VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

BAKALÁŘSKÁ PRÁCE

2011 Petr Drábek

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra měřicí a řídicí techniky

Mobilní přístroj pro zobrazení EKG signálu

Mobile ECG Signal Viewer

2011 Petr Drábek

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra měřicí a řídicí techniky

Zadání bakalářské práce

Student: **Petr Drábek**

Studijní program: B2649 Elektrotechnika

Studijní obor: 2601R004 Měřicí a řídicí technika

Téma: Mobilní přístroj pro zobrazení EKG signálu
Mobile ECG Signal Viewer

Zásady pro vypracování:

1. Obecný popis problematiky nízkoúrovňových signálů v konceptu jejich zpracování mikrokontrolérem.
2. Návrh a realizace jednotky pro měření EKG s grafickým displejem LCD.
3. Návrh možnosti ukládání dat a off-line analýzu pomocí PC.
4. Test měřicího řetězce.
5. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

1. MATOUŠEK, D. *Práce s mikrokontroléry ATMEL AVR - ATmega16.* 1. vyd. Praha: BEN - Technická literatura, 2006. 320 s. ISBN / EAN 80-7300-174-8 / 9788073001742.
2. HESSELSON, A. B. *Simplified Interpretation of Pacemaker ECGs: An Introduction.* Hoboken(USA): Wiley-Blackwell, 2007. 304 s. ISBN-10: 1405103728. ISBN-13: 978-1405103725.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Zdeněk Slanina, Ph.D.**

Datum zadání: 19.11.2010

Datum odevzdání: 06.05.2011

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA * Fakulta elektrotechniky a informatiky * OSTRAVA *

doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

PODĚKOVÁNÍ

Děkuji vedoucímu mé diplomové práce Ing. Zdeňkovi Slaninovi, Ph.D. za pomoc a podporu při tvorbě diplomové práce.

PROHLÁŠENÍ

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Datum: 6. 5. 2011 Podpis

ABSTRAKT

Měření EKG, nebo-li měření elektrické aktivity srdce je v dnešní medicíně jedno z nejčastěji skloňovaných pojmů. Měření EKG je základní vyšetřovací metoda v kardiografii, je vyžadována jak při složitých operacích, přes sledování hospitalizovaných pacientů až po měření jednotlivců při určování diagnózy. Z EKG lze vyhodnotit správnou činnost srdce, tepovou frekvenci, ale co je nejdůležitější, že z křivky lze poznat poruchy srdečního rytmu, tzv. srdeční arytmie, o kterých člověk postižený touto nemocí nemusí vědět.

Cílem této bakalářské práce je realizace mobilního zařízení, které je schopné nejen měřit a zobrazovat EKG na vlastním displeji, ale i umožnit další analýzu naměřených dat pomocí PC. Rozhodujícím kritériem je co největší spolehlivost, kvalita a nedílnou součástí při vyhodnocení jsou samozřejmě i vstupní náklady na výrobu. Po zhodnocení všech těchto parametrů se nabízí řešení pomocí mikrokontroleru ATmega16, grafického displeje s rozlišením 128x64 a integrovaného převodníku ADS 1198, který je určen přímo pro měření EKG a EEG.

ABSTRACT

The measurement of ECG, or the measurement of the electrical activity of the heart, is in modern medicine one of the most often mentioned concepts. The measurement of ECG is the basic investigative method in cardiography, and is required in cases such as before complex surgeries, or while monitoring hospitalized patients up to the measurement of individuals before the determination of a diagnosis. From the ECG, it is possible to analyze the regular activity of the heart, pulse rate, but most importantly, from the curve it is possible to recognize irregularities in heart beats, so-called cardiac arrhthymia, about which a person afflicted with this illness may not even know.

The goal of this bachelor's thesis is the implementation of a mobile appliance which is capable not only of measuring and displaying the ECG on its own screen, but also of making possible further analysis of measurement data with the help of a PC. Above all, dependability, quality, and the lowest cost possible serve as the decisive criteria. Following analysis of all of these variables, the use of a microcontroller ATmega16, a graphic display with a resolution of 128x64 and an integrated converter ADS 1198 are suggested as solutions, which is intended directly for the measurement of ECG.

KLÍČOVÁ SLOVA

EKG, mikrokontrolér, ASD1298, ATmega16, ATmega128, měření, analýza

KEY WORDS

ECG, microcontroller, ADS1298, ATmega16, ATmega128, measurement, analysis

Seznam použitých symbolů a zkratek

A/D	Analogově-digitální převodník
EKG (ECG)	Elektro-kardiografická křivka
EEG	Elektro-encefalografická křivka (vyšetření elektrické aktivity mozku)
USB	Univerzální sériová sběrnice
JTAG	Typ programování, který umožňuje sledovat procesy v MCU pomocí PC
TTL	Technologie Tranzistor Tranzistor Logic
CMOS	Technologie Complementary Metal-Oxide-Semiconductor (doplňující se o kov-oxid-polovodič)

Obsah

1 ÚVOD 1
2 EKG – parametry snímaného signálu 2
2.1 Metody měření EKG 3
3 Popis snímání nízkoúrovňových signálů 5
3.1 Zpracování signálu mikrokontrolerem 7
4 Návrh jednotky pro měření EKG s grafickým displejem LCD 9
4.1 ATmega16 MCU parametry 9
4.1.1 Programátor JTAG 10
4.2 Převodník ADS 1298 parametry 11
4.3 Parametry grafického displeje 13
4.4 Návrh ukládání dat na SD kartu 14
5 Realizace jednotky pro měření EKG s grafickým displejem LCD 16
5.1 Hardwarová část 16
5.2 Programová část 18
5.2.1 Hlavička programu 18
5.2.2 Popis základních proměnných 20
5.2.3 Software pro zobrazení průběhu EKG 20
5.2.4 Funkce tep 25
5.2.5 Displej – inicializace, funkce 26
5.2.6 Hlavní smyčka 27
5.3 Uživatelské menu 28
6 Test měřícího řetězce 32
7 Zhodnocení dosažených výsledků 35
8 Použitá literatura 36
9 Seznam příloh 37

1 ÚVOD

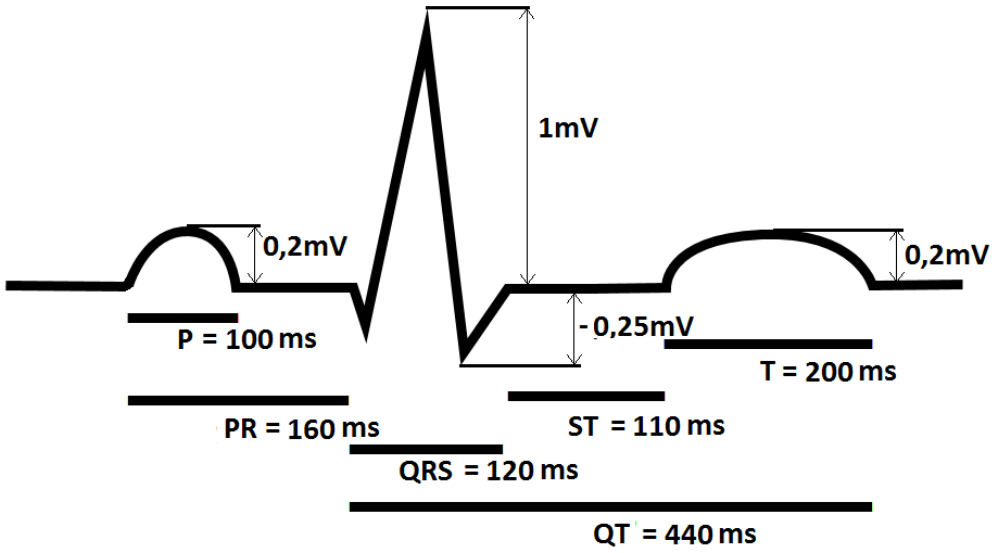
Bakalářská práce je zaměřena na sestrojení mobilního přístroje k měření elektrické aktivity srdce, který umožní jak dlouhodobé monitorování pacienta, tak i detailní zkoumání křivky EKG ať už přímo v samotném přístroji nebo v PC. Byla snaha vyvinout takový přístroj, který bude splňovat požadavky pro stanovení diagnózy lékařem, a současně bude navržen tak, aby každý pacient byl schopný v domácím prostředí změřit EKG křivku sám nebo za pomocí rodinného příslušníka. Tento způsob měření nese jak pro specialisty, tak i pro pacienty značnou míru výhod. Z pohledu specialisty přináší mobilní přístroj především možnost přenosu na místo potřeby a v případě, že měření EKG provádí sám pacient doma, je umožněno pravidelné diagnostikování pomocí PC bez osobního setkání. Tuto výhodu ocení především lidé s podezřením na vadu srdce, u kterých je nutné dlouhodobé sledování, a stejně tak pacienti po operaci srdce.

Největší nevýhodou již existujících zařízení, která fungují na tomto principu, je především jejich vysoká cena. Mnohdy jsou právě pořizovací náklady omezujícím aspektem jejich využití ať už pro zdravotnická zařízení nebo přímo pro pacienta.

Cílem bakalářské práce je proto navrhnout a vyvinout zařízení, které bude stejně funkční, přičemž jednotlivé komponenty budou cenově přijatelnější, než je tomu u běžných zařízení, neboť tato vyšetřovací metoda je jedna z nejčastěji užívaných v oblasti analýzy správné srdeční činnosti.

2 EKG – parametry snímaného signálu

Správnou funkci srdce zajišťuje převodní systém, vedení vzruchů zajišťují buňky převodního systému, které nejen vedou proud, ale i samy jeho průběh periodicky vytváří. Základní podnět vzniká v sinoatriálním uzlu, z kterého se vzruchy dále šíří přes atrioventrikulární uzlík do Hlisova svazku, z kterého se vzruch šíří do Tawarových ramének. Z grafu EKG je patrné že se jedná o nízkoúrovňový signál, jehož amplituda je řádově několik milivoltů.[1] Viz. Obr. 1



Obr. 1 Zobrazení signálu EKG zdravého srdce

Popis jednotlivých částí EKG signálu:

P – Depolarizace síní.

PR – Doba, za kterou vzruch proběhne od sinusového uzlu převodní soustavou až k pracovnímu myokardu komor.

QRS – Depolarizace komor.

QT – Celková doba aktivity srdečních komor.

ST - Doba od S po začátek T, tato část je důležitá, lze z tvaru křivky poznat infarkt myokardu.

T – Doba depolarizace komor.

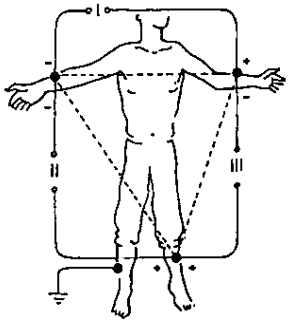
Nejkratším časovým úsekem je QRS komplex, jehož doba trvání je 60 až 120 milisekund, kde signál třikrát překmitne. [2]

Měření EKG (elektrokardiografické křivky, také ECG) dnes patří mezi jednu základních diagnostických metod určení správné nebo taky špatné činnosti srdce. Na křivce můžeme zpozorovat vady srdce, částečnou fibrilaci, nečinné ložisko buněk srdce.

