy.



Obrázek 2 – vývojový diagram zpracování stisku tlačítka v přerušení



Obrázek 3 – vývojový diagram fungování funkce GetKey



Obrázek 4 – vývojový diagram detekce držení tlačítka

2.3 ZOBRAZOVACÍ SUBSYSTÉM

Pro zobrazování dat je použit LCD displej s řadičem T6936, který umožňuje zobrazování jak textových (obsahuje HW generátor znaků), tak i grafických informací.

Protože znaky generované vnitřním generátorem znaku na displeji nejsou dobře čitelné, je třeba vytvořit generátor znaku programově.

Při zobrazování grafiky, v tomto případě jde o vykreslování počtu impulsů v jednotlivých kanálech a historie počtu impulsu ve vybrané ROI při měření SCAN. Pro tyto účely je třeba vytvořit funkce, které toto budou provádět.

Dále je třeba zvážit jakým způsobem se budou data do paměti řadiče zapisovat. Protože jak jsem se již zmínil budou znaky generovány programově, budou tedy reprezentovány jako bitmapa s různou šířkou. Z tohoto důvodu se musí při zápisu do displeje používat instrukci read-modify-write. Proto je lepší, z ohledem na rychlost, vykreslovat všechny grafické informace do bufferu v MCU a následně je všechny zapsat do paměti řadiče pomocí instrukce DATA AUTO WRITE.

Pokud podobným způsobem zvážíme vykreslování grafu na displej dojdeme ke stejnému výsledku.

2.3.1 Zobrazování textu

2.3.1.1 Znaková sada

Jednotlivé znaky ve znakové sadě jsou reprezentovány jako bitmapy s různou šířkou. Při vypisování znaku to však nevadí, vždy je třeba buď překreslit celý řádek nebo displej. Protože ve vybrané znakové sadě má nejširší znak délku větší než 16 pixelů je potřeba tyto hodnoty uložit jako 32b celé číslo bez znaménka. Z tohoto vyplývá, že počet těchto hodnot pro jednotlivé znaky bude odpovídat výšce znaku v pixelech.

Takto vytvořená znaková sada je následně reprezentována jako dvojrozměrné pole, kde první rozměr tohoto pole určuje znak ASCII – 33, první znak je tedy mezera. Další položka obsahuje hodnoty pro zobrazení s tím, že první s těchto hodnot udává šířku znaku v pixelech a hodnoty pro znaky A, B a C by vypadaly následovně.

{11,0,0,30,102,74,74,20,36,60,68,132,263,256,1536,0,0,0}, //A
{11,0,0,62,193,401,273,534,555,545,289,66,252,0,0,0,0,0}, //B
{11,0,0,30,33,65,129,128,256,256,256,256,260,136,112,0,0,0}, //C

2.3.1.2 Vykreslování znakové sady

Jak již jsem se zmínil v úvodu bude se text nejprve zapisovat do bufferu v MCU, který se následně zapíše do displeje. Tento buffer bude reprezentován dvojrozměrným polem, kde první položka tohoto pole odpovídá řádku a druhá obsahuje data pro daný řádek pro jednotlivé body displeje. Dále je třeba abychom měli informaci o aktuální pozici kurzoru. Tedy o textovém řádku, na který se bude vypisovat text a konci textu na tomto řádku v pixelech. Tyto hodnoty si nazveme *cur_line* a *cur_pos*, které jsou pro vymazání displeje nastaveny na 0,0. Dále je třeba zvážit jaké speciální znaky budeme používat. V mém případě to je to posuv na nový řádek (0xa). Ostatní speciální znaky není potřeba a předpokládá se že nebudou požívány, nebudou nějak zvlášť ošetřeny. Výpis znaku do bufferu bude tedy vypadat následovně:

```
void TERMIO_PutChar(char znak)
{
    int i;
    char pos;
    DWORD temp;
    DWORD hodnota;
    DWORD* psave;

    // posuv o řádek
    if(znak == 0xa)
    {
        cur_line++;
        cur_pos = 0;
        return;
    }
    //úprava znaku ASCII na odpovídající znak z fontu
    pos = znak - FONT_FIRST_CHAR;
    //pokud se další znak nevejde na řádek skok na nový řádek
```

```
//zde je třeba dát pozor na to že při zápisu na konci paměti
//displeje může dojít k zápisu na špatnou adresu, je teda
//třeba aby bylo pole pro řádek displeje o 4B delší
if(cur_pos + font[pos][0]) > 240)
{
    cur_line++;
    cur_pos = 0;
}
//pokud je aktuální řádek textu mimo displej nezapisuji nic
if(cur_line > 9)
    return;
//zápis bitmapy znaku do bufferu
for( i = 0; i < FONT_HEIGHT; i++)
{
    temp = 0;
    //uložení posledního vypsaného bajtu v bufferu do MSB bajtu
    //proměnné temp
    *(((char*)&temp)) = disp_buf[cur_line*FONT_LENGTH + i][cur_pos/8];
    //načtení pixelu fontu pro aktuální řádek
    hodnota = font[pos][i+1];
    //posunutí hodnoty doleva o zapsaný počet bitu v posledním
    //bajtu
    temp |= (hodnota<<(32-font[pos][0]-cur_pos%8));
    //přetypování posledního zapsaného bajtu na ukazatel DWORD*
    psave = (DWORD*)(disp_buf[cur_line*FONT_LENGTH + i] + cur_pos/8);
    //uložení hodnoty bitmapy pixelu pro aktuální řádek do bufferu
    *psave = temp;
}
//aktualizace aktuální pozice kurzoru na řádku
cur_pos += font[pos][0];
}
```

2.3.1.3 Výpis formátovaného textu

Protože jako vývojové prostředí je použit CodeWarrior, který poskytuje standardní knihovny C obsahující i funkci *printf*, která standardně pro tisk znaků používá funkci *void TERMIO_PutChar(char znak)*. Není tedy nic jednoduššího než

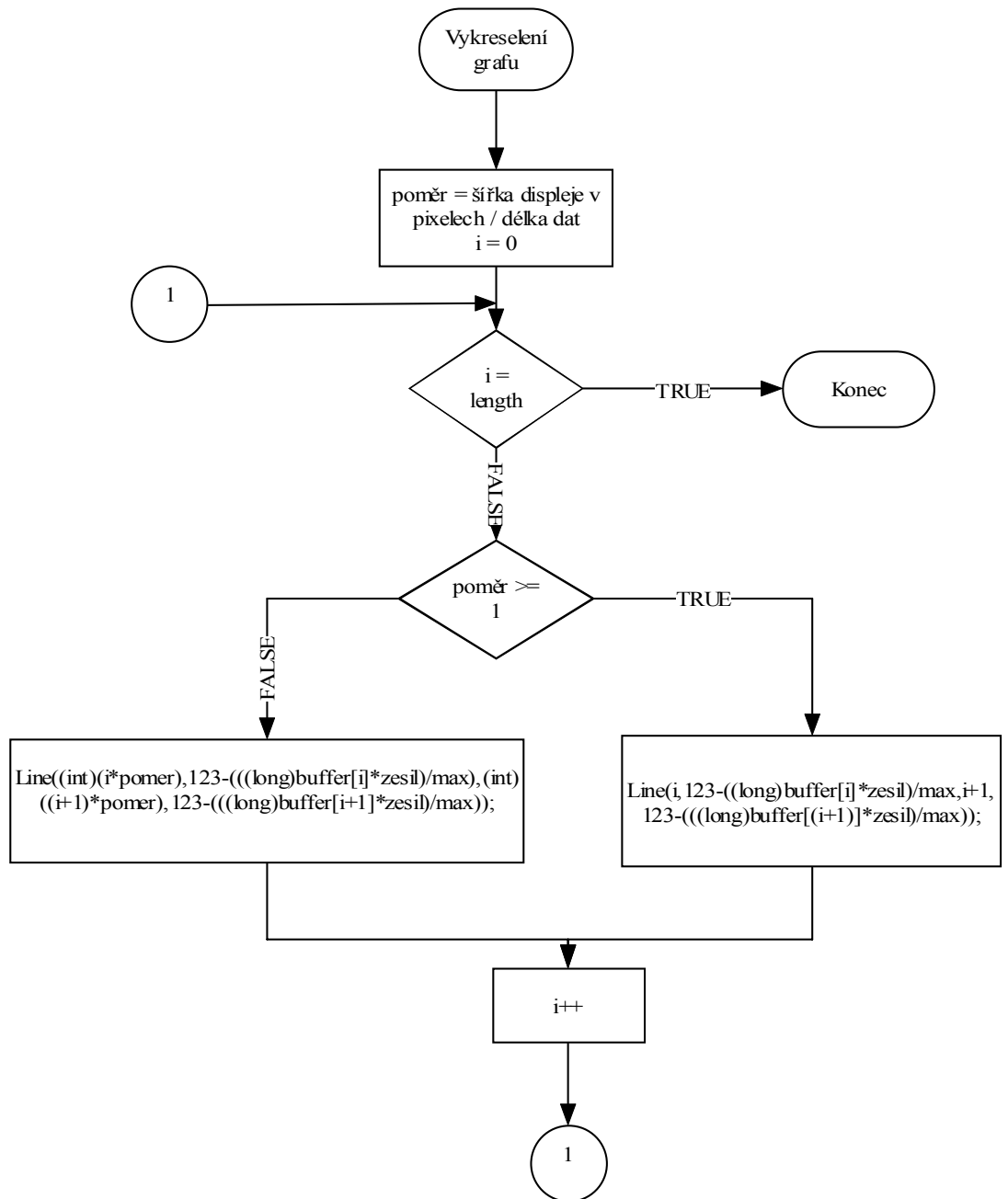
v projektu definovat tuto funkci s tím, že vypisuje znaky na displej, v mém případě do grafického bufferu, který se následně zapíše do paměti displeje.

Dále však potřebujeme minimálně další dvě funkce pro výpis znaku i grafiky. Tyto funkce jsou *ClearDisplay()* a *DisplayWrite()*. První funkce vynuluje grafický buffer a druhá ho zapíše do paměti displeje pomocí příkazu řadič DATA AUTO WRITE. Další funkce, která se používá je funkce *void WriteLine(int line)*. Tato funkce zapíše textový řádek na displej. Použití této funkce se předpokládá tam kde se mění jenom jedna hodnota ve zobrazované položce menu například čas, nebo změna stavu napětí baterie, zvýraznění nové pozice v menu.

2.3.2 Vykreslování grafiky

Pro vykreslování grafiky u tohoto přístroje potřebujeme vykreslovat graf četnosti počtu impulsu v jednotlivých kanálech. Pro tento účel si vytvoříme funkci *void ZobrazBuffer(long buffer, long length, long max, int zesil)*

Kde první položka je ukazatel na buffer s daty pro zobrazení, druhý parametr udává kolik hodnot požadujeme zobrazit, parametr max udává maximální hodnotu v datech a zesil udává zvětšení vykreslovaného grafu nad normovanou velikost. Toto funkce vypadá následovně:



Obrázek 5 – vývojový diagram zobrazení grafu

Tato funkce by měla přizpůsobit délku vykreslovaných dat šířce displeje. Nicméně v přístroji se data zobrazují buď 1:1 nebo 1:2 a pro tyto poměry zobrazení pracuje správně.

Dále je také potřeba vytvořit funkci pro zobrazování průběhu historie v režimu SCAN. Zde se postupně do pole zapisují naměřené hodnoty a pokud je toto pole plné posouvá se toto pole doleva. Pro zobrazování tohoto průběhu slouží funkce:

```
void Graf(plong buffer, int length,char krok,long max)
```

kde první hodnota je ukazatel na buffer s zobrazovanými data, druhý parametr je informace o počtu hodnot v tomto bufferu, třetí hodnota udává s jakou velikosti kroku na y-lonové souřadnici chceme data zobrazovat a poslední hodnota je maximální hodnota v zobrazovaném bufferu. Zobrazování se provádí podobně jako v předchozím případě s tím, že je třeba graf vykreslovat jen v dolní polovině displeje.

Jak je dále z *Obrázek 5* patrné používá se zde funkce *void Line(int ox, int oy,int dx, int dy)*. Tato funkce vykresluje čáru s bodu o souřadnicích ox, oy do bodu o souřadnicích dx, dy s bodem 0,0 v horním levém rohu displeje. Pro vykreslování úsečky jsem zvolil Bresenhamův algoritmus, který si vystačí jen s celočíselnou aritmetikou viz [10, 11 a 12] Tento algoritmus vychází z rovnice přímky ve tvaru:

$$y = kx + b$$

kde m je směrnice a b posun na ose y . Směrnici m tedy můžeme určit z rovnice:

$$k = \frac{dy - oy}{dx - ox} = \frac{\Delta y}{\Delta x} \quad [12]$$

$$b = \frac{dx * oy - ox * dy}{dx - ox} = \frac{dx * oy - ox * dy}{\Delta x} \quad [12]$$

Algoritmus vychází z těchto rovnic. Jedná se o přírůstkový algoritmus tedy pozice ve směru závisle osy, osy s největší délkou, se inkrementuje o 1. Výpočet potom bude probíhat následovně pokud k náleží intervalu $\langle 0,1 \rangle$

1. Vypočteme první chybový člen $e_i = 2\Delta y - \Delta x$ a zobrazím aktuální pixel
 2. $e_i < 0$ jako další bod se zobrazí $[x_i + 1, y_i]$ a $e_{i+1} = e_i + 2\Delta y$
 3. $e_i \geq 0$ jako další bod se zobrazí $[x_i + 1, y_i + 1]$ a $e_{i+1} = e_i + 2(\Delta y - \Delta x)$
 4. Opakujeme od bodu 2. dokud nedojdeme ke koncovému bodu
- Pro interval $k \in \langle 0,-1 \rangle$ je algoritmus analogický.

V programu bylo použito rozdělení souřadného systému na kvadranty a to tak, že se jako řídicí osa vybere nejprve osa s větším přírůstkem a koncový bod v této ose se praví tak aby měl větší hodnotu než počáteční.

Dále je v této funkci přidáno omezení vykreslování na y-ovou souřadnici a to tak pokud by měl být další vykreslený bod větší než 128(výška displeje v pixelech), tak se vykreslování ukončí.

3. ROZBOR MENU

Systém menu vychází z předchozího typu přístroje [8], což bylo součástí zadání.

3.1 POPIS OVLÁDACÍCH PRVKŮ PŘÍSTROJE

[ON/OFF]	Tlačítko pro zapnutí a vypnutí přístroje.
[ENTER]	Tlačítko pro potvrzení jakékoliv akce, potvrzení zadaných údajů, vstup do pod menu.
[ESC]	Tlačítko pro ukončení probíhající aktivity, bez uložení dat a návrat do nadřazeného menu.
[FUNC]	Tlačítko pro nastavení +/- při editaci numerických hodnot.
[DATA]	Tlačítko přepínající přístroj do funkce zobrazení naměřených dat: počet impulsu v jednotlivých ROI, kalkulace koncentrace radionuklidů.
[SPECT]	Tlačítko pro zobrazení naměřeného spektra, posledního nebo při procházení paměti.
[0] - [9]	Tlačítka pro zadávání číselných hodnot a skok na stejně označenou položku menu.
[^]	Tlačítko pro posuv kurzoru.
[v]	Tlačítko pro posuv kurzoru.
[>]	Tlačítko pro posuv kurzoru.
[<]	Tlačítko pro posuv kurzoru.
[NEXT]	Tlačítko použité pro skok na další položku.

