

Sem vložte zadání Vaší práce.



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
KATEDRA ČÍSLICOVÉHO NÁVRHU



Bakalářská práce

## Návrh uživatelského rozhraní pro vestavný systém

*Ladislav Kudiveis*

Vedoucí práce: Ing. Matěj Bartík

14. května 2015



# Poděkování

Děkuji panu inženýru Matěji Bartíkovi za jeho pomoc, užitečné rady a trpělivost během vedení této bakalářské práce.



# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen Dílo), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 14. května 2015

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2015 Ladislav Kudiveis. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Kudiveis, Ladislav. *Návrh uživatelského rozhraní pro vestavný systém*. Bachelářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.



# Abstrakt

Předmětem této práce je analýza možností OLED displeje řízeného řadičem SSD1306, GPIO řadiče a implementace knihoven zajišťujících jejich ovládání radiobudíkem provozovaném na 8bitovém mikrokontroléru ATtiny85. Součástí práce je také návrh jednoduchého uživatelského rozhraní pro toto zařízení.

**Klíčová slova** OLED, SSD1306, ATtiny85, radiobudík, uživatelské rozhraní

# Abstract

The object of this work is to analyze possibilities of OLED display driven by SSD1306 controller, GPIO controller of input buttons and implementation of their libraries that will be used to control radio alarm clock running on an 8-bit microcontroller ATtiny85. The work also includes design of a simple user interface for this device.

**Keywords** OLED, SSD1306, ATtiny85, radio alarm clock, user interface



# Obsah

Úvod	1
<b>1 Analýza</b>	<b>3</b>
1.1 Základní charakteristika a vlastnosti mikrokontroléru ATtiny85	3
1.2 Komunikační rozhraní I <sup>2</sup> C	6
1.3 Základní údaje o displeji a řadiči SSD1306	8
1.4 Komunikace s řadičem SSD1306	9
1.5 Konfigurace řadiče SSD1306 a výstupu OLED displeje	11
1.6 Spuštění displeje	13
1.7 Uložení obrazových dat a adresace paměti řadiče OLED displeje	13
1.8 Zajištění uživatelských vstupů a výstupů	15
<b>2 Návrh uživatelského prostředí</b>	<b>19</b>
2.1 Návrh písma	19
2.2 Jednotlivé části uživatelského rozhraní	20
<b>3 Realizace</b>	<b>25</b>
3.1 Knihovna gpio.h	25
3.2 Knihovna font.h	26
3.3 Knihovna ssd1306.h	26
<b>4 Testování</b>	<b>29</b>
<b>Závěr</b>	<b>31</b>
<b>Literatura</b>	<b>33</b>
<b>A Seznam použitých zkratk</b>	<b>35</b>
<b>B Obsah příloženého CD</b>	<b>37</b>



# Seznam obrázků

1.1	Mikrokontrolér ATtiny85 . . . . .	3
1.2	Schéma mikrokontroléru ATtiny85 . . . . .	4
1.3	Schéma zpojení mikrokontroléru ATtiny85 pro použití v radiobudíku . . . . .	5
1.4	Zapojení obvodů ke sběrnici I <sup>2</sup> C . . . . .	6
1.5	Nastavení start a stop bitu na sběrnici I <sup>2</sup> C . . . . .	7
1.6	Přenos dat po sběrnici I <sup>2</sup> . . . . .	8
1.7	Komunikace s řadičem SSD1306 po sběrnici I <sup>2</sup> C . . . . .	9
1.8	Ukládání dat v paměti . . . . .	14
1.9	Horizontální adresace paměti . . . . .	14
1.10	Vertikální adresace paměti . . . . .	15
1.11	Adresace paměti po stránkách . . . . .	15
1.12	Blokové schéma GPIO řadiče PCF8574 . . . . .	16
1.13	Schéma obvodu s připojenými tlačítky a LED diodami k GPIO řadičům PCF8574 navržené Ing. Matějem Bartíkem . . . . .	18
2.1	Podoba písma použitá pro potřeby OLED displeje . . . . .	20
2.2	Úsporný režim . . . . .	20
2.3	Hlavní obrazovka . . . . .	21
2.4	Nabídka oblíbených stanic . . . . .	22
2.5	Potvrzení akce . . . . .	22
2.6	Nastavení data a času . . . . .	23
4.1	Propojení displeje osazeného řadičem SSD1306 k Raspberry Pi . . . . .	30



# Úvod

Obor Počítačové inženýrství se v řadách zájemců o studium i stávajících studentů netěší takové přízni, jaké se dostává jiným oborům a jakou by si zasloužil. A to i přesto, že studentům nabízí širokou škálu uplatnění se a vysoký stupeň kreativity. Proto se zrodil nápad navrhnout a sestavit jednoduché vestavné zařízení s funkcí radiobudíku. Toto relativně jednoduché zařízení by bylo prezentováno veřejnosti při příležitosti dne otevřených dveří a podobných událostech a budoucím studentům by prezentovalo možnosti realizace sebe sama.

Hlavní funkcí radiobudíku je především možnost poslechu rozhlasového vysílání a nastavení automatického sepnutí zařízení v daný časový okamžik. Jedná se tedy o obvod reálného času u kterého ale nehrozí žádné katastrofické následky vlivem nedodržení časových závislostí. Celé zařízení by mělo sestávat z několika základních součástek. Vestavný systém je plánováno provozovat na 8bitovém mikrokontroléru ATtiny85 společnosti Atmel. Zařízení by vykonávalo, který by prostřednictvím sériové sběrnice komunikoval s dalšími obvody, jako je FM tuner zajišťující příjem pozemního radiového vysílání, OLED displej poskytující posluchači informace o stavu zařízení, nebo třeba čidlo snímající okolní teplotu. Možných rozšíření systému by bylo nepřeberné množství. Třeba pro budoucí studenty oboru.

Celý projekt byl rozdělen na několik ucelených částí. Každá z nich řešená v rámci samostatné bakalářské práce. V této práci je kladeno za cíl seznámit se s hardwarovým modulem obsahujícím OLED displej a jeho řadičem SSD1306. Důkladně prostudovat jeho dokumentaci, zjistit veškeré možnosti nastavení a řízení a navrhnout a realizovat knihovnu zajišťující plnohodnotné, ale přitom nijak náročné ovládání displeje. Další částí práce je prozkoumat možnosti ovládání radiobudíku přes vstupy GPIO řadiče a navrhnutí přívětivého uživatelského rozhraní s možností ovládat veškeré funkce zařízení.





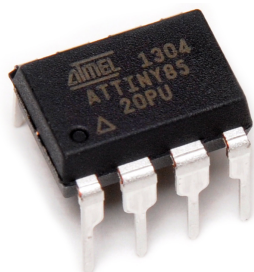
---

# Analýza

Analýza se podrobně zabývá možnostmi všech částí vestavného systému a návrhem uživatelského rozhraní z pohledu hardwarových prostředků.

