

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 10 – Elektrotechnika, elektronika a telekomunikace

Mobilní telefon s dotykovým displejem

The Mobile Phone with Touchscreen

Autor: Adam Heinrich

Škola: Mendelovo gymnázium, Opava

Opava, 2011

Mobilní telefon s dotykovým displejem

Děkuji Ing. Petru Koščákovi za velice precizní výměnu na poslední chvíli zničeného konektoru GSM modulu, bez které bych nemohl práci dokončit, Petru Zahradníkovi za poskytnutí Eagle knihovny pro konektor GSM modulu a především svým rodičům za trpělivost a podporu.

Prohlašuji, že jsem svou práci vypracoval samostatně, použité podklady jsem uvedl v Seznamu zdrojů a postup při zpracování a dalším nakládání s prací je v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V Opavě, dne 28. 3. 2011

Adam Heinrich

Mobilní telefon s dotykovým displejem

Anotace

Práce popisuje návrh, konstrukci a software mobilního telefonu. Cílem je vytvořit zařízení s grafickým uživatelským rozhraním a dotykovým displejem a v praxi si tak vyzkoušet a rozšířit znalosti z oblasti elektroniky, mikroprocesorové techniky a programování.

Klíčová slova: Dotykový displej, Mobilní telefon, Mikrořadič

Abstract

This document describes design, hardware and software of a mobile phone. The main aim was to create a mobile device with graphical user interface and touchscreen and to use and improve skills in electronics and programming.

Keywords: Touch Screen, Mobile Phone, Microcontroller

Obsah

I. Úvod.....	5
II. Elektronika.....	6
II.1 GSM modul.....	6
II.2 Modul LCD displeje.....	8
II.2.1 LCD displej.....	8
II.2.2 Řadič dotykové folie.....	9
II.2.3 Paměťová karta.....	11
II.3 Mikrokontrolér Atmega128A.....	11
II.4 Napájení.....	13
II.5 Ověření na nepáživém poli.....	15
II.6 Deska plošných spojů.....	16
II.7 Fotografie desek plošných spojů.....	17
III. Software.....	18
III.1 Kreslení BMP obrázků.....	18
III.2 Kreslení obrázků z paměti Flash.....	20
III.3 Zaslání AT příkazů.....	22
III.4 Zpracování odpovědi GSM modulu.....	23
IV. Uživatelské rozhraní.....	25
V. Závěr.....	26
VI. Fotografie hotového telefonu.....	27
VII. Významy některých použitých zkratk.....	28
VIII. Seznam použitých zdrojů.....	29
VIII.1 Články.....	29
VIII.2 Softwarové projekty.....	29
VIII.3 Dokumentace.....	29
VIII.4 Datasheety.....	29

Mobilní telefon s dotykovým displejem

I. Úvod

Spotřební elektronika prošla v posledních desetiletích prudkým vývojem. Díky malým diskretním součástkám a stále výkonnějším a komplexnějším obvodům integrovaným na malých čípech se můžeme s novinkami na trhu setkávat téměř každý týden.

Z velkých, těžkých a drahých „kufříkových“ mobilních telefonů se vyvinuly přístroje kapesních rozměrů, které postupem času získaly nové funkce, jako jsou textové zprávy, hry, prohlížení fotografií nebo natáčení videí. Vývojem prošlo také uživatelské rozhraní – původní monochromatické znakové displeje byly nahrazeny plně grafickými, které umožňovaly vykreslovat jednoduchou grafiku, důležitou například u her. Jednoznačným trendem dnešních mobilních telefonů jsou velké barevné displeje s rozlišením větším než 320×240 pixelů a dotykovou vrstvou, díky které odpadá nutnost používání klasických tlačítek a která umožňuje variabilní ovládání telefonu.

Abych se aspoň trochu moderním zařízením přiblížil, rozhodl jsem se z dostupných součástek sestavit jednoduchý mobilní telefon s grafickým uživatelským rozhraním, které je ovládáno dotykovou fólií – prstem nebo plastovým dotykovým perem (stylus). Hlavní motivací bylo využít své stávající znalosti práce s GSM moduly, naučit se pracovat s grafickým displejem a dotykovou fólií a vytvořit nenáročné grafické rozhraní, které může běžet na levném osmibitovém mikrokontroléru s omezenými prostředky, především malou pamětí.

II. Elektronika

II.1 GSM modul

GSM modul umožňuje připojení do mobilní sítě, zajišťuje hlasové hovory, odesílání a příjem SMS či práci se záznamy uloženými na SIM kartě, jako jsou přijaté zprávy SMS nebo seznam kontaktů.

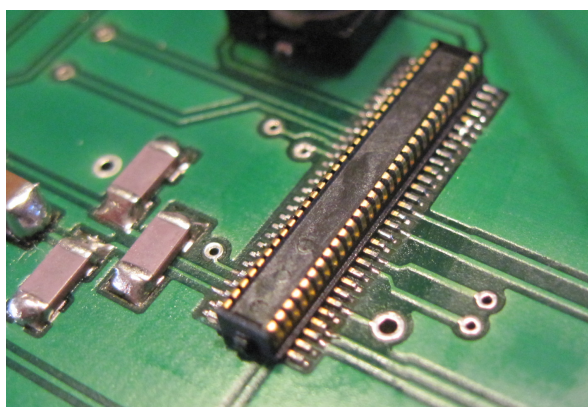
Pro svou práci jsem vybral modul **SIM100S** [SIM]¹ čínské firmy *SIMCom* – především proto, že se k desce plošných spojů připojuje konektorem. Jeho rozměry jsou 53 × 33 × 3 mm a komunikace probíhá po sériové lince rychlostí 115200 baudů pomocí AT příkazů ve formě ASCII řetězců [ATC], které zpřístupňují většinu funkcí modulu (více v části *Software*)



Obr. 1: Modul SIM100S

Kromě základní funkčnosti nabízí modul i další funkce, které jsem v mobilním telefonu využil:

- **RTC (obvod reálného času)** – k tomuto účelu je připojena 3 V zálohovací baterie **CR2032**
- **Zapnutí a vypnutí modulu tlačítkem** – funkci využívám k zapnutí celého modulu, více v části Napájení
- **Měření napětí baterie** – moduly *SIMCom* počítají s bateriovým napájením, některé z nich umožňují i správu baterie a nabíjení. SIM100S umí měřit napětí v řádech milivoltů



Obr. 2: Konektor SJ031539

1 [SIM] – odkaz na položku v *Seznamu zdrojů* uvedeném na konci práce.