Tabulka 2 – popis ovládacích prvků přístroje

3.2 POPIS MENU

V následujících podkapitolách budou popsána jednotlivá menu a význam položek. Základní filosofie systému těchto menu je taková, že aktivní řádek položky menu je zvýrazněn tak, že je tento řádek invertován. Každé menu může obsahovat dva typy položek, a to položku odkazující se na pod menu, nebo položku umožňující přímo nastavit hodnotu. Pokud se v menu nastavuje přímo položka, jde většinou o nastavení jedné hodnoty nebo výběr z několika možností. Pokud je třeba nastavit větší množství položek, je pro tento účel vytvořeno pod menu.

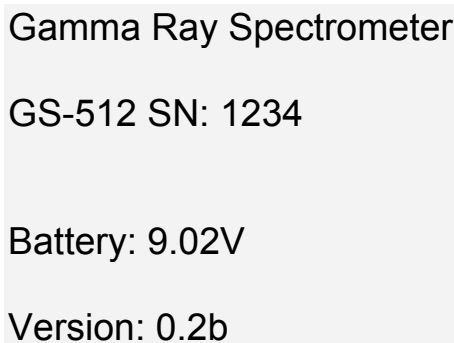
V menu se můžeme pohybovat dvěma způsoby, a to buď šipkami nahoru, dolů nebo pomocí čísla položky z numerické klávesnice. Pomocí klávesy ENTER se dostaneme do pod menu, nastavení položky a potvrzení zadaných hodnot. Tlačítko ESC má opačný význam.

Pokud v menu je položka pro nastavení hodnoty, zobrazí se při stisku tlačítka ENTER na prvním místě nastavované položky kurzor a je možno tuto položku nastavit. Podrobněji bude vysvětleno dále.

Pokud jde o položku pro výběr možnosti, zobrazí se aktuální možnost invertovaná a pomoci šipek nahoru, dolů rolují dokola všechny možnosti.

Pokud jde o menu, které slouží jen pro vstup hodnot od uživatele, je možno se pohybovat po jednotlivých položkách šipkami nebo tlačítkem NEXT. Na konci, nebo začátku se vždy roluje na opačný konec. Celková hierarchie menu viz *příloha 1*.

3.2.1 Úvodní menu



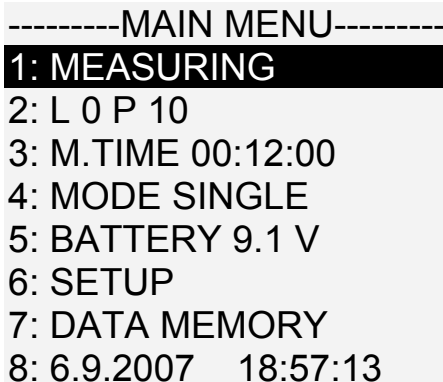
Gamma Ray Spectrometer
GS-512 SN: 1234
Battery: 9.02V
Version: 0.2b

Obrázek 6 – úvodní menu

Toto menu je zobrazeno při spuštění přístroje a jak je z *Obrázek 6* patrné je zobrazován název přístroje, jeho sériové číslo, stav baterie a verze firmware.

Z tohoto menu se dostaneme zmáčknutím jakéhokoliv tlačítka do hlavního menu.

3.2.2 Hlavní menu



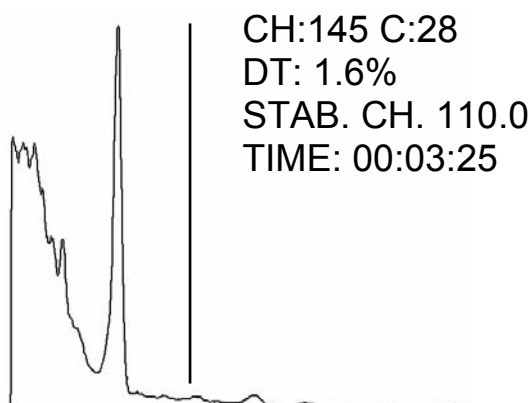
-----MAIN MENU-----
1: MEASURING
2: L O P 10
3: M.TIME 00:12:00
4: MODE SINGLE
5: BATTERY 9.1 V
6: SETUP
7: DATA MEMORY
8: 6.9.2007 18:57:13

Obrázek 7 – Hlavní menu

Na *Obrázek 7* je zobrazeno hlavní menu.

1. MEASURING - tato položka slouží pro spuštění měření.
2. LP - tato položka slouží pro vstup do pod menu pro nastavení profilu, pozice a kroku pro inkrementování pozice s novým měřením. Tyto hodnoty slouží pro označení bodu měření v terénu.
3. M. TIME – tato položka slouží pro nastavení doby měření.
4. MODE - položka slouží pro nastavení módu měření a typu ukládaných dat. Nejprve se nastaví mód měření a následně typ ukládaných dat, kromě módu SCAN.
5. BATTERY - Pátá položka zobrazuje aktuální napětí baterie, které je zobrazeno vždy, když se toto změní o 0,1V.
6. SETUP - tato položka slouží pro vstup do pod menu pro nastavení přístroje. Bude popsáno dále.
7. DATA MEMORY - položka slouží pro vstup do pod menu pro správu paměti a prohlížení uložených dat.
8. RT - položka zobrazuje aktuální datum a čas. Umožňuje také její nastavení, nejprve datum a pak čas, nezávisle.

3.2.2.1 Měření v módu SINGLE



Obrázek 8 – měření v módu SINGLE

V tomto módu měření se zobrazuje každou sekundu aktuální spektrum a k němu příslušející data. Dále se zobrazuje kurzor, kterým se může pohybovat pomocí šipek doleva a doprava, a zobrazují se příslušná data.

První řádek zobrazovaných dat přísluší kurzoru a je to poloha kurzoru a počet registrovaných impulsu na této poloze. Dalším údajem je mrtvá doba a na dalším řádku je stabilizační kanál a aktuální doba měření.

3.2.2.2 Měření v módu SCAN

```

-----SET SCANNED ROI---
1: ROI1
2: ROI2
3: ROI3
4: ROI4
5: ROI5
6: ROI6
7: ROI7
8: ROI8

```

Obrázek 9 – nastavení ROI v SCAN

Tento mód slouží pro měření počtu impulsů ve zvolené ROI a vykreslení její minulosti pomocí grafu. (ROI – region of interest – je energiové okno).

ROI je třeba vybrat před měřením viz *Obrázek 9*, pohybovat se zde můžeme pomocí šipek nahoru, dolů, nebo stisknutím příslušné číslice. Zvolenou ROI zadáme tlačítkem ENTER a následně se spustí měření, kde se zobrazuje v horní polovině displeje čas měření a naposledy změřený počet impulsů ve zvolené ROI, číslo měření, mrtvá doba, poloha cesiového píku, doba měření a v dolní polovině graf.

3.2.2.3 Měření MONITOR

```

-----MONITOR-----
START DATE   21.12.07
START TIME   08:12:34
INTERVAL     01:00:00
END DATE     13.09.07
END TIME     08:12:34

```

Obrázek 10 – nastavené měření MONITOR

MONITOR slouží pro opakované měření. Je zde třeba nastavit datum a čas prvního spuštění, pokud je toto datum a čas nižší než aktuální spustí se první měření okamžitě, interval mezi spuštěním předchozího měření a spuštěním následujícího měření a datum a čas posledního měření.

V tomto menu se můžeme po jednotlivých položkách pohybovat pomocí šipek nahoru dolů nebo pomocí tlačítka NEXT. Pokud stiskneme na poslední položce

šipku dolů nebo ladítko NEXT posune kurzor na první položku, v opačném případě je to podobné.

Mezi jednotlivými měřeními se zobrazuje čas následujícího spuštění a aktuální čas. Měření je možno ukončit pomocí tlačítka ESC.

Při měření se obrazují stejné údaje jako při měření v módu SINGLE.

3.2.2.4 Nastavení linky, pozice a kroku

LINE	+0100
POSITION	+0050
STEP	-010

Obrázek 11 – Nastavení linky, pozice a kroku

Toto menu slouží pro nastavení pozice na profilu. Po jednotlivých položkách se dá pohybovat šipkami nahoru a dolů, nebo pomocí tlačítka NEXT.

3.2.2.5 SEPUT menu

```

-----SETUP MENU-----
1: DEFAULT VALUES
2: T. OFF 00:10:00
3: ROIS
4: CONSTANTS
5: CALIBRATION
6: BEEPER OFF
7: BACKLIGHT ON
8: STABILISATION
  
```

Obrázek 12 – SETUP menu

1. DEFAULT VALUES – tato položka slouží pro tovární nastavení přístroje.
2. T. OFF – v této položce se nastavuje čas, za který se přístroj při nečinnosti vypne. Doba do vypnutí musí být ze spodu omezena na asi 10s. Nastavení probíhá stejně jako v případě nastavování doby měření.
3. ROIS – položka obsahující pod menu pro nastavení/prohlížení intervalů jednotlivých ROIs
4. CONSTANTS – tato položka obsahuje pod menu pro nastavení/prohlížení kalibračních konstant.

5. CALIBRATION – tato položka obsahuje pod menu pro provedení kalibračních měření na jednotlivých normálech a jejich zpracování.
6. BEEP – tato položka slouží pro zapnutí nebo vypnutí pípání přístroje. Nastavování této položky je vysvětleno níže.
7. BACKLIGHT – tato položka slouží pro zapnutí nebo vypnutí podsvícení. Po stisku tlačítka ENTER se invertuje aktuálně nastavená hodnota a pomocí šipek nahoru, dolů se ve smyčce nastavuje buď ON nebo OFF.
8. STABILISATION – obsahuje pod menu pro nastavení stabilizace.

3.2.2.5.1 Nastavení ROI

-----SET ROIS-----		
ROI1	140	511
ROI2	224	254
ROI3	267	305
ROI4	396	450
ROI5	104	116
ROI6	000	000
ROI7	000	000
ROI8	000	000

Obrázek 13 – nastavení ROIS

Toto menu slouží pro nastavení nebo prohlížení ROIS. První sloupec je číslo ROI druhý sloupec je počátek intervalu a třetí sloupec konec intervalů nastavované ROI. Po jednotlivých položkách se můžeme pohybovat pomocí tlačítka NEXT. Tedy počáteční interval ROI1, koncový interval ROI1, počáteční interval ROI2 atd. Nebo v jednotlivých sloupcích pomocí šipek nahoru a dolů.

3.2.2.5.2 Nastavení konstant

-----SET CONSTANTS-----			
C1	+000302	C2	+000021
C3	+000009	C4	+000004
C5	+000733	C6	+000497
C7	-000447	C8	-000017
C9	-000005	C10	+005218
C11	-003476	C12	+000013
C13	-000395	C14	+012230
DEFAULT CONSTANTS, ROIS			

Obrázek 14 – nastavení konstant

Toto menu slouží pro nastavení nebo prohlížení kalibračních konstant. Při procházení jednotlivých položek je použito stejného systému jako při procházení nastavení ROIS.

Pokud při procházení těchto konstant stiskneme tlačítko ESC dostaneme se na položku DEFAULT CONSTANTS, ROIS, zde po stisku tlačítka ENTER se provede nastavení výchozích hodnot těchto konstant a intervalu ROIS.

3.2.2.5.3 Menu CALIBRATION

```

-----CALIBRATION-----
MEASURING PAD K
MEASURING PAD U
MEASURING PAD Th
MEASURING PAD 0
MEASURING BACKGROUND
SET PAD PARAMETERS
PROCESS BACKGROUND
PROCESS CALIBR.
  
```

Obrázek 15 – CALIBRATION menu

Toto menu slouží pro provedení kalibrace přístroje.

První tři položky slouží pro měření spektra na normálových zářičích K, U, Th. Čtvrtá položka slouží pro měření na normálu bez příměsí radionuklidů a pátá položka slouží pro měření záření pozadí, tedy kosmické záření a záření vlastního přístroje.

Položka SET PAD PARAMETERS obsahuje pod menu pro nastavení parametru normálových zářičů.

Položka PROCESS BACKGROUND slouží pro výpočet prvních pěti kalibračních konstant.

Položka PROCES CALIBR. slouží pro výpočet všech kalibračních konstant přístroje z naměřených kalibračních spekter.

3.2.2.5.3.1 SET PAD PARAMETERS menu

```
SET PARAMETERS OF PADS
CON. K    U    Th
000.00   0.00  0.00
0.00     0.00  0.00
0.00     0.00  0.00
0.00     0.00  0.00
GEOMETRY  1.000
```

Obrázek 16 – SET PAD PARAMETERS menu

Toto menu slouží pro nastavení parametrů normálových zářičů K, U, Th a nulového standardu. Těmito parametry jsou koncentrace K, U a Th v těchto zářičích a jejich geometrie. Pohyb po jednotlivých položkách je stejný jako v případě nastavování kalibračních konstant.

3.2.2.5.4 STABILISATION menu

```
-----STABILISATION-----
STAB.          ON
INTERVAL      020s
GAIN          -25%
REF. CHANNEL  110.0
```

Obrázek 17 – STABILISATION menu

Toto menu slouží pro nastavení stabilizace referenčního píku při měření.

- STAB. – tato položka slouží pro zapnutí/vypnutí stabilizace. Nastavení probíhá stejně jako při nastavení podsvícení.
- INTERVAL – zde se nastavuje stabilizační interval (minimální hodnota 10s). Při stlačení tlačítka ENTER se objeví kurzor na prvním místě původně nastaveného čísla a je možno změnit hodnotu.
- GAIN – položka sloužící pro nastavení zesílení programovatelného zesilovače před ADC. Nastavení probíhá stejně jako u intervalu.
- REF. CHANNEL – položka pro nastavení referenčního kanálu na, kterém září referenční zářič umístěný v sondě. Může nabývat hodnot 40 – 140.

3.2.2.6 DATA MEMORY menu

```
-----MEMORY-----  
1: LIST  
2: ERASE  
3: TEST  
  
FREE MEMORY 1958354B  
IT IS 18650 ROI  
OR 909 SPECTRA
```

Obrázek 18 – DATA MEMORY menu

Toto menu slouží pro prohlížení paměti, mazání a test.

Položka ERASE a TEST vymaže/otestuje paměť při stisknutí ENTER na této položce se objeví hláška:

```
ALL SAVED DATA  
WILL BE LOST  
  
ENTER / ESC
```

Po potvrzení následuje ještě pro ověření další

```
ARE YOU SURE?  
NEXT / ESC
```

Po potvrzení se smaže datová paměť při tomto se zobrazuje aktuální činnost a stav operace v procentech. Po skončení se zobrazí oznámení, že daná operace skončila.

Položka list obsahuje pod menu pro prohlížení uložených dat. Pokud však nejsou žádná data uložena vypíše se pod položkou TEST, NO SAVED DATA.

Další položky zobrazují kolik je volné paměti v B, kolik měření se do paměti vejde pokud se budou ukládat jenom data a kolik se jich do paměti vejde pokud se budou ukládat i spektra.

3.2.2.6.1 Prohlížení paměti

ID	DATE/TIME	TYPE
209	17.9.07/13:44	S
208	17.9.07/12:56	R
207	16.9.07/8:01	B
206	12.9.07/16:38	U
205	12.9.07/16:05	K
204	12.9.07/15:42	T
203	12.9.07/15:29	S
202	12.9.07/15:23	S

Obrázek 19 – prohlížení paměti

Toto menu zobrazuje naměřená data. První položkou je číslo měření, druhou datum a čas kdy bylo měření provedeno a třetí položkou je typ měření.