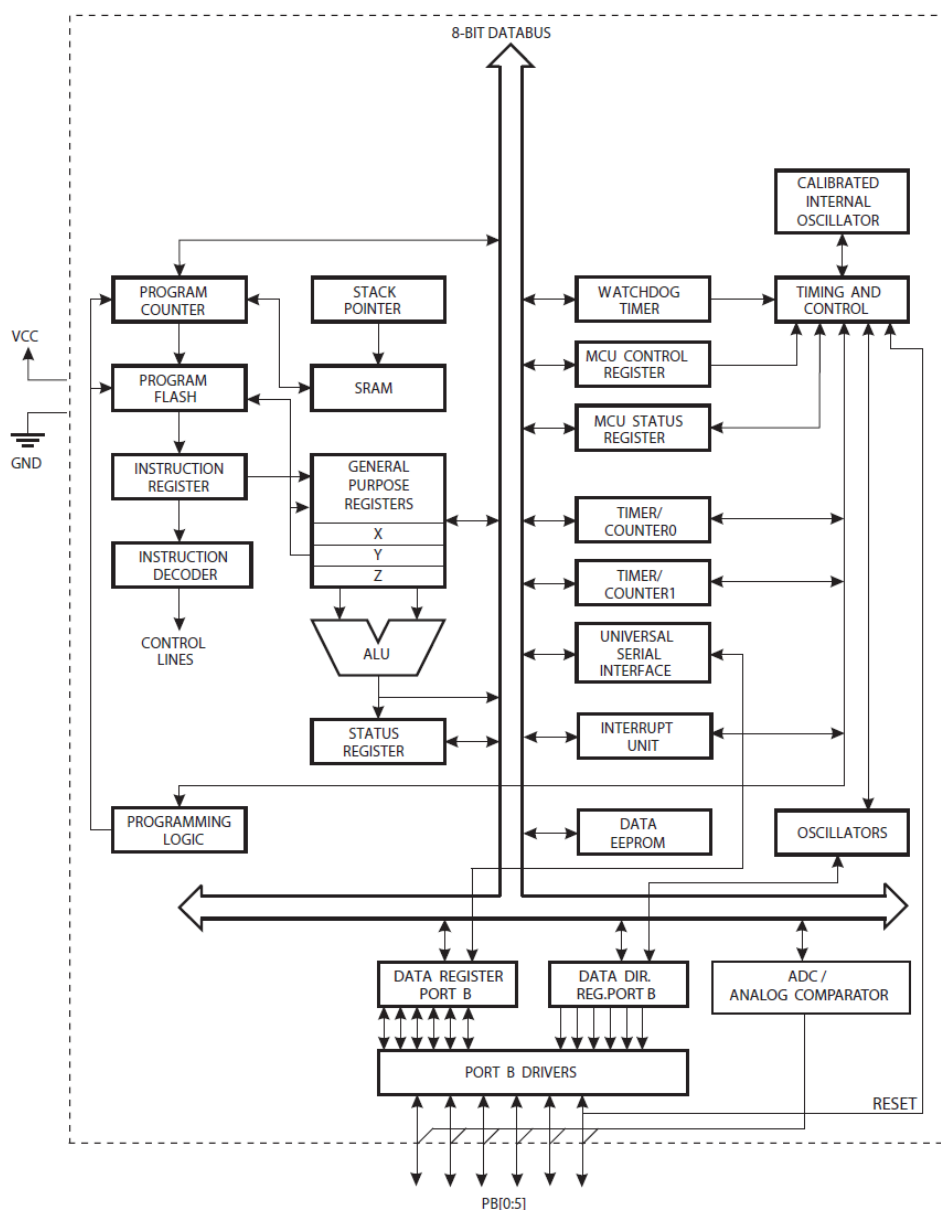
## 1.1 Zakladní charakteristika a vlastnosti mikrokontroléru ATtiny85

ATtiny85 1.2 je 8bitový mikrokontrolér založený na ARM architektuře disponující 8kB ISP flash pamětí, 512bajtovou ISP EEPROM pamětí a 512bajtovou SRAM pamětí, šesti programovatelnými vstupně-výstupními piny ??, třicetidvěma obecně použitelnými registry, obsluhou vnitřních a vnějších přerušení, 10bitovým AD převodníkem, dvěma 8bitovými čítači s předděličkou a pulzně šířkovou modulací a dalšími vestavěnými funkcemi. Maximální hodinová frekvence mikrokontroléru je 20MHz a při této rychlosti se mikrokontrolér blíží průchodnosti průchodnosti až 20 milionů instrukcí vykonaných za vteřinu.



Obrázek 1.1: Mikrokontrolér ATtiny85

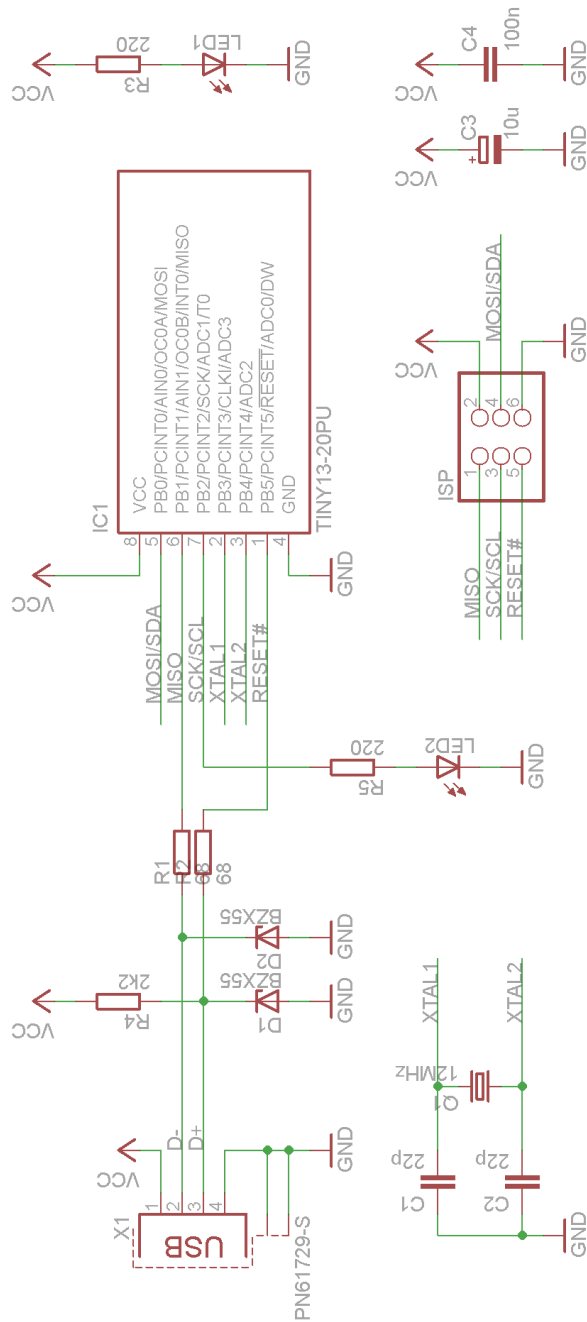
## 1. ANALÝZA



Obrázek 1.2: Schéma mikrokontroléru ATtiny85 [1]

### 1.1.1 Výhody a nevýhody použití mikrokontroléru ATtiny85

ATtiny je ve všech směrech pozoruhodný mikrokontrolér disponující celou řadou vestavěných funkcí. Jeho hlavní předností je, že i přes svou malou velikost je velice výkonný. Na druhou stranu nedisponuje příliš velkou interní pamětí a pro náročnější vestavné systémy je potřeba doplnit jej o externí paměťový modul. Pro řadu aplikací nemusí být také dostatečný počet programovatel-



Obrázek 1.3: Schéma zpojení mikrokontroléru ATtiny85 navržené Ing. Matějem Bartíkem pro použití v radiobudíku s připojenou I<sup>2</sup>C sběrnici, USB sběrnici a připojeným externím krystalovým oscilátorem

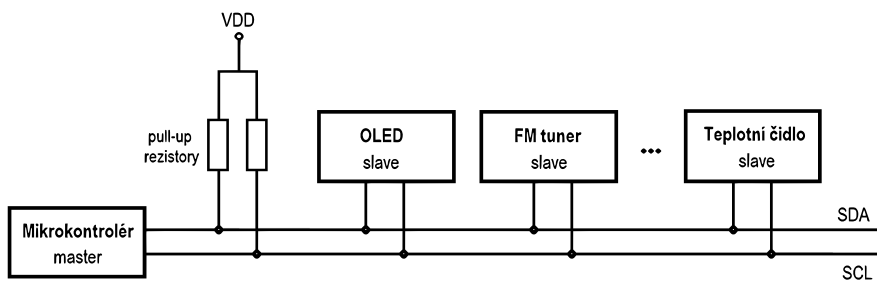
ných IO pinů. V našem případě jsou již všechny zabrány, a není tak možnost pomýšlet nad dalšími rozšířeními. Obrázek 1.3

## 1.2 Komunikační rozhraní I<sup>2</sup>C

I<sup>2</sup>C [2] je sériová datová sběrnice sloužící ke komunikaci a přenos dat mezi jednotlivými integrovanými obvody, většinou v rámci jednoho zařízení. Jedná se o široce podporované rozhraní využívané především mikrokontroléry, sériovými pamětmi, LCD, OLED, PLED displeji, různými snímači a mnohými dalšími obvody. K obousměrné komunikaci je využito pouze dvou vodičů. Po jednom z vodičů je přenášen hodinový signál (SCL), po druhém pak sériová data (SDA). Oba vodiče jsou ve stavu nečinnosti nastaveny v logické 1. Toho je docíleno tak, že každý z vodičů je připojen přes pull-up rezistor k napájení. Obrázek 1.4

K jedné sběrnici může být najednou připojeno v základní verzi až 128 zařízení. Každý z čipů však musí mít jedinečnou adresu. Z tohoto důvodu je použití stejných obvodů mnohdy limitováno počtem nastavitelných bitů adresy zařízení. Některá zařízení mají dokonce pevně nastavenou adresu. V takovém případě může být ke sběrnici připojeno pouze jediné takové zařízení. Obvody připojené ke sběrnici se dělí na takzvané master a slave, neboli řídicí a podřízená zařízení. Slave obvodů bývá připojeno ke sběrnici více, zatímco master zařízení je obvykle pouze jedno. Komunikaci vždy zahajuje master zařízení, které je rovněž zdrojem hodinového signálu. Slave zařízení odposlouchávají veškeré dění na sběrnici, až do chvíle, kdy s nimi master naváže komunikaci. Slave zařízení nemůže nikdy inicializovat přenos.