Mobilní telefon s dotykovým displejem

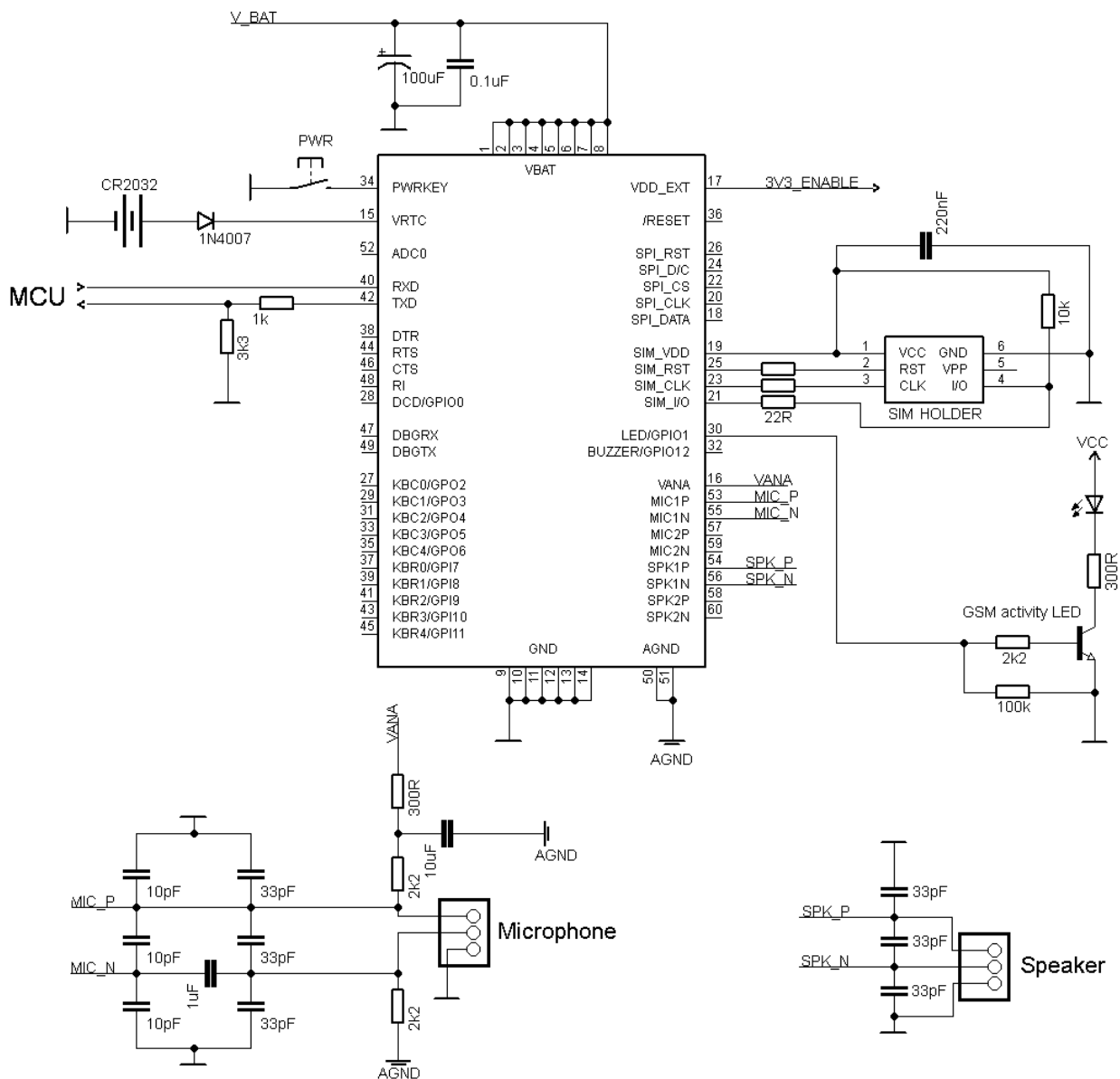


Schéma 1: Zapojení GSM modulu SIM100S

Zapojení modulu vychází z poznámek uvedených v dokumentaci, přímo k modulu jsou připojeny mikrofon, reproduktor, zálohovací baterie a zapínací tlačítko. Anténa je připojena k anténnímu konektoru na modulu, ve schématu není zakreslena. Na lince **Tx**, která slouží k přenosu dat po sériové lince z modulu směrem do mikrokontroléru, je umístěn ochranný odporový dělič. Výstup **3V3_ENABLE** je použit pro sepnutí stabilizátoru **LE33** (více v části *Napájení*).

Pro připojení k desce plošných spojů (DPS) je použit SMD konektor **SJ031539** s roztečí pinů 0,5 mm. Protože jsou na DPS umístěny prokovy a „konektorová“ část modulu je opatřena plechovým krytem, je modul připevněn oboustrannou lepicí páskou, která slouží jako izolace.

Mobilní telefon s dotykovým displejem

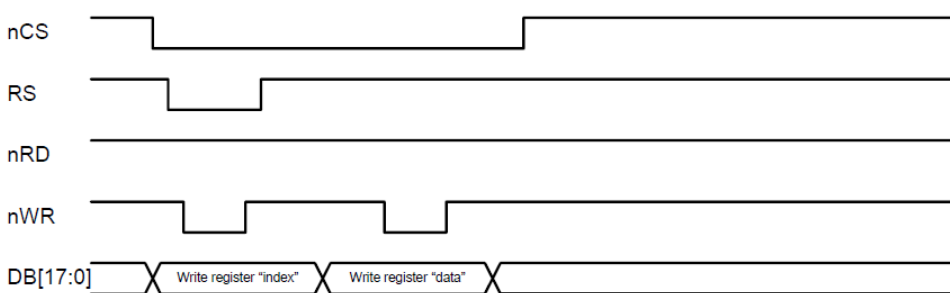
II.2 Modul LCD displeje

LCD displej s rezistivní dotykovou vrstvou slouží jako zobrazovací i vstupní zařízení. Vzhledem k cenové výhodnosti byl vybrán hotový modul z Číny s displejem **240374PQ**, slotem pro paměťovou kartu **SD** a řadičem dotykové fólie **XPT2046**. Všechna zařízení jsou určena pro napájecí napětí 3,3 V. Důležité piny jsou vyvedeny na konektoru 2×20 s roztečí 2,54 mm. Odpovídající knihovni součástku pro návrhový software Eagle jsem vytvořil podle dokumentace a rozměrového náčrtku, který prodejce poskytuje.

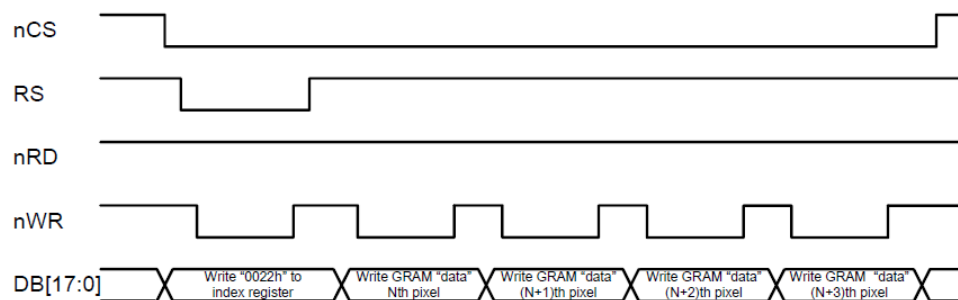
II.2.1 LCD displej

Rozlišení displeje je 320×240 pixelů a je vybaven řadičem **ILI9325** [ILI] konfigurovaným pro komunikaci po 16ti bitové paralelní sběrnici. Pro řízení komunikace slouží dále linky **WR**, **RD**, **CS** a **RS**, linka **RESET** slouží k inicializaci řadiče. Rutiny pro přístup ke konfiguračním registrům a zápis dat do řadiče vycházejí z dokumentace řadiče **ILI9325** a displeje samotného:

Zápis do registru



Zápis dat (GRAM)



Obr. 3: Časování signálů (z [ILI])

```
void LCD_WR_REG(unsigned char vh, unsigned char vl) {  
    WR_DATA_H = vh;  
    WR_DATA_L = vl;  
    WR0;  
    WR1;  
}
```


Mobilní telefon s dotykovým displejem

```
void LCD_WR_CMD(unsigned char vh, unsigned char vl) {
    RS0;
    LCD_WR_REG(vh, vl);
}

void LCD_WR_DATA(unsigned char vh, unsigned char vl) {
    RS1;
    LCD_WR_REG(vh, vl);
}
```

WR_DATA_H a **L** jsou zástupné symboly pro dva osmibitové výstupní porty, makra **RS1**, **RS0**, **WR1**, **WR0** slouží k nastavení výstupních úrovní řídicích linek displeje.

II.2.2 Řadič dotykové folie

Řadič dotykové folie (touchscreen controller) s čtyřvodičovým rozhraním **XPT2046 [XPT]** slouží k vyhodnocování dotyku a k získání polohy dotykového pera (stylus) nebo prstu. Rezistivní dotyková folie se skládá ze dvou vrstev – pružné polyesterové a pevné skleněné. Obě vrstvy jsou opatřeny tenkým vodivým povrchem. Po doteku prstu nebo pera se vrstvy přiblíží, čímž dojde k vodivému spojení. Podle odporu jednotlivých drah pak řadič vyhodnotí polohu doteku.