Po jednotlivých položkách se můžeme pohybovat šipkou nahoru a dolů. Pomocí tlačítka NEXT se můžeme posunovat o 8 položek dolů.

Pomocí tlačítek DATA se zobrazí naměřená data a pomocí tlačítka SPECT se zobrazí naměřené spektrum pokud je k dispozici.

3.2.2.6.2 Prohlížení dat

```

LINE 20 POS. 110
MT 00:05:00
DT 2,3 %
PEAK 109.9 FWHM 6.7%
COSMIC 75
TC    26.0 Ueq
K     2.5 %
U     7.8 ppm eU
Th    18.4 ppm Th

```

Obrázek 20 – Prohlížení dat obrazovka1

-----ROIS COUNTS-----	
ROI1	3925
ROI2	744
ROI3	266
ROI4	162
ROI5	0
ROI6	0
ROI7	0
ROI8	0

Obrázek 21 – Prohlížení dat obrazovka2

Do módu prohlížení dat se dá přejít pokud v hlavním menu stiskneme tlačítko DATA, kde se zobrazí data z posledního měření, nebo v menu pro prohlížení dat, kde se zobrazí data označené položky.

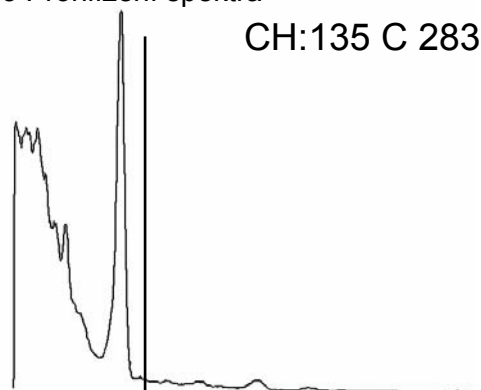
V prvním okně se zobrazují informace o měření jako profil, pozice, doba měření, mrtvá doba, referenční pík a rozlišovací schopnost (FWHM), pokud byla při měření zapnutá stabilizace, v opačném případě se na tomto řádku zobrazuje:

No stabilisation

Dále je to položka COSMIC (počet impulzů, které jsou mimo rozsah měření), celková energie spektra přepočtena na uran, koncentrace K, U, Th.

Na druhé obrazovce se zobrazuje počet impulzů v jednotlivých ROI. Přepínání mezi těmito okny probíhá pomocí tlačítka NEXT.

3.2.2.6.3 Prohlížení spektra



Obrázek 22 – prohlížení spektra

Do módu prohlížení spektra se přejde stejným způsobem jako při prohlížení dat s tím rozdílem, že je třeba zmáčknout tlačítko SPECT.

Pokud při prohlížení spektra stiskneme znovu tlačítko SPECT mění se měřítko x-ové souřadnice a to tak, že se buď o polovinu zmenší nebo zvětší v závislosti na předchozím nastavení.

Dále jde pohybovat kurzorem pomocí šipek doleva a doprava. Protože se celé spektrum nevejde na displej dochází k posuvu spektra při posuvu kurzoru na jeho okraj. S posuvem kurzoru se zobrazuje kanál ve kterém kurzor právě je a počet impulsů v tomto kanále.

Pomocí šipek nahoru a dolů se mění zvětšení spektra v y-ové souřadnici.

3.3 ROZBOR POŽADAVKŮ NA MENU

Z předchozích kapitol je patrné, že požadujeme dva typy menu:

1. menu, které umožňují jak přístup k dalším položkám tak nastavení položek obsahujících jen jednu hodnotu
2. menu, které slouží jen pro prohlížení/editaci více hodnot

Dále je nutné uvědomit si jaké hodnoty budeme požadovat aby uživatel zadával. Tyto hodnoty jsou:

1. celá čísla, jak záporná tak i jenom kladná s omezením jejich hodnoty. Nepožadujeme však různé hodnoty omezení pro kladné a záporné hodnoty.
2. desetinná čísla, pro která platí stejné požadavky jako pro celá čísla.
3. dále požadujeme od uživatele zadávání časových údajů, u kterých musíme zajistit aby uživatel nemohl zadávat neplatné hodnoty
4. zadávání data kde musíme také zajistit aby uživatel nemohl zadat neplatné hodnoty
5. a na konec je to výběr hodnoty (typ měření, zapnutí/vypnutí pípání, podsvícení)

Pro první čtyři typy si vytvoříme speciální funkce což nám umožní využívat tyto funkce dle potřeby. Dále pro poslední typ hodnotu ON/OFF si vytvoříme také samostatnou funkci, kterou můžeme použít více násobně. Pro ostatní hodnoty, které se zadávají výběrem je nutné udělat zvláštní funkce, protože tyto se používají jen jednou.

3.3.1 Společné vlastnosti prvních čtyř skupin

Společnými vlastnosti těchto funkcí a nejen v tomto přístroji je zadávání numerických hodnot což se děje prostřednictvím numerické klávesnice. Je obvyklé, že po stisku klávesy se posune kurzor na další položku a pokud je na konci neposunuje se již nikam. V případě, že je funkce použita v menu, které má za úkol získávat více položek od uživatele musíme umožnit po stisku klávesy na konci aby tato funkce vrátila uživatelem nastavenou hodnotu a ukončila se, abychom se mohli posunout na další položku.

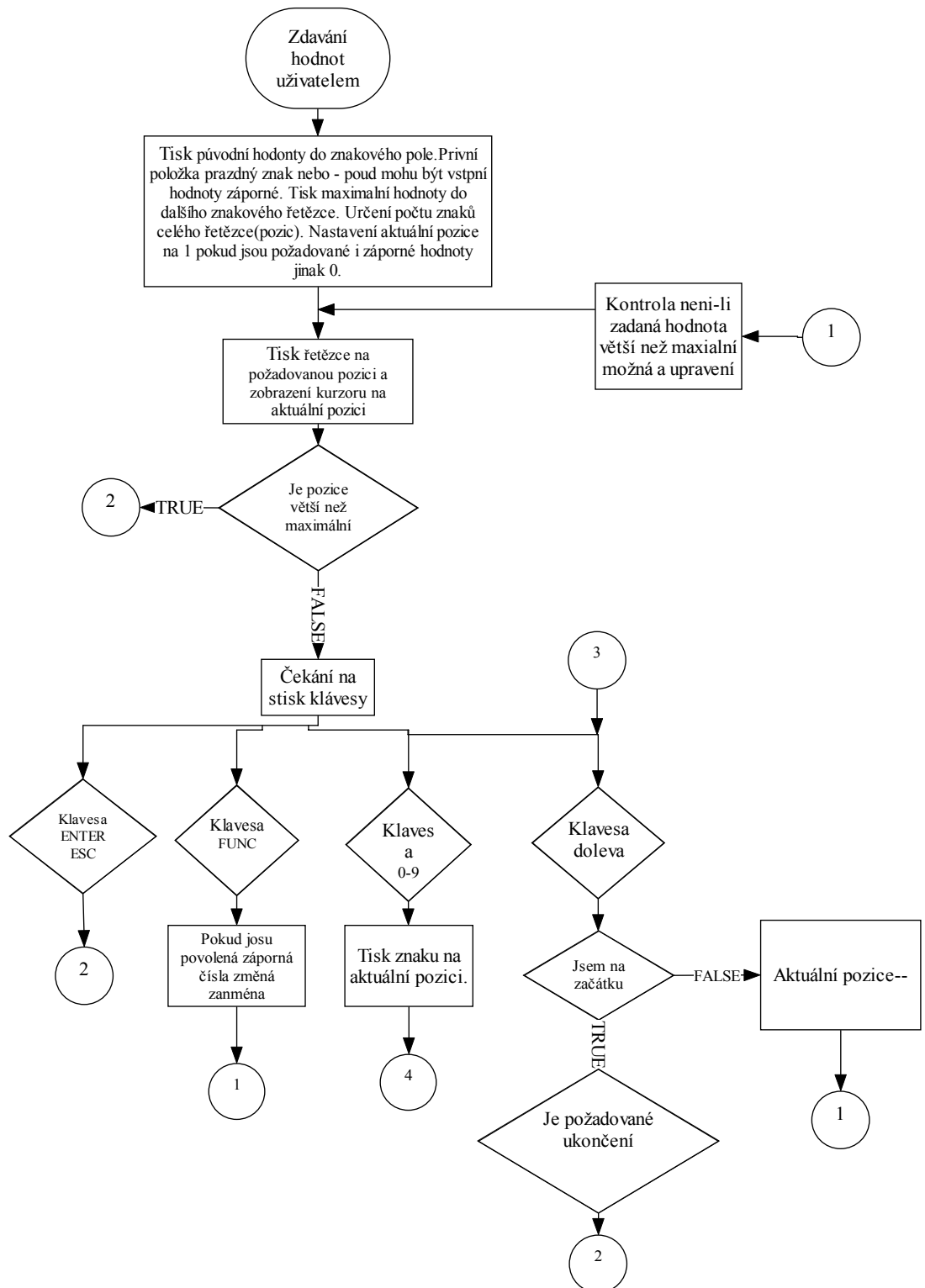
Dalším společným rysem je pohyb kurzoru pomocí šipek doleva a doprava, zde také platí, že musíme umožnit ukončení funkce pokud je kurzor na krajní pozici a funkce je použita v menu pro získávání hodnot z více položkami. Pokud je stišťena šipka nahoru nebo dolu musí funkce také skončit a vrátit nastavenou hodnotu je-li použita v menu pro zadávání více hodnot.

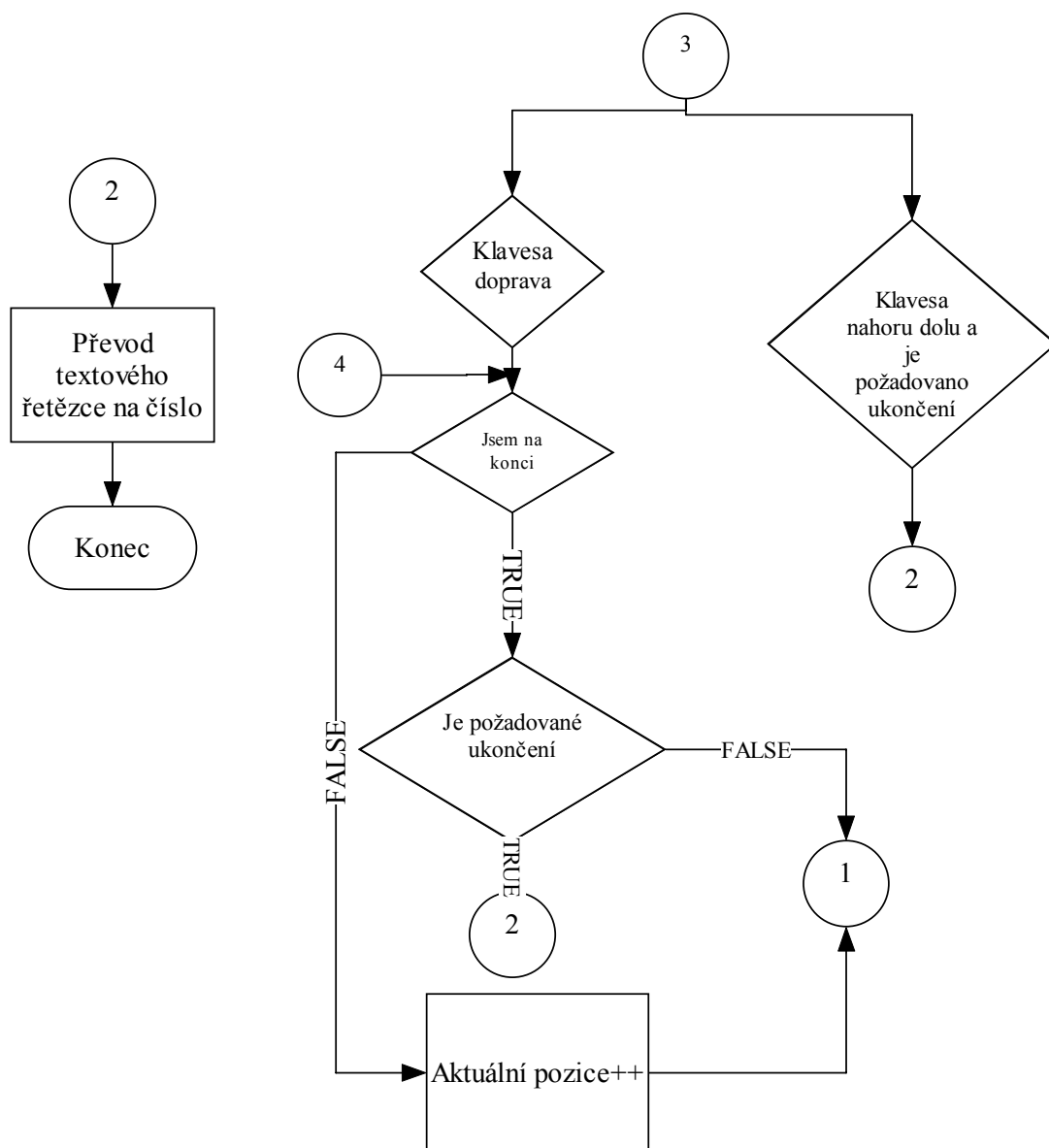
3.3.2 Zadávání celého čísla

Funkce pro zadávání celého čísla musí, aby se dala použít vícenásobně, mít parametry určující řádky a sloupec, kde se má zadávaná hodnota zobrazovat. Dále musí mít k dispozici původní hodnotu, počet míst na který požadujeme číslo zobrazovat, maximální možnou hodnotu čísla, pokud požadujeme zadávání jen kladných čísel tak příznak tohoto a příznak je-li funkce použita v menu pro získávání více hodnot (kontinuální mód). Návratová hodnota této funkce je zadaná hodnota z klávesnice uživatelem. Prototyp takové funkce v C by mohl vypadat:

```
long GetIntValue(int sloupec, int radek, long puvodni, char preciz, long max, BOOL zaporne, BOOL continus )
```

Jak by tato funkce mohla pracovat viz vývojový diagram na *Obrázek 23*.





Obrázek 23 - vývojový diagram funkce pro získání celého čísla od uživatele

3.3.3 Zadávání desetinného čísla

Funkce pro zadávání desetinného čísla je principiálně shodná s funkcí pro zadávání čísla celého s tím rozdílem, že musíme zadat ještě další parametr a tím je počet míst desetinné části čísla. Zde nám však přibyla desetinná tečka, která zvětší

počet možných pozic kurzoru o 1, proto ve funkci pro zadávání desetinného čísla musíme tuto pozici přeskokovat.

Prototyp takové funkce by potom vypadal:

```
float GetFloatValue(int sloupec, int radka,float puvodni,char preciz,char  
dprecis,float max,BOOL zaporne,BOOL continus)
```

3.3.4 Zadávání času

Tato funkce je také velmi podobná předcházejícím s tím rozdílem, že časový údaj má mezi hodinami, minutami a sekundami dvojtečku. Tyto znaky nám zvětší počet pozic o dvě a musíme je také přeskokovat.

Prototyp této funkce by mohl vypadat následovně:

```
void GetTimeValue(int sloupec, int radek,tm* settime,BOOL continus)
```

Jak je vidět první dva parametry udávají pozici kde se má vykreslovat první znak zobrazovaného času, druhý parametr je ukazatel na strukturu tm definice viz standardní knihovny C time.h. Tento parametr slouží jak pro vložení původní hodnoty tak pro získání hodnoty nastavené uživatelem. Poslední parametr určuje je li funkce součástí menu pro vstup s více položkami.