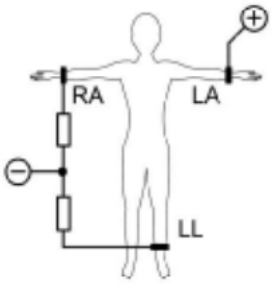
2.1 Metody měření EKG

Poprvé zavedl pojem EKG jako klinickou metodu E. W. Einthoven. Definoval EKG jako rozdílný potenciál mezi dvěma končetinami = svod.

Označíme -li pravou ruku písmenem R (right, standardně označena červenou barvou) a levou L (left, žlutá), pak signál L-R označujeme jako I. Einthovenův svod. Později byla další elektroda připevněna poblíž kotníku levé nohy F (foot, zelená) a tím pádem možnost měřit rozdíl potenciálů F-R (II. Einthovenův svod) a F-L (III. Einthovenův svod). Elektroda N (neutrální - černá) Viz. Obr. 2 [7]

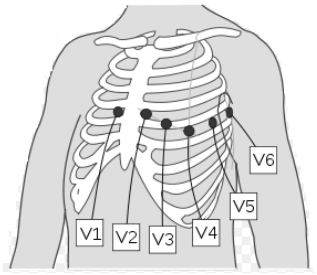


Obr. 2 Einthovenovy svody



Obr. 3 Goldbergovy svody

Jako druhý popsal měření činnosti srdce pomocí EKG Goldberg, kdy Einthovenovy svody jsou vždy doplnění o odporníky stejné velikosti a potom spojeny a tvoří vždy záporný pól. Kladný pak tvoří vždy zbývající končetina Viz. Obr. 3. Jedná se o takzvané zesílené svody. *Nejefektivnější snímání rytmů srdce je však pomocí Wilsonových hrudních svodů. Jedná se o šest svodů umístěných podél 4. a 5. mezižebří, kdy první elektroda je napravo od sterna a všechny další nalevo od sterna* Viz. Obr. 4 [8] Těchto 6 svodů se vždy měří vůči takzvané Wilsonově svorce, což jsou 3 Goldbergovy svorky, které jsou k sobě paralelně připojeny a každý svod je sériově připojen přes rezistor o velikosti 5kΩ .

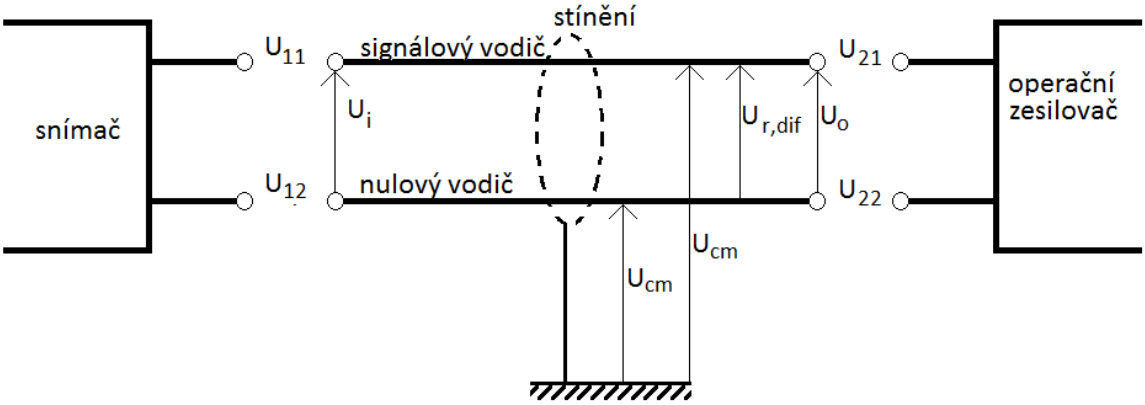


Obr. 4 Wilsonovy svody

Principy snímání a forma zobrazování EKG se od prvopočátku do dneška změnily. První záznam byl prováděn na papír, dnes už zápis EKG probíhá digitálně. Dříve se zobrazovalo například přes elektronkovou obrazovku.

3 Popis snímání nízkoúrovňových signálů

Základními prvky pro měření nízkoúrovňových signálů je snímač (v našem případě to jsou elektrody), signálové vedení, operační zesilovač s galvanickým oddělením vstupu a dolnopropustný filtr. Během měření mohou na vedení vznikat rušivá napětí, takovýto případ je znázorněn na obrázku 5.



Obr. 5 Signálové vedení

$$u_{21}(t) = u_o(t) + u_{r,dif}(t) + u_{cm}(t)$$
$$u_{22}(t) = u_{cm}(t)$$

$$u(t) = u_{21}(t) - u_{22}(t) = u_o(t) + u_{r,dif}(t) + u_{cm}(t) - u_{cm}(t)$$
$$u(t) = u_o(t) + u_{r,dif}(t) + u_{cm}(t) - u_{cm}(t)$$
$$u(t) = u_o(t) + u_{r,dif}(t)$$

Rovnice 1

Napětí na konci vedení, označené jako U_o se může výrazně lišit od napětí, které je na začátku vedení U_i, jelikož se zde uplatňuje odpor vedení a na signál působí souhlasné a nesouhlasné rušivé napětí. Chybu měření zapříčiněnou odporem vedení lze odstranit kalibrací, ale odstranění rušivých napětí je poněkud obtížnější, protože v případě, kdy je měřený signál velikosti několika milivoltů, mohou rušivá napětí nabývat několikrát vyšších hodnot než měřený signál a tím je změřené napětí U_o prakticky nepoužitelné. Proto je nutné odstranit jak souhlasné rušivé napětí, tak i nesouhlasné.

Souhlasné rušivé napětí U_{cm} se vyskytuje na signálovém i na nulovém vodiči => lze jej odstranit odečtením napětí na operačním zesilovači, výpočet je patrný z rovnice 1.

Avšak na signálu zůstává superponováno nesouhlasné rušivé napětí $U_{r,dif}$, které je dále odstraněno vhodně zvoleným filtrem, který je konstruován tak, aby odstranil nesouhlasné rušivé napětí, čímž zajistíme schopnost rekonstrukce signálu z naměřených vzorků.

Kvalita operačního zesilovače je dána poměrem zesílení diferenčního napětí a zesílení souhlasného napětí, koeficient potlačení souhlasného napětí CMR_Z Viz. Rovnice 2. [3]

$$CMR_Z = \frac{A_{u,dif}}{A_{u,cm}}$$

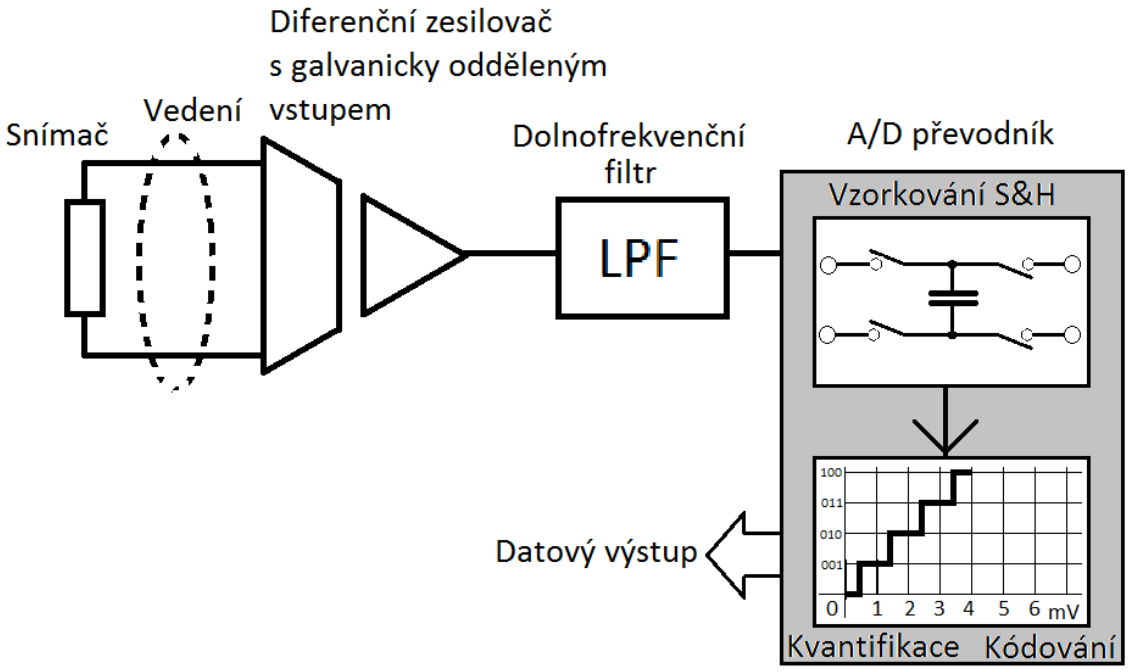
Rovnice 2

Dále mohou vznikat vlivem činnosti operačního zesilovače šumy:

Tepelný šum – je způsobený náhodným pohybem elektronů odporových struktur.

Výstřelkový šum – vzniká při průchodu proudu otevřeným přechodem PN.

Blikavý šum – vniká v oblasti přechodu báze - emitor. [4]



Obr. 6 Měřící řetězec

Zkreslený signál od snímače je přiveden stíněným vedením na svorky diferenčního zesilovače s galvanickým oddělením, kde je odstraněno souhlasné napětí, dále je signál filtrován dolnopropustným filtrem a takto upravený signál je přiveden na A/D převodník, jak je vidět na obrázku 6.

Převod z analogové veličiny na digitální má tři fáze – vzorkování, kvantifikace a kódování. Vzorkování je zajištěno obvodem Sample&Hold, má dva stavy Sampling, na vstup obvodu je připojené vstupní napětí a nabije kondenzátor, poté se odpojí vstup a připojí se

kondenzátor k A/D převodníku = nastává stav Hold a tyto dva stavy se stále opakují a časy, kdy se jednotlivé stavy mění, jsou řízeny logickou jednotkou převodníku, jedná se o vzorkovací frekvenci. Hodnota vzorkovací frekvence je dána Shannon-Kotělnikovým teorémem.

$$\omega_s \geq 2\omega_{\max}$$

Rovnice 3

Vzorkovací frekvence musí být minimálně dvakrát vyšší než nejvyšší frekvence vzorkovaného signálu, aby bylo možné signál rekonstruovat.