Komunikace probíhá po sběrnici **SPI** v režimu slave. Obvod poskytuje také výstupní pin **PENIRQ**, který indikuje dotek na folii a lze jej tak použít jako přerušení. Rutiny pro čtení polohy z dotykové folie vypadají takto (funkce **map** slouží k přepočítání naměřené hodnoty pomocí kalibračních hodnot do rozsahu 0-240 nebo 0-320 tak, aby odpovídaly souřadnicím na displeji):

```
void spi_putc(unsigned char data) {
    SPDR = data;
    while(!(SPSR & (1 << SPIF)));
    SPSR &= ~(1 << SPIF);
}

unsigned char spi_getc() {
    SPDR = 0xFF; // Send some dummy data
    while(!(SPSR & (1 << SPIF)));
    SPSR &= ~(1 << SPIF);
    return SPDR;
}

unsigned char touchRead(struct touch_struct *touch) {
    if (!XPT2046_PEN) { touch->active = 0; return 0; }

    unsigned int res;
    XPT2046_CS0;
    _delay_us(15);

    // X axis
    spi_putc(0xD8);
    while (XPT2046_BUSY);
    res = spi_getc();
    XPT2046_CS1;
}
```

Mobilní telefon s dotykovým displejem

```

res = 240-map(res, XPT_X1, XPT_X2, 240);
if (res > 240) { res = 240; } else if (res < 0) { res = 0; }
touch->x = res;

XPT2046_CS1;
_delay_us(10);
XPT2046_CS0;
_delay_us(10);

// Y axis
spi_putc(0x98);
while (XPT2046_BUSY);
res = spi_getc();
res = 320-map(res, XPT_Y1, XPT_Y2, 320);
if (res > 320) { res = 320; } else if (res < 0) { res = 0; }

touch->y = res;
touch->active = 1;

XPT2046_CS1;
return 1;
}

```

Měření polohy je zahájeno příkazem zaslaným po sběrnici **SPI**. Příkaz tvoří jeden bajt, jehož formát je popsán v následující tabulce:

Bit	7	6	5	4	3	2	1	0
Název	Start	A2	A1	A0	Rozlišení	SER/DFR	PD1	PD0
Význam	1	Adresa měřeného vstupu			1 = 8 bit 0 = 12 bit	Výběr reference	Sleep mode	

Tabulka 1: Formát příkazu pro řadič XPT2046

Adresa měřeného vstupu je vytvořena kombinací bitů *A2-A0*. Zajímavostí je, že obvod **XPT2046** umožňuje také měření napětí baterie a snímání dalšího analogového vstupu, využitelného například pro měření teploty.

	A2	A1	A0
Y	0	0	1
X	1	0	1
Z₁	0	1	1
Z₂	1	0	0

Tabulka 2: Výběr adresy vstupu

Mobilní telefon s dotykovým displejem

II.2.3 Paměťová karta

SD paměťová karta slouží k ukládání dat, která by se nevešla do paměti mikrokontroléru. Typicky jde o obrázky ve formátu BMP určené k vykreslení na displej. Karta komunikuje stejně jako řadič **XPT2046** po sběrnici SPI. Více o přístupu k SD kartě je napsáno v části *Software*.

II.3 Mikrokontrolér Atmega128A

Osmibitový mikrořadič **Atmega128A** [M128] rodiny Atmel AVR v pouzdru TQFP-64 pro povrchovou montáž (SMT) byl vybrán proto, že disponuje dostatečným počtem vstupních a výstupních pinů, které jsou z velké části využity pro řízení displeje, relativně velkou paměť (128 kB Flash, 4 kB EEPROM a 4 kB SRAM) a dostatkem dalších periférií. Paměť Flash je kromě programu využita pro uložení malých obrázků (ikonek) a textových řetězců.

Verze *A* má oproti původnímu obvodu Atmega128 větší rozsah napájecího napětí a je tedy vhodná pro použití v 3,3 V aplikaci. Mikrokontrolér je taktován krystalem na 12 MHz.



Obr. 4: ATmega128A

Programovací linky rozhraní **ISP** jsou vyvedeny na 6ti pinový konektor. Piny linek **PDO** a **PDI** jsou v mikrokontroléru sdíleny s linkami **RXD0** a **TXD0**, takže lze konektor využít k ladění aplikace prostřednictvím sériové linky **UASRT0**. Pro komunikaci s GSM modulem je použita periférie **USART1**.

Mikrokontrolér řídí podsvícení displeje PNP tranzistorem připojeným na výstupu PWM kanálu, piezo reproduktorek sloužící pro generování zvuků, rovněž připojený k PWM kanálu (vyzvánění, reakce na akce v uživatelském rozhraní) a malý vibrační motorek, spínaný NPN tranzistorem. Kromě toho čte mikrokontrolér stav pinu signalizujícího nabíjení baterie (více v části *Napájení*). Tři vstupy A/D převodníku (eventuálně digitální vstupy/výstupy) jsou vyvedeny na třipinové konektory, obsahující také napájení a zem. Dají se využít k připojení akcelerometru (senzor zrychlení), teploměru nebo senzoru osvětlení.

K signalizaci funkce zařízení jsou použity dvě LED diody (červená a zelená) a jako uživatelský vstup slouží kromě dotykového displeje tlačítko.

Z mikrokontroléru jsou vyvedeny linky **D+** a **D-** pro USB konektivitu. Ta lze využít například pro programování paměti programu za použití *bootloaderu* vycházejícího z volně dostupné knihovny **V-USB** (implementace nízkorychlostního USB zařízení metodou „bit bang“ na mikrokontrolérech AVR, které nemají USB jako periférii) [*V-USB*].

Mobilní telefon s dotykovým displejem

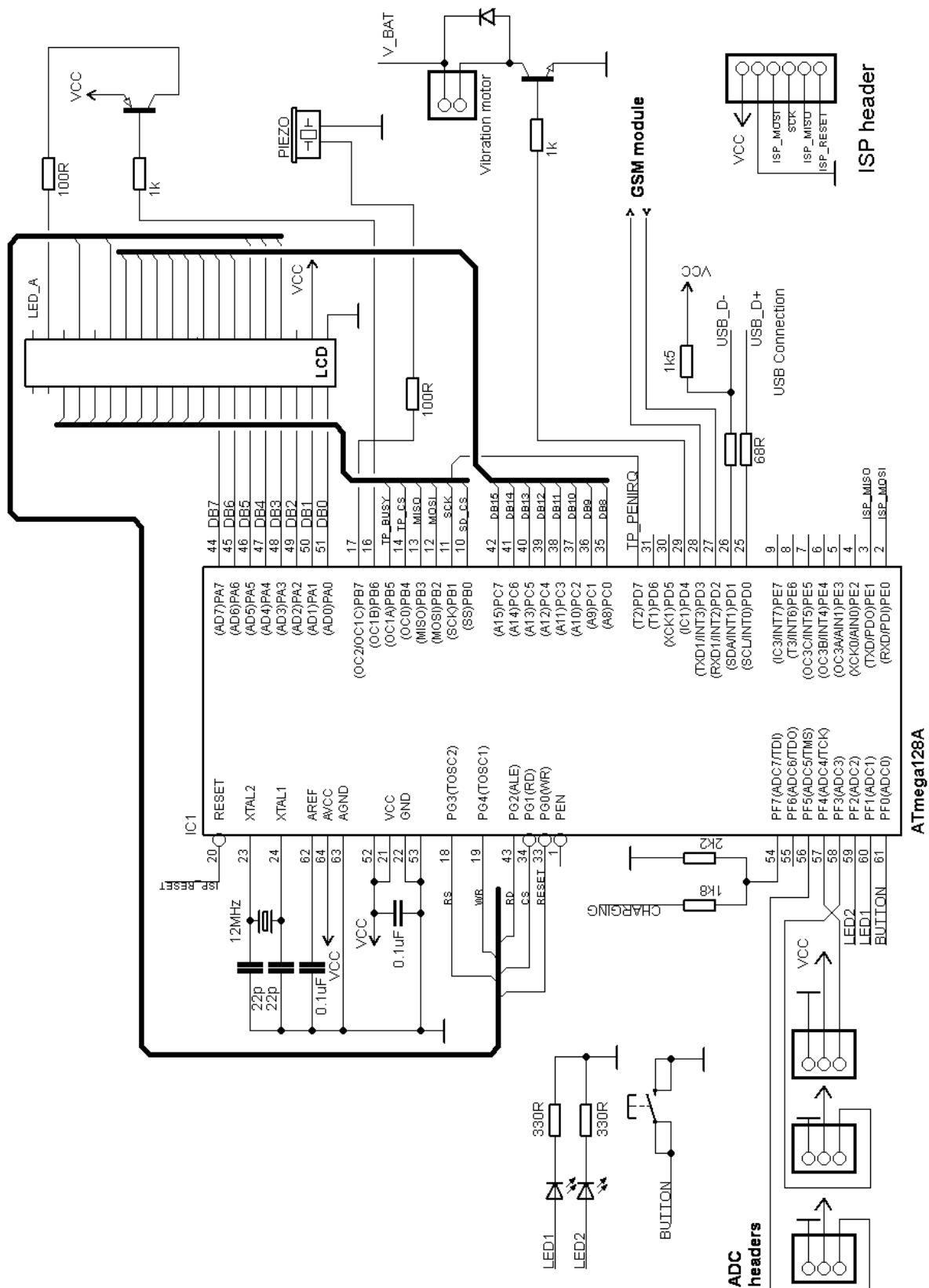
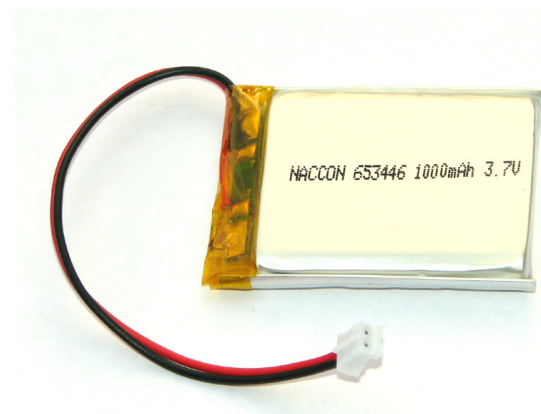


Schéma 2: Část s mikrokontrolérem Atmega128A – Ve schématu jsou vypuštěny popisy pinů u konektoru modulu LCD displeje, pro ilustraci nejsou nezbytné. Součástka pro program Eagle je samozřejmě obsahuje.