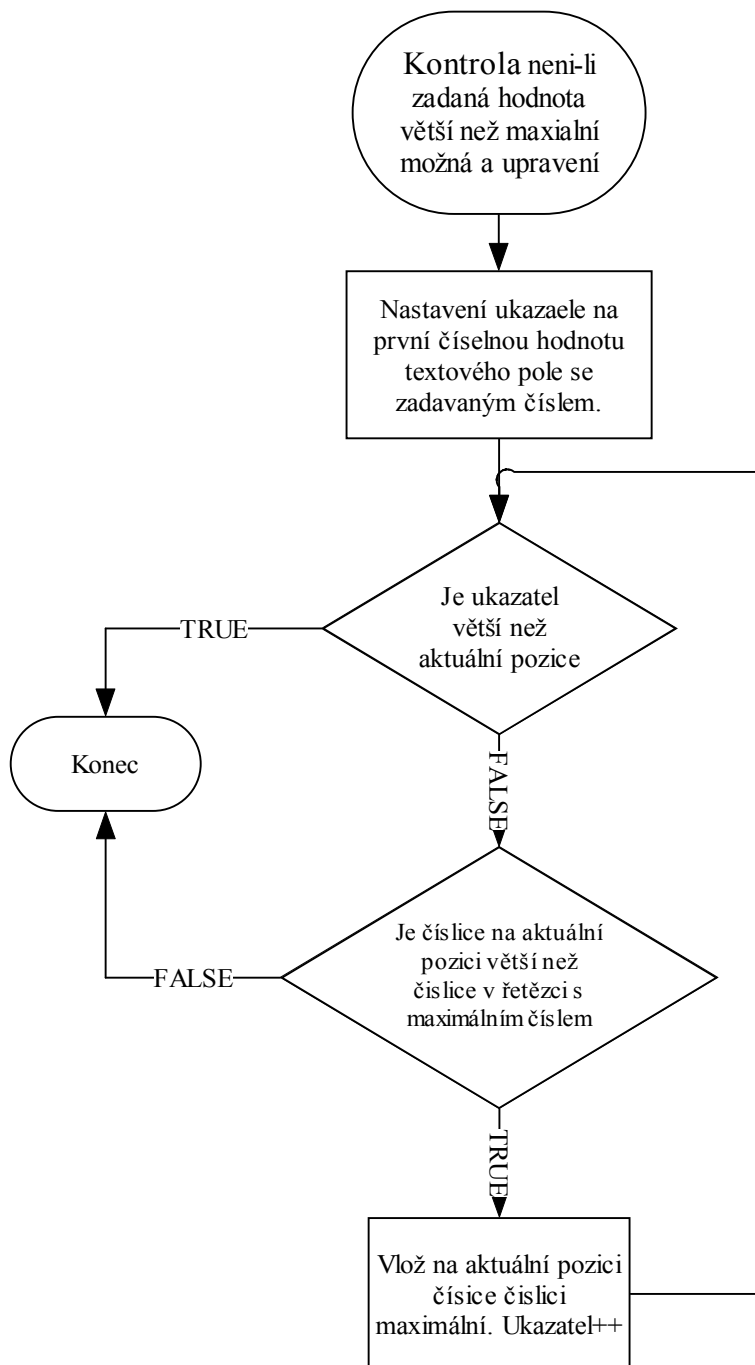
3.3.5 Zadávání data

Zde je situace stejná jako při zadávání času. Místo dvojteček jsou zde dvě tečky. Prototyp takové funkce by mohl vypadat takto:

```
void GetDateValue(int pixelpos, int line,tm* settime,BOOL continus)
```

Popis jednotlivých parametrů je stejný jako při zadávání času s tím rozdílem, že ve struktuře tm má naplněno jenom datum po ukončení funkce.

3.3.6 Řešení omezení hodnot čísel a času



Obrázek 24 - vývojový diagram omezení maximální možné zadávané hodnoty čísla

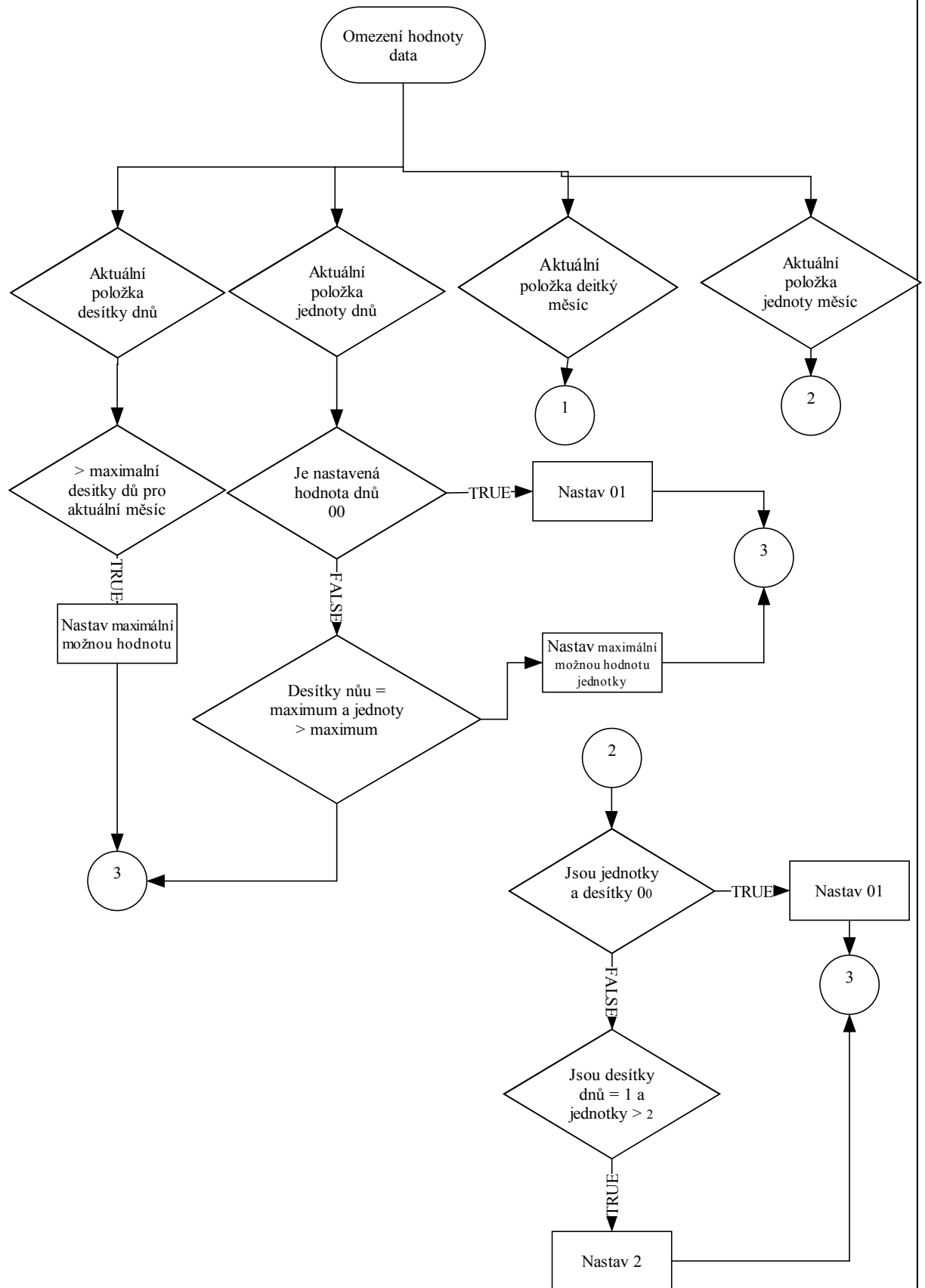
Jak je vidět z diagramu nejde o nic složitého. Tuto kontrolu je však třeba provádět po každém zadaném čísle.

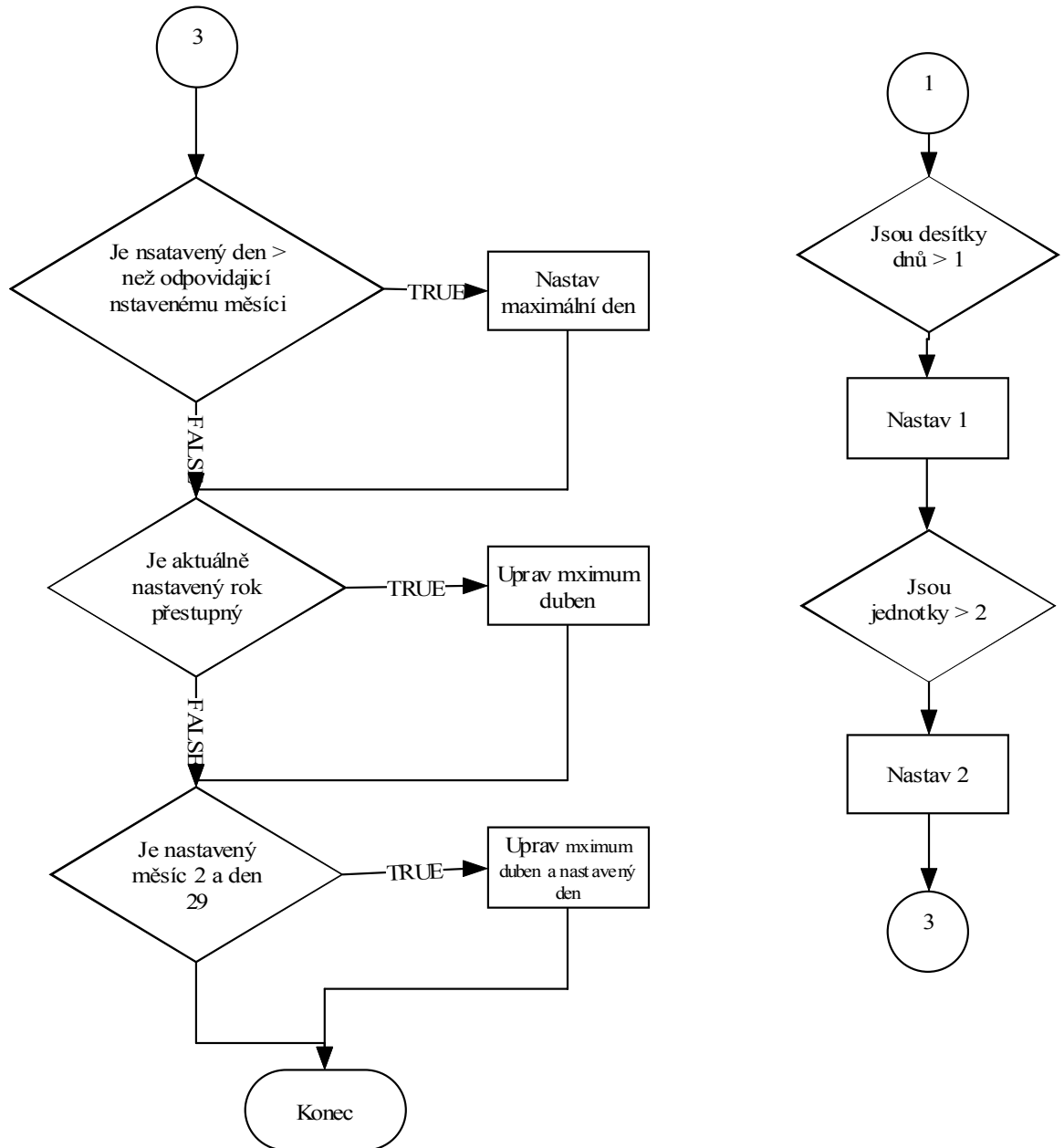
Podobným a podstatně jednodušším způsobem se provádí i kontrola časového údaje při zadávání času.

3.3.7 Řešení omezení zadávání data

Zde je třeba si uvědomit, že datum a hlavně tedy dny, mohou nabývat různých hodnot v závislosti jak na měsíci tak roku (přestupný rok). Je tedy třeba zajistit omezení těchto hodnot tak aby nedošlo k zadání špatných hodnot.

Pro účely určení maximálního počtu dnů v daném měsíci si vytvoříme pole, které obsahuje postupně jednotlivé počty dnů v měsíci jako text. Pokud je aktuální rok přestupný můžeme položku obsahující měsíc duben zvýšit nebo v opačném případě upravit zpět. Omezení hodnot udávající měsíc je podstatně jednodušší, stačí zajistit aby tato hodnota byla v intervalu 1 - 12. Pro položku roků nemusíme provádět žádné omezení pokud umožníme zadávat jenom roky v 21. století. Řešení omezení by mohlo vypadat následovně viz. *Obrázek 31*.





Obrázek 25 - vývojový diagram omezení data

3.4 ŘEŠENÍ ŘÍZENÍ MENU

Jak je z popisu menu patrné bude systém menu obsahovat několik pod menu a pokud nemá položka pod menu je třeba vykonat nějakou akci ať už je to spuštění měření, získání údajů od uživatele nebo zobrazení výsledku měření.

Takže jak řídit takový systém menu? Nejprve si vytvoříme funkce, které budou zobrazovat jednotlivá menu. Tyto funkce budou mít dva společné parametry a to číslo položky menu, která se má zvýraznit a ukazatel na proměnné prostředí.

Pokud se na problém podíváme je patrné, že každé menu má několik položek číslo položky si tedy označíme proměnou z názvem *mmpos* z hodnotami od 0 do maximálně 7 (každé menu má maximálně 8 položek), abychom věděli kolik položek jaké menu má si vytvoříme pole s názvem *menu_items*, které bude obsahovat počet položek jednotlivých menu na indexech shodných s *menu_id*. Jednotlivá menu si tedy označíme pomocí proměnné *menu_id*. Do menu zahrneme i měření abychom toto mohli rozlišit a také speciální menu pro zobrazování chybových hlášek.

