



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV RADIOELEKTRONIKY

DEPARTMENT OF RADIO ELECTRONICS

## REFERENČNÍ NÁVRH HID PERIFERIE TOUCH PAD

HID TOUCH PAD REFERENCE DESIGN

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Miroslav Děcký

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Aleš Povalač, Ph.D.

BRNO 2018



# Diplomová práce

magisterský navazující studijní obor **Elektronika a sdělovací technika**  
Ústav radioelektroniky

**Student:** Bc. Miroslav Děcký

**ID:** 164254

**Ročník:** 2

**Akademický rok:** 2017/18

**NÁZEV TÉMATU:**

## Referenční návrh HID periferie Touch Pad

### POKYNY PRO VYPRACOVÁNÍ:

Navrhněte koncepci pro realizaci Touch Padu s využitím mikrokontroléru rodiny Kinetis, periferie TSI a embedded knihovny NXP Touch. Zařízení bude připojené jako USB HID typu mouse, resp. trackpad. Realizujte hardware zařízení pomocí vhodné vývojové desky pro demonstraci a ladění.

Implementujte podporu USB HID a ověřte funkci v operačním systému Windows. Navrhněte referenční hardware, který bude obsahovat mikrokontrolér, touchpad a všechny potřebné pomocné obvody. Referenční hardware realizujte, přizpůsobte a odlaďte testovací software a ověřte celkovou funkci. Software dle potřeby rozšiřte o rozpoznávání jednoduchých gest.

### DOPORUČENÁ LITERATURA:

[1] NXP [online]. FRDM-KL26Z User's Guide, FRDMKL26ZUG [cit. 15.5.2017]. Dostupné na [www: https://goo.gl/ueMwpu](https://goo.gl/ueMwpu).

[2] FRÝZA, T. Mikroprocesorová technika. Elektronické skriptum. Brno: FEKT VUT v Brně, 2008.

**Termín zadání:** 5.2.2018

**Termín odevzdání:** 17.5.2018

**Vedoucí práce:** Ing. Aleš Povalač, Ph.D.

**Konzultant:** Michael Galda, NXP Semiconductors

**prof. Ing. Tomáš Kratochvíl, Ph.D.**  
*předseda oborové rady*

### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.



## **ABSTRAKT**

Diplomová práce se zabývá návrhem a realizací referenčního HID zařízení Touchpad. Pro realizaci touchpadu bude využito mikrokontrolerů rodiny Kinetis výrobce NXP, periferie TSI a embedded knihovny NXP Touch. Výsledné zařízení bude připojeno jako USB HID typu mouse, resp. trackpad. Hardwarová část práce je realizována přímo s procesory rodiny Kinetis na samostatně navržených deskách plošných spojů, část také s využitím vývojové platformy NXP Freedom KL25z and KE15z.

## **KLÍČOVÁ SLOVA**

Touchpad, kapacitní snímání, NXP Touch, TSI, Kinetis, FreeMASTER, HID, KE15z, KL25z, Freedom.

## **ABSTRACT**

This master's thesis deals with the draft and design of HID periphery Touchpad. The Kinetis microcontroller produced by NXP, TSI periphery and embedded library NXP Touch are used for the design. The resulting device is connected as USB HID type mouse or trackpad. Hardware part of the project is implemented by whole MCU Kinetis Family on PCBs designed by author, one part also uses evaluation kits NXP Freedom KL25z and KE15z.

## **KEYWORDS**

Touchpad, Touch Sensing, NXP Touch, TSI, Kinetis, FreeMASTER, HID, KE15z, KL25z, Freedom.

DĚCKÝ, M. *Referenční návrh HID periferie Touch Pad*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 68 s. Vedoucí diplomové práce Ing. Aleš Povalač, Ph.D..

# PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma Referenční návrh HID periferie Touch Pad jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne .....

.....

(podpis autora)

# PODĚKOVÁNÍ

Chtěl bych na tomto místě poděkovat vedoucímu práce Ing. Aleši Povalačovi, Ph.D. za konzultace a rady. Dále Ing. Pavlu Rechovi, Ph. D. a Ing. Michaelu Galdovi za praktické konzultace k návrhu touchpadu.

Experimentální část této diplomové práce byla realizována na výzkumné infrastruktuře  
vybudované v rámci projektu CZ.1.05/2.1.00/03.0072  
**Centrum senzorických, informačních a komunikačních systémů (SIX)**  
operačního programu Výzkum a vývoj pro inovace.

# OBSAH

Úvod	1
<b>1 Kapacitní snímání</b>	<b>2</b>
1.1 Principy kapacitního snímání.....	2
1.1.1 Povrchová kapacita .....	4
1.1.2 Vlastní kapacita.....	4
1.1.3 Technika vzájemné kapacity [5] .....	5
1.2 HW metody kapacitního snímání .....	6
1.2.1 Metoda GPIO [6] .....	6
1.2.2 Relaxační oscilátor a periferie TSIv4 [7].....	7
1.2.3 Metoda přenosu náboje a periferie TSIv5 [8].....	9
1.2.4 Metoda elektrického pole [9].....	10
1.3 Návrh plošného spoje pro kapacitní snímání .....	11
1.3.1 Návrh motivu plošného spoje .....	11
1.3.2 Návrh uspořádání elektrod.....	13
1.4 Knihovna NTouch.....	15
1.4.1 Nastavení knihovny NTouch .....	15
1.4.2 Použití knihovny NXP Touch.....	17
<b>2 USB</b>	<b>19</b>
2.1 Struktura USB.....	19
2.2 Třídy zařízení .....	19
2.3 Třída HID.....	20
2.3.1 HID descriptor .....	21
2.3.2 HID reporty.....	21
<b>3 Návrh HW touch padu</b>	<b>22</b>
3.1 Volba mikrokontroleru .....	22
3.2 Návrh provedení touchpadu.....	24
3.2.1 Návrh tvaru elektrod .....	24
3.2.2 Návrh motivu touchpadu pro návrhový systém PCB .....	25
3.2.3 Návrh velikosti touchpadu .....	26
3.3 Návrh plug in modulu pro FRDM-KE15Z .....	27

3.3.1	Schéma zapojení modulu .....	27
3.4	HID zařízení touchpad s KE15z + FRDM KL25z.....	29
3.4.1	Schéma zapojení touchpadu KE15z .....	29
3.4.2	Návrh plošného spoje touchpadu KE15z.....	32
3.5	HID zařízení touchpad s KL25z .....	34
3.5.1	Schéma zapojení touchpadu s KL25z .....	34
3.5.2	Návrh plošného spoje pro touchpad s KL25z .....	38
<b>4</b>	<b>Návrh softwaru touchpadu</b>	<b>39</b>
4.1	Testování HID zařízení s FRDM-KL25z.....	39
4.1.1	Volba typu USB zařízení .....	39
4.1.2	Prototypový testovací SW kompozitního HID zařízení .....	40
4.2	SW pro touchpad s KL25z.....	41
4.2.1	Inicializace knihovny NTouch.....	41
4.2.2	Řešení pro přepínání plné a „ladící“ velikosti touchpadu.....	42
4.2.3	Realizace obsluhy pohybu myši .....	44
4.2.4	Realizace scrollování .....	45
4.2.5	Realizace stisku levého tlačítka .....	45
4.2.6	Realizace zoomovacího gesta .....	47
4.3	SW pro touchpad s KE15z + FRDM KL25z .....	49
4.3.1	SW pro KE15z .....	49
4.3.2	SW pro FRDM-KL25z .....	50
<b>5</b>	<b>Nástroj FreeMASTER</b>	<b>53</b>
5.1	Použití nástroje FreeMASTER .....	55
5.1.1	FreeMASTER na straně mikrokontroléru.....	56
<b>6</b>	<b>Závěr</b>	<b>57</b>
	<b>Literatura</b>	<b>58</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>60</b>

# SEZNAM OBRÁZKŮ

Obrázek 1 Dotykové snímání [1].....	2
Obrázek 2 Model deskového kondenzátoru [2].....	3
Obrázek 3 Princip povrchové kapacity [3] .....	4
Obrázek 4 Dotykový panel [4].....	4
Obrázek 5 Uspořádání pro metodu vzájemné kapacity [5].....	5
Obrázek 6 Zapojení pro GPIO metodu [6] .....	6
Obrázek 7 Relaxační oscilátor elektrody [5] .....	8
Obrázek 8 Princip měření kapacity relaxačním oscilátorem .....	8
Obrázek 9 Principiální zapojení TSIv5 [8] .....	9
Obrázek 10 Systém s využitím elektrického pole [9] .....	10
Obrázek 11 Parazitní kapacita dvouvrstvého plošného spoje [10].....	11
Obrázek 12 Řešení zemní plochy pod elektrodami .....	11
Obrázek 13 Model vedení.....	12
Obrázek 14 Vyvedení tras od MCU [10].....	12
Obrázek 15 Motiv maticové klávesnice [1] .....	13
Obrázek 16 Analogový posuvník (nahore) a "digitální" posuvník (dole) [1].....	13
Obrázek 17 Digitální rotary (vlevo) a analogový rotary (vpravo) [10] .....	14
Obrázek 18 Modul touchpadu [10].....	14
Obrázek 19 Bloková struktura knihovny NTouch [11] .....	15
Obrázek 20 Struktura projektu s knihovnou NTouch.....	16
Obrázek 21 Platforma TOWER s TWRPI modulem [14] .....	22
Obrázek 22 Vývojový kit FRDM-KL25Z [15].....	23
Obrázek 23 FRDM-KE15Z s modulem NTouch [16] .....	23
Obrázek 24 Uspořádání elektrod modulu touchpadu .....	24
Obrázek 25 Schematická značka touchpadu.....	25
Obrázek 26 Prvky svislé elektrody .....	25
Obrázek 27 Motiv vodorovné elektrody.....	26
Obrázek 28 Detail zapojení svislých elektrod .....	26
Obrázek 29 Napěťový dělič pro identifikaci modulu .....	28
Obrázek 30 Motiv dceřiného modulu touchpad s konektory pro FRDM KE15z.....	28
Obrázek 31 SWD konektor a RESET obvod.....	30

Obrázek 32 Zapojení Arduino konektorů .....	30
Obrázek 33 Rozmístění Arduino kompatibilních konektorů .....	32
Obrázek 34 Touchpad KE15z s možností připojení FRDM platformy .....	33
Obrázek 35 Schématická značka touchpadu 9×7 .....	34
Obrázek 36 Obvod pomocného kontaktu pro povolení ladění .....	35
Obrázek 37 Zapojení lineárního stabilizátoru pro KL25z .....	36
Obrázek 38 Zapojení USB mini B konektoru pro touchpad [17] .....	36
Obrázek 39 Touchpad s KL25z .....	38
Obrázek 40 Testovací přípravek pro ladění USB .....	40
Obrázek 41 Vývojový diagram funkce pro detekci kliku myši .....	46
Obrázek 42 Realizace dvouprstového zoomování .....	47
Obrázek 43 Vývojový diagram příjmu paketu od KE15z .....	51
Obrázek 44 Záložka Controls ve FreeMASTERu pro MATRIX .....	54
Obrázek 45 Nastavení výstupního souboru aplikace ve FMSTR .....	56



## SEZNAM TABULEK

Tabulka 1 Příklad tříd USB zařízení.....	20
Tabulka 2 Zapojení elektrod modulu.....	27
Tabulka 3 Zapojení pinů touchpadu s KE15z.....	31
Tabulka 4 Zapojení elektrod touchpadu s KE15z.....	31
Tabulka 5 Zapojení touchpadu s KL25z.....	34
Tabulka 6 PinMux SWD pinů [18].....	35
Tabulka 7 Zapojení pinů s mikrokontrolerem KL25z .....	37

## SEZNAM PŘÍLOH

Příloha 1 Schéma zapojení modulu pro KE15z.....	61
Příloha 2 Motiv plošného spoje touchpadu s KE15z.....	62
Příloha 3 Osazovací plán touchpadu s KE15z.....	63
Příloha 4 Motiv TOP plošného spoje touchpadu s KL25z .....	64
Příloha 5 Osazovací plán touchpadu s KL25z.....	65
Příloha 6 Strana BOTTOM motivu plošného spoje KL25z .....	66
Příloha 7 Záložka Modules ve FreeMASTERu.....	67
Příloha 8 Záložka Controls ve FreeMASTERu .....	68

# ÚVOD

Tato diplomová práce se zabývá referenčním návrhem HID periferie touchpad, jakožto zařízení pro prezentační účely rodiny mikrokontrolerů Kinetis od firmy NXP Semiconductors a její knihovny NXP Touch.