Mobilní telefon s dotykovým displejem

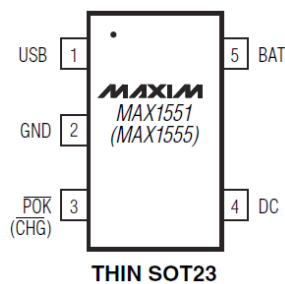
II.4 Napájení

GSM modul **SIM100S** vyžaduje napájecí napětí v rozsahu 3,4 – 4,5 V, odběr může být ve špičkách až 2 A. Z tohoto důvodu je napájen přímo jednočlánkovým Li-Pol akumulátorem, vyráběným čínskou firmou *Naccon*, s nominálním napětím 3,7 V a kapacitou 1000 mAh. Akumulátor je vybaven obvodem, který zajišťuje nadproudovou ochranu a kontrolou podvybití (3 V). Akumulátor je přímo od výrobce dodáván s nakrimpovaným zámkovým **konektorem JST**. Piny konektoru mají rozteč 2 mm a zámek zabraňuje přepólování baterie.

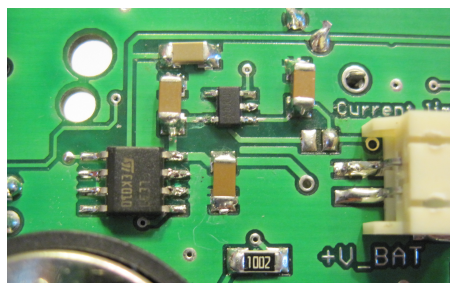


Obr. 5: Li-Pol článek 3,7 V

Nabíjení akumulátoru zajišťuje integrovaný obvod **MAX1555** firmy *Maxim* v pěťvývodovém pouzdru SOT-23-5 pro povrchovou montáž. Použití obvodu je velice jednoduché a zapojení vychází z dokumentace výrobce [MAX]. Výstup pro připojení signalizační LED (ta signalizuje, zda nabíjení právě probíhá nebo zda je dokončeno) je připojen na vstup mikrokontroléru Atmega128A. **MAX1555** umožňuje nabíjení ze dvou zdrojů, **USB** (5 V) nebo **DC** (3,7 – 7 V). V případě, že jsou připojeny oba zdroje, obvod automaticky zvolí zdroj DC. Při použití zdroje USB je použit nabíjecí proud 100 mA, pro nabíjení ze zdroje DC je nastaven proud 280 mA. Na desce mobilního telefonu jsem k nabíjení použil pouze Mini-USB konektor, lze jej ale použít i jako napájecí zdroj DC a zvýšit tak nabíjecí proud (k tomu slouží jumper „**Current limit**“ realizovaný dvěma pájecími ploškami na desce).



Obr. 6: MAX1555

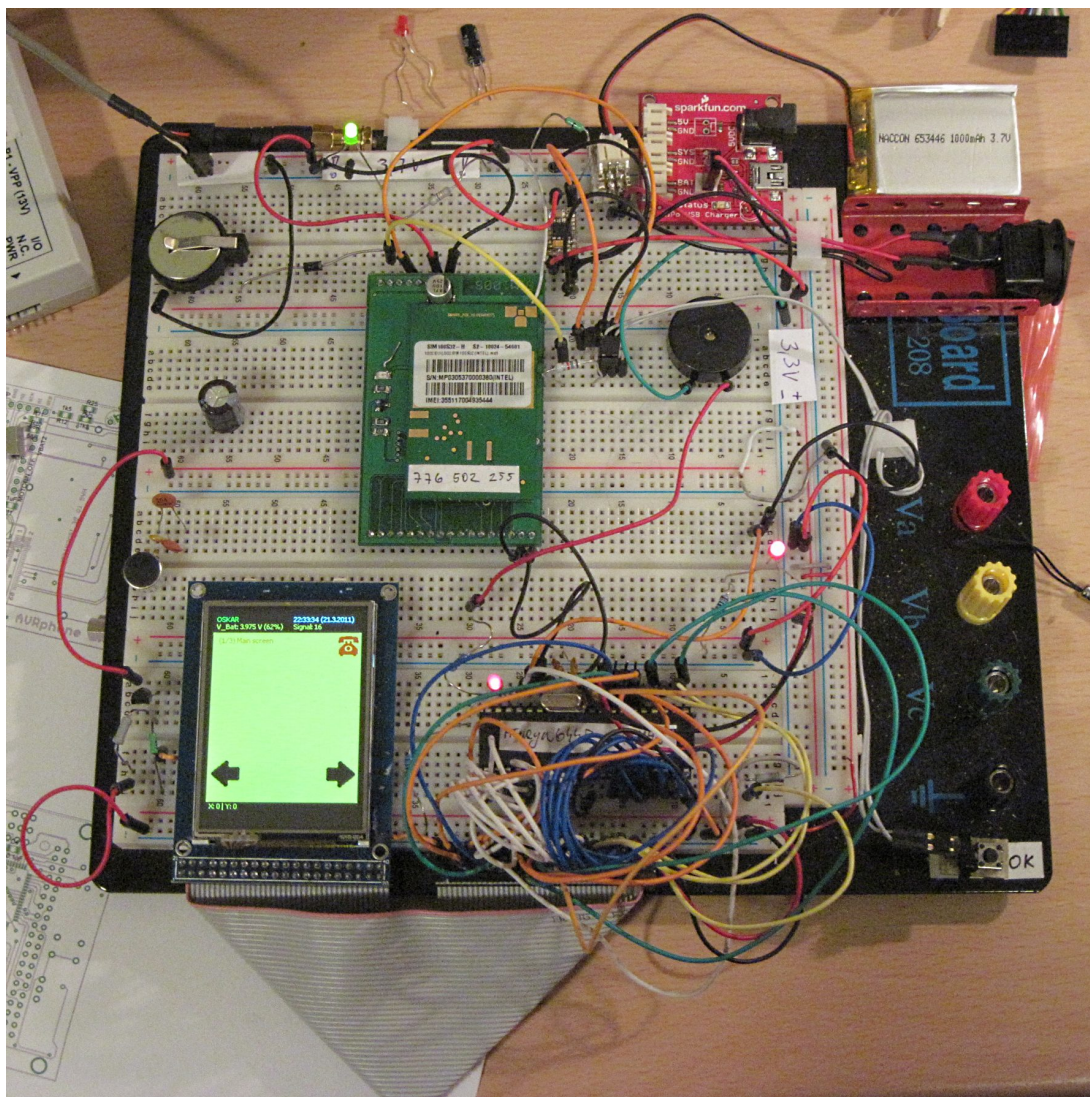


Obr. 7: Fotografie napájecí části, vpravo napájecí konektor

Mobilní telefon s dotykovým displejem

II.5 Ověření na nepájivém poli

Pro ověření všech funkcí byly základní části mobilního telefonu odzkoušeny na univerzálním nepájivém poli s mikrokontrolérem **ATmega644P** [M644]. Nabíjecí obvod **MAX1555** byl připojen díky desce vyráběné americkou firmou *Sparkfun*, pro GSM modul **SIM100S** jsem navrhl a nechal vyrobit jednoduchou tzv. „break out“ desku se signalizační LED diodou, konektorem pro SIM kartu ze spodní strany a s důležitými piny vyvedenými na 2,54 mm lišty. Obvod **LE33** je umístěn na destičce univerzálního prototypovacího plošného spoje s roztečí plošek 1,27 mm, protože pin **INHIBIT** je dostupný jen ve verzi určené pro povrchovou montáž.