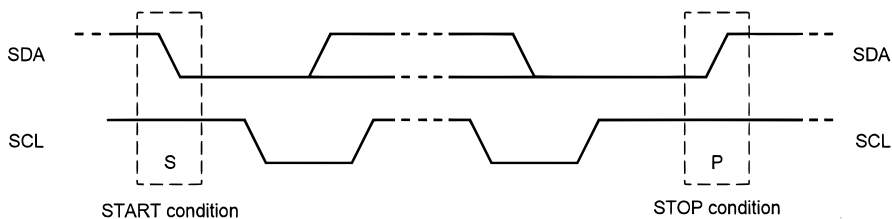
Standardní hodinová frekvence I<sup>2</sup>C sběrnice je 100kHz, ale existují i rychlejší verze s frekvencí 400kHz (fast mode) a 3,4MHz (high-speed mode). Ne však všechna zařízení podporují vyšší rychlosti. Rychlost komunikace musí být přizpůsobena nejpomalejšímu z obvodů.



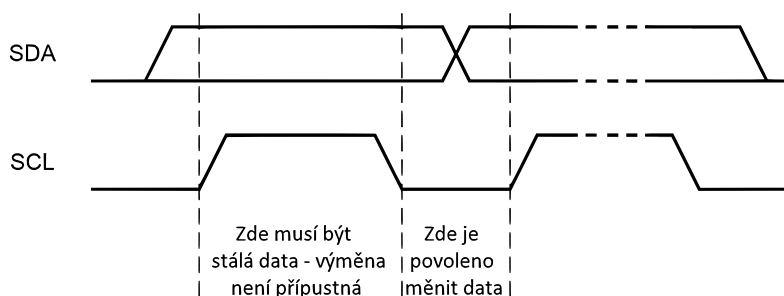
Obrázek 1.4: Zapojení obvodů ke sběrnici I<sup>2</sup>C

### 1.2.1 Přenos dat

- Ve stavu klidu na sběrnici neprobíhá žádný přenos ani hodinový signál a na obou vodičích je trvale vystavena logická jednička i ve chvíli, kdy je zařízení ve stavu vysoké impedance. Toho je docíleno připojením vodičů přes pull-up rezistory k napájení.
- Zahájení přenosu inicializuje řídicí zařízení vysláním start bitu, který je vygenerován změnou logického stavu datového vodiče z logické jedničky do nuly, zatímco vodič hodinového signálu je stále nastaven na logickou jedničku jak je tomu vidět na obrázku 1.5. Od té doby je pravidelně generován hodinový signál. Pouhé nastavení start bitu by ovšem nestačilo. Před přenosem dat je potřeba určit, které ze slave zařízení má s řídicím obvodem komunikovat a jakým směrem bude přenos probíhat. Proto je po nastavení start bitu vysláno sedm bitů adresy cílového zařízení a jeden bit určující směr přenosu (0 pro zápis, 1 pro čtení) načež by mělo master zařízení obdržet potvrzovací ACK bit.
- Přenos dat probíhá po jednotlivých bitech. Každý bit je přenesen při jednom hodinovém pulzu. Během doby, kdy je vodič hodinového signálu v logické jedničce, nesmí docházet na datovém vodiči ke změně logické hodnoty. Hodnota se mění v době mezi jednotlivými hodinovými pulzy, viz obrázek 1.6.
- Poté je příjemcem vygenerován ACK bit. Pokud je ACK nulový, znamená to, že přenos proběhl v pořádku a zařízení je připraveno pokračovat v komunikaci. Pokud přenos selhal, je ACK nastaven na 1.
- Nastavením vodičů SDA a SCL do logické jedničky dojde k vygenerování stop bitu, který přenos ukončuje. Obrázek 1.5



Obrázek 1.5: Nastavení start a stop bitu na sběrnici I<sup>2</sup>C [3]

Obrázek 1.6: Přenos dat po sběrnici I<sup>2</sup>C [3]

## 1.3 Základní údaje o displeji a řadiči SSD1306

### 1.3.1 OLED displej

O OLED displejích jsou často považovány za technologii budoucnosti, která je ovšem dostupná již dnes. OLED displeje mohou být velice tenké, mají dlouhou životnost, rychlou odezvu a nevydávají ultrafialové ani infračervené záření. Díky využitým materiálům může být OLED panel i částečně ohebný. OLED displej je složen z několika polovodiivých vrstev z organických materiálů naskládáných na sobě mezi dvěma elektrodami, přičemž alespoň jedna z nich je průhledná. Stejně jako u LCD displejů je každý pixel složený z tzv. subpixelů. Každý subpixel vyzařuje světlo jiné barvy – červenou, zelenou nebo modrou. Výsledná barva pixelu je potom utvářena vyzařováním všech tří barev v různém poměru intenzity. Narozdíl od LCD panelu, kde je zdroj světla společný pro všechny pixely, vyzařuje každý pixel světlo sám a je tedy možné docílit daleko věrohodnějšího podání barev, především pak černé.

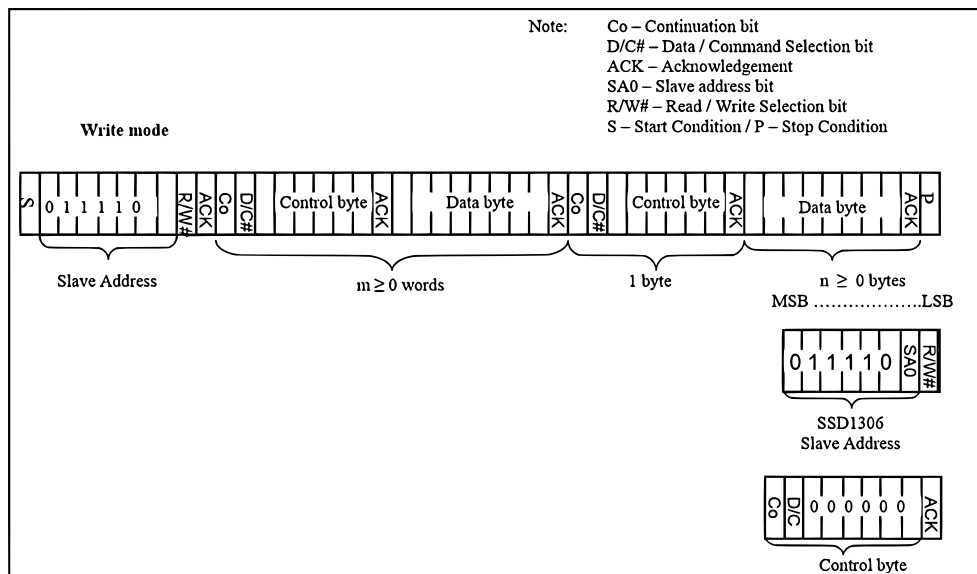
### 1.3.2 Řadič SSD1306

SSD1306 [4] je jednočipový CMOS řadič OLED/PLED displejů s rozlišením  $128 \times 64$  obrazových bodů. Obraz je ukládán do vnitřní paměti SRAM o velikosti  $8 \times 128$  bajtů. Výstup displeje je monochromatický, nelze tedy ovlivnit jeho barvu <sup>1</sup>. Proto k ovládní každého bodu stačí pouze jeden jediný bit uložený ve vnitřní paměti. Data a příkazy jsou řadiči posílány po sériové sběrnici I<sup>2</sup>C. Obvod je napájena napětím od 1,65 do 3,3 voltů. SSD1306 je osazen vlastním oscilátorem a regulátorem jasu displeje s 256 nastavitelnými kroky.