Kvantifikace je přiřazení k příslušné analogové hodnotě kvantifikační hladinu. Počet kvantifikačních hladin je dán vztahem **Chyba! Objekty nemohou být vytvořeny úpravami kódů polí.**, neboť jedna hladina je vyhrazena pro nulu. Kvantifikační interval je nejmenší hladina, je to „krok“ převodníku a je definován vztahem $q = \frac{FS}{2^n}$, kde FS je měřitelný rozsah převodníku. V procesu kódování je přiřazen konkrétní kvantifikační hladině binární kód. [3]

3.1 Zpracování signálu mikrokontrolerem

V návrhu je použit mikrokontrolér značky ATMEL, řada ATmega16. MCU je vybaveno základním měřícím řetězcem. Měřící řetězec se skládá z diferenčního vstupu, předzesilovače (gain), jednotky Sample-Hold a integrovaného 10 -bitového A/D převodníku. Signál může být zesílen 10x nebo 200x ale v případě, že použito maximální zesílení na úroveň 200x, pak amplituda signálu bude 0,2V. Převodník by měl rozlišovací schopnost pouze 7 bitů, pak by však na celý rozsah 5V zbylo jen 127 kvantifikačních úrovní a tím citlivost by byla jen 0,0393 V na jednu úroveň jak je patrné z rovnice 4.

$$\pm 1mV \cdot 200 = 0,2V$$
$$2^7 - 1 = 128 - 1 = 127$$
$$q = \frac{5}{127} = 0,0393V / úroveň$$

Rovnice 4

Z výše uvedených výsledků je patrné, že neupravený signál integrovaným převodníkem nelze zpracovat. Nabízí se řešení předzesílení signálu, následné upravení na 0-5V a zpracovávat již takto upravený signál přímo pro MCU, pak by měl převodník rozlišovací schopnost 10 bitů, a 1023 kvantifikačních hladin. Signál EKG nabývá jak kladných tak i záporných hodnot napětí. Podle rovnice 5 by tím bylo určeno 511 kvantifikačních hladin na kladné hodnoty a 511 na záporné.

$$2^{10} - 1 = 1024 - 1 = 1023$$

$$\frac{1023}{2} = 511$$

$$q = \frac{5}{1023} = 0,00488V / úroveň$$

Rovnice 5

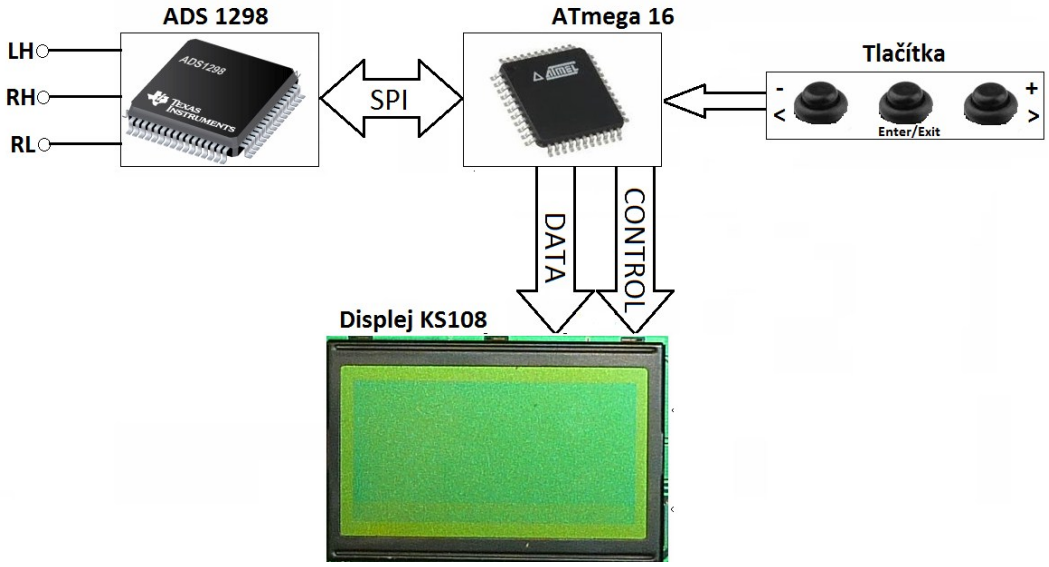
Návrh zesilovače, který by optimálně zesílil jednotky milivoltů na 0 – 5V bez šumu a zkreslení signálu by byl složitý. Proto je EKG měřeno integrovaným obvodem ADS 1298, který je vyvinut pro měření EKG a EEG, tudíž již obsahuje předzesilovače, filtry i A/D převodník. Z převodníku ADS 1298 jsou data přenášena pomocí rozhraní SPI.

4 Návrh jednotky pro měření EKG s grafickým displejem LCD

MCU ATmega16 je jádrem celé měřící jednotky, řídí veškeré procesy. Signál je snímán specifickými elektrodami, umístěných na člověku například na levé ruce, pravé ruce a pravé noze. Převodník ADS 1298 převádí signál EKG na data. MCU pomocí sběrnice SPI řídí příjem naměřených dat z převodníku.

SPI je sériové periferní rozhraní, jež se používá mezi MCU, který řídí komunikaci (master), a více periferiemi (slave). Synchronizace je zajištěna pomocí hodinového signálu SCLK zařízení master. Periferie, s kterou je třeba komunikovat je aktivována masterem, který nastaví CS (chip select) na logickou 0, samotná komunikace mezi dvěma zařízení je uskutečněna přes MOSI a MISO.[5]

Na displej jsou zobrazována data dvěma porty, jeden datový a druhý řídící. Celá jednotka je ovládána pomocí 3 tlačítek, dvě pro pohyb v menu a jedno na potvrzení příkazu, vstup do menu a dále pro únik z menu Viz. Obr. 7



Obr. 7 Blokové měřícího zařízení EKG

4.1 ATmega16 MCU parametry

ATmega16 je nízkoodběrový CMOS 8 -bitový mikrokontroler postaven na RISC architektuře, která umožňuje větší rychlost zpracovávání než CISC, ale díky 131 instrukcím je mikrokontroler na hranici mezi RISC a CISC. K aritmetické logické jednotce (Arithmetic Logic Unit) je hardwarově implementováno 32 registrů. Dále je mikrokontroler vybaven širokou škálou periferii např.: časovači, PWM kanály nebo integrovaným 10 -bitovým převodníkem.

Důležité parametry pro konkrétní aplikaci:

-Taktovací frekvence 16MHz

-16kB programovatelné FLASH paměti

-1kB interní SRAM paměti

-512B paměti EEPROM

-32 libovolně programovatelných vstupů/výstupů

-SPI rozhraní

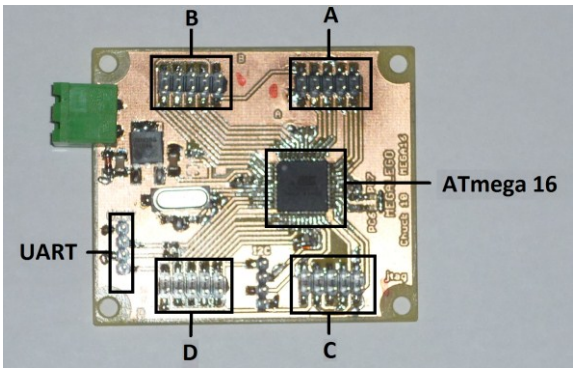
-JTAG programovací rozhraní

-napájecí napětí: $V_{cc} = 4,5 - 5,5V$

-odběr: $I_{cc} = 25mA$

[9]

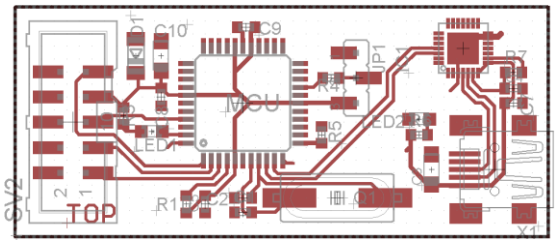
Pro vývoj bakalářské práce je používán vývojový kit, což je DPS, na které jsou vyvedené jednotlivé porty A až D (viz. Obr. 8), dále je deska opatřena stabilizátorem napájení a oscilátorem. Dále je na desce vyvedeno rozhranní UART. Tímto pro počáteční vývoj kit plně vyhovuje svými vlastnostmi.



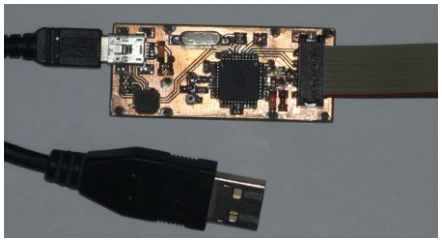
Obr. 8 Vývojový kit

4.1.1 Programátor JTAG

Rozhraní, které se řídí normou IEEE 1149.1. [9] Pomocí tohoto typu programátoru lze přes USB nahrát program do mikrokontroleru a následně jej ladit za chodu programu, postupně krokovat nebo sledovat hodnoty proměnných. Návrh DPS a vyhotovený programátor je zobrazený na obrázcích 9 a 10.



Obr. 9 DPS programátoru



Obr. 10 Funkční programátor JTAG

4.2 Převodník ADS 1298 parametry

Integrovaný obvod ADS 1298 od firmy Texas Instrument je plnohodnotným převodníkem už z výroby určeným k měření EKG a EEG. Jelikož velikost signálu jsou řádově jednotky mV, je nutné před vzorkováním signál zesílit. Převodník ADS1298 vyžaduje převod z logiky TTL na CMOS, aby bylo možné komunikovat a posílat data pomocí SPI.

Důležité parametry pro konkrétní aplikaci:

- 24 -bitový převodník
- CMRR: -115dB
- Šum µ4V
- SPI rozhraní
- Logika TTL
- Příkon 0,75mW/kanál
- 8 kanálů
- předzesilovač 1x – 12x

[10]

Převodník je řízen pomocí několika bitů:

START – zahájení posílání dat přes SPI

CS – chip select - příznakový bit, jenž určuje, že konkrétní zařízení je aktivní na sběrnici SPI

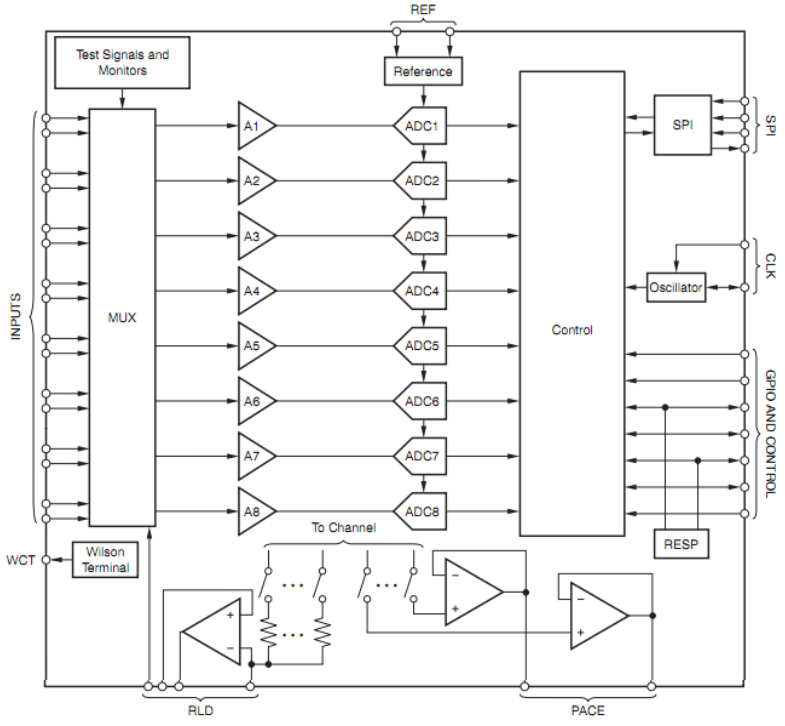
SCK – zdroj hodinového signálu

DOUT – linka určená pro posílání dat z převodníku

DIN – linka určená pro posílání dat do převodníku

DRDY- data ready – signalizuje, že převodník je připraven posílat data

RESET – system reset – převodník musí být při každém přenosu resetován



Obr. 9 Blokové schéma převodníku ADS1298

V převodníku je na každý kanál individuálně nastavitelný předzesilovač a A/D převodník. Dále převodník obsahuje zesilovač referenčních svodů RLD a Wilsonoův terminal viz. Obr. 12

Podle Shannon-Kotělnikova teorému je minimální vzorkovací frekvence určena frekvencí QRS komplexu Viz. Rovnice 6.

$$f_{QRS} = 50Hz \Rightarrow f_{vz,\min} = 2 \cdot f_{QRS} = 100Hz$$

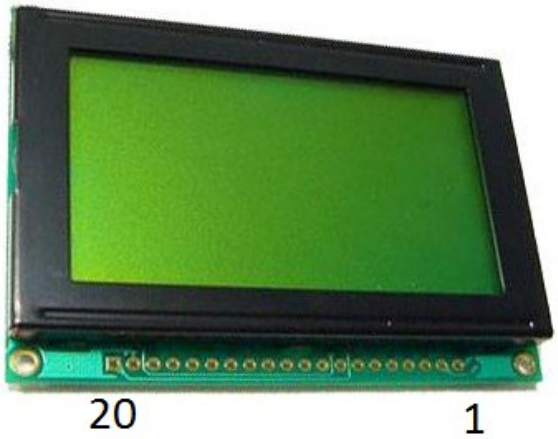
Rovnice 6

4.3 Parametry grafického displeje

Jako zobrazovací prvek je použit grafický displej s rozlišením 128x64 pixelů, skládá se ze dvou dílčích displejů o rozlišení 64x64 pixelů a zobrazení zajišťují dva řadiče KS108. Připojení na MCU je patrné z tabulky 1.