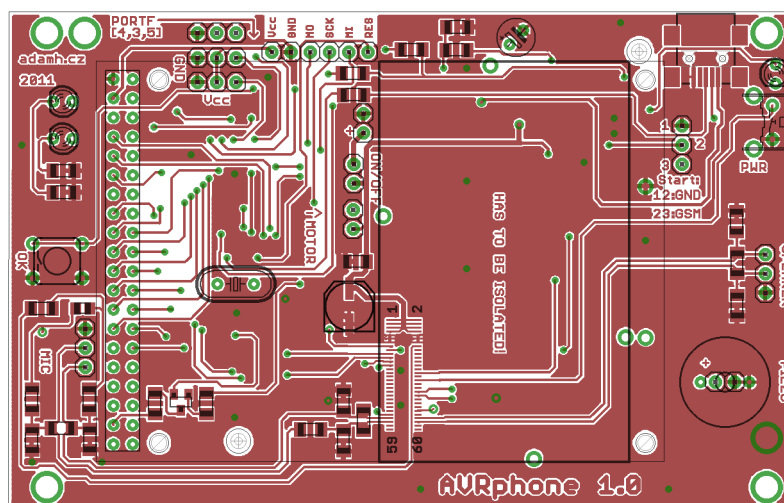
Obr. 8: Zkušební zapojení na nepájivém poli

Mobilní telefon s dotykovým displejem

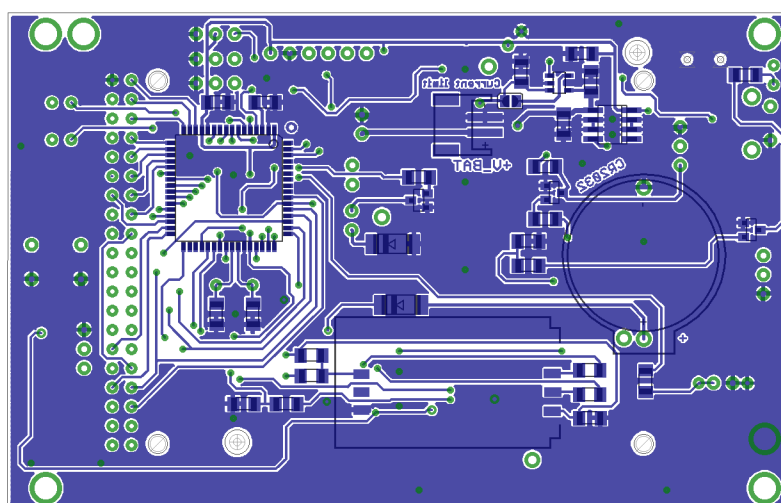
II.6 Deska plošných spojů

Deska plošných spojů byla navržena v programu *Eagle* jako oboustranná s prokovy a je vybavena nepájivou maskou a bezolovnatým pokovením (HAL). Diskrétní součástky jsou většinou v pouzdech pro povrchovou montáž, SOT-23 pro tranzistory a 1206 pro rezistory a blokovací kondenzátory. Výjimku tvoří tlačítka a signalizační LED diody, pro které jsem zvolil klasická „through-hole“ pouzdra, aby dosahovaly nad úroveň potenciální krabičky. Ze stejného důvodu jsou mikrofón a reproduktor připojeny kablíkem zakončeným konektorem.

Návrh počítá s otvory pro uchycení GSM modulu, distančními sloupky pro uchycení displeje a celé desky a s otvory použitými pro průchod kabelů (vypínač, vibrační motorek, anténa apod.).



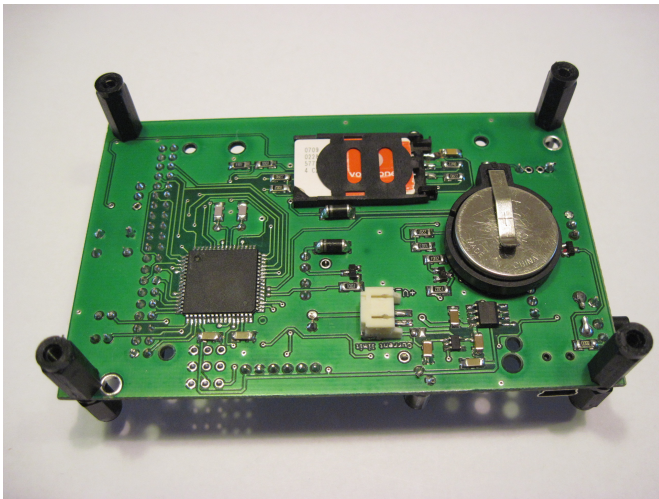
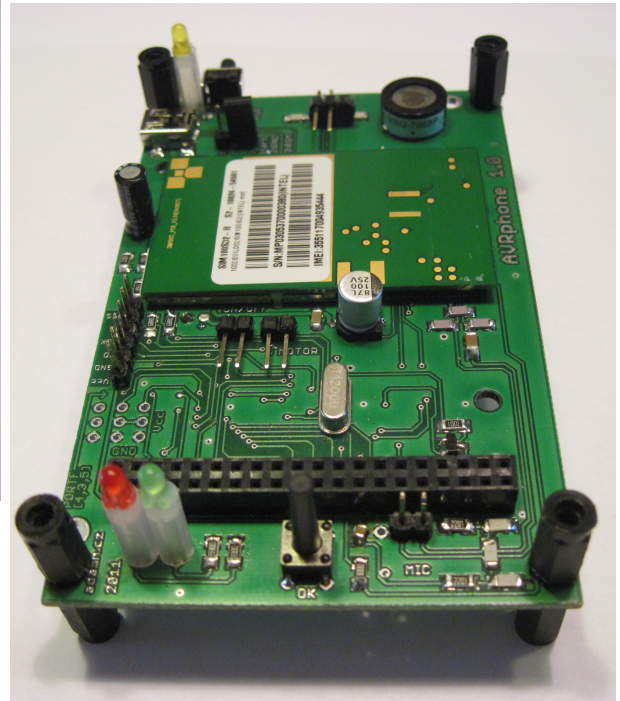
Obr. 9: Deska plošných spojů – vrchní strana (TOP)



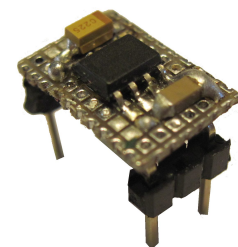
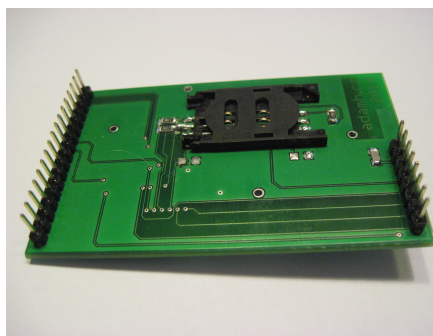
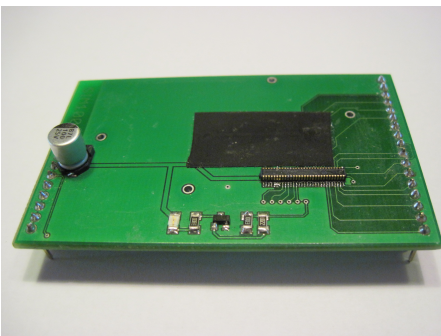
Obr. 10: Deska plošných spojů – spodní strana (BOTTOM)

Mobilní telefon s dotykovým displejem

II.7 Fotografie desek plošných spojů



Přípravky pro ověření funkčnosti na nepájivém poli (SIM100S a LE33C):



III. Software

Program pro mikrokontrolér ATmega128A byl napsán v jazyce C s využitím volně dostupného toolchainu **WinAVR**², obsahujícího kompilátor založený na GCC a knihovny avr-libc [*LIBC*]. Rutiny pro přístup k perifériím mikrokontroléru byly napsány podle dokumentace firmy *Atmel*. Výjimkou je již hotová knihovna pro práci s paměťovou kartou SD.

Tato část práce není a také vzhledem k rozsahu ani nemůže být detailním popisem celého programu a elementárních rutin pro práci s perifériemi (sériová linka, časovače, A/D převodník atd.), jsou v ní zmíněny ty části programu, které jsou nějakým způsobem zajímavé.

Výpisy zdrojových kódů jsou oproti programu zkrácené (aby nezabraly příliš mnoho místa) a doplněny o komentáře v češtině, anglické názvy proměnných a funkcí byly zachovány.