<sup>1</sup>To že je obraz monochromatický neznamená, že výstup musí být pouze v jediné barvě. V našem případě je displej rozdělen na dvě oddělené části a každá vyzařuje odlišnou barvu. Menší část o výšce 16 bodů je provedena v oranžové barvě, zatímco větší část vyzařuje světlo azurové barvy

## 1.4 Komunikace s řadičem SSD1306

Řadič komunikuje s vnějším světem přes rozhraní I<sup>2</sup>C. Data odeslaná řadiči jsou interpretována dvěma různými způsoby – jako příkazy, nebo jako obrazová data. Příkazy nastavují chování řadiče, potažmo výstup displeje. O tom, zda-li se jedná o příkaz nebo data, rozhoduje řídicí byte následující po adrese čipu. Data i příkazy mohou být odesílány v sériích a není tedy potřeba pokaždé znovu vystavovat na sběrnici adresu ani kontrolní bajt. Viz obrázek 1.7



Obrázek 1.7: Komunikace s řadičem SSD1306 po sběrnici I<sup>2</sup>C [4]

- Přenos je zahájen vysláním start bitu, viz kapitola 1.2. Za start bitem následuje 7 bitů adresy řadiče. Adresa je nastavena na hodnotu 3Ch, ale lze ji změnit manuálním připojením posledního bitu adresy k logické jedničce na hodnotu 3Dh. Další bit v pořadí určuje, zda bude probíhat zápis nebo čtení. Pro zápis nastavíme R/W# bitu na logickou 0. Po odeslání posledního bitu je řadičem vygenerován ACK bit.
- Poté je zapotřebý odeslat řídicí bajt a určit tak druh přenášených dat. V případě přenosu příkazů je odeslán kontrolní bajt 00h, po kterém následuje jeden nebo více příkazů. Pokud budou přenášena obrazová data, je odeslán řídicí bajt 40h a za ním následují data. Platí, že po každém odeslaném bajtu je přijat ACK bit. Po odeslání posledního bajtu je přenos ukončen posláním stop bitu.

## 1. ANALÝZA

Tabulka 1.1: Tabulka příkazů řadiče SSD1306

Příkaz	Délka v bajtech	Popis
<b>Základní příkazy</b>		
81h A[7:0]	2	Nastavení jasů displeje v rozmezí 0–255
A4h / A5h	1	Povolení / ignorování obsahu RAM paměti
A6h / A7h	1	Normální / invertovaný výstup RAM paměti
A Eh / AFh	1	Vypnutí / zapnutí displeje
<b>Nastavení adresování</b>		
00h–0Fh	1	Nastavení spodních čtyř bitů adresy sloupce
10h–1Fh	1	Nastavení horních čtyř bitů adresy sloupce
20h A[1:0]	2	Nastavení způsobu adresace paměti A[1:0]=00b, horizontální adresace A[1:0]=01b, vertikální adresace A[1:0]=10b, adresace po stránkách
21h A[6:0] B[6:0]	3	Nastavení počáteční a koncové adresy sloupce A[6:0]: index počátku řádku v rozmezí 0–127 B[6:0]: index konce řádku v rozmezí 0–127
22h A[2:0] B[2:0]	3	Nastavení počáteční a koncové stránky A[2:0]: index počáteční stránky v rozmezí 0–7 B[2:0]: index koncové stránky v rozmezí 0–7
B0h–B7h	1	Nastavení indexu stránky
<b>Hardwarová konfigurace</b>		
40h–7Fh	1	Nastavení indexu řádku paměti v rozsahu 0–63, od kterého se má indexovat
A0h / A1h	1	Zakáže / povolí vertikální otočení
A8h A[5:0]	2	Nastavení multiplexeru na hodnotu 0Fh–3Fh
C0h / C8h	1	Zakáže / povolí horizontální otočení
D3h A[5:0]	2	Nastavení vertikálního posuvu v rozmezí 0–63
DAh A[5:4]	2	Konfigurace COM pinů
<b>Nastavující časování a řízení</b>		
D5h A[7:0]	2	A[3:0]: clock divide ratio A[7:4]: nastavení hodinové frekvence oscilátoru
8Dh 10h / 14h	2	Vypnutí / zapnutí charge pump regulátoru



## 1.5 Konfigurace řadiče SSD1306 a výstupu OLED displeje

Ovládání OLED displeje prostřednictvím řadiče je prováděno sadou jednoduchých příkazů. Pro odeslání příkazu je potřeba navázat nové spojení s řadičem a vyslat řídicí bajt určující přenos příkazů, viz kapitola 1.4. Příkazy jsou až na výjimky obvykle jedno, dvou až třibajtové a lze je posílat v sérii za sebou. Příkazy lze rozdělit do pěti kategorií podle jejich funkce. Jsou to základní příkazy, příkazy umožňující automatický posuv obrazu<sup>2</sup> neboli srollování, příkazy ovlivňující adresaci vnitřní paměti, příkazy nastavující hardwarovou konfiguraci řadiče a příkazy nastavující časování a řízení řadiče. Nejčastěji využívané příkazy naleznete v tabulce příkazů<sup>3</sup> 1.1 Podrobnější vysvětlení některých příkazů lze nalézt níže.

### 1.5.1 Nastavení spodních čtyř bitů adresy sloupce (00h–0Fh)

Pomocí tohoto příkazu můžeme nastavit spodní 4 bity aktuálního indexu sloupce.

### 1.5.2 Nastavení horních čtyř bitů adresy sloupce (10h–1Fh)

Pomocí tohoto příkazu můžeme nastavit horní 4 bity aktuálního indexu sloupce.

### 1.5.3 Nastavení způsobu adresace pamět (20h)

Tímto třibajtovým příkazem lze nastavit jeden ze tří způsobů přístupu a zápisu do paměti. Možné přístupy k adresaci:

- horizontální adresace paměti
- vertikální adresace paměti
- adresace po stránkách

Více informací lze nalézt v kapitole 1.7

### 1.5.4 Nastavení počáteční a koncové adresy sloupce (21h)

Tento třibajtový příkaz umožňuje nastavit počáteční a koncový index sloupce. Při zápisu nebo čtení dat poté dojde při přetečení čítače sloupce přes námi nastavenou koncovou hodnotu k nastavení indexu na počáteční adresu. Výchozí nastavení je 0–127. Tento příkaz lze použít poze v případě horizontální nebo vertikální adresace paměti.

---

<sup>2</sup>V této práci není tato skupina příkazů vysvětlena a pro bližší seznámení se s nimi je potřeba nahlédnout do produktového manuálu řadiče SSD1306.

<sup>3</sup>Kompletní tabulku lze nalézt v dokumentaci řadiče.

### 1.5.5 Nastavení počáteční a koncové adresy stránky (22h)

Tento třibajtový příkaz umožňuje nastavit počáteční a koncový index stránky. Při zápisu nebo čtení dat poté dojde při přetečení čítače stránky přes námi nastavenou koncovou hodnotu k nastavení indexu na počáteční adresu. Výchozí nastavení je 0–7. Tento příkaz lze použít poze v případě horizontální nebo vertikální adresace paměti.

### 1.5.6 Nastavení počátečního indexu řádku paměti (40h–7Fh)

Tento příkaz umožňuje nastavit počáteční řádek paměti v rozmezí 0–63, od kterého bude indexováno.

### 1.5.7 Nastavení jasu displeje (81h)

Tento dvoubajtový příkaz umožňuje nastavit jas displeje na jednu z 256 hodnot. Čím vyšší je jas, tím vyšší je odebíraný proud.

### 1.5.8 Nastavení vertikálního otočení (A0h/A1h)

Tento příkaz umožňuj vertikálně otočit zobrazení displeje. V kombinaci s horizontálním otočením lze dosáhnout pootočení displeje o 180°.

### 1.5.9 Nastavení indexu stránky (B0h–B7h)

Tento příkaz umožňuje nastavit aktuální index stránky. Tento příkaz je určen pouze pro způsob adresace paměti po stránkách.

### 1.5.10 Nastavení vertikálního otočení (C0h/C8h)

Tento příkaz umožňuje horizontálně otočit zobrazení displeje. V kombinaci s vertikálním otočením lze dosáhnout pootočení displeje o 180°.

### 1.5.11 Nastavení vertikálního posuvu (D3h)

Tento dvoubajtový příkaz umožňuje přemapovat začátek paměti RAM na jiný řádek displeje. Výstup je pak posunutý o 0–63 řádek.

## 1.6 Spuštění displeje

Displej nemůžeme jednoduše používat ihned po připojení. Před prvním použitím je potřeba správně nastavit registry zadáním inicializační sekvence příkazů. Pořadí příkazů je přesně specifikováno v dokumentaci a mělo by být dodrženo, aby funkce řadiče byla správná. Některé hodnoty závisí na potřebách konkrétního systému. Pro naše použití bude nastavení vypadat následovně.