Touchpad je polohovací zařízení založené na kapacitním snímání. První kapitola této diplomové práce se zabývá popisem principů a metod kapacitního snímání, nejdříve obecně, poté je zde popsán přehled možných metod kapacitního snímání, dále následuje stručný výtah pravidel pro návrh plošného spoje, na kterém je využito kapacitní snímání. Následuje popis možných uspořádání elektrod pro jednotlivé použití snímačů pro knihovnu NXP Touch. Na závěr je rozebrána knihovna NXP Touch a nastíněn postup její úpravy pro potřeby nového projektu.

Druhá kapitola se zabývá stručným popisem sériové sběrnice USB a třídy zařízení HID, kterou výsledný touchpad využívá pro komunikaci s počítačem.

Třetí kapitola se zabývá kompletním návrhem HW touchpadu. Nejdříve je popsán výběr použitých mikrokontrolerů, následně obecný postup návrhu samotného bloku součástky touchpad. Poté jsou v této kapitole popsány návrhy a výroba tří touchpadů – touchpadu 9×8 jako dceřiné rozšiřující desky pro FRDM-KE15z, dále návrh touchpadu s mikrokontrolerem KE15z, ke kterému je možno pomocí Arduino kompatibilních konektorů připojit platformu NXP Freedom, a na závěr návrh samostatného HID touchpadu s mikrokontrolerem Kinetis KL25z

Čtvrtá kapitola se věnuje softwarové části diplomové práce. Jsou zde rozebrány jednotlivé kroky vývoje, nejdříve testování HID zařízení s FRDM-KL25z, poté samotný SW pro HID touchpad s KL25z s popisem postupu implementace jednotlivých funkcí touchpadu. Poslední část čtvrté kapitoly se věnuje přenosu funkcionality touchpadu s KL25z na verzi s KE15z a jejím nutným úpravám.

Poslední pátá kapitola se krátce věnuje ladicímu nástroji FreeMASTER, který vytvořila firma NXP pro real-time ladění programů v mikrokontrolerech. Tento nástroj umožňuje přes sériovou linku vyčítat obsah paměti mikrokontroleru a dle připravených šablon jej vizualizovat. Je zde také popsán postup nutných úprav projektu pro vizualizaci aplikací NXP Touch, které bylo nutné pro vizualizaci touchpadu v tomto nástroji provést.

# 1 KAPACITNÍ SNÍMÁNÍ

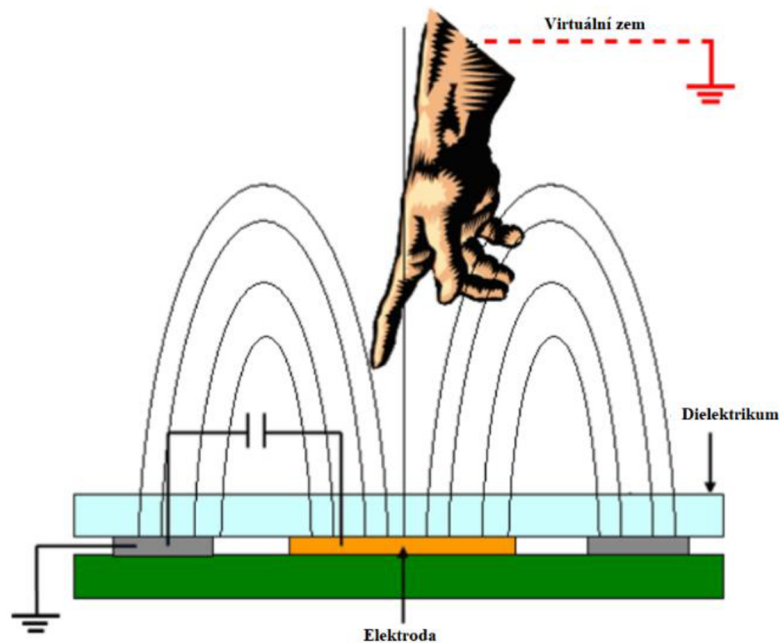
Dotykové ovládání zařízení využívající různé dotykové panely, displeje, klávesnice nebo enkodéry se objevuje čím dál častěji. Tato technologie nahrazuje klasické mechanické spínače nebo membránové klávesnice. Lze je najít na širokém spektru zařízení od monitorů počítačů až po miniaturní MP3 přehrávače, Bluetooth sluchátka nebo chytré telefony.

Dotykové snímání neobsahuje žádné pohyblivé mechanické prvky (jazýčkové kontakty atd.), a tedy odbourává mechanické problémy s únavou materiálů, případně s oxidací a korozi kontaktů. Díky dotykovému snímání jsou moderní zařízení tišší, mohou být tenčí a výroba dotykového snímače je výrazně levnější a technologicky méně náročná než výroba klasického miniaturního mikrosvínače.

## 1.1 Principy kapacitního snímání

Hlavní výhodou kapacitního snímání je možnost oddělit elektroniku od uživatele bez nutnosti využití pohyblivých částí. Uživatel se dotýká snímače, který může být průhledný, jako v případě dotykového displeje, nebo skrz nevodivý materiál se servisním potiskem.

Možné zapojení kapacitního systému je uvedeno na obrázku 1 níže:

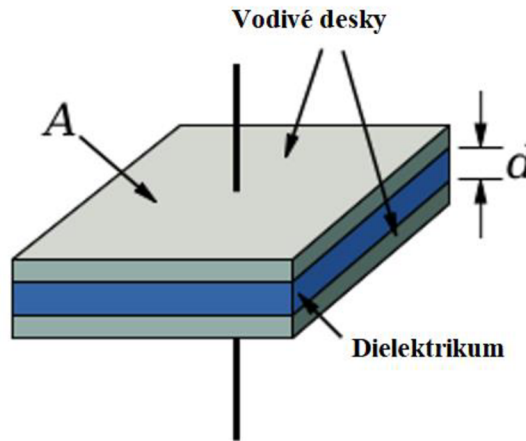


Obrázek 1 Dotykové snímání [1]

Uživatel se přiblížením k dotykovému panelu stává součástí obvodu, svým prstem změni kapacitu dotykového snímače, kterou zachytí měřicí elektronika a vyhodnotí jako

dotek. Díky metodě měření kapacity se tento systém snímání nazývá kapacitní snímání.

Chování elektrody – kapacitního snímače, ovlivňují parametry jako tloušťka pokovení plošného spoje, materiál substrátu, velikost a tvar snímací elektrody a v neposlední řadě také vedení od elektrody k pinu mikrokontroleru. Kapacitní snímání je založeno na principu deskového kondenzátoru, jeho model je na obrázku 2 dále.



Obrázek 2 Model deskového kondenzátoru [2]

$$C = \frac{k \cdot \epsilon_0 \cdot A}{d} \quad (1)$$

- $C$  – kapacita [F]
- $A$  – plocha elektrod [ $\text{m}^2$ ]
- $d$  – vzdálenost mezi elektrodami [m]
- $k$  – dielektrická konstanta materiálu mezi elektrodami
- $\epsilon_0$  – permitivita vakua ( $8,854 \cdot 10^{-12}$  F/m)

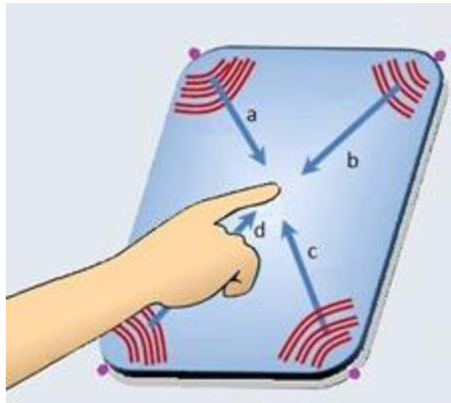
Model deskového kondenzátoru vykresluje princip fungování kapacitní elektrody, kdy elektroda na plošném spoji je jednou polovinou tohoto kondenzátoru a lidský prst druhou.

Ze vzorce (1) vyplývá, že pokud je požadavek na maximální změnu kapacity, je zapotřebí co největší plocha elektrod, pro větší spolehlivost musí být ovšem plocha elektrody úměrná detekovatelnému objektu. Větší elektroda je citlivější, ale také náchylnější na rušení, ovšem tento efekt se využívá pro tzv. proximity elektrody, tedy elektrody, které mají za úkol detekovat objekty na větší vzdálenosti. Tyto proximity elektrody slouží například pro probouzení mikrokontroleru z režimu spánku při přiblížení ruky uživatele a podobně.

Parametr vzdálenosti elektrod  $d$  klade omezující podmínky na krytí elektrod dielektrickým materiálem, což omezuje použití krycích plexiskel apod.

### 1.1.1 Povrchová kapacita

Metoda využívající povrchovou kapacitu se využívá například u dotykových displejů, jelikož umožňuje přímou detekci prostorové 2D souřadnice. Princip metody dobře ilustruje obrázek 3. Je patrné, že základem zařízení může být například sklo, které je pevně spojeno s vodivou folií. Tento systém je v rozích napájen měřicími signály, které vytvoří homogenní elektrostatické pole a v klidovém stavu jsou tedy tyto rohové zdroje v rovnováze.

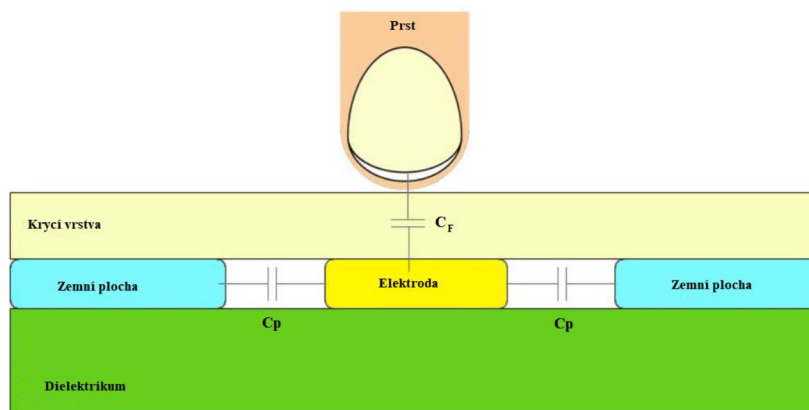


Obrázek 3 Princip povrchové kapacity [3]

Při doteku prstu dojde k vytvoření kondenzátoru mezi vodivou plochou a lidským tělem. Jednotlivé zdroje v rozích dotykové plochy „uvidí“ v závislosti na vzdálenosti doteku od příslušného zdroje různou kapacitu a díky tomu bude také proud těchto zdrojů rozdílný. Z poměrů proudů je poté možno vypočítat pozici prstu na dotykové ploše.