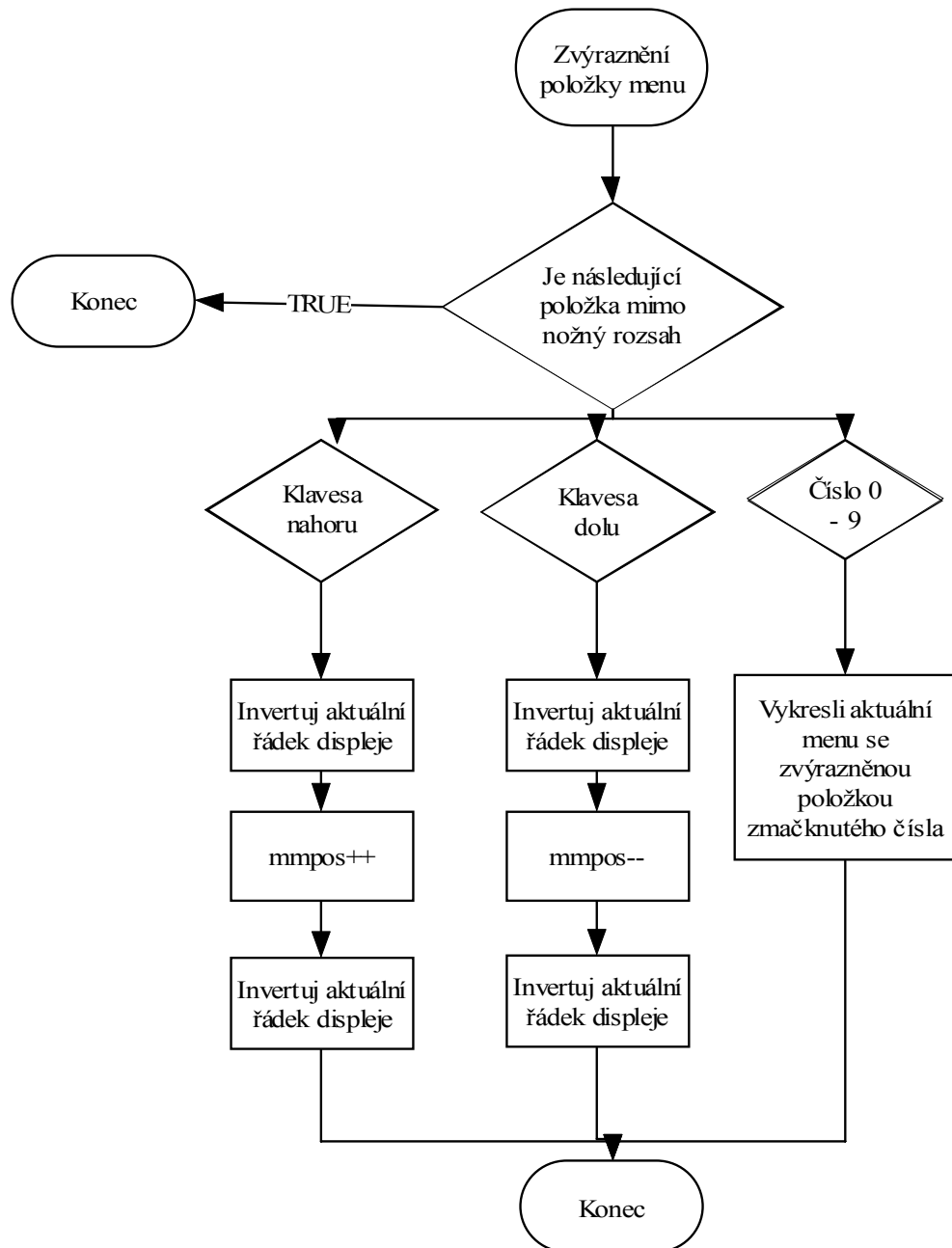
Dále je patrné, že většina akcí je řízena stiskem tlačítka, nastavenými parametry a několika stavy, hlavně při manipulaci z daty, tyto stavy jsou *probíhá měření, jsou v RAM data*.

Pro řízení menu si vytvoříme funkci například *Klávesa(kód_klávesy)*. V této funkci si vytvoříme blok *switch(kód_klávesy)* pro klávesy, pro každou klávesu vytvoříme další blok *switch(menu_id)*, abychom věděli pro jaké mnu se má požadovaná akce provádět, dále vytvoříme pro jednotlivé *menu_id* další blok *switch(mmpos)*, abychom určili pro jakou položku je akce určena. Z hierarchie menu viz jejich popis je patrné jaké akce se budou provádět při stisku tlačítka ENTER, ESC a další by mělo být jasné z popisu menu.

3.4.1 Řízení zvýraznění položky menu

Pro zvýraznění položky bychom mohli použít funkce pro zobrazení požadovaného menu s požadovanou zvýrazněnou položkou. Tato možnost má nevýhodu v tom, že zápis na displej (pracuje v grafickém režimu) by trval relativně dlouho (při rychlém mačkání šipek nahoru dolu by to nevypadalo dobře), a v tomto případě není ani zapotřebí přepisovat celý displej, budeme provádět jen inverzi

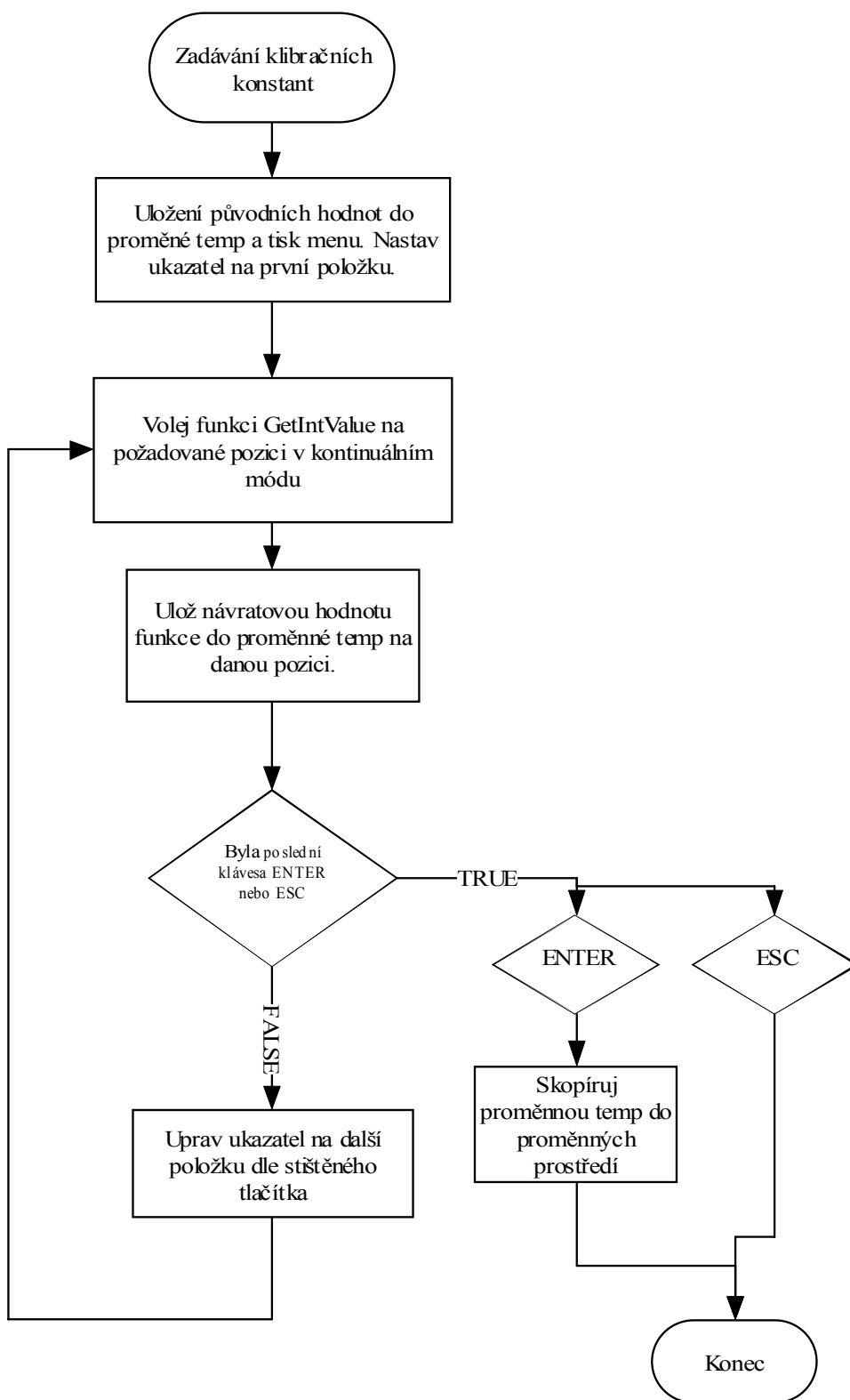
položek menu na displeji. Pokud se zmáčkne číslo položky překreslíme celé menu s označenou odpovídající položkou (v tomto případě to nevypadá špatně).



Obrázek 26 - vývojový diagram řízení označení položky

3.4.2 Řízení menu pro vstup více položek

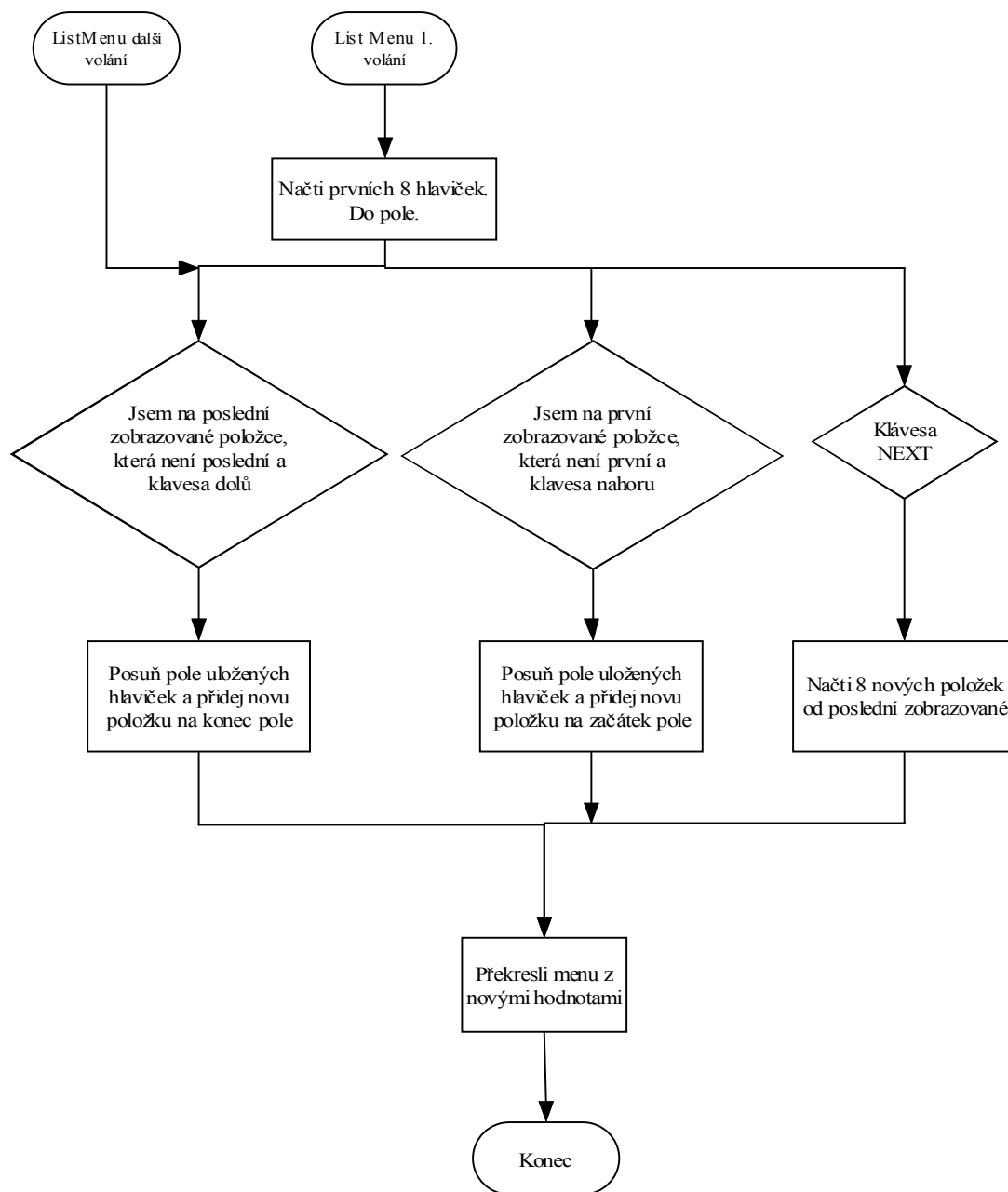
Jak jsem se již zmínil pro tento účel si pro jednotlivá takováto menu vytvoříme funkce jejichž parametrem je ukazatel na proměnné prostředí. Tyto funkce jsou pro různé hodnoty principiálně shodné proto popíšu zadávání hodnot kalibračních konstant.



Obrázek 27 - vývojový diagram pro vstup vícenásobných položek

3.4.3 Zobrazení uložených položek

Menu pro zobrazování uložených hodnot může současně na displeji zobrazit jenom 8 položek (více se jich na displej nevejde) z tohoto důvodu je třeba zajistit rolování těchto položek. Z tohoto důvodu je systém, načítání uložených dat vytvořen tak, že umožňuje načíst základní údaje, hlavičku uložené položky, id měření, čas a datum uložení, typ uložené položky, odkaz na další položku. Z odkazu na následující položku můžeme postupně načítat jednotlivé položky dle potřeby. Fungování zobrazování vypadá následovně.



Obrázek 28 - vývojový diagram zobrazování uložených hodnot

Protože zobrazování seznamu uložených hodnot je také menu zvýraznění aktuální položky probíhá stejně jako pro jakékoliv jiné menu s tím rozdílem, že v poli obsahujícím počet položek pro toto menu se v případě že je třeba zobrazit méně než 8 položek toto promítne. Proměnná *mmpos* v tomto případě označuje index

do pole s datovými hlavičkami, takže můžeme jednoduše pomoci odpovídající hlavičky načíst uložené údaje.

3.4.4 Zobrazení naměřených dat

Pro zobrazení naměřených dat si také vytvoříme funkci, jde v podstatě o další menu. Protože zobrazovaná data se na displej nevejdou jsou rozděleny na dvě obrazovky. Z tohoto důvodu bude mít tato funkce parametr, který udává jakou obrazovku chceme vykreslovat. Tato funkce, pokud nejsou k dispozici žádná data vypíše odpovídající hlášení. Dále musíme vytvořit proměnnou, která obsahuje číslo aktuální obrazovky, které budeme rolovat při stisknutí tlačítka NEXT.

3.4.5 Zobrazení spektra

Spektrum se zobrazuje při měření, dále je možno zobrazit poslední měřené spektrum v hlavním menu a samozřejmě při procházení naměřených údajů pokud bylo uloženo i spektrum. Při měření se zobrazují u grafu naměřeného spektra i údaje o probíhající měření viz Měření v módu SIGLE a je také třeba zajistit změnu y-lonového měřítka z novými daty a to tak aby maximum bylo vždy v druhé horní polovině displeje. Při měření je třeba obrazovku aktualizují každou sekundu. Pokud jde o zobrazení spektra při procházení naměřených hodnot zobrazuje se jen kanál ve kterém je kurzor a počet impulsů. Protože se spektrum nevejde celé na displej je nutné zajistit rolování displeje ve vodorovném směru. Dále je možno měnit x-ové měřítka a to buď v poměru 1:1 nebo 1:2 pomocí tlačítka SPECT. Y-lonové měřítka můžeme měnit pomocí šipek nahoru, dolů a kurzor se posunuje šipkami doleva a doprava.