1	Vss	GND	GND
2	Vcc	5V	+5V
3	V0	Udává kontrast	
4	RS	Rozlišení instrukce/data	PB0
5	R/W	Rozlišení zápis/data	PB1
6	E	Hodinový signál	PB2
7	DB0	Datový bit 0	PA0
8	DB1	Datový bit 1	PA1
9	DB2	Datový bit 2	PA2
10	DB3	Datový bit 3	PA3
11	DB4	Datový bit 4	PA4
12	DB5	Datový bit 5	PA5
13	DB6	Datový bit 6	PA6
14	DB7	Datový bit 7	PA7
15	CS1	Zvolen řadič 1	PB3
16	CS2	Zvolen řadič 2	PB4
17	RST	Reset	
18	Vout	Výstup	
19	LED+	Anoda LED	
20	LED-	GND	

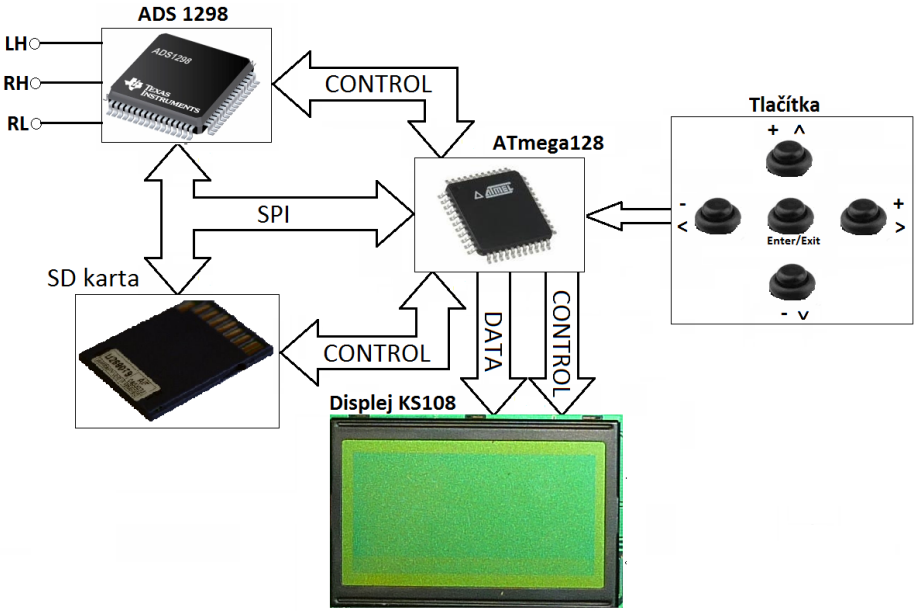
Tab. 1



Obr. 10 Displej KS108

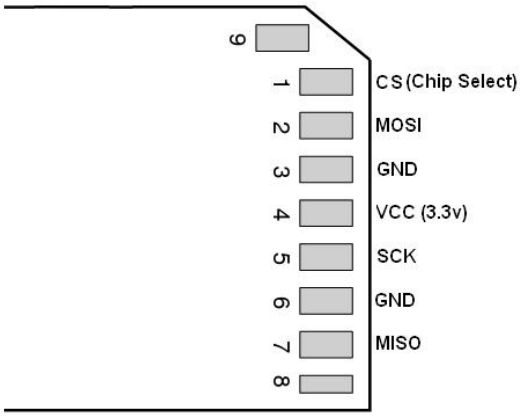
4.4 Návrh ukládání dat na SD kartu

Ukládaní dat na SD kartu by probíhalo pomocí standardního rozhraní SPI a řídícího bitu CS, kterým je určeno, zda-li posílaná data přes sběrnici SPI jsou určena právě pro SD kartu. Blokové schéma celého zařízení je znázorněno na obrázku 26.



Obr. 11 Blokové schéma Měřícího zařízení s SD kartou

Standardní SD karta je opatřena 9 piny, z nichž pouze 7 je zapojených. Z obrázku 27 je patrné rozložení jednotlivých pinů. Nutno podotknout, že SD karta musí být napájena 3,3V, takže při realizaci by do obvodu musel být zahrnut regulátor napětí a samozřejmě obvody pro převod z TTL na CMOS a naopak.



Obr. 12 Popis rozložení pinů SD karty

Kapacita karty v závislosti na čase:

- kapacita karty vždy závisí vzorkovací frekvenci a velikosti jednoho vzorku. V konkrétním případě je vzorkovací frekvence 500 Hz velikost jednoho vzorku 3 B. Pak přenosová rychlost je $12 \cdot 10^3 B \cdot s^{-1}$ a čas, za který se karta zaplní daty je rovna podílu velikosti paměti SD karty a přenosové rychlost, např. u karty, jež paměť je 256 MB, je čas zaplnění roven 5,9 hodinám, výpočty jsou uvedeny v rovnici 7.

$$f_{vz} = 500Hz$$
$$v_p = 3 \cdot 500 \cdot 8 = 12 \cdot 10^3 B \cdot s^{-1}$$
$$t = \frac{256 \cdot 10^6}{12 \cdot 10^3} = 21,3 \cdot 10^3 s = 5,9h$$

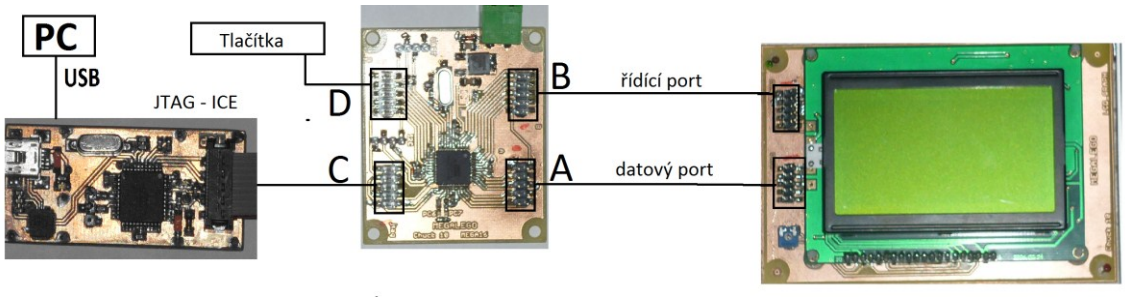
Rovnice 7

5 Realizace jednotky pro měření EKG s grafickým displejem LCD

- Realizace je rozdělena na dvě základní části, hardware a programovou část

5.1 Hardwarová část

ATmega16 má 4 porty A, B, C a D, které jsou obsazeny periferiemi, jak je patrné z obrázku 14. Displej má dva porty - datový port a řídící port. Datový port tvoří 8 bitů. Na tento port jsou posílána data, tzn. instrukce, nebo konkrétní hodnoty k zobrazení. Řídící port je tvořen 5 bity CS1, CS2, E, RS,R/W, pomocí kterých je displej řízen. Bity CS1 a CS2 určují, která část displeje je aktivní, pro který displej jsou data určena (skládá se z 2 displejů). Bit E je zdroj hodinového signálu, který je nutný k synchronizaci MCU s displejem. Bit RS určuje, zda-li jsou na datový port posílána data nebo instrukce. R/W je bit, podle kterého se řídí displej, jestli má zapisovat hodnoty, nebo vyčítat zobrazené hodnoty. Přes port C, kde je umístěn interface pro JTAG, bylo programováno MCU. Na port D jsou připojena 3 tlačítka pro ovládání jednotky. Po odladění programu bude řídící port připojen na port C a na port B, kde je interface pro SPI bude připojen převodník ADS 1298, pro přenos dat je již předepsaný kód, takže propojení převodníku a MCU lze považovat za jednoduché, ovšem na řídící port displeje pak musí být přivedený hodinový signál



Obr. 13 Připojení jednotlivých periferii

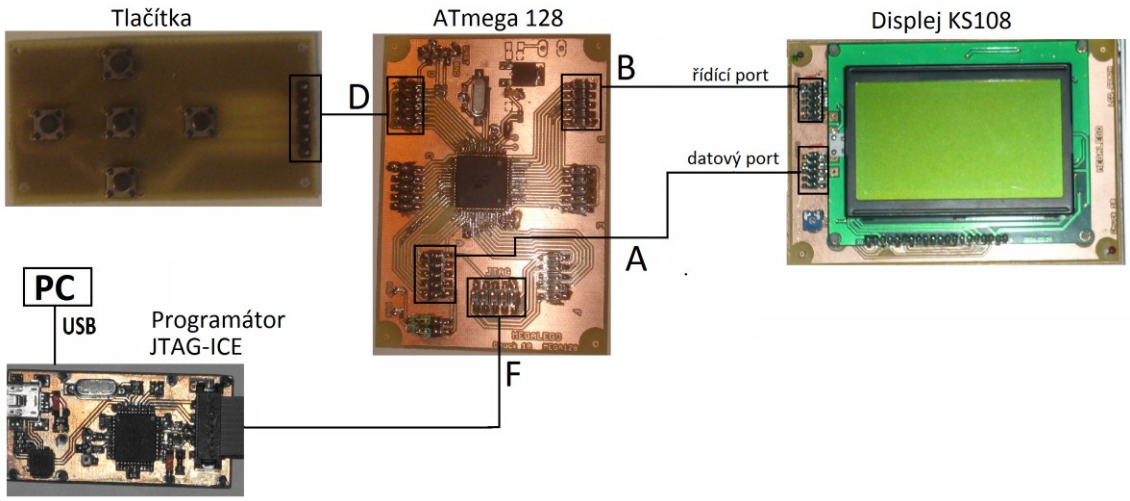
Při postupném vyvíjení softwaru se požadavky na paměť změnily a to v takové míře, že MCU ATmega16 již nebyl dostačující. Po složitém výběru byl vybrán mikrokontrolér ATmega128, který již svými parametry splňoval požadavky paměti FLASH a SRAM.