III.1 Kreslení BMP obrázků

Pro kreslení barevných obrázků na displej jsem zvolil poměrně rozšířený **formát BMP** (Bitmap, bitová mapa). Obrázky jsou uloženy přímo na paměťové kartě se souborovým systémem FAT. Pro přístup ke kartě a souborovému systému jsem použil již hotovou knihovnu [*RIE06*].

Aby bylo možné obrázky upravovat přímo po vložení paměťové karty do čtečky počítače, použil jsem asi nejrozšířenější formát BMP obrázků podporovaný například oblíbeným programem *Malování*, který je součástí operačního systému *Windows* – nekomprimovaná bitová mapa s 24 bity na pixel. Jako zdroj informací o formátu mi při implementaci posloužil článek [*KIR01*].

Nevýhody tohoto formátu jsou dvě:

1. Barevné složky pro každý pixel je nutné přepočítat (ze tří bajtů na dva).
2. Bitmapy jsou ukládány „vzhůru nohama“ – při přechodu na další obrazový řádek je tedy nutné posunout ukazatel GRAM paměti LCD řadiče „ručně“, nelze využít automaticky inkrementovaného ukazatele při běžném zápisu dat „shora dolů“.

Offset	Počet bajtů	Význam
0x0012	4	Šířka (W)
0x0016	4	Výška (H)
0x001c	2	Počet bitů na pixel
0x0036	$(W \times H) \cdot 3$	Obrazová data [B, G, R]

Tabulka 3: Užitečná data BMP souboru

BBBBBBBB	GGGGGGGG	RRRRRRRR	
RRRRR	GGG	GGG	BBBBB

Tabulka 4: Přepočítání 24-bitového formátu na 16-ti bitové schéma 5-6-5

² WWW stránka projektu WinAVR: <http://winavr.sourceforge.net/>

Mobilní telefon s dotykovým displejem

Následující výpis zdrojového kódu ukazuje přepočítání barevných složek každého pixelu a jejich vykreslení na LCD displej (data jsou načítána do vyrovnávacího bufferu **pixelBuffer** po blocích o šířce 300 B):

```
int w = x+width-1;
int bytes = 100; // Počet pixelů v bufferu (100*3 = 300 B)

// Načtení 300 B bloku dat
while (fat_read_file(fd, pixelBuffer, (bytes * 3)) == (bytes * 3)) {
    int p;
    unsigned char *b;
    for (p = 0; p < (bytes * 3); p += 3) {
        // Převod z 24-bit na 16-bit RGB formát
        b = &pixelBuffer[p];
        *(b) >>= 3; // Modrá (B - 5 bajtů)
        *(b + 1) >>= 2; // Zelená (G - 6 bajtů)
        *(b + 2) >>= 3; // Červená (R - 5 bajtů)

        LCD_WR_DATA(((*(b+2) << 3) | (*(b+1) >> 3)),
                    ((*(b+1) << 5) | *(b)));

        // Posun souřadnice X a „ruční“ přepnutí řádku (souřadnice Y)
        xx++;
        if (xx > w) {
            xx = x;
            yy--;
            LCD_setXY(xx, yy);
            if (yy < y) {
                break;
            }
        }
    }
}
```

Mobilní telefon s dotykovým displejem

III.2 Kreslení obrázků z paměti Flash

Pro kreslení malých jednobarevných ikon (například tlačítka) jsou BMP obrázky na paměťové kartě nevhodné, proto jsem zvolil načítání a vykreslování ikon z paměti Flash, které je řádově rychlejší.

Obrázky jsou uloženy jako pole bajtů v programové paměti (realizováno funkcemi z knihovny `<avr/pgmspace.h>`). První čtyři bajty jsou hlavička, obsahující rozměry obrázku, po ní následují data:

Bajt	0	1	2	3	4
Popis	Šířka obrázku		Výška obrázku		Obrazová data

Tabulka 5: Popis formátu pro uložení ikonky

Pixely jsou ukládány od levého horního po pravý dolní roh – tak, jak se inkrementuje ukazatel obrazové paměti **GRAM** v řadiči displeje. Každému pixelu odpovídá jeden bit, černí pixely jsou uloženy jako 1 a bílé jako 0. Ikonka s rozměry 39×39 pixelů tak v paměti Flash (celkem 128 kB) zabere 4+39·39, tedy 195 Bajtů. Funkce pro vykreslení obrázku umožňuje nastavit barvu popředí i pozadí, případně je možné nastavit pozadí průhledné (`BG_TRANSPARENT`):

```
#define BG_TRANSPARENT 1

// Ukázka obrázku uloženého v programové paměti
const unsigned char imgSquare[] PROGMEM = {0x27, 0x00, 0x27, 0x00, ...};

void drawImage(unsigned int x, unsigned int y, const unsigned char *image,
               unsigned int color, unsigned int bgColor) {

    unsigned long pixel = 0;
    int i;

    // Zjištění šířky a výšky
    unsigned int width = (pgm_read_byte(image+1) << 8);
    width += pgm_read_byte(image);

    unsigned int height = (pgm_read_byte(image+3) << 8);
    height += pgm_read_byte(image+2);

    // Celkový počet pixelů
    unsigned long pixels = width * height;
    unsigned int byte = 4;

    // Nastavení oblasti pro zápis do LCD na rozměry obrázku
    LCD_setWrap(x, y, x + width, y + height - 1);

    // Znovupoužití starých proměnných (ulehčení RAM)
    width += x;
    height = x;
}
```

Mobilní telefon s dotykovým displejem

```
// Pro každý pixel (bit) v bloku dat (1 bajt):
for (i = 7; pixel < pixels; i--) {
    LCD_setXY(x+1, y); // Změna pixelu, do kterého se bude zapisovat

    if (i < 0) { i = 7; byte++; }

    // Vykreslení barvou popředí nebo pozadí (pokud není průhledné)
    if ((pgm_read_byte(image+byte) & (1 << i)) > 0) {
        LCD_WR_DATA((color >> 8), color);
    } else if (bgColor != BG_TRANSPARENT) {
        LCD_WR_DATA((bgColor >> 8), bgColor);
    }

    // Inkrementace pozice pixelu
    pixel++; x++; if (x == width) { x = height; y++; }
}

// Obnovení oblasti pro zápis do LCD na 320x240
openWrap();
}
```

Funkce má i svou obdobu, která do prostoru tlačítka vykreslí pro větší efekt místo jedné barvy brevný přechod (gradient). Pro převádění černobílých obrázků z formátu PNG jsem ve skriptovacím jazyce PHP napsal jednoduchý program **png2code**. Jeho výstupem je textový soubor, který obsahuje jednotlivé bajty v hexadecimálním zápisu oddělené čárkami.



```
C:\Příkazový řádek
C:\Documents and Settings\Adam Heinrich\Dokumenty\programovani\eclipse-workspace\ATmega644P_Test\util\png2code>C:\wamp\bin\php\php5.2.8\php.exe -f png2code.php phone.png

Data saved to phone.txt: Width: 39, Height: 39, Size: 195 Bytes
C:\Documents and Settings\Adam Heinrich\Dokumenty\programovani\eclipse-workspace\ATmega644P_Test\util\png2code>
```

Obr. 11: Převodní obrázku programem png2code

Mobilní telefon s dotykovým displejem

III.3 Zasilání AT příkazů

Pro komunikaci s GSM modulem je použita periferie **USART1**. Modulu jsou zasilány textové AT příkazy a odpovědi jsou zachyceny v přerušení do globálního pole (*buffer*) *uart* o šířce 350 B.

Globální proměnná *uartLen* uchovává délku přijaté zprávy, proměnná *uartReceived* je příznak (*flag*) přijaté zprávy – je v přerušení nastavena na 1, pokud byla přijata celá odpověď GSM modulu (odpověď vždy končí „odřádkováním“ – dvojicí znaků <CR><LF>). Do přijaté zprávy nejsou tyto znaky počítány, aby se usnadnila možnost porovnávání odpovědi funkcí **strcmp** z knihovny <*string.h*>.

Funkce **atc** slouží k zaslání AT příkazu a vyčkání na odpověď, aby mohlo následovat další zpracování (pokud je potřeba). Čas strávený čekáním na přijatou odpověď je pro uživatelské rozhraní zanedbatelný, proto je použita čekací smyčka.