1. Příkazem A8h nastavíme hodnotu multiplexeru na 3Fh
2. Příkazem D3h nastavíme posunutí řádků na 0
3. Příkazem 40h nastavíme, aby se první řádek paměti zobrazoval na prvním řádku displeje
4. Příkazem A1h nastavíme horizontální otočení, abychom zabránili zrcadlovému otočení
5. Příkazem C8h nastavíme vertikální otočení, abychom menší, barevně odlišenou část displeje měli nahoře
6. Příkazem DAh nakonfigurujeme nastavení COM pinů na hodnotu 12h
7. Příkazem 81h nastavíme kontrast displeje (doporučená hodnota je 127)
8. Příkazem A4h povolíme výstup z RAM na OLED displej
9. Příkazem A6h nastavíme normální výstup (logická jednička znamená že pixel vyzařuje světlo)
10. Příkazem D5h nakonfigurujeme oscilátor na hodnotu 80h
11. Nastavením registru 8Dh na hodnotu 14h povolíme charger pump regulátor
12. Příkazem AFh zapneme displej

## 1.7 Uložení obrazových dat a adresace paměti řadiče OLED displeje

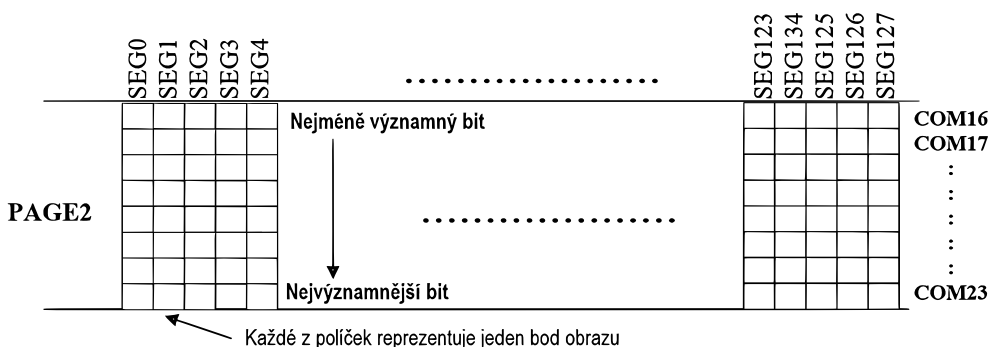
Obrazu je udržována ve vnitřní paměti, kde každý bit reprezentuje stav konkrétního bodu na displeji. Paměť je rozdělena na osm stránek označených jako PAGE0–PAGE7. Každá stránka sestává ze 128 sloupců SEG0–SEG127, kde každý sloupec obsahuje 8 řádků.

Ukládání nebo čtení dat probíhá po bajtech. Při zápisu je tedy vždy nastaveno 8 bodů najednou. Při zápisu nebo čtení je vybrán jeden sloupeček z aktuální stránky. Uložení ve sloupci je shora dolů od nejméně významného

## 1. ANALÝZA

bitu po nejvýznamnější bit, jak je tomu vidět na obrázku 1.8.

Bajty jsou zapisovány do paměti sekvenčně a posun indexů záleží na zvoleném způsobu adresace paměti. Nastavení adresace paměti lze provést příkazem 20h. Paměť lze adresovat třemi způsoby. Zapicovat nebo číst lze jednu stránku stále dokola, dokud uživatel adresu stránky nezmění sám. K tomu slouží adresace po stránkách. Dále je možné ukládat data po řádcích - horizontální adresace, nebo po sloupcích - vertikální adresace. O posun po řádcích a sloupcích se stará sám řadič, který podle zvoleného způsobu adresace indexy inkrementuje.



Obrázek 1.8: Ukládání dat v paměti [4]

### 1.7.1 Horizontální adresace paměti (00h)

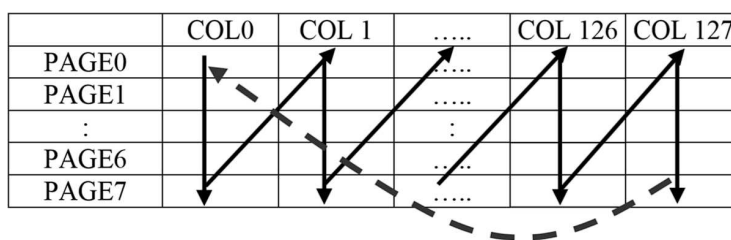
Při zápisu nebo čtení v tomto režimu adresace je po každém zapsaném nebo přečteném bajtu automaticky inkrementován index sloupce. Ve chvíli, kdy číslo sloupce přesáhne index konce řádku, je toto číslo nastaveno zpět na nastavenou počáteční hodnotu a index stránky je navýšen. Pokud i index stránky přesáhne za poslední stránku, je taktéž nastaven na počáteční hodnotu a data se čtou/přepisují od začátku. Viz obrázek 1.9. Počáteční a koncový index sloupce a stránky lze nastavit příkazy 21h a 22h.



Obrázek 1.9: Posuv indexů v horizontálním režimu adresace paměti [4]

### 1.7.2 Vertikální adresace paměti (01h)

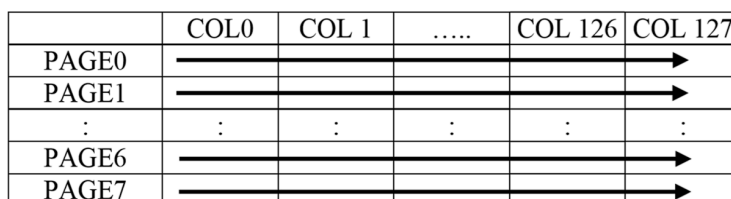
V tomto režimu je při každém zápisu nebo přečtení jednoho bajtu automaticky posunut index stránky o hodnotu 1. Ve chvíli, kdy číslo stránky přesáhne index koncové stránky je toto číslo nastaveno zpět na nastavenou počáteční hodnotu a index sloupce je zvýšen. Při překročení indexu koncového sloupce jsou oba indexy nastaveny na počáteční hodnoty, viz obrázek 1.10. Počáteční a koncový index sloupce a stránky lze nastavit příkazy 21h a 22h.



Obrázek 1.10: Posuv indexů ve vertikálním režimu adresace paměti [4]

### 1.7.3 Adresace po stránkách (02h)

V tomto režimu je vždy zapisována pouze jediná stránka, takže při dosažení konce řádku je index sloupce nastaven zpět na počáteční hodnotu, číslo stránky zůstává stejné. Pro přechod na jinou stránku je potřeba nastavit index stránky manuálně příkazem B0h–B7h. Obrázek 1.11.



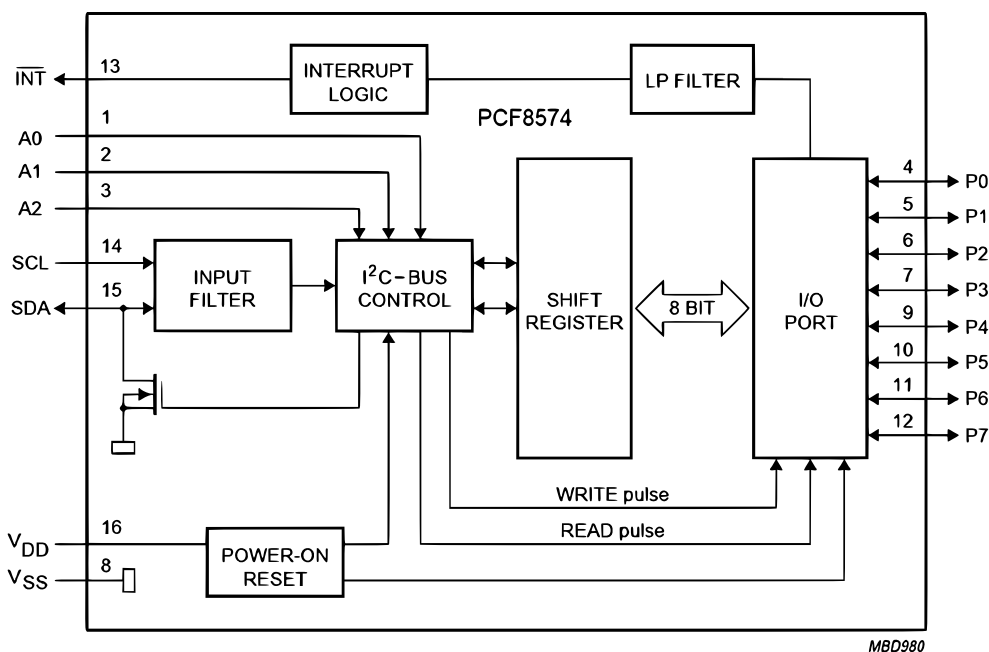
Obrázek 1.11: Adresace paměti po stránkách [4]

## 1.8 Zajištění uživatelských vstupů a výstupů

Cílová platforma ATtiny85 nedisponuje dostatečným počtem vstupně-výstupních pinů. Řešením tohoto problému může být použití mikrokontroléru ze stejné rodiny, ale s větším počtem pinů. Druhým způsobem je připojení I<sup>2</sup>C expandéru, který zprostředkovává další piny přes sběrnici I<sup>2</sup>C.