Typ mikrokontroléru:	ATmega16	ATmega128
FLASH paměť:	16kB	128kB
SRAM paměť:	1kB	4kB
EEPROM paměť:	512B	4kB
Počet portů:	4(A až D)	6(A až G)
Počet vstupů/výstupů:	32	53

Tab. 2 Srovnání mikrokontrolérů ATmega16 a ATmega128

Z tabulky 2 je zřejmý rozdíl jak ve velikostech jednotlivých pamětí, tak i v počtu portů. ATmega128 má navíc tu výhodu, že lze rozšířit paměť o 64kB. [6]

Dále se byla přidána další 2 tlačítka, tudíž pro ovládání měřícího zařízení EKG slouží 5 tlačítek. Tlačítka byla přidána z důvodu rychlejšího pohybu v nabídce menu, zvláště pak v části Zobrazení EKG. V případě původní varianty 3 tlačítek uživatel musel zvolit typ akce, tzn. Zoom, Posuv nebo Kurzor, a dále vstoupit do dalšího dílčího menu, kde by zvolil, zda se jedná o akci v ose x nebo v ose y. Takto na rozdíl od ovládání 5 -ti tlačítky uživatel zvolí pouze typ akce a následně už používá čtyři tlačítka, dvě v ose x a dvě v ose y, což v závěru zjednodušuje celý postup z minimálních 7 kliknutí na 3. Nové hardwarové vybavení přístroje je znázorněno na obrázku 15.



Obr. 14 Nová verze hardwaru

5.2 Programová část

ATmega 128 má 7 portů A až G, které jsou obsazeny periferiemi. Jako programovací prostředí bylo užito softwaru AVR Studio 4, který je spolu s knihovnami WinAVR je určeno pro programování mikrokontrolérů v jazyce C. Dále je možné programovat také v assembleru. Velikou výhodou byla možnost sledovat chod programu a nabývání jednotlivých hodnot proměnných pomocí JTAG ICE. Software prošel od původní verze až po konečnou verzi mnoha změnami, co se týče jak vizualizace, tak výpočetních algoritmů.

Při zprovoznění displeje se vyskytly značné potíže, pro ovládání displeje jsou knihovny volně ke stažení, ale jejich nevýhodou je velikost, neboť po vložení knihoven do projektu byla překročena jak paměť FLASH tak i paměť SRAM. Tudíž bylo nutné knihovny upravit tak, aby bylo vůbec možné zprovoznit displej. Knihovna font, kde je umístěná znaková sada byla ponechána původní. Knihovna glcd měla největší velikost a zároveň byla nejdůležitější, neboť jsou zde napsány funkce pro inicializaci displeje, vyčištění displeje, předpis pro bod, přímku, odkazování na font pro vkládání znaků a také mnoho další funkcí. Veškeré nepotřebné funkce byly odstraněny a dále byly definovány příslušné bity na příslušných portech. Ke knihovnám byl přiložen kód jak jednotlivé funkce používat, ale v kódu bylo mnoho chyb, takže jednotlivé byly testovány jednotlivé funkce knihoven, k čemu která slouží.

5.2.1 Hlavička programu

Software je napsaný tak, aby se dal použít i v dalších aplikacích podobného charakteru, tzn. v případě jiné aplikace s displejem řízeným řadiči KS108. Na počátku kódu je úvodní hlavička, kde se nadefinují veškeré potřebné konstanty.

Hlavička:

Zadání parametrů vstupních dat:

`#define SAMPLEF 100` - vzorkovací frekvence (ovlivnění měřítek na časové ose).

`#define GAIN 1` - zesílení převodníku (ovlivnění měřítka na y-ose).

`#define POCETVZ 200` - velikost hlavního pole vzorků, které si MCU pomocí SPI příjme od převodníku, vždy však musí být vyšší než je šířka okna grafu + 1.

Zadání parametrů displeje:

`#define WD 127` - šířka displeje (128 x 64)

`#define HD 63` - výška displeje

Zadání velikosti grafu:

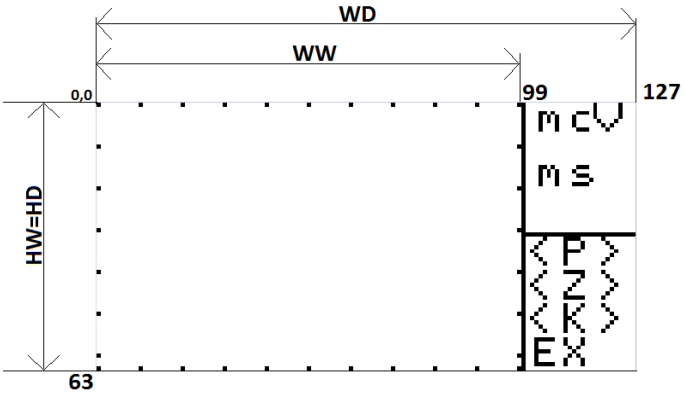
`#define WW 99` - šířka okna grafu (100 x 64)

`#define HW 63` - výška okna grafu

`#define XW 0` - počáteční souřadnice okna grafu x

`#define YW 0` - počáteční souřadnice okna grafu y

Jednotlivé hodnoty konstanty rozměrů displeje a části určené pro zobrazení křivky jsou znázorněny na obrázku 16.



Obr. 15 Rastr pro vykreslení grafu

Zadání pozice ovládacího menu

`#define PLM2 4` - pozice menu grafu řádek (umístění textu do příslušného řádku)

`#define PXM2 100` - pozice menu grafu x

Přiblížení, natavení kolikrát se zvětší signál:

`#define XZ 2` - krok přiblížení x

`#define YZ 2` - krok přiblížení y

Kurzor, nastavení o kolik pixelů se posune kurzor:

`#define XK 3` - krok kurzor x

`#define YK  3` - krok kurzor y

Posun, nastavení o kolik pixelů se posune graf:

`#define XP  5` - krok posun x

`#define YP  5` - krok posun y

5.2.2 Popis základních proměnných

Většina proměnných je deklarovaná jako globální proměnná z důvodu nutnosti přístupu v kterékoliv části programu. Hodnoty proměnných je třeba v některých situacích ponechat a jiných vynulovat, v souvislosti s tím volána funkce `nulovani()`.

Základní proměnné:

`vzorky[]` - pole, které je určené pro vzorkovaná data, hodnoty z tohoto pole nejsou přepisovány během vizualizace.

`pole[]` - pole, jehož rozměr je definován jako šířka okna grafu v pixelech.

Hodnoty pole symbolizují hodnoty v ose y a index pole symbolizuje hodnotu v ose x.

`maximum` - maximální kladná hodnota signálu v daném poli vzorků.

`minimum` - minimální (záporná) hodnota signálu v daném poli vzorků.

`zmenseni` - poměr zmenšení signálu vůči skutečné hodnotě.

`tp, tpe, tv` - proměnné určující akci ze vstupu od uživatele, tzn. tlačítek.

`c, cu, cc, ct, co, cul` - proměnné určující příkazy na první úrovni menu.

5.2.3 Software pro zobrazení průběhu EKG

Po přijmutí nadefinovaného počtu vzorku konstantou `POCETVZ` je zavolána funkce `zpracovani()`, která projde přijatá data s názvem `vzorky[]` a vyhledá `minimum` a `maximum`. Dále ze součtu maxima a absolutní hodnoty minima určí rozkmit signálu. Pomocí cyklu `while` zjistí poměr zmenšení signálu, tzn. takový poměr, kdy minimum a maximum je menší nebo roven výšce displeje `HW`. Další výstup funkce je zmenšený offset, který je v další části programu přičten, aby všechny vzorky, jejichž hodnota je záporná, byly rovny nule nebo větší.

```
void zpracovani(void)
{
          int i;
          zmenseni=1;
```

```
        for (i=0;i<=POCETVZ-1;i++)
        {
              if (maximum<vzorky[i])
              {
                    maximum=vzorky[i];
              }

              if (minimum>vzorky[i])
              {
                    minimum=vzorky[i];
              }
        }

        spicky=maximum+abs(minimum);

        while ((spicky/zmenseni)>=HW)
        {
              zmenseni=zmenseni+1;
        }

        zmensoffset=abs(minimum)/zmenseni;
}
```

5.2.3.1 Vykreslení hodnot na displej

Po základním zpracování pole vzorků je zavolána funkce `krivka()`, která v prvním cyklu `for` novému poli, jehož velikost je určena šířkou zobrazovaného okna `WW`, přiřadí hodnotu. Hodnota konkrétního prvku pole je určena podílem skutečné naměřené hodnoty a zmenšení, dále se přičte zmenšený offset. V této fázi pole prvků tvoří zobrazovaný signál, ale jelikož počátek displeje (bod [0,0]) se nachází v levém horním rohu, je spočítaná hodnota odečtena od výšky okna `HW`. Body křivky jsou propojeny přímkami podle příkazu z knihovny `glcd.h`: `line(x1,y1,x2,y2,0,1)`.

```
void krivka(void)
{
      int i;

      for (i=0;i<=WW;i++)
      {
            pole[i]=HW-(vzorky[i]/zmenseni+zmensoffset);
      }
      for (i=0;i<=WW-1;i++)
      {
            line(i,pole[i],i+1,pole[i+1,0,1);
      }
}
```

5.2.3.2 Funkce zoom

Funkce `zoom()` zajišťuje přiblížení a oddálení signálu jak v ose x tak i v ose y. V počátku funkce je ošetření proti násobení a dělení nulou. Prvky pole pro zobrazení na displej se počítají z naměřených vzorků, tzn. je zachována největší možná přesnost. Prvkům, jejichž hodnota je vyšší než rozlišení okna grafu je přiřazena nejvyšší hodnota rozlišení okna `HW` a prvkům, jejichž hodnota je záporná je přiřazena hodnota 0. Přiblížení v ose x je realizováno násobením x–ových souřadnic ve funkci `line(x1,y1,x2,y2,0,1)`, kde je opět ošetřeno podmínkou, která zajišťuje, aby hodnota `x2`, na kterou se program odkazuje, nebyla vyšší než rozměr `WW`.

```
void zoom()
{
            int i;
            if (hz==0)
                  hz=1;

            if (vz==0)
                  vz=1;

            zmenseni=zmenseni/hz;
            cyklus krivka() s omezením:
                              if (pole[i]>HW)
                                    pole[i]=HW;

                              if (pole[i]<0)
                                    pole[i]=0;

            for (i=0;i<=WW-1;i++)
            {
                  if(vz*i<=WW-1)
                  {
                        line(vz*i,pole[i],vz*(i+1),pole[i+1],0,1);
                  }
            }
}
```

5.2.3.3 Funkce posun

Posun signálu je realizován pomocí funkce posuv. V ose y je posun zajištěn přičtením proměnné `posuvy` k hodnotě každému prvku v poli, které je určené k zobrazení. Krok posunutí v pixelech určuje konstanta `YP`. V ose x se funkce odkazuje na indexy v poli vzorky určené proměnnou `posuvx` a krok posunutí je definován konstantou `XP`.

Na obrázku 17 jsou znázorněny 3 stavy, které mohou nastat při zobrazování :

1) `(posuvx+i>=0&&posuvx+i+1<POCETVZ-1)`
   První prvek křivky i poslední prvek se nachází v poli vzorků.