V případě, že doba čekání na reakci přesáhne 2 vteřiny, je procesor resetován pomocí *watchdog* časovače (*WDT*). V programu je pro úsporu RAM použita podobná funkce používající řetězce uložené v programové paměti – Flash (knihovna <*avr/pgmspace.h*>). To lze využít u „statických“ AT příkazů bez parametru nebo s neměnnými parametry (například *AT+CCLK?* pro zjištění času, *ATH* pro zavěšení hovoru, *AT+CMGF=1* pro nastavení ASCII formátu SMS zpráv apod.).

```
void atcn(char *str) {
    uartLen = 0;
    uartReceived = 0;
    wdt_enable(WDTO_2S); // Nastavení WDT na 2 vteřiny

    while (*str) { // Odeslání příkazu
        uartPutc(*str++);
    }
    uartPutc(13); // Odřádkování (<CR><LF>)
    uartPutc(10);

    while (!uartReceived); // Čekání na odpověď
    wdt_disable();
}
```

Mobilní telefon s dotykovým displejem

III.4 Zpracování odpovědi GSM modulu

Odpověď GSM modulu pokaždé nestačí pouze porovnat, v některých případech je potřeba informace v ní obsažené uložit a dále zpracovávat. Proto jsem vytvořil několik funkcí, které informace z odpovědi ukládají do samostatných řetězců, případně je převádějí na čísla.

Jednoduchá funkce `parseTime` slouží k uložení času do struktury, se kterou se v programu dále pracuje. Odpověď po zaslání AT příkazu `AT+CCLK?` je ve formátu:

```
+CCLK: "94/05/06,22:10:00+00"
```

a je tedy možné číselné údaje „vytáhnout“ z pevně daných pozic, které se nemění (jednociferná čísla jsou zarovnána nulami):

```
void parseTime(char *buffer, struct time_struct *time) {
    time->hours = ((buffer[17]-48)*10)+buffer[18]-48;
    time->minutes = ((buffer[20]-48)*10)+buffer[21]-48;
    time->seconds = ((buffer[23]-48)*10)+buffer[24]-48;

    time->year = ((buffer[8]-48)*10)+buffer[9]-48+2000;
    time->month = ((buffer[11]-48)*10)+buffer[12]-48;
    time->day = ((buffer[14]-48)*10)+buffer[15]-48;
}
```

Složitější funkce `parseSms` slouží k získání informací o SMS zprávě, které jsou zaslány jako odpověď na AT příkaz `AT+CMGR=<ID zprávy>` (v případě, že je zvolen ASCII formát SMS zpráv) ve formátu:

```
+CMGR: "REC UNREAD", "+8613918186089", ["Jmeno"], "02/01/30,20:40:31+00"
Text zprávy...
```

Celou odpověď je tedy nutné projít znak po znaku a orientovat se podle počtu čárek, který je pevně daný. První řetězec značí, zda je zpráva přečtená či nová, nepřečtená, druhý řetězec obsahuje telefonní číslo odesilatele a v poli „jméno“ je uloženo jméno odesilatele, pokud je nalezeno v seznamu kontaktů na SIM kartě (hranatí závory značí, že se pole v odpovědi nemusí objevit vždy a je tudíž „nepovinné“).

Mobilní telefon s dotykovým displejem

Následuje zkrácený výpis zdrojového kódu funkce doplněný o české komentáře. Argumenty **buffer** a **len** jsou vstupní řetězec a jeho délka, **number** a **name** jsou řetězce pro uložení telefonního čísla a jména odesilatele, **time** je struktura pro uložení času doručení zprávy a **contentOffset** je číslo bajtu, kterým začíná text zprávy (pro vypsání na displej). V případě, že je zpráva přečtená, vrací funkce hodnotu 1, v opačném případě 0.

```
unsigned char parseSms(char *buffer, int len, char *number, char *name,
                      struct time_struct *time, unsigned char *contentOffset) {

    unsigned char i, n, read, commaCount, start, dateTimeStart;
    read = 0; commaCount = 0; start = 0; dateTimeStart = 0;

    strcpy(name, "Unknown"); // Výchozí hodnota neuloženého jména

    for (i = 0; i < len; i++) {
        if (buffer[i] == ',') { // Počítání čárek
            commaCount++; start = 0;
        } else if (buffer[i] == '"' &&!start) { // Detekce řetězce
            start = 1; n = 0;
        }

        if (start && buffer[i] != '"') {
            switch (commaCount) {
                case 0: // Přečtená/nepřečtená
                    if (buffer[i+4] == 'R') { read = 1; }
                    start = 0;
                    break;
                case 1: // Tel. číslo
                    number[n++] = buffer[i]; number[n] = '\0';
                    break;
                case 2: // Jméno
                    name[n++] = buffer[i]; name[n] = '\0';
                    break;
                case 3: // Čas
                    dateTimeStart = i; start = 0; *contentOffset = i+21;
                    break;
            }
        }
    }

    // ... Uložení času - Podobný kód jako u funkce parseTime

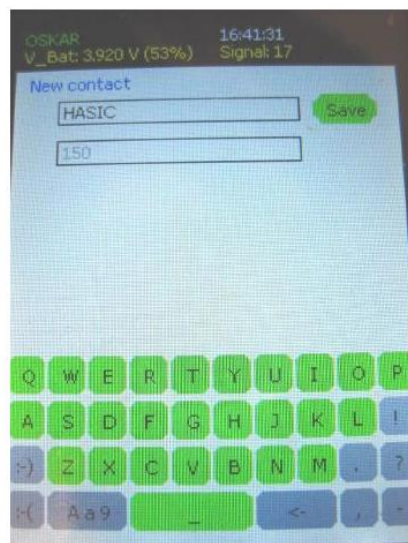
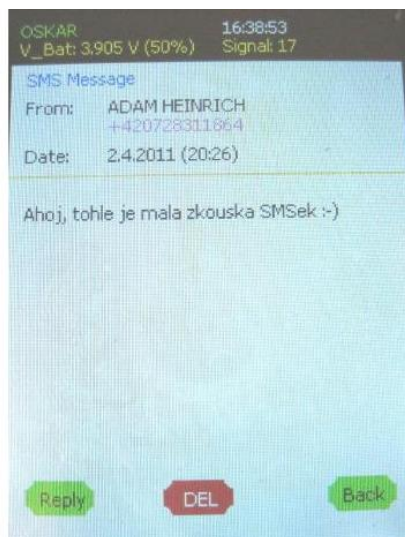
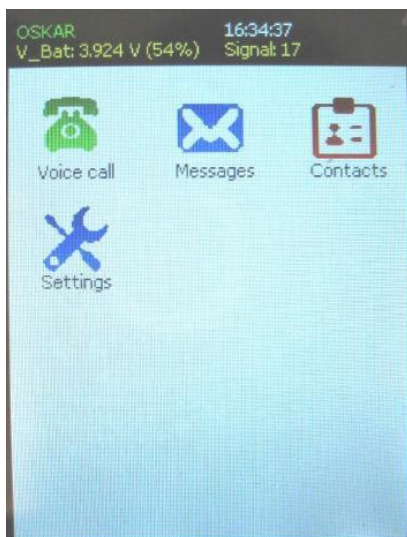
    return read;
}
```


Mobilní telefon s dotykovým displejem

IV. Uživatelské rozhraní

Uživatelské rozhraní je navrženo jako dotykové. Ve většině případů lze obsluhovat prsty, pro zadávání textu „klasickou“ QWERTY klávesnicí je ale vhodné použít stylus, který má tenčí hrot (použití obyčejné tužky je výrobcem v dokumentaci [LCD] LCD displeje zakázáno). Veškeré popisky a informační hlášky v uživatelském rozhraní jsou anglicky, stejně jako zdrojové kódy mobilního telefonu.

Na následujících šesti fotografiích jsou vyobrazeny ukázky z uživatelského rozhraní: **hlavní menu** s volbou „aplikace“, **vytočení telefonního čísla** pomocí numerické klávesnice, zobrazení **přijaté SMS zprávy**, zobrazení **seznamu kontaktů** na SIM kartě, vytvoření **nového kontaktu** – zadávání jména a **startovací obrazovka** – program čeká na inicializaci GSM modemu a paměťové karty, provede počáteční konfiguraci (přihlášení do sítě, volba operátora, nastavení formátu SMS zpráv atd.) a pro přehlednost vypisuje informace o právě prováděných operacích. Celý start trvá asi tři vteřiny.



Mobilní telefon s dotykovým displejem

V. Závěr

Podařilo se mi navrhnout, vyrobit a naprogramovat jednoduchý mobilní telefon s funkčním grafickým rozhraním, který lze dále rozšiřovat – především po stránce softwarové, protože jsem kvůli kompilacím s čínským prodejcem modulu LCD displeje (správný modul jsem obdržel až po více než dvou měsících od objednání) mohl návrh elektroniky dokončit až v půlce března roku 2011.