Protože jsou vždy požadované stejné funkce vytvoříme si samostatnou funkci pro zobrazení spektra, která bude mít také jako parametr ukazatel na proměnné prostředí. Protože máme k dispozici funkci pro vykreslování grafu,

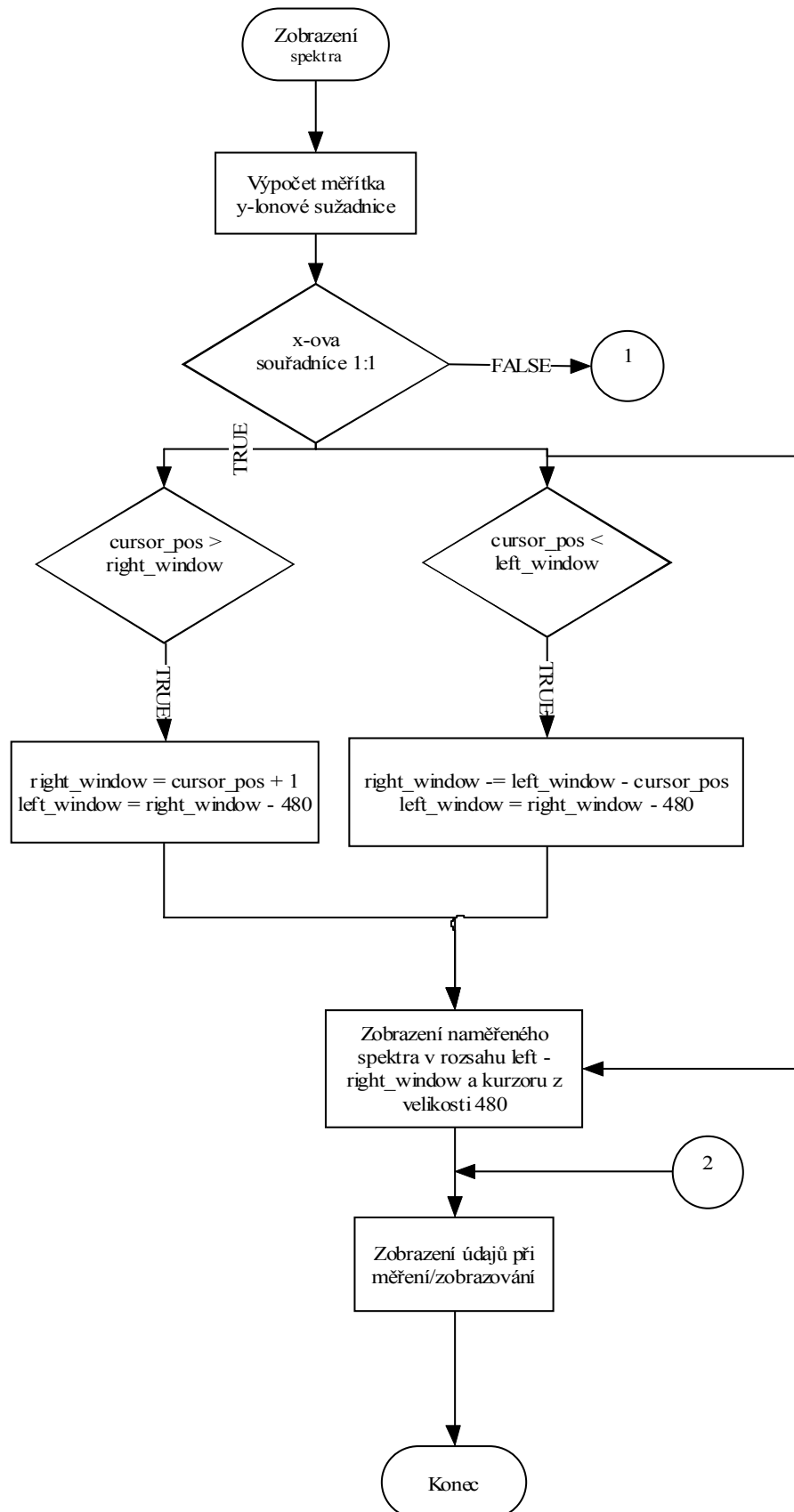
```
void ZobrazGrtaf(plong buffer, long length, long max, int zesil)
```

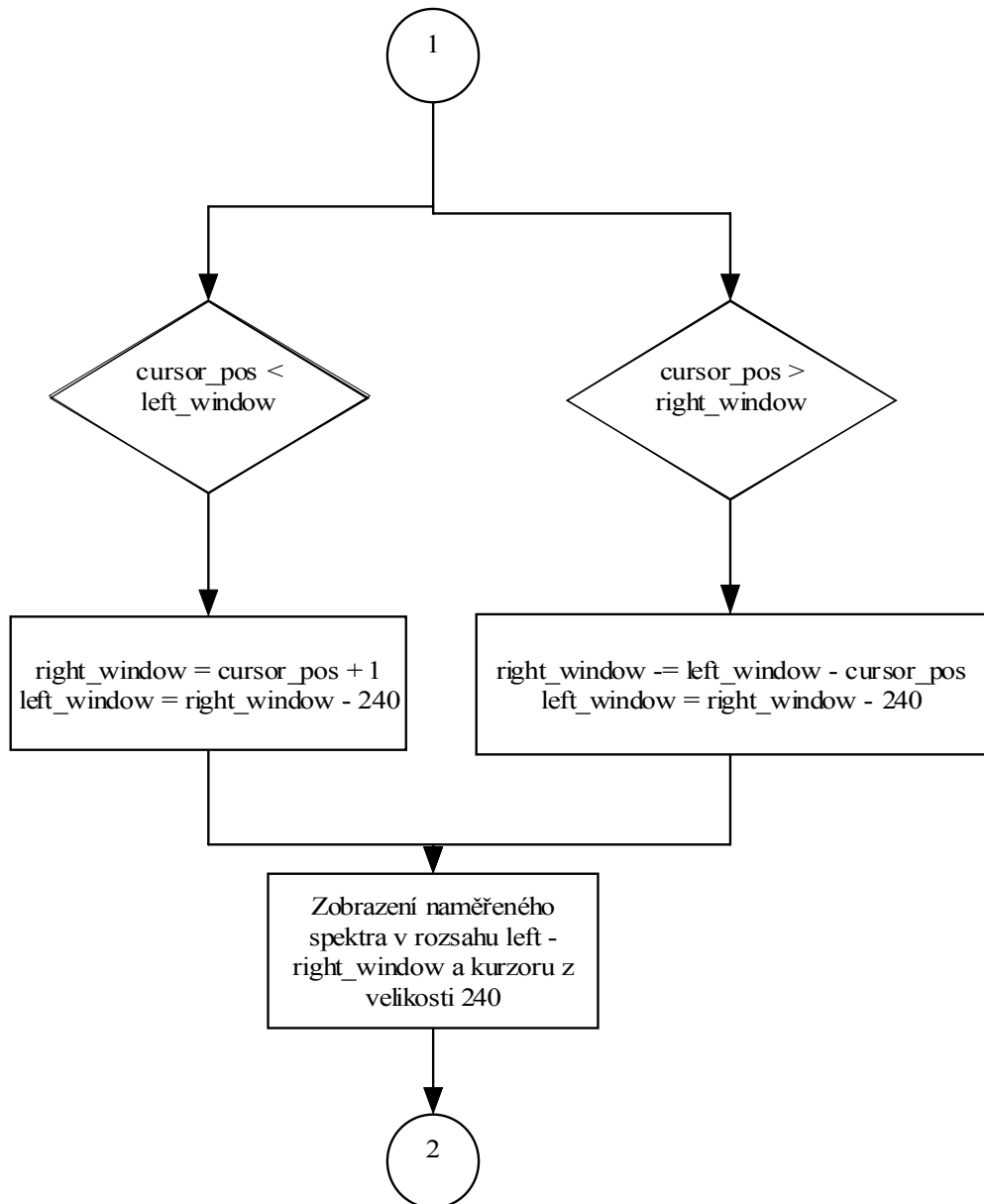
kteřá má jako první parametr ukazatel na buffer s naměřenými hodnotami, druhý parametr určuje kolik hodnot se má zobrazit. Pokud je počet hodnot větší než

šířka displeje funkce sama vybere hodnoty, které se budou vykreslovat. Další dvě hodnoty slouží pro přizpůsobení y-lonového měřítka.

Pro účely řízení zobrazování budeme muset vytvořit několik proměnných a to *left_winow* a *right_window* pro okraje zobrazovaného spektra, stav x-ovového měřítka *male_spect*, *násobitel* pro zvětšení y-lonového měřítka a proměnnou *kursor_pos* pro aktuální pozici kurzoru.

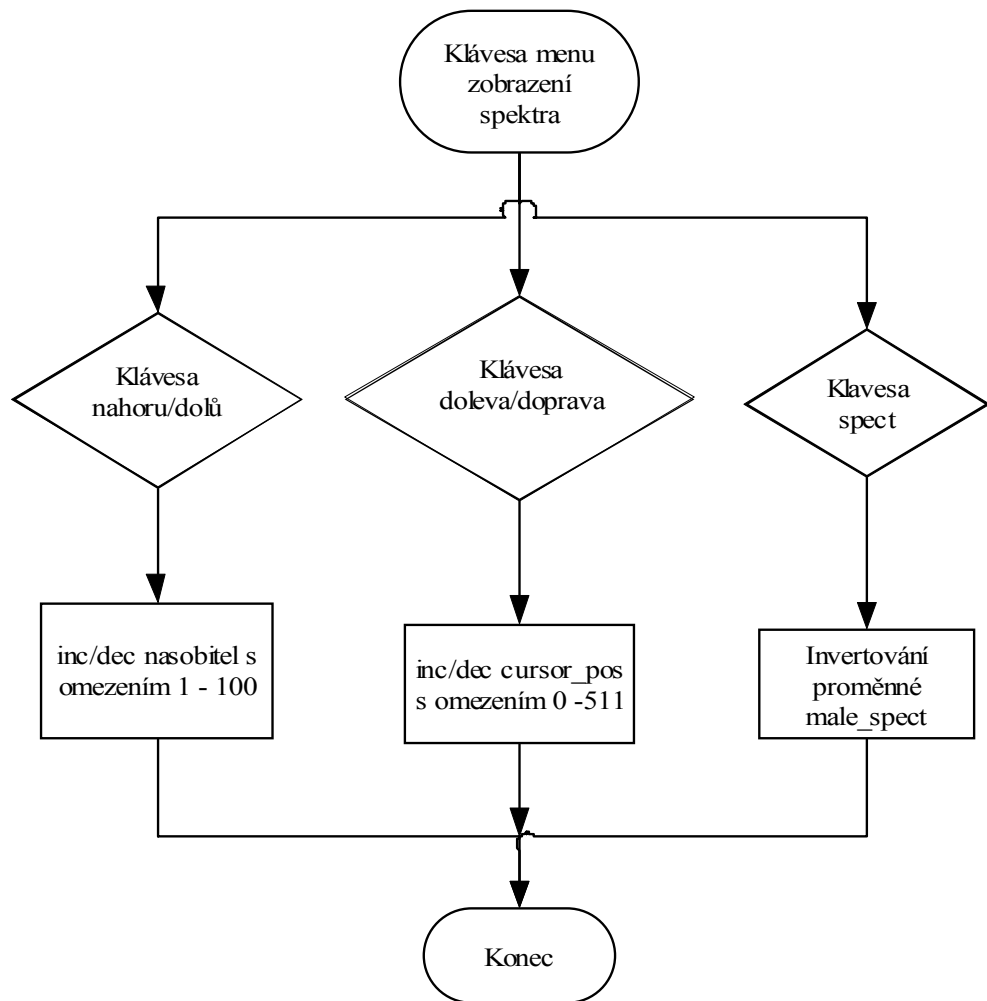
Vykreslování spektra by mohlo vypadat následovně.





Obrázek 29 – vývojový diagram zobrazení spektra

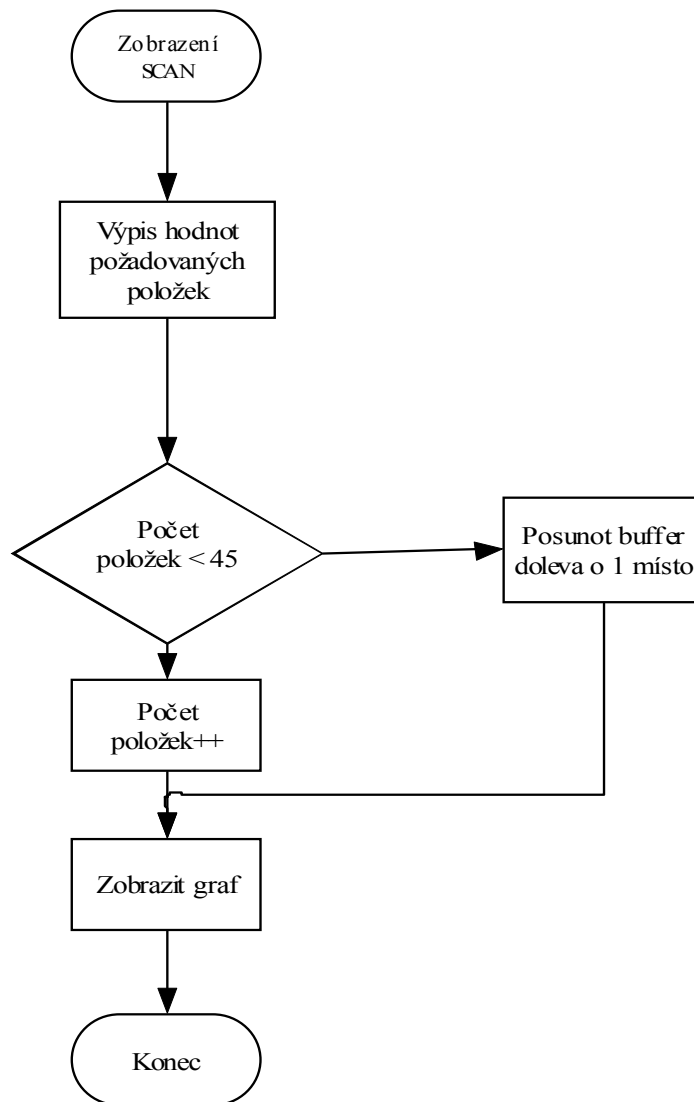
A řízení zobrazení



Obrázek 30 – řízení zobrazení spektra

3.4.6 Zobrazení měření v módu SCAN

Pro toto měření si vytvoříme pro zobrazování naměřených údajů zase zvláštní funkci, která bude vykreslovat požadované hodnoty viz Popis měření v módu SCAN. Pro zobrazení těchto hodnot budeme potřebovat buffer pro naměřené hodnoty vybrané ROI o velikosti 44 položek (body jsou od sebe o 5 pixelů na x-ové souřadnici), dále budeme potřebovat položku s počtem již naměřených hodnot a samozřejmě položku určující, jakou ROI chceme měřit. Tyto hodnoty se zobrazí po skončení měření a následně se spustí nové. Pokud jsme již zaplnili celý buffer s naměřenými hodnotami ROI musíme tento posunout o jednu položku doleva abychom udělali místo pro novou hodnotu.



Obrázek 31 – vývojový diagram zobrazení SCAN

3.4.7 Zobrazení chybových hlášek, varování a stavů

Zde máme dvě možnosti, využít speciální menu pro chybové hlášky a varování, nebo zobrazit hlášení a čekat na stisk klávesy. Tak či tak se musí stejně počkat na stisk klávesy. Pro přehlednost je dobré chybové hlášky mít pohromadě proto vytvoříme pro tyto účely speciální menu. Chybové hlášky budeme do aplikace posílat pomocí funkce *SendMessage* chybovou zprávou. V těle funkce zpracovávající tuto zprávu se nastaví odpovídající *menu_id* a vypíše se chybové hlášení. Pro

vartování a stavy jako výsledek testu, vymazání paměti, neexistenci dat aj. je lepší použít druhého způsobu, protože si myslím, že je to v těchto situacích přehlednější.

3.4.8 Zobrazování časově proměnných hodnot

V podstatě jde o dvě hodnoty v hlavním menu a to o stav napětí baterie a zobrazování času. Zde je samozřejmě z časových důvodů nevhodné přepisovat celé menu, ale stačí přepisovat jen dané údaje.

Protože systém, ve kterém se řídí menu podporuje oznámení při změně napájení o desetinu voltu tak můžeme pokud aktuální *menu_id* odpovídá hlavnímu menu vypsat aktuální hodnotu napětí baterie.