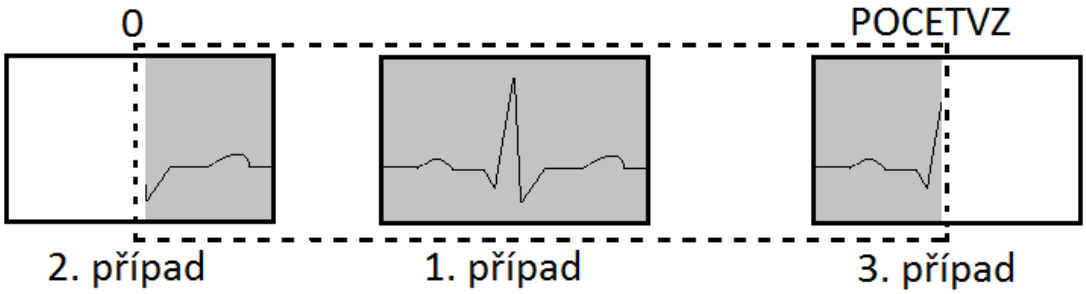
Pak odkázaní na konkrétní prvek je:
`pole[i+posuvx]`

2) `(posuvx+i<0)`
   Část křivky se už nenachází v poli vzorků, jelikož index je záporný.

Je třeba odkázat na konec pole vzorky:
`pole[i+posuvx+POCETVZ]`

3) `(posuvx+i+1>POCETVZ-1)`
   Index je větší než konstanta `POCETVZ`, tudíž větší než index pole.

Je třeba odkázat na počátek pole vzorky:
`pole[i+posuvx-POCETVZ]`



Obr. 16 Příklady funkce posuv

5.2.3.4 Závěrečná verze funkce pro vykreslení

Aby bylo možné signál přiblížit, oddálit nebo posunout, bylo nutné všechny výše uvedené funkce implementovat do jedné funkce, neboť v jiném případě by bylo třeba více pomocných polí, jelikož posloupnost jednotlivých funkcí a počet jejich vykonání by musel být zaznamenáván a poté opakován. Další nevýhodou by byl vyšší počet cyklů, jelikož musel být v každé funkci, tzn. `zoom`, `posun` a `krivka`. V každém cyklu by dílčí funkce prošly celé pole určené k zobrazení a upravily ho, což by bylo mnohem náročnější, jelikož tento postup by vyžadoval o mnoho více taktů mikrokontroléru. Další varianta by byla zpracovat referenční pole `vzorky[]` funkcí `zpracovani()` a dále pracovat pouze s takto upraveným polem, ale tímto by byla zobrazovaná data značně zkreslená, například ve funkci `zoom` by se zvětšení rozlišení realizovalo pomocí násobení prvků v poli k zobrazení, tímto by se jednalo pouze o zvětšení na úrovni vizualizace, tzn. násobení y-ových hodnot pixelů (Rovnice 7), ale nikoliv na úrovni datové tzn. skutečné naměřené hodnoty (Rovnice 8). Rozdíl je zřejmý na výpočtech, již při čtyřnásobném přiblížení je v prvním případě rozdíl hodnot 20 procent skutečné hodnoty.

$$vzorky[i] = 45mV$$
$$zmenseni = 20$$
$$pole[i] = vzorky[i] / zmenseni$$
$$pole[i] = 2$$

$$zoomY = 4$$
$$pole[i] = 8$$
$$zobrazená_hodnota = 40mV$$

Rovnice 8

$$vzorky[i] = 45mV$$
$$zoomY = 4$$
$$zmenseni = 20$$
$$zmenseni = zmenseni / 4 = 5$$
$$pole[i] = vzorky[i] / zmenseni$$
$$pole[i] = 9$$

$$zobrazená_hodnota = 45mV$$

Rovnice 9

V konečné fázi probíhá zpracování funkcí `zpracovani()` a dále veškeré další výpočty a vykreslení na displej provádí funkce `krivka()`. Funkce uvedené výše jsou implementovány do funkce `krivka()`.

5.2.3.5 Funkce graf

S funkcí `krivka()` úzce souvisí funkce graf. Tato funkce zajišťuje zobrazení rastru (Obr. 18) pro graf a je volána vždy po vykreslení křivky, resp. přímo ve funkci `krivka()`. Dále zde probíhá výpočet měřítek os, z kterých lze určit, v jakém čase má signál určitou hodnotu.

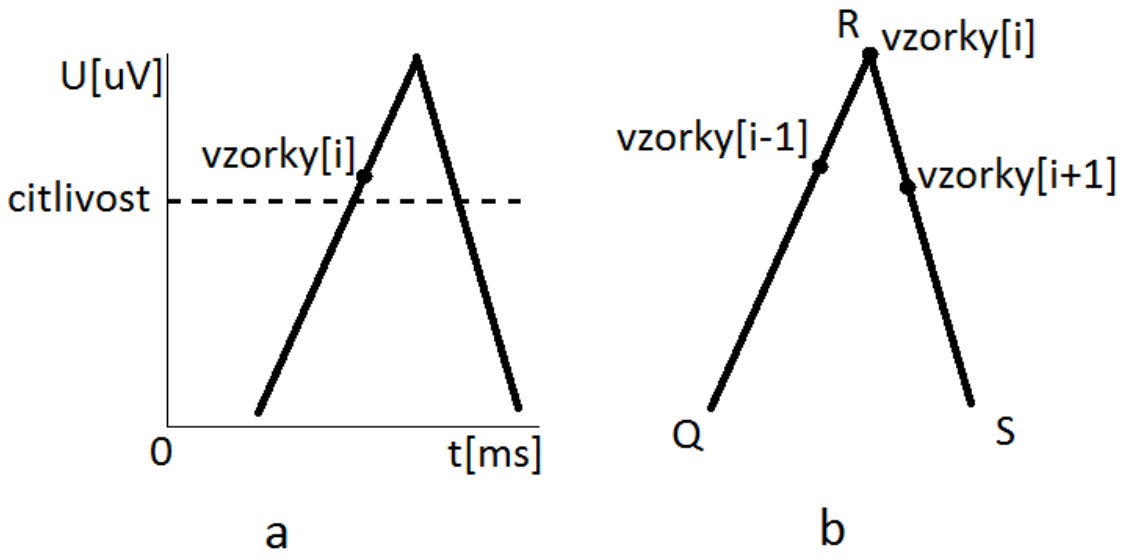
Obr. 17 Rastr připravený pro zobrazení grafu

5.2.3.6 Funkce kurzor

Funkce zobrazuje kurzor-kříž a v závislosti na aktuální poloze kurzoru pomocí algoritmů zobrazuje na displej do pravého horního rohu napětí a časový úsek od počátku vzorkování. V ose x se podílí na výsledku proměnná `kx` a krok kurzoru je definován konstantou `XK`, která určuje o kolik pixelů se posune kurzor při stisku tlačítka. Podobně tak tomu je i v ose y s proměnnou `ky` a konstantou `YK`. Tyto hodnoty jsou násobeny poměrem skutečné veličiny a zmenšení. V závislosti na směru pohybu kurzoru jsou hodnoty buďto odečítány nebo přičítány. Funkce kurzor může být volána jen tehdy, když není volána funkce `graf()`.

5.2.4 Funkce tep

Algoritmus v této funkci projde celé pole vzorky a každý prvek porovná s proměnnou `citlivost`. Tep je detekován pomocí QRS komplexu, což je část křivky, která má největší amplitudu. V případě, že hodnota prvku `vzorky[i]` je vyšší než `citlivost`, jsou dále porovnány s prvky před `vzorky[i-1]` a za `vzorky[i+1]` prvkem, jenž byl vyšší než proměnná `citlivost` a pokud je splněna podmínka `vzorky[i-1] < vzorky[i]` a `vzorky[i] > vzorky[i+1]` jedná se o vrchol R z QRS komplexu (Obr. 19). Je spuštěna inkrementace v cyklu, jenž prochází pole `vzorky[]`. Inkrementace probíhá tak dlouho než je detekován další puls z QRS komplexu. Z počtu prvků proběhlých od prvního pulzu po druhý a vzorkovací frekvence je spočítán tep. Tento údaj je dále porovnán s nejnižším, nejvyšším tepem, na konci algoritmu je připočítán k váženému průměru.



Obr. 18 a) Detekce hodnoty > citlivost b) Detekce tepu

5.2.5 Displej – inicializace, funkce

MCU zobrazuje elementární prvky jako např. bod, přímku nebo text pomocí dvou knihoven, glcd a font. Knihovna font je pouze zapsaná ASCII tabulka ve dvojrozměrném poli, knihovna má funkci předpisu, jak které písmeno zobrazit. Prostřednictvím knihovny glcd je provedena inicializace displeje, je však nutné nadefinovat přímo v knihovně ovládací port a port, přes který budou posílána data a určit správně bity `CS1, CS2, DI, RW` a `EN` v závislosti na zapojení. Dále je třeba v hlavní funkci `main` povolit příslušný počet bitů na příslušných portech.

```
#define DATAPORT PORTA
#define CONTROLPORT  PORTB
#define DATADDR DDRA
#define DATAPIN PINA
#define CS1 (1<<3)
#define CS2 (1<<4)
#define DI  (1<<0)
#define RW  (1<<1)
#define EN  (1<<2)
DATADDR = 0xff;
DDRB = 0x3F;
```

Základní funkce, pomocí kterých je uskutečňována vizualizace, jsou bod, přímka, čtverec, text, zapnutí displeje a vymazání displeje. Souřadnice [0,0] se na displeji nachází v levém horním rohu.

Bod:

```
void point_at(unsigned int x,unsigned int y,byte color)
```

`x` – x-ová souřadnice bodu (0 až 127)
`y` – y-ová souřadnice bodu (0 až 63)
`color` – barva bodu (0 nebo 1)

Přímka:

```
void line(unsigned int x1,unsigned int y1,
          unsigned int x2,unsigned int y2,
          byte s,byte c)
```

`x1,y1` – souřadnice prvního bodu (0 až 63, 0 až 127)
`x2,y2` – souřadnice druhého bodu (0 až 63, 0 až 127)
`s` – vynechávání bodů, tzn. pomocí tohoto parametru lze udělat přerušovanou čáru (0 nebo 1)
`c` – barva (0 nebo 1)

Vkládání řetězce znaků:

```
void glcd_puts(char *c,int x,int y,unsigned char l)
```

`*c` – řetězec, který má být zobrazen
`x` – x-ová souřadnice (0 až 127 – 8 x počet vkládaných znaků)
`y` – y–ová souřadnice, avšak lze vkládat pouze na řádky (0 až 7)
`l` – určení jazykové sady, v případě latinky l = 0, v případě arabštiny l=0.