### 1.8.1 I<sup>2</sup>C expandér PCF8574

K ovládání radiobudíku 1.13 máme k dispozici obvod osazený dvěma 8bitovými GPIO řadiči PCF8574. K jednomu z řadičů je připojeno 8 LED diod, k druhému pak 5 tlačítek a jeden rotary switch. Oba řadiče jsou připojeny ke sběrnici I<sup>2</sup>C.



Obrázek 1.12: Blokové schéma GPIO řadiče PCF8574 [5]

### 1.8.2 Výstup LED diod

Pro rozsvícení konkrétních LED stačí na příslušný GPIO řadič zapsat hodnotu, kde pro každou rozsvícenou LED bude bit v logické jedničce. Zápis dat probíhá standardním způsobem pro I<sup>2</sup>C sběrnici. Nejprve je odeslána adresa. V tomto případě jsou adresní piny GPIO řadiče LED diod nastálo připojeny k zemi a jeho adresa je 20h. R/W bit bude nastaven do logické nuly pro zápis dat. Poté je ji možné jednoduše zapsat požadovanou hodnotu. Přečtením hodnoty GPIO řadiče je možné získat informace o aktuálně rozsvícených LED diodách.

### 1.8.3 Detekce stisknutých tlačítek

Stav tlačítek je možné jednoduše zjistit přečtením hodnoty GPIO řadiče s adresou 21h. Stisk konkrétního tlačítka zjistíme pouhým vymaskováním daného bitu, kdy při stisknutém tlačítku je na dané pozici logická jednička. Vzhledem

k nedokonalostem kontaktů je vždy vhodné tlačítka ošetřit proti zákmitům, aby nedošlo k několikanásobnému vyhodnocení stisku tlačítka.

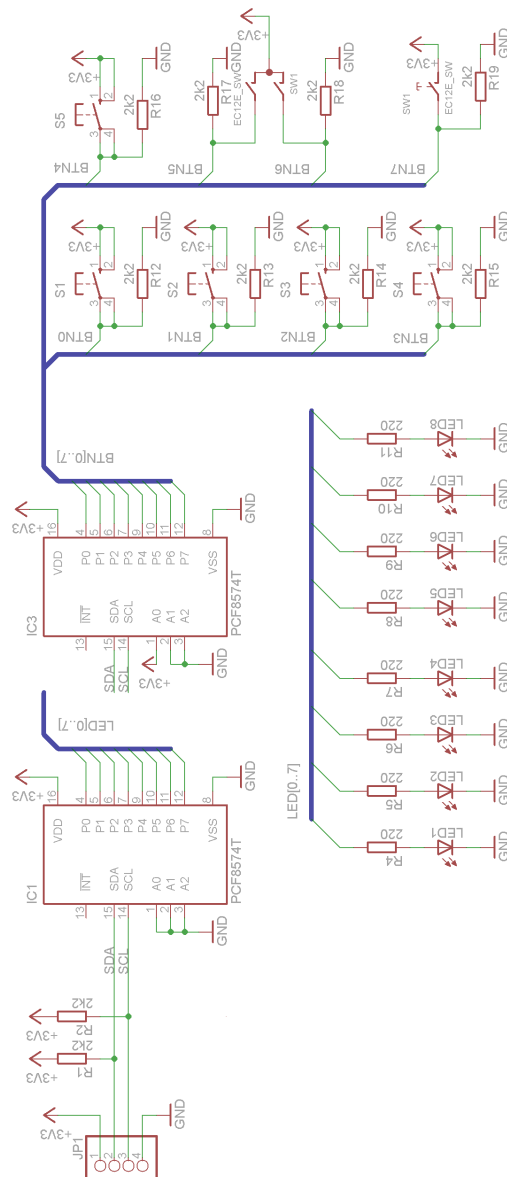
#### 1.8.4 Vyhodnocení směru otáčení rotary switche

S osazeným rotary switchem lze donekonečna otáčet a s každým pootočením lze zjistit pohyb a směr otáčení. Rotary switch nastavuje druhý a třetí nejvýznamnější bit GPIO řadiče a generuje následující řady v binární podobě (Grayův kód):

- 00 10 11 01 00 10 11 01 ... při otáčení po směru hodinových ručiček
- 00 01 11 10 00 01 11 10 ... při otáčení opačným směrem.

## 1. ANALÝZA

---



Obrázek 1.13: Schéma obvodu s připojeními tlačítka a LED diodami k GPIO radičům PCF8574 navržené Ing. Matějem Bartíkem



## Návrh uživatelského prostředí

Návrh uživatelského prostředí [6] není nikterak jednoduchá záležitost. Je nutné přihlídnout na všechny aspekty vestavného systému. Je nutné promyslet veškeré možnosti a myslet i na možná rozšíření systému do budoucnosti. Ovládání UI musí být přehledné, pohyb v něm musí být snadný a nijak zdoluhavý. Uživatelé bývají často na něco zvyklí a neradi se učí novým věcem. Proto je mnohdy potřeba přihlídnout ke stávajícím řešením a nechat se inspirovat, byť daný systém nemusí být nejideálnější. Uživatelské prostředí bude poskytovat uživateli radiobudíku možnosti zařízení ovládat a nastavovat. Veškeré potřebné informace bude mít zobrazeny na displeji. K ovládání radiobudíku bude sloužit 5 tlačítek. Rotary switch lze příhodně využít k ladění frekvence. Je potřeba navrhnout ergonomii ovládání, aby bylo využito všech možností, a zároveň nebylo ovládání nijak složité.

### 2.1 Návrh písma

Aby bylo vůbec možné na displej něco vypsát, bylo nejprve potřeba navrhnout podobu písma. Písmo bylo navrženo tak, aby bylo dobře čitelné, znaky byly dobře rozpoznatelné a zároveň aby jejich způsob zapsání do obrazové paměti byl co nejjednodušší. S ohlédnutím na možnosti řadiče bylo základní písmo navrženo na výšku řádku 8 bodů včetně meziřádkové mezery a 6 bodů na šířku včetně meziznakové mezery. Znak lze poté relativně jednoduše zvětšit na výšku řádku z řady 16, 32, 64 bodů. Tyto rozměry vycházejí z vlastností řadiče SSD1306 a zároveň z pozorování v reálném nasazení. Jednotlivé bajty obrazu znaku jsou skládány za sebou po sloupcích od nejvyššího po nejnižší. Pro zápis znaku do paměti pak stačí jednoduše nastavit vertikální režim adresace paměti 1.10.

Aby bylo dosaženo plné kompatibility, je potřeba přenastavit šířku displeje ze 128 na 120 bodů. Toho lze docílit konfigurací řadiče SSD1306. Při velikosti obrazu 120 bodů lze na displeji zobrazit až 160 znaků základní výšky 8 bodů v osmi řádcích, 40 znaků výšky 16 bodů ve čtyřech řádcích, nebo 10 znaků

## 2. NÁVRH UŽIVATELSKÉHO PROSTŘEDÍ

---

výšky 32 bodů ve dvou řádcích. Se znaky větší velikosti se nemá cenu již zabývat, neboť monost zobrazit nějakou srozumitelnou informaci zaniká.