### 1.1.2 Vlastní kapacita

Další metodou kapacitního snímání je metoda vlastní kapacity. Při této metodě je využíván kondenzátor vznikající mezi elektrodou a prstem uživatele. Možné zapojení elektrody udává obrázek 4 níže:



Obrázek 4 Dotykový panel [4]

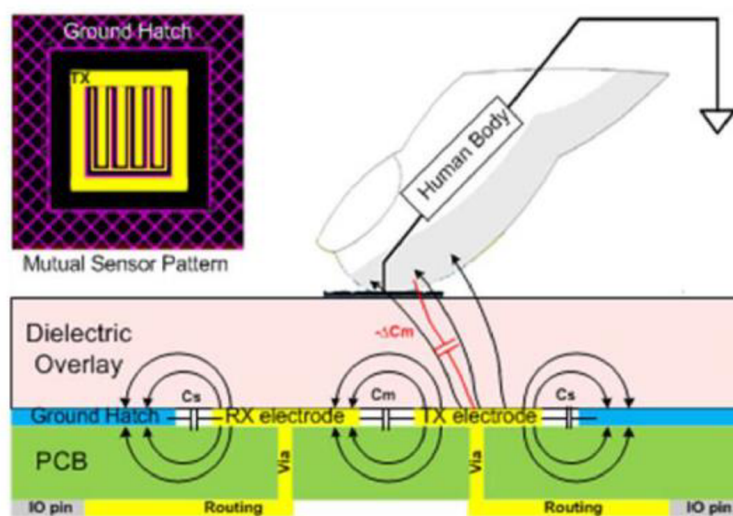
Při této metodě mají senzory vlastní parazitní kapacitu tvořenou kapacitou mezi senzorem a vylitou zemní plochou. Při přiblížení prstu k elektrodě se vytvoří přidaná kapacita mezi senzorem a prstem uživatele, a tím se změní kapacita systému:

$$C = C_p + C_f \quad (2)$$

Dotek je detekován pomocí periodického měření kapacity senzoru, náhlá změna kapacity znamená dotek uživatele. V případě využití metody vlastní kapacity dochází při doteku ve většině případů ke zvýšení kapacity systému (požadavek na shodný zemní potenciál uživatele a zařízení). Každý dotek ovšem vyvolá jiný přírůstek kapacity, a tak je nutné nastavit určitý práh citlivosti, který musí kapacita překročit pro potvrzený dotek. Tento práh je nutné periodicky upravovat, jelikož parazitní kapacita systému  $C_p$  se v čase také mění.

### 1.1.3 Technika vzájemné kapacity [5]

Zatímco pro dotykovou elektrodu využívající vlastní kapacity postačuje jeden pin mikrokontroleru, pro metodu vzájemné kapacity je nutné využít piny dva. Příklad konfigurace snímacího systému je na obrázku 5 dále:



Obrázek 5 Uspořádání pro metodu vzájemné kapacity [5]

Při metodě vzájemné kapacity je snímací systém připojen ke dvěma pinům mikrokontroleru. Jeden pin slouží jako vysílací (budicí) prvek a druhý pin jako snímací. Budicí pin generuje do systému impulzy, které jsou skrz vzájemnou kapacitu elektrod přeneseny do přijímací elektrody. Při dotyku prstu dojde k přidání kapacitní vazby mezi vysílačem a prstem uživatele, čímž se zmenší přenos pulzů z vysílací elektrody na přijímací, tento pokles vazební kapacity je detekován v mikrokontroleru.

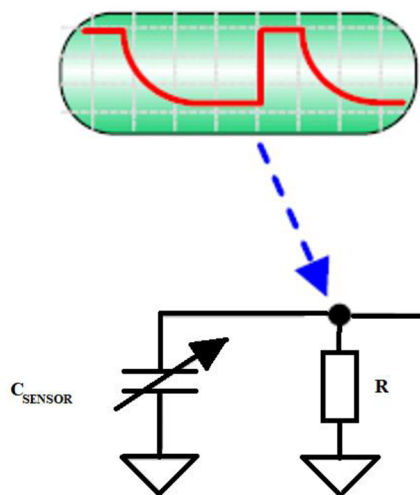
## 1.2 HW metody kapacitního snímání

Kapacitní snímání je možné implementovat různými způsoby, nejstarší metodou je metoda s pomocí GPIO pinů, další metodou je metoda využívající vlastností relaxačního oscilátoru, případně metoda přenosu náboje či detekce elektrického pole.

Firma NXP vyvinula pro své mikrokontroléry rodiny Kinetis periferii TSI – Touch Sensing Interface, tato periferie umožňuje přímou implementaci kapacitního snímání s pomocí hardware periferie mikrokontroleru, není tedy nutné pro připojení snímacích elektrod vytvářet žádné pomocné meziobvody. TSI periferie má za sebou několik vývojových stupňů, aktuálně nejmodernější je TSIv5 (jaro 2018), která je použita na mikrokontrolerech řady Kinetis KE, starší verzi je TSIv4, kterou je možno nalézt například na mikrokontrolerech Kinetis KL25z.

### 1.2.1 Metoda GPIO [6]

GPIO metoda využívá externího rezistoru pro vybíjení/nabíjení externího kondenzátoru a čítače měřícího čas nabíjení/vybíjení. Jednoduché schéma zapojení pro GPIO metodu s využitím vybíjení kondenzátoru je na následujícím obrázku 6:



Obrázek 6 Zapojení pro GPIO metodu [6]

Principem metody je v tomto zapojení měření doby vybíjení kondenzátoru připojeného k libovolnému portu. GPIO port je v první fázi nastaven jako výstupní a pomocí log. 1 je nabit kondenzátor. Poté je nutné přepnout GPIO pin do vstupního módu, kde díky vysoké impedanci dochází k majoritnímu vybíjení přes externí rezistor. Při přepnutí do vstupního módu je vynulován čítač a dochází k měření doby do poklesu úrovně na logickou 0, a tedy odpovídající kapacity.

Kapacita elektrody senzoru je přibližně 10 pF, dotek lidského prstu může přidat přídavnou kapacitu do 3 pF. Problém metody spočívá v korektní volbě vybíjecího rezistoru, který musí zajistit vybití systému v rozumném čase z log. 1 na log. 0. V 3,3 V

logice to znamená, že napětí na kondenzátoru musí klesnout z VCC na hodnotu menší než 0,8 V, což činí pokles na 24 % původní hodnoty.

$$u = VCC \cdot e^{-\frac{t}{\tau}} \quad (3)$$

Ze vzorce pro vybíjení kondenzátoru (3) výše vyplývá, že pro dosažení vybití na 24 % původní hodnoty musí platit:

$$\frac{t}{\tau} > 1,5 \quad (4)$$

Pro vybíjecí odpor tedy platí:

$$\frac{t}{1,5} = R \cdot C \quad (5)$$

$$R = \frac{t}{1,5 \cdot C} \quad (6)$$

Uvažujeme-li kapacitu dotknuté elektrody  $C = 13 \text{ pF}$  a periodu vzorkování  $T_{vz} = 250 \text{ } \mu\text{s}$ , poté výsledný vybíjecí odpor musí mít hodnotu:

$$R = \frac{250 \cdot 10^{-6}}{1,5 \cdot 13 \cdot 10^{-12}} \sim 13 \text{ M}\Omega \quad (7)$$

Jak vplývá z obrázku 6 výše, rezistor je v obvodu pevně dán při výrobě plošného spoje. Jakákoli adaptace tohoto obvodu v čase s pomocí SW úprav není možná, a je tedy problém navrhnout korektní hodnotu rezistoru. Druhým omezením je samotná vysoká hodnota rezistoru, na které se může snadno indukovat rušení. GPIO metoda trpí stejnými neduhy jako HW odstranění zákmitů tlačítek, a tak je i tato metoda nahrazena vhodnějším, softwarově říditelným řešením.

## 1.2.2 Relaxační oscilátor a periferie TSIv4 [7]

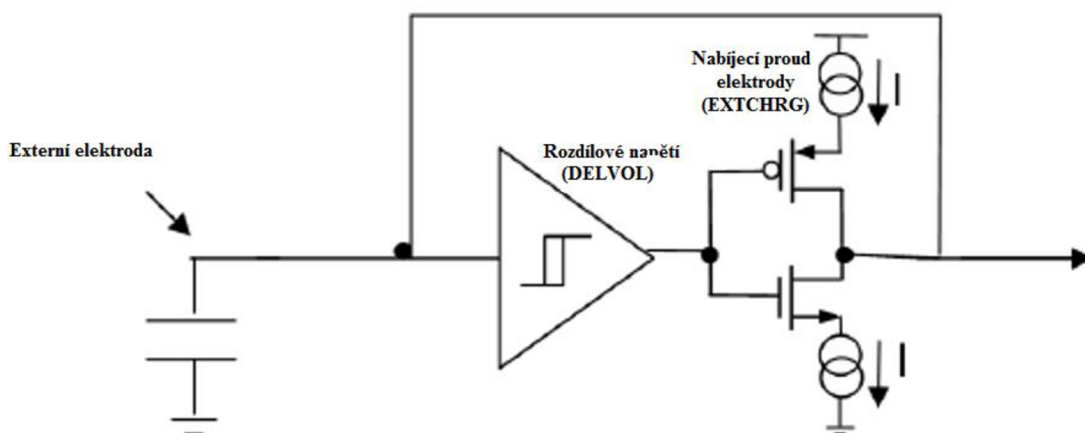
Základní princip TSIv4 je založen na aplikaci relaxačního oscilátoru. Relaxační oscilátor je oscilátor, který kmitá díky nelineárnímu prvku, a tedy není schopen kmitat samostatně, nýbrž vyžaduje trvalý přísun energie. Díky požadavku na budící signál je možné jeho kmity dobře řídit.

Celý systém periferie TSIv4 pracuje se dvěma signály. Jako relaxační oscilátor je zapojen obvod se snímanou elektrodou, která produkuje signál elektrody.

Druhý signál je interní a slouží jako referenční, pro dobrou funkci porovnávání je nutné, aby interní referenční signál byl výrazně rychlejší než signál relaxačního oscilátoru elektrody. Při této konfiguraci dvou dostupných signálů mikrokontroler vyhodnocuje počet period referenčního signálu za dobu jedné periody signálu relaxačního oscilátoru elektrody.