Dále systém do aplikace posílá každou sekundu zprávu pro tyto účely, dá se vypnout a zapnout, takže zde můžeme přepisovat datum a čas.

4. KOMUNIKACE Z NADŘAZENÝM SYSTÉMEM

V komunikaci s nadřazeným systémem jde u tohoto přístroje jen o to přenášet naměřené hodnoty do počítače a případně kalibrační konstanty z počítače do přístroje pomocí USB.

Pro zajištění komunikace přístroje s PC pomocí USB je v přístroji obvod FTDI FT245BM, který integruje na čipu celý USB protokol. Z tohoto důvodu se není nutné starat o opravy chyb, adresaci, atd., podrobný popis vlastnosti tohoto obvodu viz [9].

Tento obvod je připojen k MCU přístroje pomocí 8b sběrnice a dále jsou připojeny signály RD#, WR, RXF#, TXE#, a PWREN#. Signál PWREN# je použit k detekci připojení, odpojení USB kabelu (USB blok je napájen jen z USB) a signál RXF# je připojen na vstup vnějšího přerušení. Pomocí toho přerušení se dá detekovat jestli přišla nějaká data, v případě že vnitřní buffer pro příjem obvodu FTDI FT245BM je prázdný.

Dále je nutné vyřešit komunikaci s tímto obvodem ze strany PC. Výrobce k tomuto obvodu dodává ovladače pro Windows a to dvou druhů VCP a D2XX viz [9].

Ovladače VCP emulují po připojení zařízení s obvodem FTDI FT245BM jako sériový port, což umožňuje jednoduchou komunikaci s přístrojem. Na druhé straně nejsme schopni bez nutnosti navázání komunikace s přístrojem zjistit o jaký přístroj jde a jestli to není přístroj jiného výrobce s tímto obvodem. Další nevýhodou tohoto řešení, je že tento virtuální COM port se mapuje na první nalezený volný COM port což následně komplikuje nastavení komunikace, hlavně pro uživatele, kteří výpočetní technice moc nerozumí.

Ovladače D2XX, umožňují přímý přístup k USB zařízení pomocí funkcí DLL knihovny, popis těchto funkcí viz [9]. Základní výhodou těchto ovladačů je možnost vyhledávat připojená zařízení s USB obvodu FtdiChip a připojit se k vybranému zařízení podle jména nebo sériového čísla, což je oproti ovladačům VCP podstatná výhoda. Abychom mohli využít těchto výhod je nutné aby k obvodu FTDI FT245BM byla připojená sériová EEPROM připojení a parametry viz [9]. Do této EEPROM je

následně možné zapsat výrobce, jméno přístroje a sériové číslo. Toto lze provést programem MProg 3.0a dodávaným výrobcem, který umožňuje uložení výše zmíněných údajů přímo přes USB do konfigurační EEPROM.

4.1 KOMUNIKACE PŘÍSTROJE S PC

Hlavním požadavkem na přístroj je přenést uložená data do PC a kalibrační konstanty z PC do přístroje.

Původní typ Spektrometru používal pro komunikace sériové rozhraní pomocí COM portu a přenášená data byla v textovém formátu aby se výstup z přístroje nemusel zpracovávat nejdříve dodávaným programem pro přenos dat, ale přímo zpracovávat speciálními zpracovatelskými programy. Z tohoto důvodu bylo požadováno zachování této koncepce i při komunikaci pomocí USB. Tato koncepce se nakonec opustila z důvodu nevýhod VCP ovladačů.

Pokud použijeme ovladače V2XX není nutné trvat na přenosu dat v textové podobě, ale můžeme přenášet data binárně. Výhodou této koncepce je menší zatížení procesoru přístroje při přenosu protože se nemusí naměřené hodnoty převádět na text, toto se provede až v PC, a není nutná výrazná manipulace s uloženými daty posílají se přímo tak jak jsou uložena jen s malými úpravami.

Jak již sem zmínil je nutné při odesílání dat do PC provést drobné úpravy, které je nutné vykonat ze dvou důvodů:

1. Přístroj používá Big Endian a Windows Little Endian
2. 32b platformy x86 vyžadují zarovnání dat na 4B hranici tedy začátek každé proměnné nebo pole musí začíná na adrese kde nejnižší dva bity jsou 00b

Ad. 1) Tento problém se dá vyřešit s minimálním nebo žádným zatížením procesoru. Pokud budeme data v přístroji odesílat a přijímat po bajtech od MSB k LSB. V případě složitějších datových typů jako pole a struktury můžeme použít stejnou koncepci s tím, že uspořádání položek v poli a struktuře bude v opačném pořadí, tedy hodnoty které byly původně na začátku pole budou nyní na jeho konci.

Ad. 2) Řešení tohoto problému bude potřebovat jistý čas procesoru přístroje. Nejednodušeji jak je tento problém možno vyřešit je transformovat uložená data tak

aby jednotlivé položky různých typu byly zarovnány na 4B hranici (jedná se jen o základní data).

4.2 FORMÁT ODESÍLANÝCH A PŘIJÍMANÝCH DAT

Uložená data v přístrojích mohou být dvojího typu a to buď jenom hlavička obsahující základní data, nebo i naměřené spektrum.

Základní data jsou reprezentována následující strukturou:

```
struct
{
    tm          m_end_time; //čas kdy bylo ukončeno měření
    UCHAR      m_doba_s;    //doba měření sekundy
    UCHAR      m_doba_m;    //minuty
    UCHAR      m_doba_h;    //hodiny
    int         line;       //linka na, které byla data měřena
    int         pos;         //pozice na lince
    float       ded_time;    //celková mrtvá doba měření v %
    WORD        cosmic_ch;   //počet impulsů nad rozlišovací schopností ADC
    float       polosirka;   //střední hodnota šířky stabilizačního píku
    float       peak;        //poloha stabilizačního píku
    long        ROI[8];      //počet impulsů v jednotlivých ROI
    WORD        gain;        //poslední nastavené zesílení předzesilovače
}ZAKL_DATA;
```

Spektrum je reprezentováno polem 4B celých čísel o velikosti 512 hodnot. Protože jak jsem se již zmínil musí být položky stejného typu, které následují za sebou zarovnány na 4B hranici, je třeba tuto strukturu před osedláním zkopírovat do jiné, která tuto podmínku splňuje a vypadá následovně:

```
typedef struct
{
    char        typ;
    UCHAR      balast2;
    tm         m_end_time;
    UCHAR      balast;
    UCHAR      m_doba_s;
    UCHAR      m_doba_m;
    UCHAR      m_doba_h;
    int        line;
    int        pos;
    float      ded_time;
    DWORD     cosmic_ch;
    float      polosirka;
    float      peak;
    long      ROI[8];
    DWORD     gain;
}ZAKL_DATA_SEND;
```

Jednotlivé položky mají stejný význam jako předchozí. Rozdíl je jen ve dvou věcech a to

1. je přidaná položka typ určující o jaký typ dat se jedná(jenom data, data i spektrum a nebo kalibrační spektra, K, U, Th, 0 a B
2. Jsou přidány dvě položky balast a balast2, sloužící pro zarovnání na 4B hranici. V PC tyto položky není nutné definovat

Protože hodnoty odesílaného spektra jsou 4B hodnoty nemusíme se starat o zarovnání a odesílat je přímo.

V programu pro PC musí být tato struktura definována takto:

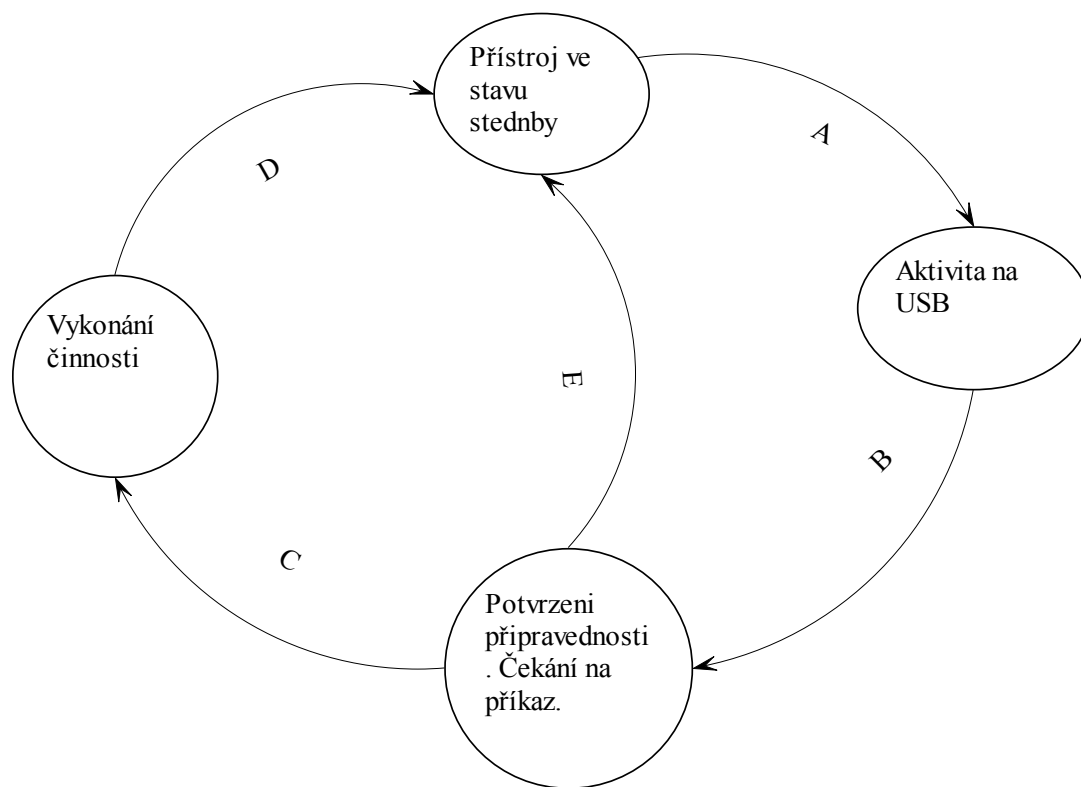

```
typedef struct
{
    DWORD    gain;
    long     ROI[8];
    float    peak;
    float    polosirka;
    DWORD    cosmic_ch;
    float    ded_time;
    short    pos;
    short    line;
    UCHAR    m_doba_h;
    UCHAR    m_doba_m;
    UCHAR    m_doba_s;
    tm       m_end_time;
    char     typ;
}ZAKL_DATA_SEND;
```

Jak je patrné položky jsou uspořádány v opačném pořadí a jsou odstraněny položky balast a balast2. Je zřejmé, že je nutné definovat v PC strukturu tm s položkami v opačném pořadí popis této položky viz standardní knihovna C time.h

Pokud posíláme do přístroje kalibrační konstanty, jedná se o hodnoty rozsahu ROI což je 16 2B hodnot a kalibrační konstanty 16 4B hodnot. Kalibrační konstanty můžeme odesílat bez problému se zarovnáním na 4B hranici. Hodnoty rozsahu ROI budeme posílat jako pole, které obsahuje na první pozici začátek ROI1, konec ROI1 atd. Zde není nutné se zabývat zarovnáním na 4B hranici, protože celé toto pole již je zarovnáno na tuto hranici.

4.3 KOMUNIKAČNÍ PROTOKOL

Jak již jsem se zmínil byla původně požadovaná koncepce pomoci virtuálního sériového portu a proto komunikační protokol vychází z původní koncepce komunikace pomoci sériového portu. Tato koncepce vypadá následovně:

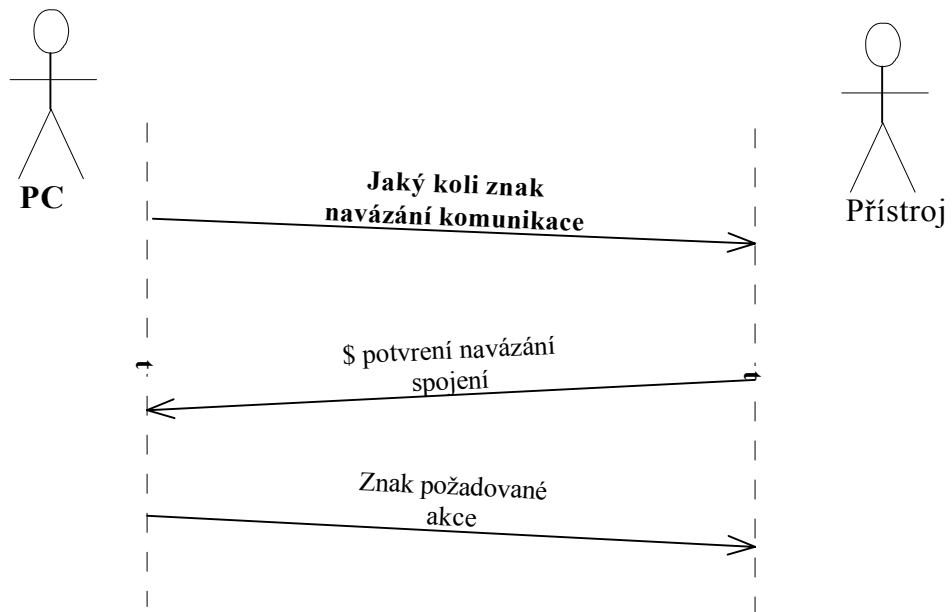


Obrázek 32 – Stavový automat komunikace s přístrojem

- A. přerušení od obvodu FTDI FT245BM, že přišla nějaká data (vnitřní buffer musí být prázdný). Z PC je tedy nutno poslat přes USB jakýkoliv 1B.
- B. oznámení aplikace že došlo k aktivitě na USB. Potvrzení že je přístroj připraven pro komunikaci, odesláním nějakého znaku např. „\$“ a čeká na příkaz.
- C. vykonání příkazu dle příkazu. Příkazy viz Tabulka 3.
- D. po skončení přenosu přechází přístroj do stavu Standby.
- E. neznámý příkaz. Přístroj přechází do stavu Standby.