Všechny potřebné funkce k převodu znaků na jeho obrazovou podobu jsou implementovány v knihovně `font.h`.

```
! " # $ % & ' ( ) * + , - . /
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [ \ ] ^ _
` a b c d e f g h i j k l m n o
p q r s t u v w x y z { | } ~
```

Obrázek 2.1: Podoba písma použitá pro potřeby OLED displeje

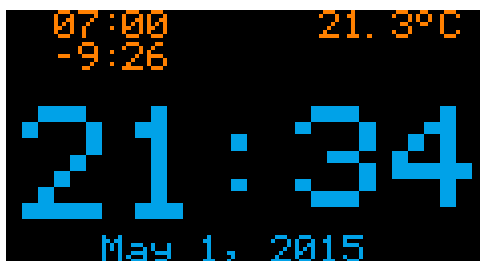
## 2.2 Jednotlivé části uživatelského rozhraní

### 2.2.1 Úsporný režim

Toto je implicitní režim, do kterého bude radiobudík spuštěn při připojení napájení. Na displeji je v hlavní části zobrazen velkým písmem aktuální čas a pod ním datum a rok. V horní liště je vlevo zobrazen nastavený čas budíku a pod ním zbývající čas do spuštění. Vpravo je zobrazen údaj o okolní teplotě. Pokud není budík nastaven, je toto místo prázdné.

**Podržení prostředního tlačítka** zapnutí radiobudíku

**Otáčení rotary switchem** Zvýšení/snížení jasu displeje



Obrázek 2.2: Úsporný režim

### 2.2.2 Hlavní obrazovka

Do tohoto režimu se uživatel dostane manuálním spuštěním, nebo automatickým spuštěním. Obrazovka poskytuje uživateli informaci o prave naladene frekvenci a případný název poslouchané stanice. V horní liště jsou vlevo zobrazeny informace o síle signálu a hlasitosti poslechu, uprostřed je zobrazena teplota okolí a vpravo se nachází hodiny.

**Podržení prostředního tlačítka** vypnutí radiobudíku

**Prostřední tlačítko** vyvolání nabídky oblíbených radiostanic

**Horní tlačítko** zvýšení hlasitosti

**Dolní tlačítko** snížení hlasitosti

**Levé tlačítko** nabídka menu

**Pravé tlačítko** nabídka oblíbené

**Podržení levého tlačítka**

automatické vyhledání směrem k nižším frekvencím

**Podržení pravého tlačítka**

automatické vyhledání směrem k vyšším frekvencím

**Otáčení rotary switchem** manuální ladění stanic



Obrázek 2.3: Hlavní obrazovka

### 2.2.3 Nabídka oblíbených stanic

Do této nabídky se uživatel dostane stisknutím prostředního tlačítka na hlavní obrazovce. Zde může uživatel projíždět všechny své uložené stanice, může některou smazat, nebo si jednu z nich vybrat a spustit její přehrávání.

**Prostřední tlačítko** vybrat stanici a opustit nabídku

**Horní tlačítko** přejít o stanici výše

**Dolní tlačítko** přejít o stanici níže

**Levé tlačítko** opuštění nabídky bez výběru stanice

**Pravé tlačítko** smazání uložené stanice (nutné potvrdit)

**Otáčení rotary switchem** procházení nabídky



Obrázek 2.4: Nabídka oblíbených stanic



Obrázek 2.5: Potvrzení akce

### 2.2.4 Nabídka menu

V této nabídce bude uživatel moci nastavovat radiobudík, jako nastavení kontrastu displeje, nastavení data a času, nastavení budíku apod.

**Prostřední tlačítko** vybrat položku menu / potvrdit

**Horní tlačítko** přejít o nabídku výše / zvýšit hodnotu

**Dolní tlačítko** přejít o nabídku níže / snížit hodnotu

**Levé tlačítko** opuštění nabídky menu / posunutí doleva

**Pravé tlačítko** vybrat položku menu / posunutí doprava

**Otáčení rotary switchem** procházení nabídky / změna hodnoty



Obrázek 2.6: Nastavení data a času



---

## Realizace

Realizace a implementace probíhala v podobě vývoje knihoven pro ovládání periférií. Pro řadič SSD1306 existuje mnoho kompletních řešení. Z důvodu co největší úspory paměti a výpočetních prostředků byli naprogramovány vlastní knihovny v jazyce C. Knihovny samozřejmě poskytují pouze omezené vlastnosti a nejsou tak komplexní jako jiná řešení. Nicméně pro potřeby vestavného systému jakým by měl náš radiobudík být, by mělo být dané řešení plně dostačující.

### 3.1 Knihovna gpio.h

Knihovna zprostředkováváající informace od GPIO řadičů. Pomocí jejích funkcí může uživatel rozsvítet nebo zhasínat LED diody, nebo zjišťovat stav tlačítek a rattary switche.

```
void setLEDs(uint8_t);
```

Tato funkce umožňuje nastavit LED diody. Pro zapnutí konkrétní diody musí být bit na dané pozici nastaven na logickou jedničku.

```
uint8_t getRotaryDirection(void);
```

Tato funkce vrací informaci, zda a jakým směrem bylo pootočeno rotary switch. Funkce vrací hodnotu 0, pokud nedošlo k pootočení. Hodnotu 1 v případě, že bylo pootočeno protisměru hodinových ručiček a hodnotu 2, pokud bylo pootočeno ve směru hodinových ručiček.

```
uint8_t getPressedButtons(void);
```

Tato funkce vrací aktuální stav všech tlačítek. Každé stisknuté tlačítko je signalizováno jedničkou na příslušné pozici.

```
uint8_t isPressed(uint8_t);
```

Tato funkce uvěřuje, zda je zadané tlačítko/tlačítka stisknuto. pokud ano, vrací 1, jinak 0.

## 3.2 Knihovna font.h

Funkce této knihovny zajišťují uživateli možnost snadno získat obrazovou podobu ASCII znaku. Knihovna implementuje všechny potřebné tišitelné znaky. Funkce umožňují převést znak na obraz výšky 8, 16 a 32 bodů.

```
uint8_t getChar(char c, uint8_t charSize, uint8_t * buffer);
```

Tato funkce převede zadaný znak na jeho binární podobu o velikosti zadané parametrem `charSize` a uloží jej do bufferu. Návrátová hodnota je počet zapsaných bajtů.

```
int8_t getChar8(char c, uint8_t * buffer);
```

Tato funkce převede zadaný znak na jeho binární podobu o výšce řádku 8 bodů a uloží jej do bufferu. Návrátová hodnota je počet zapsaných bajtů.

```
uint8_t getChar16(char c, uint8_t * buffer);
```

Tato funkce převede zadaný znak na jeho binární podobu o výšce řádku 16 bodů a uloží jej do bufferu. Návrátová hodnota je počet zapsaných bajtů.

```
uint8_t getChar32(char c, uint8_t * buffer);
```

Tato funkce převede zadaný znak na jeho binární podobu o výšce řádku 32 bodů a uloží jej do bufferu. Návrátová hodnota je počet zapsaných bajtů.

## 3.3 Knihovna ssd1306.h

Funkce jež tato knihovna obsahuje sprostředkovávají základní ovládání OLED displeje. Aby vše fungovalo tak jak má, musí být k dispozici knihovna `i2c.h`, která zprostředkovává komunikaci po I<sup>2</sup>C sběrnici, a knihovna `font.h`, kde je uložena podoba ASCII znaků.

### 3.3.1 Funkční rozhraní knihovny

```
void OLED_init(void);
```

Tato funkce provede veškerá potřebná nastavení řadiče jak je tomu uvedeno v kapitole 1.6 a uvede jej do provozu. Zavolání této funkce je nezbytně nutné a musí předcházet před voláním kterékoli jiné funkce z této knihovny. Typické je tuto funkci zavolat na začátku aplikace. V případě problémů nastalých ruční konfigurací řadiče lze opětovným zavoláním této funkce obnovit počáteční konfiguraci.

```
void OLED_on(void);
```

Tato funkce provede zapnutí displeje z úsporného režimu. Její volání při



prvním spuštění není nutné, neboť toto volání proběhne v inicializační části.

**void OLED\_of(void);**

Tato funkce provede vypnutí displeje do úsporného režimu.

**void OLED\_setContrast(uint8\_t level);**

Tato funkce nastaví na hodnotu předanou parametrem level.

**void OLED\_clear(void);**

Tato funkce provede vymazání celé paměti. Tuto funkci je vhodné zavolat ihned po inicializaci displeje, neboť paměť SRAM může na počátku zaplněna náhodnými daty.

**void OLED\_fillRect(uint8\_t row, uint8\_t col,  
uint8\_t nrows uint8\_t ncols, uint8\_t fill);**

Touto funkcí lze vykreslit na pozici určenou číslem stránky a sloupce obdélník specifikované výšky a délky vyplněný daným vzorem.

**void OLED\_setContrast(uint8\_t row, uint8\_t col);**

Nastavení pozice kurzoru na displeji.