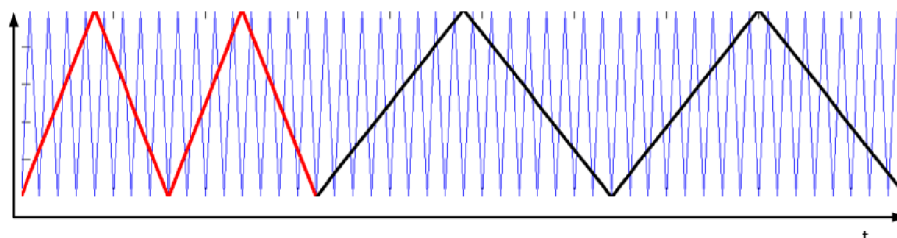


Na obrázku 7 níže lze vidět principiální zapojení modulu TSIv4 a elektrody:



Obrázek 7 Relaxační oscilátor elektrody [5]

Z principiálního schématu na obrázku 7 je patrné, že když dojde k doteku elektrody, dojde zároveň ke zvýšení kapacity obvodu. Díky nabíjení systému zdroji proudu dojde k prodloužení potřebného času nabíjení a tím i periody signálu oscilátoru. Díky násobně rychlejšímu referenčnímu signálu je možné detekovat změnu počtu kmitů referenčního signálu za periodu signálu oscilátoru a tím určit dotek. Zobrazení průběhu interního signálu a signálu elektrody bez doteku i s dotekem je naznačeno na obrázku 8 dále:



Obrázek 8 Princip měření kapacity relaxačním oscilátorem

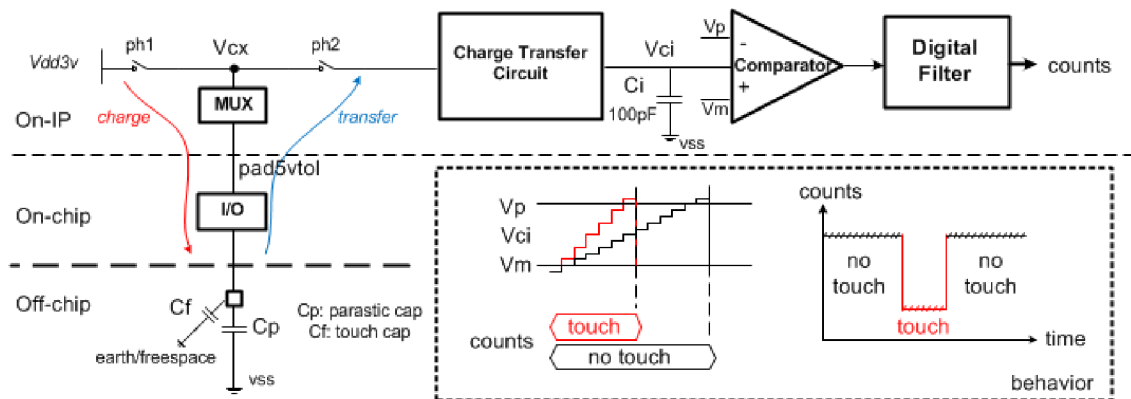
- Modrý průběh – interní referenční signál
- Červený průběh – signál relaxačního oscilátoru elektrody bez doteku
- Černý průběh – signál relaxačního oscilátoru elektrody při doteku

Pro zvýšení citlivosti na změnu kapacity je možné zvýšit frekvenci vnitřního oscilátoru. Díky říditelným zdrojům je možné lépe nastavit také dobu nabíjení/vybíjení obvodu podle potřeby a vlastností připojené elektrody.

Metoda s relaxačním oscilátorem oproti GPIO metodě využívá konkrétní periférii mikrokontroleru s říditelnými proudovými zdroji a vnitřní referenční oscilátor, v případě mikrokontrolerů Kinetis se jedná o periférii TSI verze 4. Díky možnosti softwarového nastavení je tato metoda vhodnější, jelikož je schopna reagovat na změny aplikačního prostředí pouze změnou parametrů periferie TSI bez nutnosti fyzického zásahu do plošného spoje, což umožňuje i lépe pokrýt výrobní tolerance.

### 1.2.3 Metoda přenosu náboje a periferie TSIv5 [8]

TSI verze 5 je aktuálně (jaro 2018) nejnovější verzí TSI periferie, oproti TSI verze 4 nevyužívá relaxačního oscilátoru, ale metody přenosu náboje. Principiální blokové schéma je uvedeno na obrázku 9 dále. Systém využívá dva nepřekrývající se signály  $ph1$  a  $ph2$ , které řídí nabíjení kapacitní elektrody a přenos náboje na referenční kondenzátor, tedy její vybíjení.



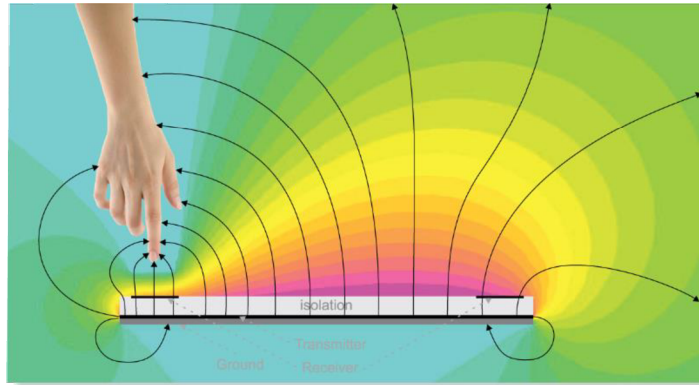
Obrázek 9 Principiální zapojení TSIv5 [8]

Náboj kapacitní elektrody je ve fázi  $ph2$  přiveden na CTC (viz obrázek 9 výše), který převádí náboj elektrody na nabíjecí proud pro referenční kondenzátor  $C_i$ . Počet pulzů  $ph2$  signálu je zaznamenáván, při doteku elektrody dojde ke zvýšení kapacity, díky nabíjení kondenzátoru napěťovým zdrojem i ke zvýšení náboje a výsledkem je snížení počtu pulzů nutných pro dosažení referenčního napětí na referenčním kondenzátoru  $C_i$ . Po dosažení tohoto napětí dojde k vybití referenčního kondenzátoru  $C_i$  a následuje další cyklus snímání.

### 1.2.4 Metoda elektrického pole [9]

Výše popsané metody jsou vhodné zejména pro kontaktní snímání, při dodržení jistých konstrukčních požadavků elektrod je ale možné docílit tzv. proximity snímání, tedy snímání ve vzdálenějším prostoru. Přímou pro proximity snímání bylo firmou Microchip patentováno řešení pro využití elektrického pole.

Typové zapojení uvádí obrázek 10 dále.



Obrázek 10 Systém s využitím elektrického pole [9]

Podobně jako metoda snímání vzájemné kapacity využívá metoda elektrického pole budicí a snímací elektrodu. Vysílací elektroda je buzena střídavým napětím o frekvenci cca 100 kHz, díky čemuž dojde k vytvoření téměř homogenního elektrického pole nad budicími i snímacími elektrodami.

Vysílací elektroda má rozměry řádově  $20 \times 20$  cm, ovšem budicí signál má frekvenci 100 kHz, vlnová délka signálu tedy je:

$$\lambda = \frac{c}{f} = \frac{3 \cdot 10^8}{1 \cdot 10^5} = 3000 \text{ m} \quad (8)$$

Díky tomu, že elektroda, v tomto případě ji můžeme nazvat anténou, je výrazně menší než vlnová délka, výsledná magnetická složka pole je nulová, a tak nedochází ke generování elektromagnetické vlny.

Obdobně jako když při metodě vzájemné kapacity došlo ke snížení vazební kapacity přiložením detekovaného prstu, zde dochází při přiblížení uživatelské ruky do elektrického pole k jeho částečnému uzemnění přes tělo uživatele. Díky tomu se intenzita elektrického pole měřená na přijímacích elektrodách zmenší a tato změna informuje systém o přiblížení uživatele k zařízení. Tato situace je zachycena na obrázku 10 výše.

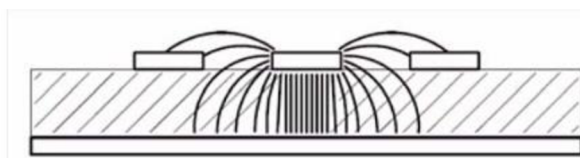
## 1.3 Návrh plošného spoje pro kapacitní snímání

Klasický mikrosvínač je velice jednoduché zařízení. Pro detekci jeho stavu se používají GPIO piny mikrokontroleru a jeho rozepnutí je dobře detekovatelné, a tedy mechanický mikrosvínač neklade téměř žádné dodatečné požadavky na konstrukci plošného spoje, s výjimkou případných obvodů pro potlačení zákmitů.

Oproti tomu kapacitní snímač neobsahuje žádný mechanický prvek, je tedy jednodušší po stránce konstrukční, ovšem oč je jednodušší po mechanické stránce, o to větší nároky klade na návrh a výrobu plošného spoje.

### 1.3.1 Návrh motivu plošného spoje

Jedním z omezujících faktorů je parazitní kapacita při použití vícevrstvého plošného spoje. Při vícevrstvěm plošném spoji vzniká parazitní kondenzátor mezi elektrodou a zemní deskou. Příklad uspořádání je možno vidět na obrázku 11.



Obrázek 11 Parazitní kapacita dvouvstvého plošného spoje [10]

Tento jev vnáší do měření jistou základní úroveň kapacity elektrody. Je možno uvažovat úplné vynechání zemní plochy pod elektrodami, avšak tento přístup vnáší do obvodu náhodnost, kdy kapacita obvodu může značně kolísat v závislosti na okolních podmínkách, například při pohybu plošného spoje s elektrodou a podobně.

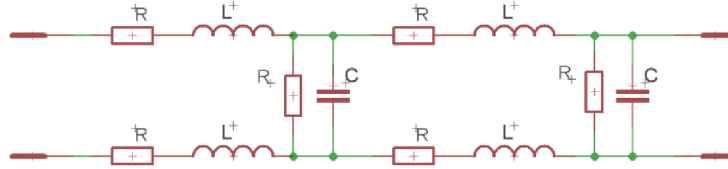
Proti tomuto jevu je možné bojovat pouze kompromisem, a tím je zmenšení plochy parazitní elektrody – zemní plochy. Tohoto je možné dosáhnout použitím šrafovaného vzoru s nízkým poměrem plnění v okolí elektrod. Možné provedení je ukázáno na obrázku 12 níže.



Obrázek 12 Řešení zemní plochy pod elektrodami

Jako u každého řešení jsou podmínky protichůdné, čím menší plnění země bude zvoleno, tím více bude systém náchylný na elektromagnetické rušení, ale naproti tomu bude také citlivější. Je tedy nutné zvolit vhodný kompromis pro každou jednotlivou aplikaci. Obecně je doporučováno plnění mezi 20 až 30 % [10].

Druhým výrazným faktorem návrhu plošného spoje je samotné vedení mezi elektrodou a pinem mikrokontroleru, model vedení dvouvrstvé desky plošného spoje je vyobrazen na obrázku 13 níže:

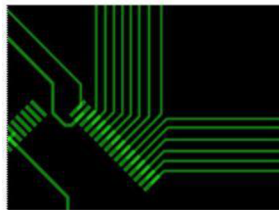


Obrázek 13 Model vedení

Vzdálenost jak mezi jednotlivými cestami, tak mezi cestami a zemní plochou ovlivňuje velmi výrazně kapacitní vazbu mezi těmito trasami. V aplikacích kapacitního snímání je jakákoliv parazitní kapacitní vazba kritickým prvkem, který ovlivňuje citlivost elektrod.

Pro minimalizaci výše zmíněných nežádoucích jevů jsou doporučovány následující postupy:

- **Šířka trasy** – šířka trasy pro kapacitní snímání by měla být volena co nejtenčí, doporučená je 5 až 7 mil (kapacitní vazba o polovinu tenčí trasy je poloviční)
- **Rozestup mezi trasami** - pro zachování kvality signálu je doporučeno dodržovat rozestup alespoň 10 mil pro paralelní trasy. Nejlepším přístupem je udržovat cesty co nejdále od sebe. V místě připojení tras k mikrokontroleru, kde se musí trasy k sobě přiblížit, se doporučuje uspořádání připomínající hrdlo láhve, viz obrázek 14.



Obrázek 14 Vyvedení tras od MCU [10]

- **Rozmíst'ování komponent** – při kapacitním snímání je nejlepší, když v okolí elektrod nejsou umístěny žádné komponenty. Dále je doporučeno vést spoje od elektrod v protilehlé vrstvě, než ve které se nachází snímací elektrody, ovšem vedení by se nemělo nacházet přímo pod jinými elektrodami. Elektrody musí být od sebe vzdáleny tak daleko, aby jejich poloha minimalizovala možnost nežádoucí detekce dotyku na více nezávislých elektrodách, pokud tento stav není pro danou aplikaci žádoucí.