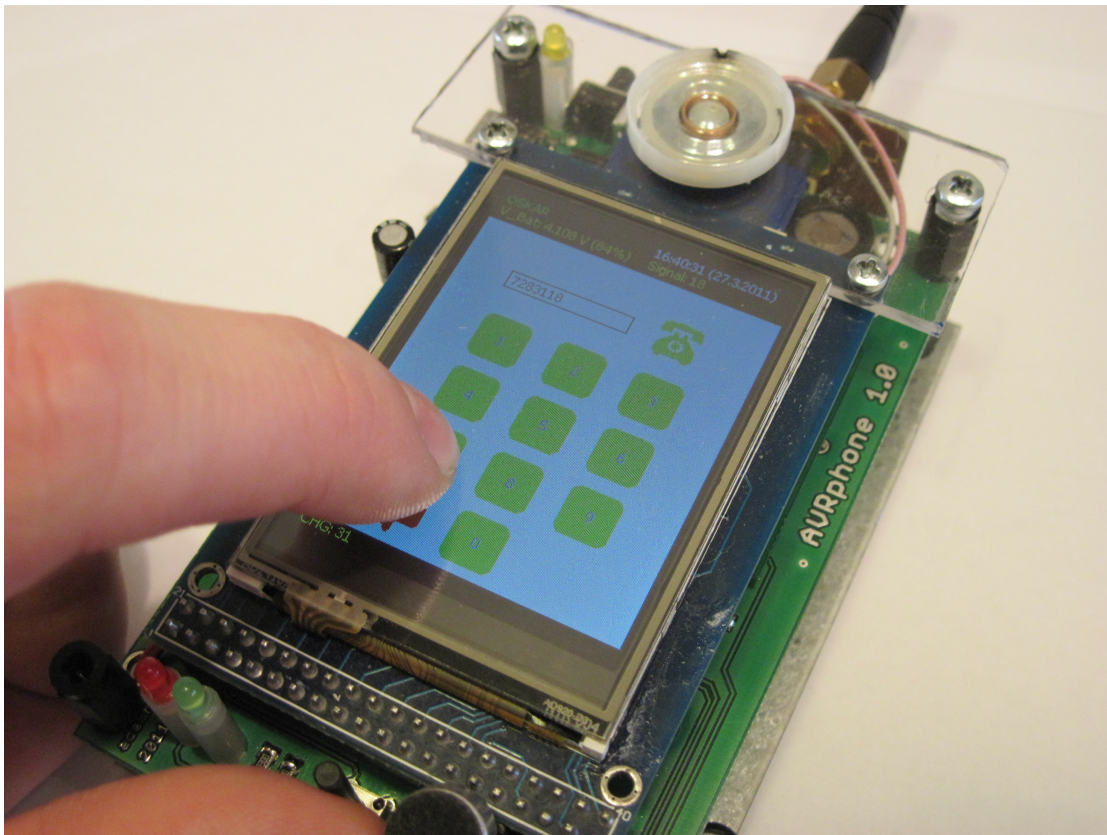
Mezi další plánovaná rozšíření patří:

- Plastová nebo plexisklová **krabička**
- **Prohlížení fotografií** na paměťové kartě – zde zvažuji místo formátu BMP použít vlastní formát, ve kterém by byly barvy uloženy v 16ti bitovém formátu 5-6-5 a který by tak načítání a vykreslování obrázků zrychlil.
- **Jednoduché hry**
- Dopracování, vylepšení a **zpřehlednění grafického rozhraní**, například vytvoření grafických indikátorů stavu baterie a síly signálu nebo sjednocení nabídek a tlačítek
- **Připojení akcelerometru** jako vstupního zařízení, případně **teploměru** pro měření pokojové teploty

Text práce, schémata, užitečné části zdrojových kódů a informace o vývoji projektu jsou k dispozici na WWW stránkách www.projects.adamh.cz/AvrPhone, aby mohly posloužit jako zdroj informací inspirace a usnadnily tak dalším zájemcům například práci s grafickým displejem a implementaci grafického rozhraní. Na této stránce budou rovněž přibývat aktuální fotografie.

Mobilní telefon s dotykovým displejem

VI. Fotografie hotového telefonu



VII. Významy některých použitých zkratk

- **GSM** Groupe Spécial Mobile (fr.) – komunikační *standard používaný mobilními telefony (1982)*
- **SIM** Subscriber Identity Module – *čipová karta pro identifikaci účastníka v síti mobilního operátora*
- **LCD** Liquid Crystal Display – *displej tvořený maticí tekutých krystalů*
- **LED** Light-Emitting Diode – *světlo vyzařující dioda*
- **Tx** Transmit – *přenos*
- **Rx** Receive – *příjem*
- **DC** Direct Current – *stejnosměrný proud*
- **PWM** Pulse Width Modulation – *pulzně šířková modulace*
- **ISP** In-System Programming – *programování v systému*
- **SPI** Serial Peripheral Interface – *Sériové rozhraní pro komunikaci mezi čipy*
- **UART** Universal Asynchronous Receiver/Transmitter – *asynchronní sériová linka*
- **ADC** Analog to Digital Converter – *analogově/digitální převodník*
- **AVR** Advanced Risc Machine – *mikroprocesorová řada výrobce Atmel*
- **SMD** Surface Mount Device – *součástka pro povrchovou montáž*
- **SMT** Surface Mount Technology – *technologie povrchové montáže*
- **USB** Universal Serial Bus – *vysokorychlostní diferenciální sériové rozhraní*
- **SD** Secure Digital (Memory Card) – *typ paměťové karty*
- **FAT** File Allocation Table – *souborový systém používaný firmou Microsoft*
- **Li-Pol** Lithium-ion Polymer – *druh baterie, vyznačuje se nízkým samovybitím*

VIII. Seznam použitých zdrojů

VIII.1 Články

- [KIR01] KIRKBY, David. *Bitmap format* [online], 2001 [cit. 27. 03. 2011].
Dostupné na: <http://atlc.sourceforge.net/bmp.html>

VIII.2 Softwarové projekty

- [RIE06] RIEGEL, Roland. *MMC/SD/SDHC card library* [online], 2006 [cit. 27. 03. 2011].
Dostupné na: <http://www.roland-riegel.de/sd-reader/>

VIII.3 Dokumentace

- [LIBC] FREE SOFTWARE FOUNDATION. *AVR Libc Home Page* [online], 2008 [cit. 27. 03. 2011]. Dostupné na: <http://www.gnu.org/savannah-checkouts/non-gnu/avr-libc/>
- [V-USB] OBJECTIVE DEVELOPMENT SOFTWARE GMBH. *V-USB: A Firmware-Only USB Driver for Atmel AVR Microcontrollers* [online], 2008 [cit. 27. 03. 2011].
Dostupné na: <http://www.obdev.at/products/vusb/index.html>

VIII.4 Datasheety

- [M128] ATMEL. *Datasheet ATmega128A* [online], 2010 [cit. 27. 03. 2011].
Dostupné na: http://atmel.com/dyn/resources/prod_documents/8151.pdf
- [M644] ATMEL. *Datasheet ATmega644P* [online], 2010 [cit. 27. 03. 2011].
Dostupné na: http://atmel.com/dyn/resources/prod_documents/doc8011.pdf
- [LE33] STMICROELECTRONICS. *Datasheet LE33* [online], 2005 [cit. 27. 03. 2011].
Dostupné na:
http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00000545.pdf
- [MAX] MAXIM. *Datasheet MAX1555* [online], 2003 [cit. 27. 03. 2011].
Dostupné na: <http://datasheets.maxim-ic.com/en/ds/MAX1551-MAX1555.pdf>
- [LCD] *Datasheet XPT24037PQ* [online], [cit. 27. 03. 2011].
Dostupné na: <http://iteadstudio.com/produce/2-4-tft-lcd-screen-module-itdb02/>
- [XPT] SHENZHEN XPTEK TECHNOLOGY CO.,LTD. *Datasheet XPT2046* [online], 2007 [cit. 27. 03. 2011].
Dostupné na: <http://www.xptek.cn/en/upfiles/2007122417224111856.pdf>
- [ILI] ILITEK. *Datasheet ILI9325*, 2007 [cit. 27. 03. 2011]
- [ATC] SIMCOM. *SIM100S AT Commands Set* [cit. 27. 03. 2011]
- [SIM] SIMCOM. *SIM100S Hardware Specification* [cit. 27. 03. 2011]