**void OLED\_setFont(uint8\_t font);**

Nastavení písma. Více podrobností v ssd1306.h.

**void OLED\_putChar(char c);**

Tato funkce vloží na předem specifikovanou pozici znak. V případě že kurzor dojde na konec displeje, vrátí se zpět na začátek a začne přepisovat obsah od začátku.

**void OLED\_putString(char \* str);**

Tato funkce vloží na předem specifikovanou pozici řetězec. Pokud se řetězec nevejde na řádek, je na konci řádku zalomen a pokračuje na novém. V případě že kurzor dojde na konec displeje, vrátí se zpět na začátek a začne přepisovat obsah od začátku.



---

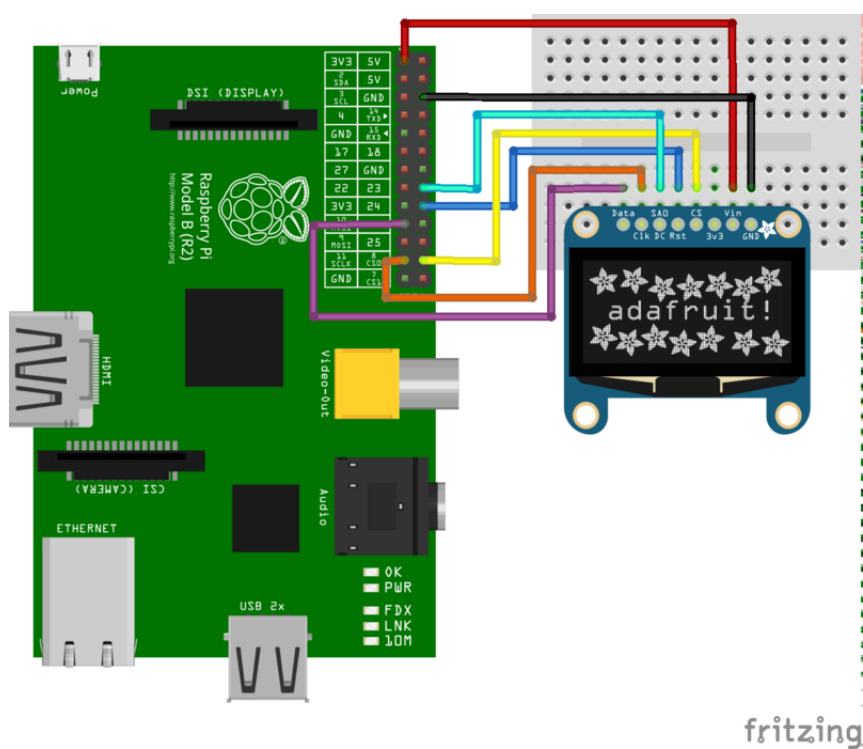
## Testování

Prvotní testování probíhalo platformě Raspberry Pi z důvodu snadné laditelnosti a jednoduchosti vývoje. Cíle tohoto způsobu testování bylo odladit uživatelské rozhraní a programové části kódu před portací na cílovou platformu ATtiny85. Cílová platforma nedisponuje mnoha prostředky pro ladění. Obě tyto platformy disponují rozhraním I<sup>2</sup>C, proto by neměl být problém přenést uživatelské rozhraní z jedné platformy na druhou.

Během přenosu na cílovou platformu se projevil nedostatek vnitřní paměti. Cílová aplikace potřebuje pro svůj běh vyčítat a ukládat do paměti množství textových řetězců a obrazových dat. Proto bylo nutné připojit externí I<sup>2</sup>C EEPROM paměť, do které by byla veškerá potřebná data ukládána. Nahrávání dat z externí paměti přes rozhraní I<sup>2</sup>C by bylo neefektivní a mohli by vést ke značnému zatížení a zpomalení celého vestavného systému. Kód realizující načítání dat je poměrně složitý a je příčinou, proč se nepovedlo dokončit portaci na cílovou platformu včas. Z tohoto důvodu se nepovedlo otestovat uživatelské rozhraní na cílové platformě.

#### 4. TESTOVÁNÍ

---



Obrázek 4.1: Propojení displeje osazeného řadičem SSD1306 k Raspberry Pi

# Závěr

V této závěrečné práci jsem navrhl a realizoval uživatelské rozhraní (UI) vestavného systému – Radiobudíku. Projekt radiobudíku se skládá z více bakalářských prací, které se zabývají USB komunikací, knihovnamí pro periferní obvody, návrhem desky plošných spojů a podobně.

V návrhu UI jsem se zabýval vhodnou velikostí písma a ovládacích prvků, dále způsobem zadávání uživatelských vstupů. Použitý mikrokontrolér AT-Tiny85 nedisponuje dostatečným množstvím volných I/O pinů, proto bylo nutné navrhnout alternativní způsoby pro získání vstupů od uživatele. UI je uzpůsobeno pro použití na displejích s úhlopříčkou okolo jednoho palce (do 3 cm).

V průběhu realizace a testování na Raspberry Pi jsem zjistil další nedostatky použitého mikrokontroléru. Z nedostatku paměti na cílové platformě ATtiny85 je nutné UI uložit mimo programovou pamět. Navržený způsob (použití EEPROM na I<sup>2</sup>C sběrnici) se nepovedlo včas implementovat pro svojí složitost. UI je proto realizované a otestované pouze na platformě Raspberry Pi.

Jako možné budoucí práce se nabízejí integrace UI s výsledky ostatních bakalářských prací z projektu radiobudíku a testování UI na cílové skupině uživatelů.



# Literatura

- [1] Atmel: *ATtiny85 Datasheet [online]*. Srpen 2013, [cit. 2015-05-06]. Dostupné z: [http://www.atmel.com/Images/Atmel-2586-AVR-8-bit-Microcontroller-ATtiny25-ATtiny45-ATtiny85\\_Datasheet.pdf](http://www.atmel.com/Images/Atmel-2586-AVR-8-bit-Microcontroller-ATtiny25-ATtiny45-ATtiny85_Datasheet.pdf)
- [2] Olejár, M.: Stručný popis sběrnice I<sup>2</sup>C a její praktické využití k připojení externí eeprom 24LC256 k mikrokontroléru PIC16F877 [online]. *HW.cz*, Květen 2000, [cit. 2015-05-01]. Dostupné z: <http://www.hw.cz/navrh-obvodu/strucny-popis-sbornice-i2c-a-jeji-prakticke-vyuziti-k-pripojeni-externi-eeeprom-24lc256>
- [3] Philips Semiconductors: *THE I2C-BUS SPECIFICATION [online]*. Druhé vydání, Leden 2000, [cit. 2015-05-01]. Dostupné z: <http://i2c2p.twibright.com/spec/i2c.pdf>
- [4] Solomon Systech: *SSD1306 [online]*. Duben 2008, [cit. 2015-05-01]. Dostupné z: <https://www.adafruit.com/datasheets/SSD1306.pdf>
- [5] Philips Semiconductors: *PCF8574 Remote 8-bit I/O expander for I<sup>2</sup>C-bus [online]*. Druhé vydání, Listopad 2002, [cit. 2015-05-01]. Dostupné z: [http://www.nxp.com/documents/data\\_sheet/PCF8574.pdf](http://www.nxp.com/documents/data_sheet/PCF8574.pdf)
- [6] Park, J.; Baek, N.: A Text-Based User Interface Style Toolkit for Low-Tier Embedded Systems. In *IT Convergence and Security (ICITCS), 2013 International Conference on*, Dec 2013, s. 1–4, doi:10.1109/ICITCS.2013.6717794.





## Seznam použitých zkratek

**ARM** Advanced RISC architecture

**EEPROM** Electrically erasable programmable read-only memory

**FM** Frekvenční modulace

**I<sup>2</sup>C** Internal-integrated circuit

**ISC** In-system programming

**GPIO** General purpose input/output

**LCD** Liquid crystal display

**OLED** Organic light-emitting diode

**PLED** Polymeric light-emitting diode

**RISC** Reduced instruction set computing

**SRAM** Static random access memory

**UI** User interface



---

## Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
exe .....	adresář se spustitelnou formou implementace
src	
_ impl.....	zdrojové kódy implementace
_ thesis .....	zdrojová forma práce ve formátu L <sup>A</sup> T <sub>E</sub> X
text .....	text práce
_ BP_Kudiveis_Ladislav_2015.pdf.....	text práce ve formátu PDF
_ BP_Kudiveis_Ladislav_2015.ps.....	text práce ve formátu PS