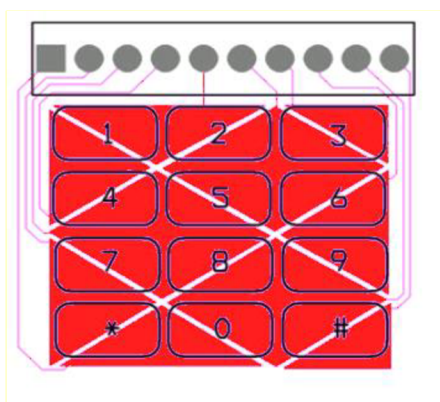
### 1.3.2 Návrh uspořádání elektrod

Stejně jako je možné použít různé mechanické ovládací prvky jako spínače, maticové klávesnice nebo Nkodéry, je možné tyto prvky nahradit kapacitními snímači.

Pro simulaci různých mechanických prvků je nutné různé množství a uspořádání elektrod. Knihovna NTouch umožňuje použít různé „controls“ bloky, které budou rozebrány dále:

#### 1. Klávesnice

Stejně jako u mechanické klávesnice, kde je z důvodu úspory pinů mikrokontroleru použito maticování, se stejného efektu dosahuje u kapacitního snímání, kdy detekcí dotyku dvou elektrod je jasně definována pozice. Rozložení elektrod, které je s výhodou využito pro eliminaci falešného doteku, je naznačeno na obrázku 15.



Obrázek 15 Motiv maticové klávesnice [1]

Použití modulu klávesnice je samozřejmě možné i v případě, že jsou jednotlivé klávesy od sebe vzdáleny, poté postačí jedna elektroda na klávesu. Ovšem vzdálenost těchto elektrod musí být dostatečná pro zamezení přeslechů mezi klávesami.

#### 2. Posuvník – „Slider“

Modul posuvníku umožňuje využít dvě varianty. Varianta na obrázku 16 dole je digitální posuvník. U digitálního posuvníku je možné detekovat směr pohybu prstu uživatele a pozici s jistým krokem. Horní část obrázku 16 zobrazuje analogový posuvník, kdy je pozice prstu dána osmibitovou hodnotou. Tento posuvník je možné použít jako náhradu potenciometru.

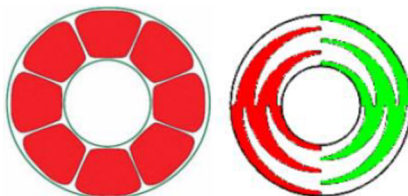


Obrázek 16 Analogový posuvník (nahore) a "digitální" posuvník (dole) [1]



### 3. Rotary

Uspořádání rotary je použitelné pro simulaci enkodérů, opět jsou v knihovně k dispozici dvě varianty – digitální a analogový. Oba typy dávají k dispozici informaci o doteku a směru pohybu. Analogový disponuje, stejně jako analogový posuvník, osmibitovou hodnotou pozice. Možné uspořádání elektrod ukazuje obrázek 17.

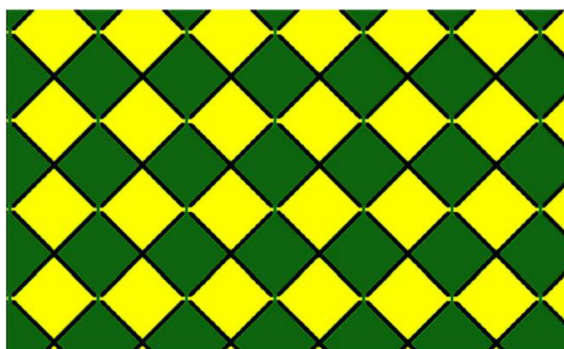


Obrázek 17 Digitální rotary (vlevo) a analogový rotary (vpravo) [10]

### 4. Matrix

Uspořádáním elektrod do mřížkové struktury, případně včelího plástu, je získána možnost detekce prostorové souřadnice. Algoritmus pro matrix control je v době psaní této diplomové práce ve fázi prototypování. Aktuálně poskytuje digitální pozici doteku jednotlivých elektrod, tedy XY souřadnici.

Možné uspořádání mřížkové struktury je na obrázku 18 níže, žluté elektrody jsou propojeny v řádcích, zelené elektrody jsou propojeny do sloupců.



Obrázek 18 Modul touchpadu [10]

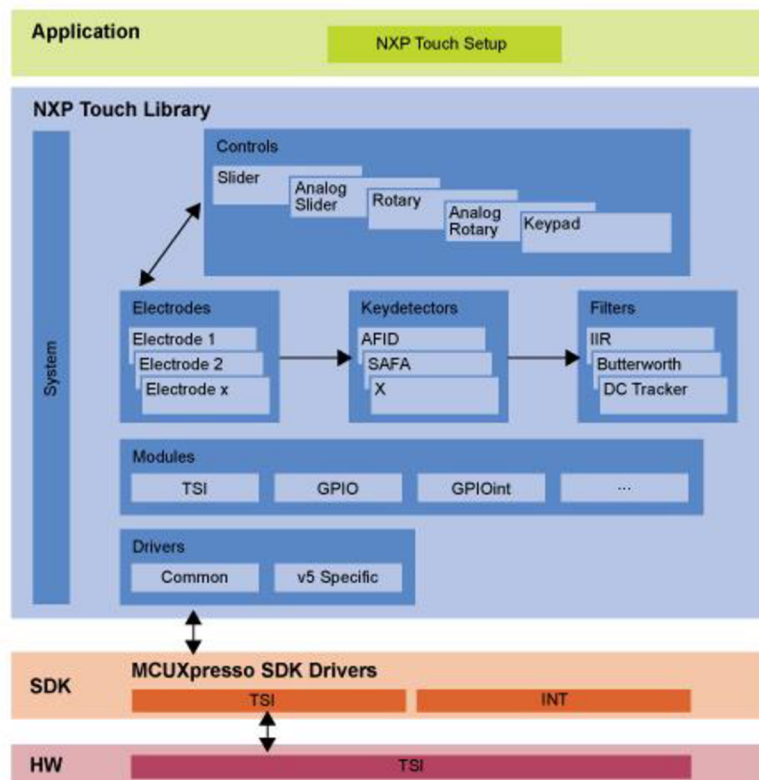
Velikost elektrod je nutné volit odpovídající snímanému objektu, tedy lidskému prstu, který by měl překrýt většinu plochy elektrody. V případě systémů, které se vyznačují více elektrodami na „klávesu“, je nutné, aby prostor jedné klávesy odpovídal snímanému prstu. U touchpadu je tato problematika rozebrána dále v kapitole 3.

Další upřesnění a podrobnosti je možné nalézt v [10].

## 1.4 Knihovna NTouch

Softwarová podpora kapacitního snímání je taktéž od firmy NXP k dispozici. V době psaní této práce (jaro 2018) je stále možné na stránkách NXP nalézt také starší a již nepodporovanou knihovnu TSS – Touch Sensing Software. Ovšem tato knihovna je v plné míře nahrazena knihovnou NXP Touch.

Knihovna NXP Touch je distribuována v podobě zdrojových kódů. Knihovna je volně ke stažení z webových stránek výrobce, viz [11]. NXP Touch je architekturou orientována objektově, k dispozici dává množství nastavení. Blokovou strukturu knihovny NXP Touch představuje obrázek 19.



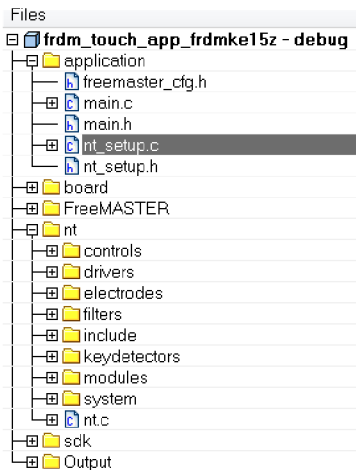
Obrázek 19 Bloková struktura knihovny NTouch [11]

### 1.4.1 Nastavení knihovny NTouch

V balíčku knihovny NTouch je taktéž sada demonstračních projektů, které je možné využít pro první seznámení s knihovnou.

Jak vyplývá z blokové struktury na obrázku 19 výše, do aplikace je nutné přidat soubory knihovny a také soubory pro nastavení, konkrétně `nt_setup.c` a `nt_setup.h`. Výsledná souborová struktura projektu pro knihovnu NXP Touch s mikrokontrolerem KE15z je zobrazena z prostředí IAR na obrázku 15:





Obrázek 20 Struktura projektu s knihovnou NTouch

Jak vyplývá z blokového diagramu (obr. 19), téměř veškerá uživatelská nastavení pro použití kapacitního snímání se odehrávají v souborech `nt_setup`. Při vlastní implementaci knihovny NTouch je výhodné použít jako výchozí bod tyto soubory z některého vzorového projektu, případně použít konfigurační nástroj ve FreeMASTERu, který je ovšem aktuálně ve vývoji a měl by být v červnu 2018 k dispozici v balíku NTouch na webu [11]. Při použití již existujícího souboru je nutné tento soubor uzpůsobit, základní úpravy jsou popsány dále.

Jako hlavní stavební kámen je struktura `nt_elektrode`, která definuje jednu klávesu, zápis této struktury je k vidění na následujícím zdrojovém kódu. Tímto způsobem je nutné nadefinovat veškeré použité elektrody v návrhu.

```
const struct nt_electrode electrode_0 =
{
    .pin_input = BOARD_TSI_ELECTRODE_1,
    .keydetector_interface = &nt_keydetector_safa_interface,
    .keydetector_params.safa = &keydec_safa,
};
```

Po definování elektrod je nutné konkrétní elektrody sdružit do modulu:

```
/* Modules */
const struct nt_electrode * const module_0_electrodes[] = {&electrode_0,
&electrode_1,
NULL};
```

Poté následuje struktura pro nastavení parametrů modulu, viz soubor `nt_setup.c` v projektových souborech v elektronické příloze této diplomové práce.

Další strukturou je struktura `nt_control`, která v sobě nese nastavení pro jednotlivé typy uspořádání elektrod (viz kapitola 1.3.2). Ukázka pro analogový posuvník je v následujícím zdrojovém kódu:

```

const struct nt_control aslider_0 =
{
    .interface = &nt_control_aslider_interface,
    .electrodes = control_0_electrodes,
    .control_params.aslider = &aslider_params,
};

```

Jelikož je možné mít v aplikaci více struktur `nt_control`, například více posuvníků, je nutné, byť i jediný posuvník umístit do pole:

```

/* System */
const struct nt_control * const controls[] = {&aslider_0, NULL};
const struct nt_module * const modules[] = {&tsi_module, NULL};

```

Následuje struktura `nt_system`, která sdružuje pole `controls` a `modules`. Některé struktury byly vynechány, podrobnější popis těchto struktur, ve kterých se jedná o nastavení samotné funkce knihovny a přidruženého hardwaru je možné nalézt v referenčním manuálu NXP Touch [1]. Tyto kroky výrazně ulehčí použití demo aplikací, které obsahují stejné controls moduly, jaké jsou zamýšleny v nově vyvíjeném projektu.

## 1.4.2 Použití knihovny NXP Touch

Po korektním nastavení HW v souborech `nt_setup.c` a `nt_setup.h` je ještě zapotřebí implementovat knihovnu do aplikace. Do projektu je nutné vložit veškeré zdrojové soubory knihovny plus výše zmíněné nastavovací soubory.