Čtverec:

```
void rectangle(unsigned int x1,unsigned int y1,
               unsigned int x2,unsigned int y2,
               byte s,byte c)
```

`x1,y1` – souřadnice prvního bodu (0 až 63, 0 až 127)
`x2,y2` – souřadnice druhého bodu (0 až 63, 0 až 127)
`s` – vynechávání bodů, tzn. pomocí tohoto parametru lze udělat přerušovanou čáru (0 nebo 1)
`c` – barva (0 nebo 1)

Zapnutí displeje:

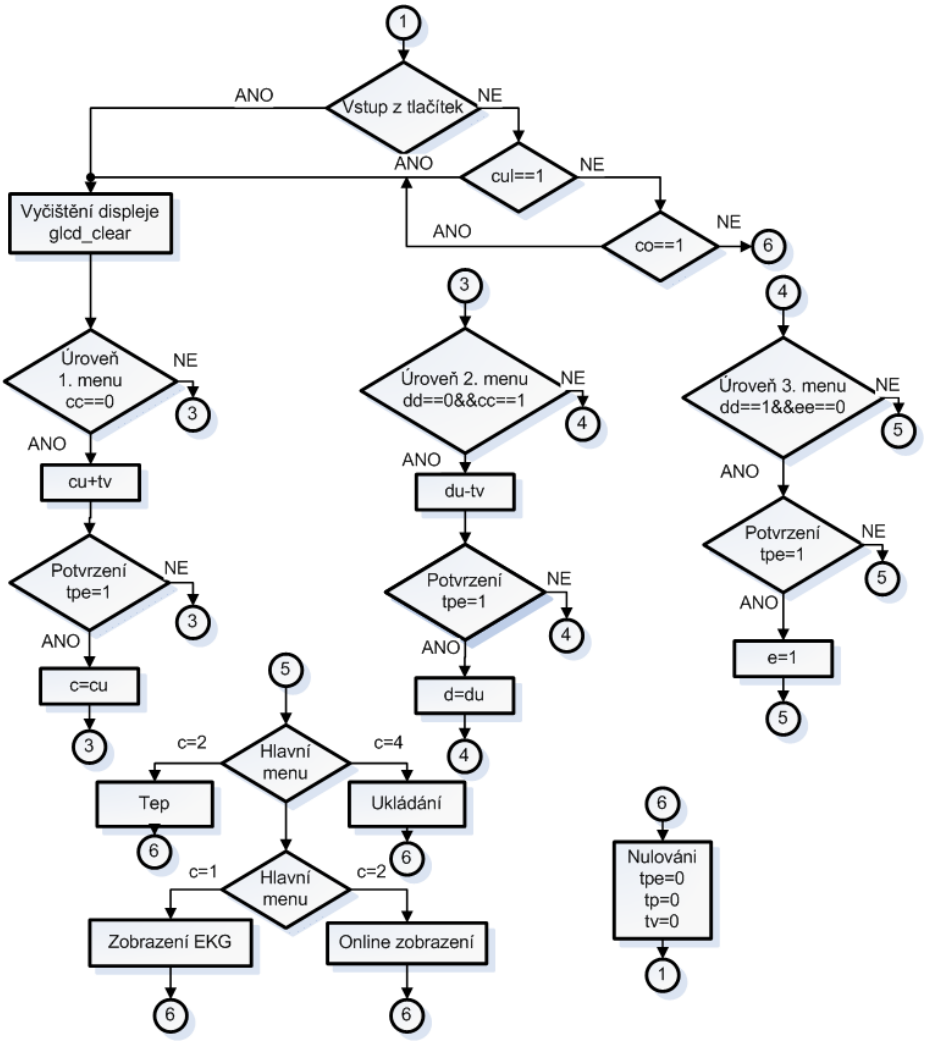
```
void glcd_on()
```

Vyčištění displeje:

```
Void glcd_clear()
```

5.2.6 Hlavní smyčka

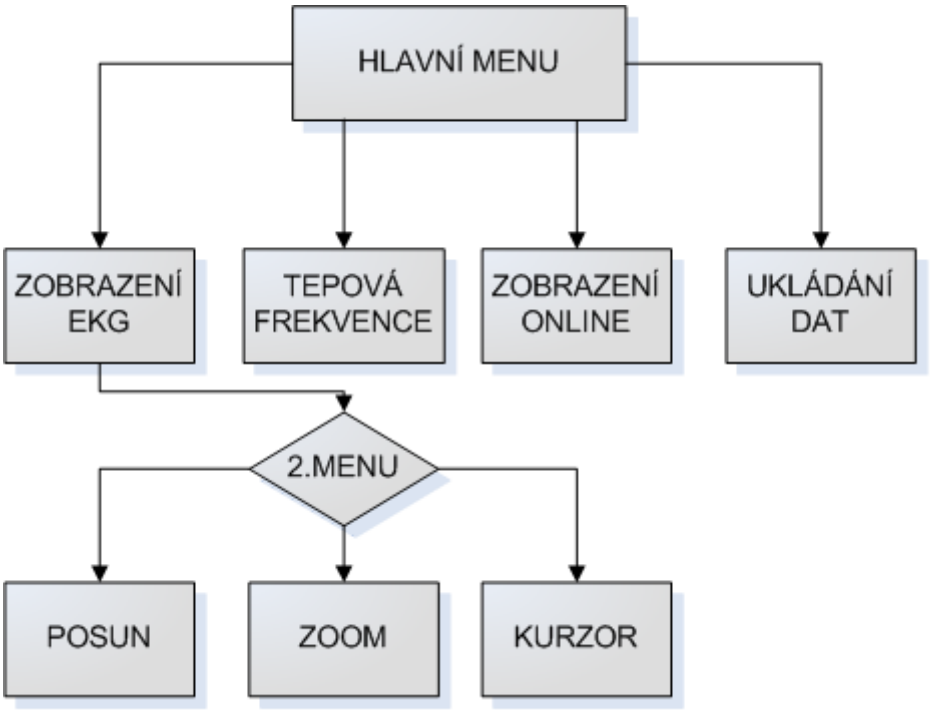
Na začátku hlavní smyčky jsou tři podmínky, vstup z tlačítek, `cul` a `co`, v případě splnění jedné mikrokontrolér vyčistí obrazovku displeje. Proměnné `cul` a `co` slouží k používání periodicky se opakujících funkcí, tzn. Ukládání dat a Online zobrazení. Po vyčištění pomocí proměnných `cc`, `dd` a `ee` zjistí, která úroveň menu je právě aktivní a popřípadě, že bylo stisknuto tlačítko pro posun v menu, přičte ukazovací proměnné (`cu` nebo `du`) proměnnou `tv`. V případě, že bylo stisknuto potvrzovací tlačítko, tzn. hodnota `tpe=1`, příkazová proměnná (`c` nebo `d`) převezme hodnotu ukazovací proměnné a v příslušném dílčím algoritmu (Tep, Ukládání, Zobrazení EKG, Online zobrazení) je hodnota proměnné cc přepsána přepsaná z 0 na 1, čímž je v následujícím cyklu otevřené další menu a menu předcházející je zablokováno. Na konci každého cyklu jsou proměnné signalizující vstup z tlačítek (`tp, tv` a `tpe`) nulovány. Celý tento algoritmus je znázorněn na obrázku 20.



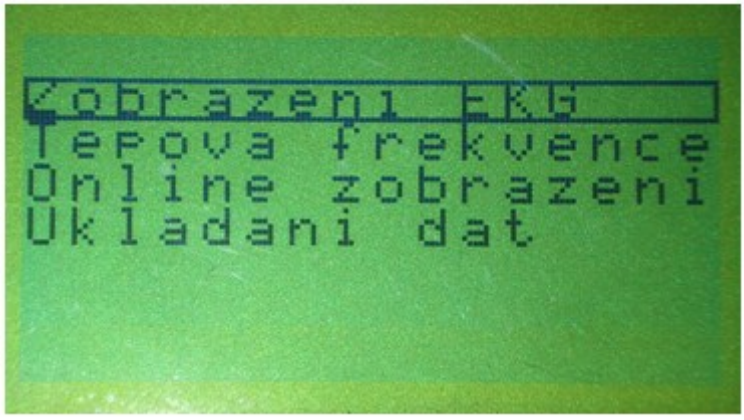
Obr. 19 Vývojový graf hlavní smyčky

5.3 Uživatelské menu

Při tvorbě softwaru muselo být od počátku jasné, jaké jsou požadavky na zařízení a tím i software a co lze spočíst a vyvodit ze získaných vzorků a také co by měl mít uživatel k dispozici v nabídce menu, aby bylo zařízení účelné (Obr. 21). Vzhledem k těmto parametrům je vytvořeno hlavní menu tak, aby se v něm uživatel dobře orientoval a aby bylo jasné a srozumitelné. V hlavní menu má uživatel na výběr 4 možnosti, Zobrazení EKG, Tepová frekvence, Zobrazení online, ukládání dat Viz. Obr 22.



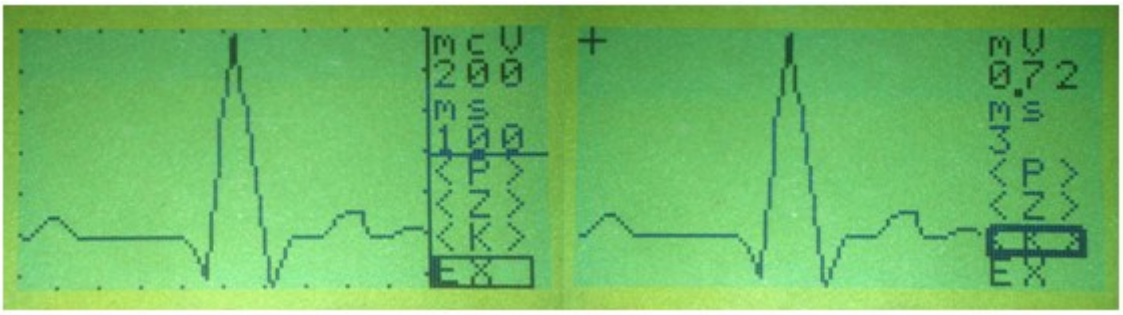
Obr. 20 Schéma menu



Obr. 21 Hlavní menu

Zobrazení EKG:

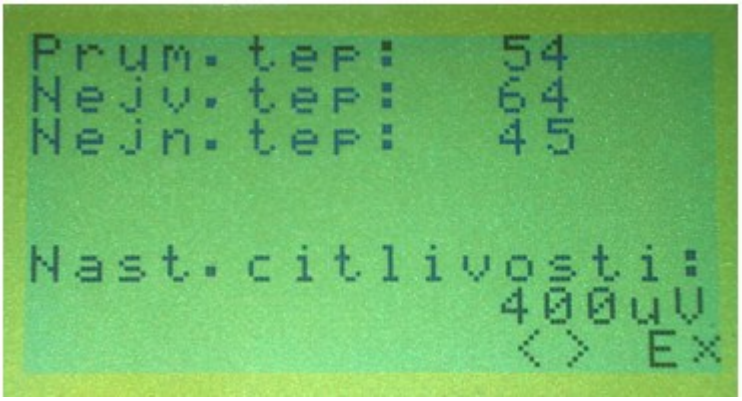
- zde lze detailně sledovat signál EKG, kromě toho je možné průběh posunout nebo přiblížit podle potřeby jak v ose x tak v ose y (Obr. 23a). Dále je zde funkce kurzor, kde lze vyčíst konkrétní parametr signálu v čase (Obr. 23b). Zobrazený signál je vzorkován po potvrzení vstupu uživatelem do tohoto bloku, po opuštění a novém vstupu se signál bude vzorkovat znovu. Délka vzorku je dána nastavením počtu vzorků v programu.



Obr. 22 a) Zobrazení EKG b) Zobrazení kurzor

Tepová frekvence:

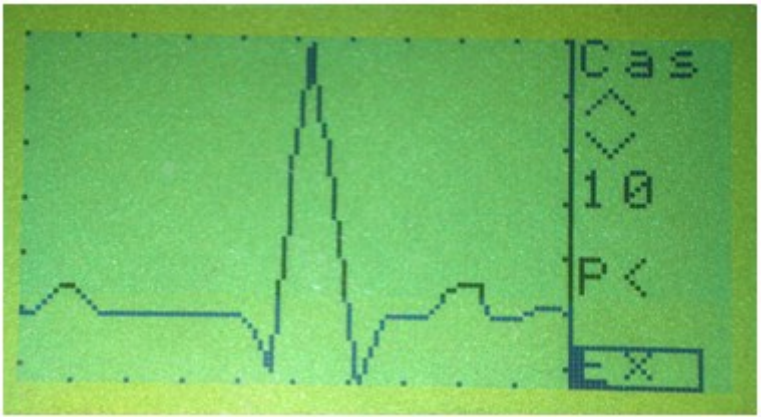
- tento blok slouží k vyčíslení důležitých hodnot, jako jsou nejvyšší tep, nejnižší tep, průměrný tep a citlivost, jak je znázorněno na obrázku 24. Citlivost je hranice, odkud začíná zařízení zaznamenávat tep a lze ji nastavit přímo v menu. Průměrný tep se počítá z váženého průměru.