Znak	Požadovaná činnost
"d"	Přenos uložených dat z přístroje do PC
"c"	Přenos kalibračních konstant a rozsahu ROI s PC do přístroje

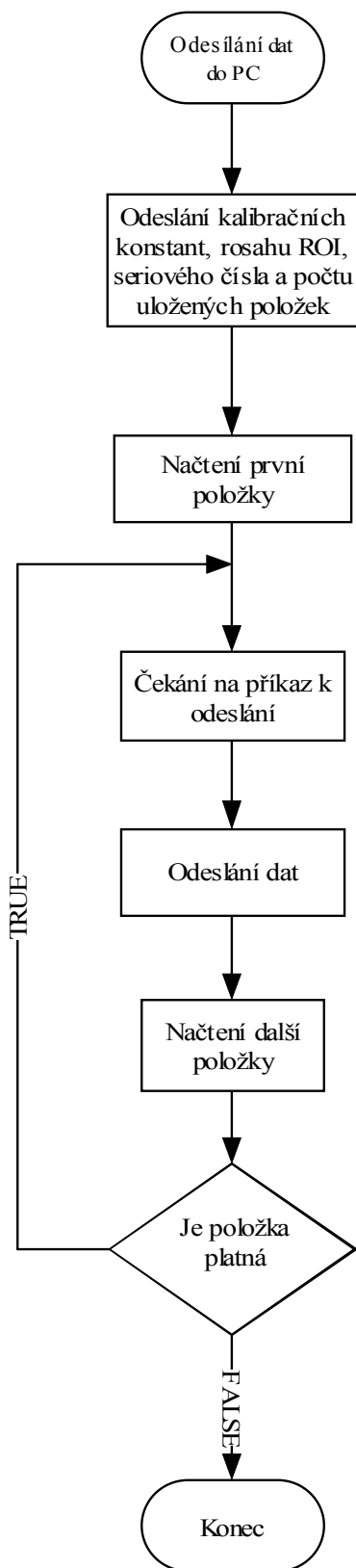
Tabulka 3 – příkazy činnosti komunikace



Obrázek 33 - navázání komunikace

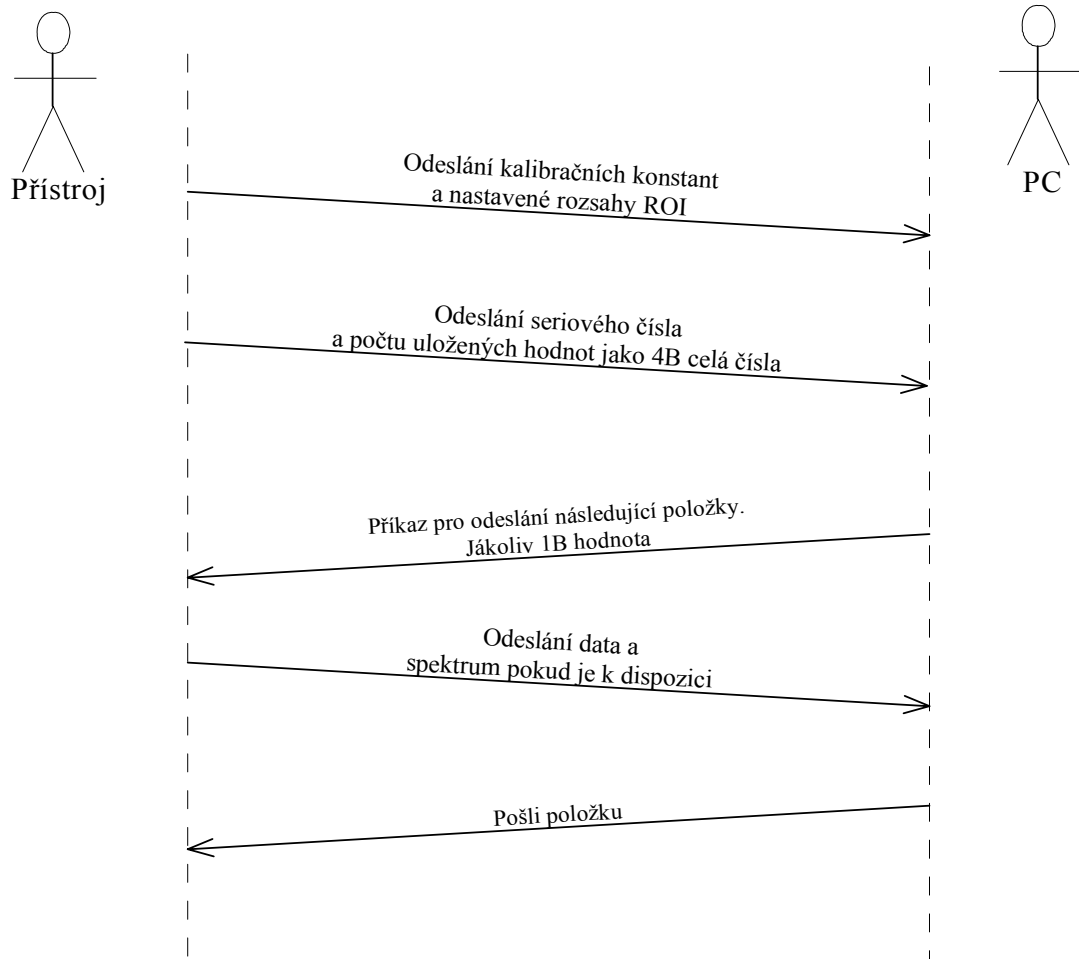
4.4 ODESÍLÁNÍ DAT DO PC

Pro tuto činnost si vytvoříme funkci, která se bude starat o odesílání dat do PC. Tato funkce se bude volat jako odezva na odpovídající příkaz viz předchozí kapitola. Přenos probíhá následovně s tím, že sériové číslo a počet položek jsou 4B celá čísla.



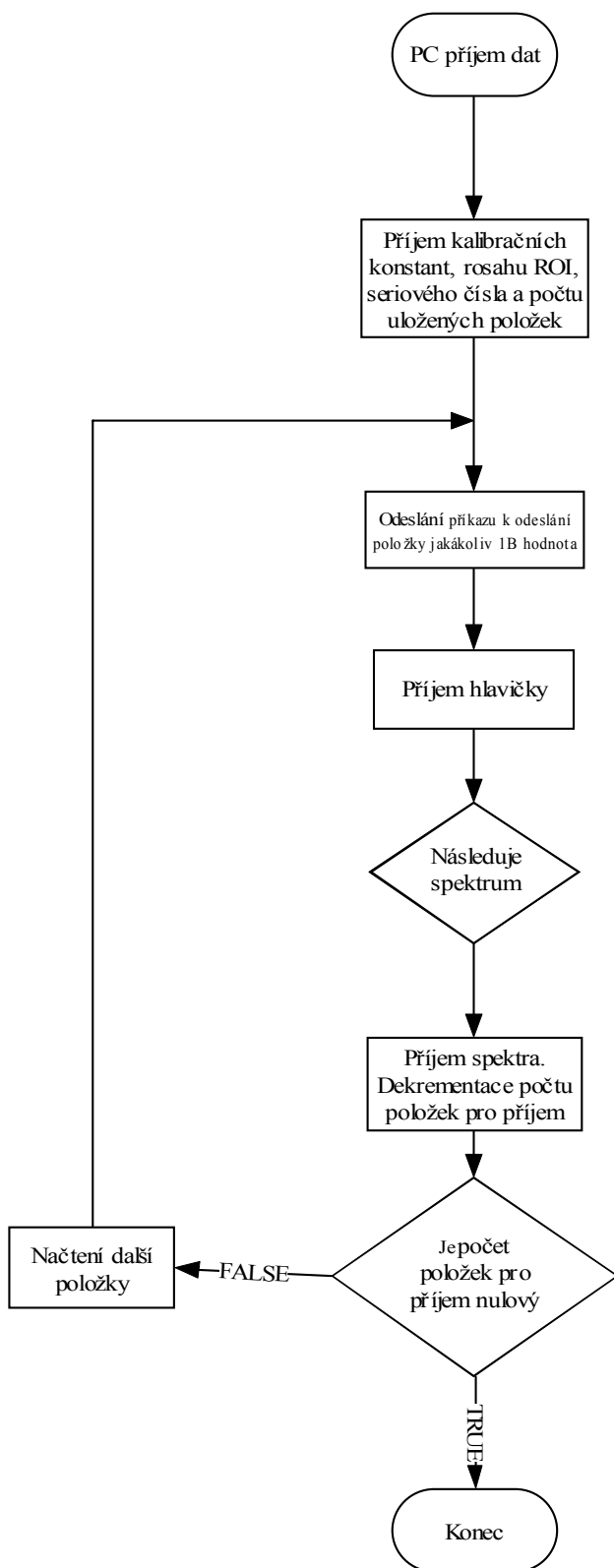
Obrázek 34 – vývojový diagram odesílání dat do PC

Výměna dat na USB bude pak vypadat takto:



Obrázek 35 - odesílání dat do PC

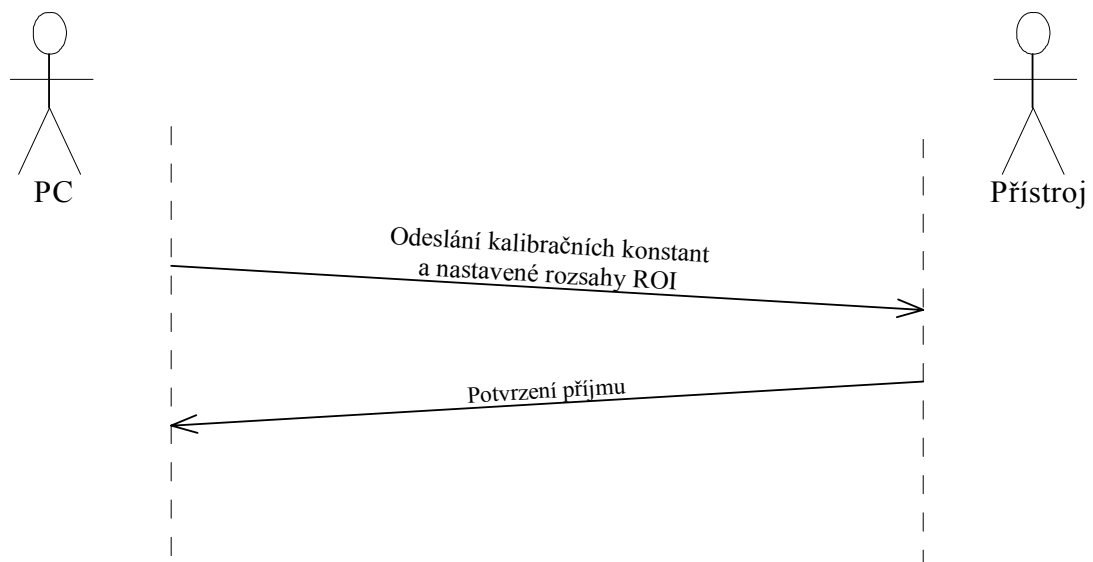
Příjem na straně PC bude vypadat potom takto:



Obrázek 36 – stavový diagram příjmu dat na straně PC

4.5 PŘÍJEM KALIBRAČNÍCH KONSTANT Z PC

Po příjmu příkazu pro příjem kalibračních konstant a rozsahu ROI je třeba tyto konstanty přijmout dle *Obrázek 37* a následně tyto hodnoty uložit.



Obrázek 37 - příjem kalibračních konstant z PC

5. ZÁVĚR

Celý systém menu je vytvořen modulárně a je vícenásobně použitelný, umožňuje jednoduché rozšíření přidáním další položky case pro *menu_id* a bloku switch pro položky menu *mmpos*. Z popisu řízení menu je patrné, že navržený systém řízení menu je jednoduchý a je závislý na třech hodnotách. Řízení menu je nutné řešit bloky switch, protože toto je podstatně rychlejší a hlavně přehlednější než použití if.

V práci jsem také částečně popsal programové jádro do kterého je celý systém menu a komunikace zařazen. Detailně jsou v *kapitole 2* popsány postupy jak řešit vstup klávesnice, výstup textu a zobrazení grafiky.

V *kapitole 3* je proveden rozbor menu, principy ovládání tohoto menu a řešení jednotlivých požadavků na toto menu. V této kapitole jsou také popsány funkce použité pro vstup údajů od uživatele. Jsou to čísla, jak celá tak desetinná, dále je to datum, čas a údaje jako ON/OFF.

Funkce pro vstup hodnot od uživatele jsou popsány univerzálně a mohou být implementovány v různých zařízeních. Jejich implementace závisí na typu použitého displeje. Pracuje-li displej v textovém režimu, není třeba žádná výrazná úprava, pokud se funkce pro řízení kurzoru jmenují stejně. Tyto funkce jsou také navrženy tak, aby mohly být využity v menu pro vstup více hodnot, což zvyšuje jejich použitelnost.

Vykreslování jednotlivých menu je velice jednoduché, protože tisk znaků na displej je propojen s funkcí printf. V tomto případě se však znaky nejprve tisknou do paměti v CPU, protože jsou generovány SW a nemají stejnou šířku, což vyžaduje častý přístup k řadiči displeje, jak je popsáno v *kapitole 2*. Z tohoto důvodu je třeba po vykreslení menu zapsat tato data do displeje.

Protože jsou znaky generovány programově, je zde i možnost jednoduše změnit jak velikost písma tak i fonty bez nutnosti velkých úprav.

Komunikace z nadřazeným systémem je v tomto případě velice jednoduchá, protože je požadován jen přenos dat do PC a kalibračních konstant a rozsahu ROI do přístroje z PC. Protože je použit obvod FTDI FT245BM nemusíme se starat o

protokol USB ani zabezpečení dat při přenosu. Jak je v *kapitole 4* popsáno, jsou využity ovladače D2XX, které umožňují přímou komunikaci mezi PC a přístrojem. Tento typ ovladačů také umožňuje komunikačnímu programu vyhledat zařízení připojené k PC pomocí jeho jména, což je pro uživatele oproti předchozí verzi přístroje velká výhoda. V minulosti bylo pro komunikaci použito RS-232. Dále jsou v této kapitole rozebrány formáty dat a řešení problému se zarovnáním proměnných na zásobníku přístroje a PC.

Na závěr bych se chtěl říci, že téma své diplomové práce jsem si vybral tak, abych mohl spojit studium a práci. Popsána je jen část programu gama spektrometru GS-512, nicméně jsem jej udělal celý a ověřil jsem, že funguje. Přístroje s tímto programem se prodávají do celého světa a doposud se neobjevila žádná výrazná závada.

6. LITERATURA

- [1] Doc. Ing. Josef Šeda, CSc, Ing. Josef Sabol, aj.: Jaderná elektronika, SNTL 1977
- [2] P. W. Nicholson: Nuclear Electronics, John Wiley & sons Ltd 1974, ISBN 0 471 63697 5
- [3] FreeScale: MS9S12XDP Data Sheet, Rev. 2.15, July 2006, [on-line] www.freescale.com , 12.5.2008
- [4] TOSHIBA: T6936 DOT MATRIX CONTROL LSI, 22.8.2001, [on-line] <http://www.515ic.com/down.aspx?id=107> , 12.5.2008
- [5] EPSON TOYOCOM: Real Time Clock Module RX-4045SA/NB, [on-line] http://ndap3-net.ebz.epson.co.jp/w/www/PDFS/epdoc_qd.nsf/WE_rtc_sif/880ED1FD9343C0F94925707D003F3958?OpenDocument , 22.8.2008
- [6] RNDr. Jana Křížová, aj.: Gama Ray Spectrometer GS-512, SatisGeo s. r. o. Brno 2008
- [7] FAIRCHILD SEMICONDUCTOR: MM74C923 20-Key Encoder, revize duben 2001 [on-line] <http://ee.usc.edu/library/ee459/datasheets/NN74C922.pdf> , 16.5.2008
- [8] RNDr. Jana Křížová, aj.: Gama Ray Spectrometer GS-512, SatisGeo s. r. o. Brno 2005
- [9] FtdiChip: Dokumentace a popis obvodu FT245BM aj. [on-line] <http://www.ftdichip.com/Products/FT245BM.htm>, 10.3.2008
- [10] Výheň: Besenhamův algoritmus [on-line] <http://nuane.com/vyhen/07/bres.html> , 16.5.2008
- [11] Wiki knihy: Rasterizace [on-line] <http://cs.wikibooks.org/wiki/Rasterizace> , aktualizace 1.12.2007, 16.5.2008
- [12] Počítačová grafika: Generování úsečky v rastru, [on-line] <http://pocitacova-grafika.kvalitne.cz/dda-bresenham-usecky> , 16.5.2008

SEZNAM PŘÍLOH

- Příloha 1 Hierarchie položek menu, List A3
- Příloha 2 CD se zdrojovými kódy a elektronickou verzí tohoto dokumentu
a elektronická verze stavových diagramů