Pro korektní funkci periferie a knihovny jsou v hlavním programu nutné dvě funkce, a to „`nt_task()`“ a „`nt_trigger()`“, kdy „`nt_task()`“ je volán v hlavní smyčce programu neustále a je hlavní vykonávací funkcí knihovny NXP Touch.

Oproti tomu „`nt_trigger()`“ je nutné volat z přerušení časovače a zajišťuje získávání dat z TSI periferie. Tato funkce vrací „`NT_FAILURE`“, pokud je například vzorkování rychlejší než snímání pomocí TSI periferie. Čas, který je potřebný k získání jednoho vzorku z elektrod celého modulu, je závislý na počtu použitých elektrod a nastavení TSI periferie. Jednotlivé snímání kanálů je docíleno multiplexováním nabíjecího obvodu a elektrody jsou tedy snímány sériově.

Z toho důvodu je vhodné experimentálně nastavit čas trigrování, jelikož pokud vzorkování bude trvat jen o málo více, bude výsledek získán za dvojnásobnou dobu. V dostupných vzorových příkladech je k tomu to účelu využít PIT časovač.

```

/* Init PIT */
PIT_Init(PIT, &my_pit_config);
/* Set PIT0 period 32ms */
PIT_SetTimerPeriod(PIT, kPIT_Chnl_0, 5*CLOCK_GetBusClkFreq()/1000);
/* Enable PIT interrupt */
PIT_EnableInterrupts(PIT, kPIT_Chnl_0, PIT_TCTRL_TIE_MASK);
EnableIRQ(PIT_IRQn);
/* Start the PIT timer */
PIT_StartTimer(PIT, kPIT_Chnl_0);

```

Po jeho inicializaci je v jeho přerušení volána funkce „*nt\_trigger()*“. Pro ladění je vhodné využívat návratové hodnoty pro korektní nastavení doby vzorkování.

```
void PIT_IRQHandler(void)
{
    int32_t result = 0;
    /* Clear interrupt flag.*/
    PIT_ClearStatusFlags(PIT, kPIT_Chnl_0, 1);
    result = nt_trigger();
}
```

Taktéž samotná TSI periferie generuje při své činnosti přerušení, jedná se o následující vektor přerušení:

```
void TSI0_IRQHandler(void)
{
    TSI_DRV_IRQHandler(0);
}
```

Pro prvotní využití knihovny jsou distribuované příkladové projekty velice dobrým odrazovým můstkem, který poskytuje potřebnou funkční kostru projektu. Tento balík je dostupný z nástroje MCUXpresso SDK z [19].

## 2 USB

USB, tedy univerzální sériová sběrnice, je standardem, který definuje komunikační protokol, kabeláž a také napájení mezi počítačem a zařízeními. Je také univerzálním nástupcem řady sériových sběrnic jako paralelního a sériového portu, PS/2 a dalších. Hlavním benefitem oproti zmíněným standardům je funkce Plug-and-Play a možnost napájení připojených zařízení. V dnešní době se jedná o jeden z nejrozšířenějších standardů, který se ovšem dále rozvíjí a postupně jsou tímto standardem nahrazovány také rychlejší sběrnice.

### 2.1 Struktura USB

USB se vyznačuje stromovou strukturou. USB zařízení jsou spojena vždy v zapojení bod-bod, nejedná se tedy o klasickou sběrnici se společnými vodiči (oproti např. I<sup>2</sup>C).

USB obsahuje 3 typy zařízení:

- **USB hostitel (HOST)** – řídí veškerou komunikaci na „sběrnici“, skládá se z HW a ovladačů, bývá součástí operačního systému
- **USB zařízení (Device)** – zařízení připojené k hostiteli pomocí USB
- **USB rozbočovač (HUB)** – druh USB zařízení, který vytváří další přípojné místa ke sběrnici (uzly)

K jednomu hostiteli je možno připojit až 127 zařízení, do tohoto počtu se zahrnují i případné HUB zařízení. Díky tomuto je tedy možné vytvořit složitou strukturu limitovanou pouze délkami propojovacích kabelů.

### 2.2 Třídy zařízení

Všechny moderní počítačové myši posílají do počítače informaci o pohybu a stisku tlačítek, tiskárny přijímají data k tisku a do počítače posílají informace o svém stavu. Díky těmto podobnostem v zařízeních má smysl definovat společný protokol pro komunikaci těchto zařízení s počítačem. Na základě této úvahy vznikly třídy zařízení USB.

Třída zařízení USB je specifikací, kterou může výrobce při vývoji zařízení použít a získat tím výhodu kompatibility se systémem. Zařízení kompatibilní s některou ze tříd zařízení nepotřebují ke své funkci specifický ovladač dodávaný výrobcem, jelikož ovladače kompatibilní s třídami USB obsahují operační systémy. Příklad tříd zařízení je v tabulce 1 dále:

Tabulka 1 Příklad tříd USB zařízení

Třída	Descriptor	Popis
0x01	Interface	Audio
0x02	Both	Komunikační a CDC protokol
0x03	Interface	HID - pro interakci s člověkem
0x07	Interface	Tiskárny
0x08	Interface	Paměťová zařízení
0x09	Device	Hub
0x0E	Interface	Video
0x10	Interface	Audio/Video zařízení

Kompletní výčet dostupných tříd je možné získat z webu [usb.org](http://usb.org), viz [12].

## 2.3 Třída HID

Třída HID je jedna z prvních tříd podporovaných operačním systémem Windows. Je určena pro zařízení interagující s člověkem, patří zde například klávesnice, polohovací zařízení, herní konzole, čtečky čárových kódů a všechna další zařízení, která nemusí pracovat přímo s člověkem, ale možnosti pro přenos dat v této třídě jim vyhovují. Díky tomu je možné využít HID třídu taktéž pro různá zařízení pro posílání informací do hostitelského počítače.

HID zařízení mají určitá specifika a omezení. Veškerá data jsou ve třídě HID přenášena ve strukturách, které mají pevnou délku, zvaných reporty. Hostitelské zařízení posílá nebo si vyžádá report od HID zařízení. Formát reportu je závislý na typu přenášených dat, formát těchto reportů je popsán v tzv. *descriptorech*.

HID zařízení musí dále obsahovat právě jeden vstupní endpoint s přenosem v přerušení (endpoint je buffer sloužící k dočasnému uložení dat) pro posílání vstupních reportů a maximálně jeden výstupní endpoint pro přijímání dat od hostitele. Pokud zařízení vyžaduje více endpointů, je nutné jej konstruovat jako kompozitní zařízení s více HID jednotkami. O komunikaci v módu přerušení se jedná, protože HID zařízení potřebuje posílat data k hostiteli v předem neznámých časech, a jelikož komunikaci po USB řídí hostitel (popis viz [13]), nemá HID zařízení možnost si komunikaci vynutit, je nutné, aby se hostitel periodicky dotazoval zařízení na data.

### 2.3.1 HID descriptor

HID descriptor říká hostiteli vše, co potřebuje vědět o komunikaci se zařízením. Obsahuje tedy kompletní identifikaci zařízení a také popis struktur dat – reportů, která zařízení vyžaduje. Každé zařízení HID musí obsahovat obecný descriptor pro třídu HID, jehož součástí je specifický descriptor pro HID, descriptor pro vstupní, případně také výstupní endpoint v módu přerušení a descriptor s informací o formátu a použití dat v reportu.

Hostitel si pomocí „Get Descriptor request“ vyžádá postupně veškeré informace o zařízení. Přesný popis descriptorů a USB komunikace je možné nalézt v [13].

Výrobci čipů obvykle dodávají ke svým mikrokontrolérům demonstrační projekty pro použití USB. Firma NXP Semiconductors, jejichž produkty jsou v řešení této práce použity, není výjimkou. V knihovním balíku SDK je k dispozici vzorový projekt, který byl využit pro tuto práci. Descriptory pro HID zařízení jsou tedy k dispozici v elektronické příloze práce.

### 2.3.2 HID reporty

Report je descriptorem popsaná struktura, která nese data. HID descriptor reportu dává hostiteli informaci o struktuře dat, která HID posílá a přijímá. Descriptor tedy popisuje, jak bude budoucí zasláný report vypadat, jeho obsah a datové typy, jak jej má hostitel interpretovat.

Report může obsahovat struktury vlastností, jako jsou vlastnosti například tlačítka, přepínače a další prvky zařízení. Cokoliv dalšího jsou datové reporty.

K vytvoření descriptoru pro požadovaný report je možné použít utilitu od asociace USB-IF „HID Descriptor Tool“ který je dobrou pomůckou k vytvoření descriptoru (pokud není k dispozici descriptor ze vzorového projektu).

## 3 NÁVRH HW TOUCH PADU

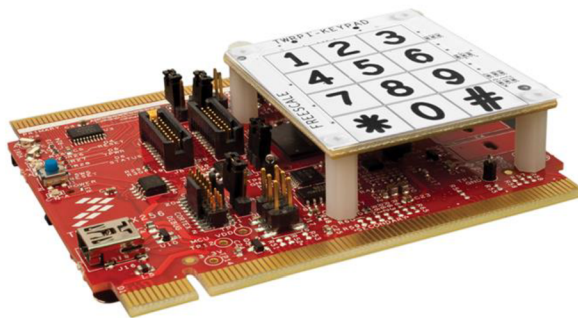
Touchpad je 2D polohovací zařízení, které snímá pohyb prstů uživatele po dotykové ploše a souřadnice předává do řídicího systému. Touchpad nemusí být pouze pro přenosné počítače, ale je využitelný pro širokou škálu zařízení, dotyková zařízení jako jsou chytré telefony, tablety a MP3 přehrávače využívají specifickou verzi touchpadu, kdy referencí pro uživatele není pohyb kurzoru, ale přímo pohyb prstu po displeji.

Návrh referenčního zařízení Touchpad je prováděn za účelem vytvoření demonstračního zařízení s mikrokontroléry Kinetis firmy NXP Semiconductors, z tohoto důvodu budou použity výhradně produkty firmy NXP Semiconductors.

### 3.1 Volba mikrokontroleru

Pro mikrokontroléry rodiny Kinetis dodává NXP demonstrační sady, tyto sady jsou k dispozici ve dvou platformách, a to Tower a Freedom. Tower je stavebnicová platforma umožňující připojování různých rozšiřujících kitů do postranních konektorů. Na obrázku 21 níže je vidět modulová tower deska s připojeným rozšiřujícím modulem dotykové klávesnice.

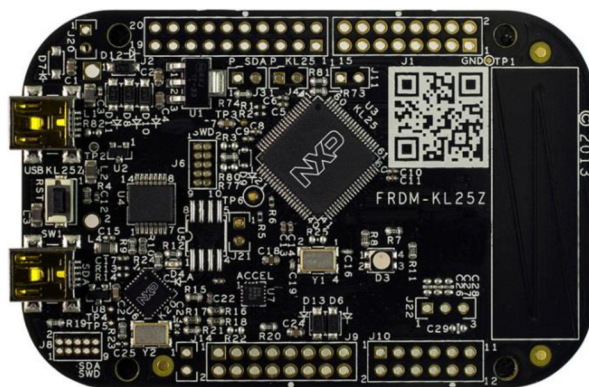
TWRPI-TOUCH-STR: Attached to Board



Obrázek 21 Platforma TOWER s TWRPI modulem [14]

Platforma Freedom je s rozměry přibližně odpovídající platební kartě menší než TOWER. K připojování rozšiřujících desek je možno využít Arduino kompatibilního uspořádání konektorů. Na obrázku 22 níže je zobrazena velice rozšířená verze FRDM-KL25Z.