Obr. 23 Zobrazení tep

Zobrazení online:

- signál je vzorkován, upraven a zobrazen na displej po dobu jedné sekundy. Uživatel zde může nastavit stabilní zpoždění a tím zajistit určitou synchronizaci a pomocí tlačítka vlevo signál jednorázově zpozdit, takže v případě, že se první synchronizuje zobrazení signálu, pak tlačítkem vlevo lze signál posunout Viz. Obr. 25



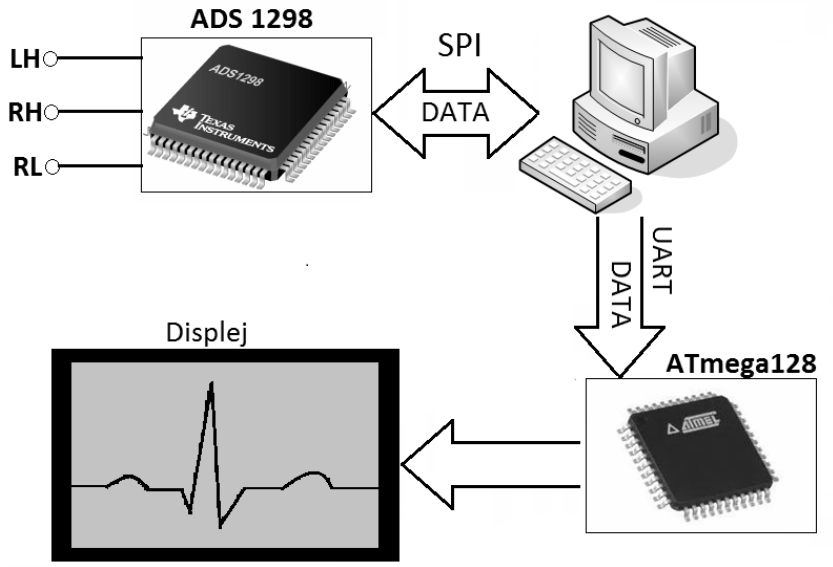
Obr. 24 Online Zobrazení

Ukládání dat:

- tato část menu je vyhrazena do budoucna pro ukládání dat na SD kartu, kdy mikrokontrolér bude pouze načítat data z převodníku do paměti SRAM a po načtení budou naměřené veličiny ukládány pomocí sběrnice SPI na SD kartu.

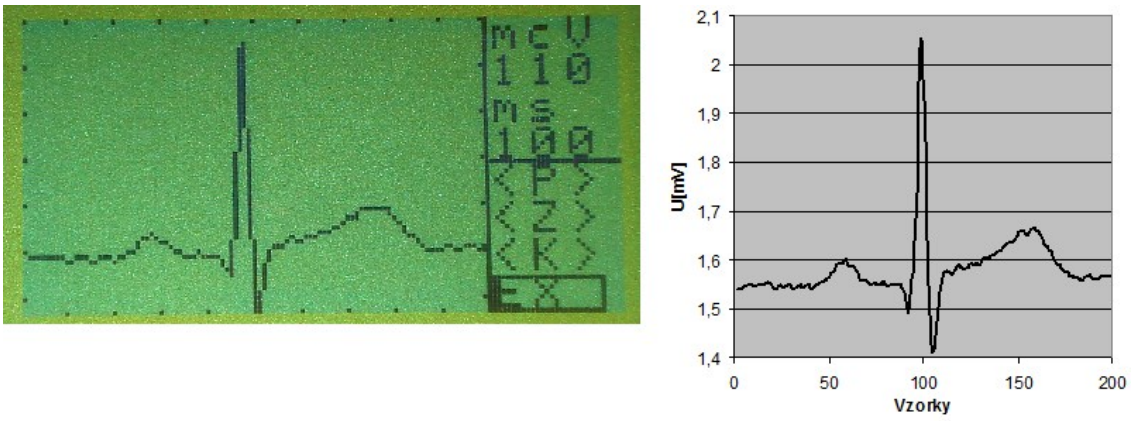
6 Test měřícího řetězce

- přes převodník ADS1298 byla vzorkována data do PC. Pomocí rozhraní UART byla tyto data nahrána do mikrokontroléru a poté už byla zpracována výše uvedenými algoritmy, jak je zřejmé z obrázku 28.



Obr. 25 Blokové schéma Testování

Do MCU byla nahrána data po dobu měření dvou sekund, převodník vzorkoval data s frekvencí 250Hz a na obrázku 30 je možno vidět průběh EKG jak na displeji, tak i v počítači zobrazený pomocí Microsoft Office Excel. Tímto bylo v testování ověřeno, že zpracování a zobrazení naměřených dat proběhlo úspěšně.



Obr. 29 Zobrazení průběhu EKG

Inicializace rozhraní UART:

- Inicializace probíhá pomocí 3 registrů UCSRA (Tab. 3), UCSRB (Tab. 4) a UCSRC (Tab. 5).

RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM
0	0	0	0	0	0	1	0

Tab. 3

RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8
1	0	0	1	0	0	0	0

Tab. 4

URSEL	UMSEL	UPM1	UMP0	USBS	UCSZ1	UCSZ0	UCPOL
1	0	0	0	0	1	1	0

Tab. 5

O stránku níže je uvedena část kódu, kde se realizuje přenos dat pomocí rozhraní UART a jejich následné zpracování a vykreslení na displej, v přerušení jsou do buffer načítána data. V momentě, kdy proměnná `buffCt` je rovna 250, jsou data překopírována do pole `vzorky[]`, následně se buffer načte znovu. Dále pomocí funkce `USARThandle()` jsou data upravena a zobrazena na displej, tato funkce je volána v hlavní smyčce, kde je uvedena dále inicializace periferie -UART a definování přenosové rychlosti baudrate.

```
volatile uint8_t buffer[250];
volatile uint8_t buffCt = 0;
volatile uint8_t baud_set = 51;

SIGNAL (SIG_UART_RECV)
{
      int i;

      buffer[ buffCt ] = UDR;
      uffCt++;

      if (uffCt==250)
      {
            for (i=0;i<=249;i++)
            {
                  buffer[i]=vzorky[i];
            }
            buffCt=0;
      }
}

void USARThandle( void )
{
      zpracovani();
      krivka();
      nulovani();
}

int main (void)
{
      ...
      // Inicializace UART
      UBRRL=(uchar)baud_set;
      UCSRA=0b00000000;
      UCSRB=0b10010000;
      UCSRC=0b10000110;
      sei();
      ...
      while(1);
      {
            ...
            //zavolání funkce pro vykreslení dat
            USARThandle();
            ...
      }
}
```

7 Zhodnocení dosažených výsledků

Měření EKG signálu je v dnešní době jedna z nejčastěji používaných metod, nicméně je stále málo používaná v souvislosti s dlouhodobým měřením pacientů doma. Tato problematika je úzce spjata s cenou zařízení, které by bylo schopno měřit elektrickou aktivitu srdce a zaznamenávat její průběh na datové médium. Vzhledem k tomuto problému jsem se pokusil realizovat takovéto měřící zařízení, s lepší cenovou relací vstupních nákladů tak, aby bylo účelné jak pro odborného lékaře v měření EKG a stanovení diagnózy pacienta, tak v domácím prostředí pacientů ve spojitosti s dlouhodobým sledováním činnosti srdce. Měřící zařízení spolu se softwarem je konstruováno tak, aby v budoucnu bylo možné na bázi této bakalářské práce pokračovat ve skutečné realizaci a případného uvedení do praxe.

Jádrem měřícího zařízení je mikrokontrolér ATmega128 vystavěný na 8-mi bitové RISC architektuře. Jako měřící člen zde slouží převodník ADS1298 a získaná data jsou zobrazována na grafický displej. Po spuštění lze zařízení ovládat pomocí tlačítek. Program je vytvořen v programovacím jazyku C a jeho struktura je popsána v bakalářské práci.

Pro aplikovatelnost zařízení v praxi jsem vytvořil čtyři základní funkce, z nichž tři byly testovány, je to vizualizace průběhu EKG a jeho detailní analýza, dále zobrazení a výpočet tepu a zobrazení aktuálního průběhu EKG. Ukládání dat na SD kartu bylo dle zadání navrženo.

Průběh práce byl intenzivně konzultován s vedoucím bakalářské práce. Stanovené cíle práce se podařilo úspěšně splnit.

Vzhledem k tomu, že MCU musí zpracovávat 24 -bitová data z převodníku s vzorkovací frekvenci 250 Hz a vyšší, je celý proces měření, zpracování a zobrazení výpočetně velmi náročný a mikrokontrolér je značně vytížen. Z tohoto důvodu bych do budoucna doporučil nahradit mikrokontrolér jiným nejlépe 32-ti bitovým typem nebo signálovým procesorem.

8 Použitá literatura

[1] PENHAKER, Marek. *Elektrokardiografie* [online]. Ostrava: 2006. 22 s. Skripta. VŠB. URL: <http://fei1.vsb.cz>.

[2] HESSELSON, A. B. *Simplified Interpretation of Pacemaker ECGs*: An Introduction.[/i] Hoboken (USA): Wiley-Blackwell, 2007. 304 s.

[3] M. POKORNÝ, P. NEVŘIVA, J. ŠVÍGLER. *Elektroktrotechnická měření a přístroje* [online]. Ostrava: 2005. 142 s. Skripta. VŠB. URL: <http://fei1.vsb.cz>.

[4] PUNČOCHÁŘ, Josef. *Operační zesilovačev elektronice*. Praha : BEN, 1999. 496 s.

[5] MATOUŠEK, David. *Práce s mikrokontroléry ATMEL AVR : ATmega16*. Praha: BEN, 2006. 320 s.

[6] HÁJEK, Karel ; KREJČÍŘEK, Jaromír. *Jednočipové mikropočítače ATMEL : ATmega128 / 128L*. Brno: Univerzita obrany, 2006. 92 s.

[7] *Wikipedie* [online]. 2010-04-24 [cit. 2011-10-08]. Elektrokardiogram. URL: <http://cs.wikipedia.org>.

[8] *PŘEHLED SVODŮ UŽÍVANÝCH V EKG* [online]. 2010-01-13 [cit. 2011-02-05]. Svody. URL: <http://ekg.kvalitne.cz>.

[9] *Atmel* [online]. 2010 [cit. 2010-10-15]. ATmega16 overview. URL: <http://www.atmel.com>.

[10]*Texas Instruments* [online]. 2010 [cit. 2010-12-04]. ADS1298 . URL: <http://focus.ti.com>.

9 Seznam příloh

Příloha I - Popis algoritmu a vývojový diagram Zobrazení EKG..... I
Příloha II - Měřící zařízení II