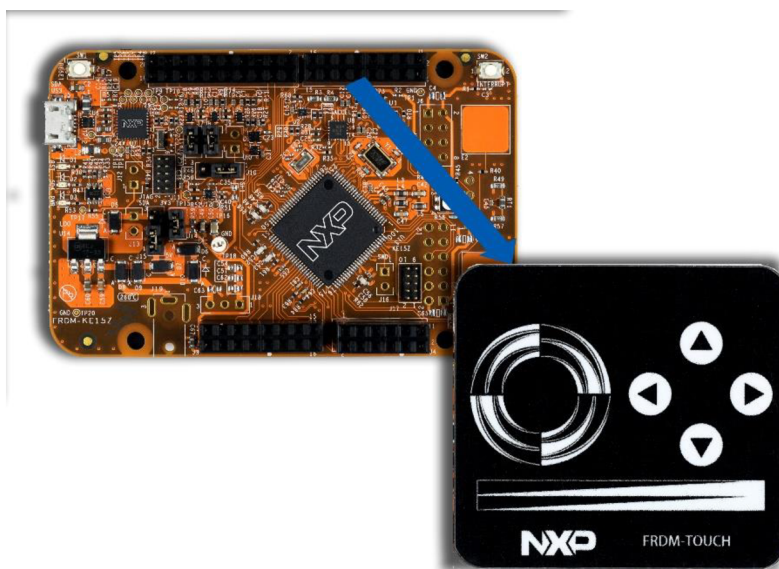


Obrázek 22 Vývojový kit FRDM-KL25Z [15]

FRDM-KL25Z obsahuje mikrokontroler KL25Z, jedná se o mikrokontroler z rodiny Kinetis s jádrem Cortex-M0+. Tento mikrokontroler je vybaven taktéž podporou USB na čipu, je tedy vhodný pro demonstraci funkce HID zařízení. KL25Z obsahuje 16 TSI kanálů díky periférii TSI verze 4.

Tato platforma byla v první fázi tohoto projektu dále použita pro demonstraci a ladění USB komunikační části. Jelikož hlavním problémem této vývojové desky je, že na Arduino kompatibilní konektory má vyvedeno pouze 10 TSI kanálů, maximální možná velikost touchpadu by tedy byla 5×5 elektrod. Ve druhé části je tato platforma využita pro jednu z verzí referenčního touchpadu pro USB komunikaci.

FRDM-KE15Z je taktéž vývojová deska platformy Freedom, dodává se spolu s dotykovým panelem, jak je možné vidět na obrázku 23 níže, a je zvolena jako hlavní platforma pro dotykové snímání s knihovnou NXP Touch.



Obrázek 23 FRDM-KE15Z s modulem NTouch [16]

KE15Z je taktéž mikrokontroler řady Kinetis s jádrem Cortex-M0+, ovšem není vybaven podporou USB. Pro HID zařízení je tedy nevhodný, nicméně účel



demonstračního touchpadu je firmou NXP dán, jakožto demonstrace možnosti procesoru KE15Z. Platforma FRDM-KE15Z má taktéž Arduino kompatibilní konektory, v základní verzi se do těchto konektorů umísťuje destička FRDM-TOUCH, jak je vidět na obrázku 23 výše. Tato platforma má na tyto konektory vyvedeno 20 TSI kanálů.

V přípravné fázi projektu byly využity výše zmíněné vývojové kity pro ověření funkčnosti zvolených metod. Ve druhé fázi byly výše zmíněné mikrokontroléry využity samostatně pro návrh konkrétních zařízení.

## 3.2 Návrh provedení touchpadu

Jak bylo zmíněno výše, touchpad je 2D snímací zařízení pro snímání pozice, případně pohybu prstů uživatele. Pro jeho konstrukci je využito dotykového snímání s periferií TSI. Jedna TSI periferie umožňuje zpracovat až 25 dotykových kanálů. Množství použitelných elektrod pro touchpad je tedy omezeno tímto počtem. Samozřejmě mikrokontroler technicky může obsahovat více TSI periferií, ovšem takovýto mikrokontroler není v aktuální nabídce firmy NXP Semiconductors (jaro 2018).

### 3.2.1 Návrh tvaru elektrod

Touchpad je zařízení, které dává systému informaci o XY pozici prstu. Z tohoto důvodu je nutné navrhnout uspořádání XY sítě elektrod tak, aby snímání bylo co nejefektivnější.

V přírodě nejefektivnější využití prostoru s téměř „kruhovou“ podstatou prezentují včelí plástve. Včelí plástve mají šestiúhelníkový tvar a tento tvar zaručuje maximální využití prostoru při potřebě vytvořit téměř kruhový půdorys. Ovšem pro návrh motivu plošného spoje by byl komplikovaný. Z tohoto důvodu byla zvolena „aproximace“ čtvercovým tvarem elektrod.

Výsledné uspořádání elektrod zobrazuje obrázek 24 níže. Z obrázku je patrné, že vodorovné elektrody jsou propojeny přímo v horní vrstvě a tvoří jednodolý blok. Svislé elektrody je nutné propojit ve spodní vrstvě.



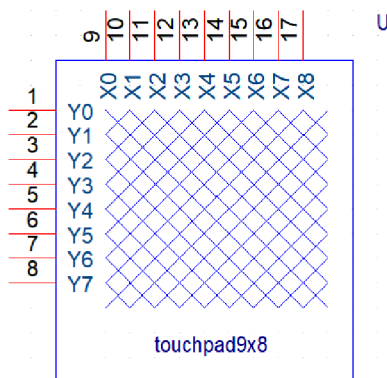
Obrázek 24 Uspořádání elektrod modulu touchpadu

Moduly elektrod jsou čtverce otočené o úhel 45°. Tento tvar byl zvolen z důvodu nutnosti detekce pozice prstu pomocí vyvolání doteku současně na X a Y elektrodě.

### 3.2.2 Návrh motivu touchpadu pro návrhový systém PCB

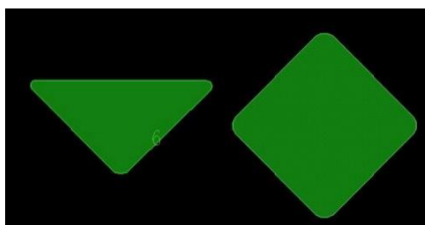
Po teoretickém návržení rozložení elektrod bylo nutné tento motiv přenést do návrhového prostředí Allegro pro návrh plošného spoje. Systém pro návrh plošného spoje vyžaduje vytvoření schématické značky nové součástky „touchpadu“ pro její umístění do schématu a „pouzdra“ pro motiv plošného spoje. Jako rozměr touchpadu byl zvolen širokoúhlý rozměr 9×8 (Š × V) elektrod.

Schématická značka touchpadu se skládá pouze z mřížky znázorňující maticové uspořádání elektrod a přípojovacích pinů s popisy X0 až X7 pro svislé elektrody a Y0 – Y8 pro vodorovné elektrody, viz obrázek 25 níže:



Obrázek 25 Schématická značka touchpadu

„Pouzdro“ touchpadu je pouze motiv elektrod, které budou prezentovány měděným povrchem. Tyto elektrody, jak bylo zmíněno výše, mají čtvercový tvar se zaoblenými rohy. Svislé elektrody (X0–X8) se skládají s jednotlivých čtvercových a trojúhelníkových dílů. Tvar těchto dílů ukazuje obrázek 26 dále.



Obrázek 26 Prvky svislé elektrody

Vodorovné elektrody Y0 až Y7 byly vytvořeny jako kompaktní bloky složené z osmi čtverců a dvou trojúhelníků, příklad takovéto elektrody je na obrázku 27 dále:

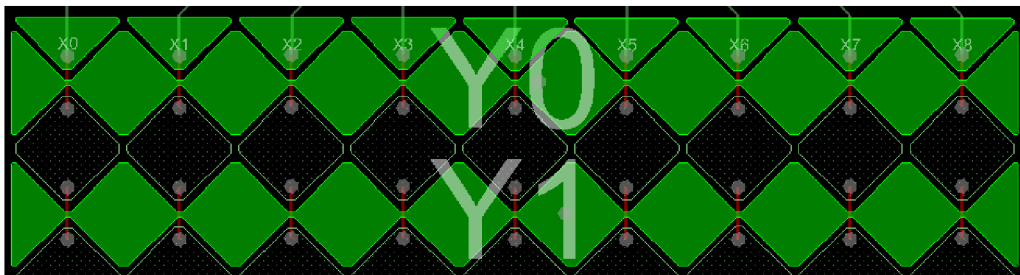


Obrázek 27 Motív vodorovné elektrody

Pomocí těchto stavebních prvků byl složen výsledný motiv touchpadu. Tento motiv elektrod byl následně uložen jako pouzdro součástky touchpad, které bylo spojeno se schematickou značkou.

Po vložení takto předpřipraveného pouzdra je nutné propojit svislé elektrody, jelikož „pinem“ svislé elektrody je pouze vstupní trojúhelníkový element. Pro plnou funkčnost je nutné pomocí příkazu „assign NET“ přiřadit i zbývajícím prvkům svislých elektrod příslušnou cestu (X0–X8) a propojit tyto prvky ve spodní vrstvě plošného spoje.

Detail výsledného propojení části pouzdra součástky je na obrázku 28 níže:



Obrázek 28 Detail zapojení svislých elektrod

### 3.2.3 Návrh velikosti touchpadu

Jak bylo již popsáno výše, pro větší citlivost je nutné volit větší elektrody, nejvíce efektivní je pro přímý dotyk elektroda, která přibližně odpovídá snímanému předmětu, v našem případě touchpadu se jedná o prst.

Při návrhu modulu touchpadu bylo po předchozí zkušenosti s malými elektrodami (které ale zase umožňují relativně velké rozlišení na malé ploše) jít cestou větších elektrod. Z principu fungování maticového snímání je pro korektní identifikaci pozice doteku nutné vyvolat „dotek“ na dvou elektrodách současně, přesněji na jedné X elektrodě a jedné Y elektrodě. Velikost elektrod tedy musí umožňovat jedním prstem překrýt více než jedno čtvercové pole. S ohledem na toto omezení byla volena výsledná velikost prvků elektrod.

Před finálním návrhem motivu touchpadu bylo experimentálně testováno několik papírových modelů s různými velikostmi prvků elektrod. Velikostí prvku elektrody se myslí velikost jednoho čtvercového prvku, ze kterých jsou jednotlivé elektrody složeny (viz obrázky 26 a 27 výše). Tyto papírové modely byly vytvářeny stejně jako výsledný funkční prototyp v návrhovém prostředí Allegro PCB designer 16.6.

Postupně byl navržen kompletní motiv touchpadu s úhlopříčkou čtverce 8 mm, 12 mm a 16 mm. Po konzultaci těchto tištěných prototypů s kolegy z týmu bylo rozhodnuto vyrobit verzi 12 mm, jelikož 16 mm verze již nezajistila spolehlivý dotyk svislé a vodorovné elektrody, 8 mm verze by sice umožňovala větší plošné rozlišení, ale s podobnou velikostí již byly z minula zkušenosti, a tak byla zvolena střední velikost.

### 3.3 Návrh plug in modulu pro FRDM-KE15Z

V době řešení diplomové práce se ve firmě NXP Semiconductors v Rožnově pod Radhoštěm pracovalo na implementaci knihovnic funkcí pro maticové uspořádání elektrod – matrix control. Pro potřeby testování těchto knihovnic funkcí a následné demonstrace funkčnosti bylo nutné navrhnout funkční HW.

Pro prvotní vývojové potřeby byla navržena dceřiná deska s Arduino kompatibilními konektory pro připojení k vývojovému kitu FRDM-KE15Z.

FRDM-KE15z byl vybrán z marketingových důvodů firmy NXP, kde rodina KE je aktuálně propagovanou rodinou mikrokontrolerů pro dotykovou knihovnu NXP Touch.

#### 3.3.1 Schéma zapojení modulu

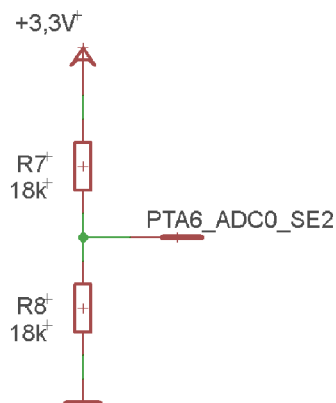
Modul byl navrhován ve vývojovém prostředí Allegro 16.6 od společnosti Cadence. Jelikož dceřiný modul touchpadu pro svou činnost nevyžaduje žádné pomocné obvody, obsahuje schéma zapojení pouze schématickou značku touchpadu, připojovací HDR konektory a pro případnou možnost detekce verze destičky je přidán odporový dělič. Kompletní schéma zapojení modulu je v příloze 1 této práce.

Jednotlivé elektrody jsou přímo napojeny na Arduino kompatibilní HDR konektory. Na tyto konektory má FRDM-KE15Z vyvedeny GPIO piny s možností připojení k TSI kanálům, využité piny a k nim připojené příslušné elektrody popisuje následující tabulka:

Tabulka 2 Zapojení elektrod modulu

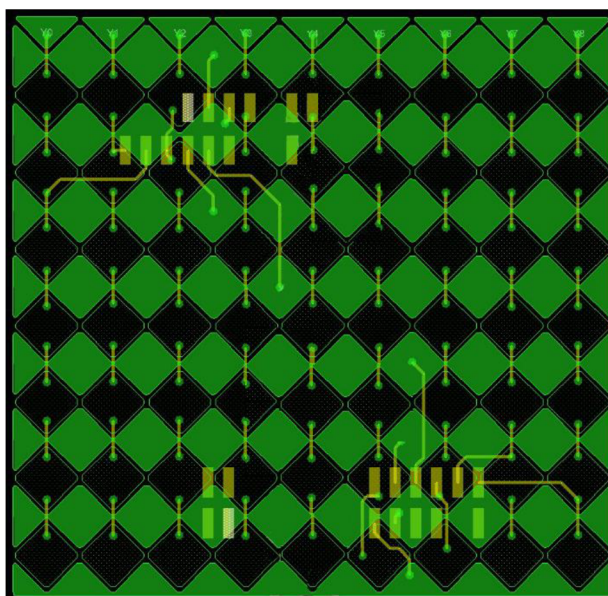
Elektroda	Port	TSI kanál	Elektroda	Port	TSI kanál
<b>Y0</b>	PTE0	CH13	<b>X0</b>	PTE10	CH2
<b>Y1</b>	PTE1	CH14	<b>X1</b>	PTE3	CH24
<b>Y2</b>	PTD6	CH7	<b>X2</b>	PTE11	CH3
<b>Y3</b>	PTD5	CH6	<b>X3</b>	PTE2	CH19
<b>Y4</b>	PTB5	CH9	<b>X4</b>	PTE8	CH11
<b>Y5</b>	PTE4	CH1	<b>X5</b>	PTE5	CH0
<b>Y6</b>	PTA1	CH18	<b>X6</b>	PTB4	CH8
<b>Y7</b>	PTA0	CH17	<b>X7</b>	PTD7	CH10
			<b>X8</b>	PTD1	CH5

Jelikož k platformě NXP Freedom existuje několik rozšiřovacích modulů, speciálně pro KE15Z výše zmíněný FRDM-Touch modul a nyní taktéž Touchpad bylo využito pinu PTA6, který je připojen k AD převodníku, kde je jednoduchým odporovým děličem zajištěna možnost detekování konkrétního modulu po připojení. Díky tomuto označení modulu je v případě potřeby možné demonstrační firmware v mikrokontroleru udržovat kompatibilní s oběma demonstračními moduly současně. Zapojení je na obrázku 29.



Obrázek 29 Napěťový dělič pro identifikaci modulu

Motiv plošného spoje dceřiného modulu je zobrazen na obrázku 30 níže:



Obrázek 30 Motiv dceřiného modulu touchpad s konektory pro FRDM KE15z

Výsledný motiv modulu má rozměry 75×73 mm, překrývá tedy svou velikostí platformu Freedom. Připojovací konektory tedy musí být umístěny na spodní straně plošného spoje – viz obr. 30 výše. Toto s sebou přináší omezení v podobě pozice konektorů, jejich pozice je pevně dána a díky tomu zasahují prokvy svislých elektrod do pinů konektorů.

Tento fakt byl eliminován úpravou připojovacích konektorů, kdy byly z jejich motivů vypuštěny nepoužité piny, které zároveň překážely routování. Výsledné tvary konektorů jsou patrné na obr. 30 výše. Jednalo se pouze o úpravu knihovního pouzdra pro patřičné součástky.

Kompletní výrobní podklady pro plug in modul jsou k dispozici v elektronické příloze této diplomové práce.

### **3.4 HID zařízení touchpad s KE15z + FRDM KL25z**

Jak již bylo zmíněno výše, v době vypracování této práce je jako hlavní demonstrační procesor firmou NXP pro technologii kapacitního snímání zvolen mikrokontroler Kinetis KE15z. Z tohoto důvodu došlo k návrhu referenčního touchpadu s tímto procesorem, ovšem aby bylo možné demonstrovat funkci HID zařízení, bylo ke KE15z nutné přidat druhý mikrokontroler, který je na rozdíl od KE15z vybaven podporou USB.

Variant bylo několik, aby byla zachována univerzálnost vytvářeného demonstračního HW, byl navrhnut systém připojení tohoto druhého mikrokontroleru pomocí Arduino kompatibilních konektorů ze spodní strany navrhovaného plošného spoje, do kterých může být nasunuta libovolná verze platformy Freedom obsahující USB periférii. Pro účely tohoto projektu je použita vývojová deska FRDM-KL25z. Tento systém proti přímému osazení umožní buďto využít pouze modul touchpadu s připojením přes sériovou linku například k real-time debuggeru FreeMASTER (viz dále kapitola 5), nebo naopak připojit jiný vývojový kit z platformy Freedom, například K64f pro využití Ethernetu nebo BlueTooth připojení.

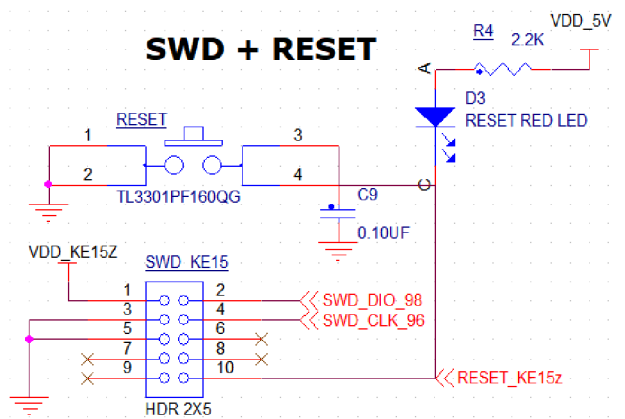
#### **3.4.1 Schéma zapojení touchpadu KE15z**

Schéma zapojení touchpadu neobsahuje příliš mnoho prvků, základním blokem je samotný mikrokontroler a jeho pomocné obvody.

Pro generování hodinového signálu pro mikrokontroler KE15z byl zvolen externí krystal 8 MHz. Tato hodnota krystalu je běžně dostupná, a právě tato hodnota je taktéž použita na vývojových kitech FRDM-KE15z, což případným uživatelům usnadní přenos aplikace z této vývojové platformy na demonstrační touchpad.

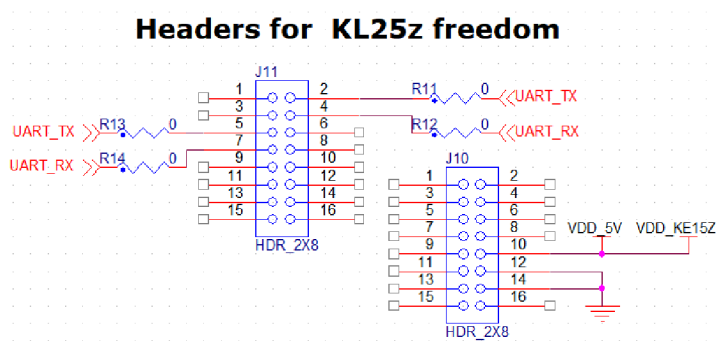
Pro naprogramování mikrokontroleru je potřeba externí debugger, pro tento debugger je připraven desetipinový konektor SWD. Spolu s konektorem SWD je na obrázku 31 níže zachycen obvod RESET tlačítka se signalizační červenou led diodou. Jelikož je toto schéma vyvíjeno pro interní potřeby firmy NXP Semiconductors, jsou schématické značky dle americké normy, z důvodů využití interní databáze firmy NXP a vývojového prostředí Allegro pro potřeby výroby.





Obrázek 31 SWD konektor a RESET obvod

Pro připojení komunikační platformy slouží Arduino kompatibilní konektory z platformy Freedom. Pro snazší nasunutí jsou využity pouze dva 16 pinové konektory. Zapojení konektorů ukazuje obrázek 32 dále:



Obrázek 32 Zapojení Arduino konektorů

Z obrázku 32 výše je patrná možnost volby, kam je připojen signál z mikrokontroleru KE15z. Pomocí osazení dvojice propojek (R11 a R12 případně R13 a R14) je možné zvolit mezi dvěma sériovými kanály, které jsou vyvedeny z platformy FRDM. Tyto kanály jsou připojeny na piny periferie LPUART0 mikrokontroleru KE15z, viz tabulka 3 dále.

Kromě samotného touchpadu a výše zmíněných nezbytných obvodů byly doplněny signalizační diody, a to dvě klasické a jedna RGB LED dioda, pro nezávislou komunikaci také druhá sériová linka. Zapojení těchto obvodů popisuje následující tabulka:

Tabulka 3 Zapojení pinů touchpadu s KE15z

<b>Pin MCU</b>	<b>Reference ve schématu</b>	<b>Popis</b>
PTB11	D1	Červená LED
PTB9	D2	Zelená LED
PTB16	D5	RGB – červená
PTB17	D5	RGB – zelená
PTA17	D5	RGB – modrá
PTC8	J11	LPUART0_RX
PTC9	J11	LPUART0_TX
PTA2	UART1-2	LPUART1_RX
PTA3	UART1-4	LPUART1_TX
-	UART1-1	GND
-	UART1-3	VCC

Sériový komunikační kanál LPUART1 je vyveden na čtyř pinový HDR konektor ze spodní strany plošného spoje. Tento kanál je určen pro vyvedení ladicích informací, primárně pro nástroj FreeMASTER (viz kapitola 5 dále) a umožňuje použít touchpad pouze pro demonstraci dotykového snímání bez HID připojení.

Samotný touchpad má stejný počet elektrod jako dceřiný modul pro KE15z, ovšem s ohledem na fyzické umístění elektrod a mikrokontroleru je zapojení elektrod odlišné. Zapojení elektrod k pinům mikrokontroleru KE15z je v následující tabulce:

Tabulka 4 Zapojení elektrod touchpadu s KE15z

<b>Elektroda</b>	<b>Port</b>	<b>TSI kanál</b>	<b>Elektroda</b>	<b>Port</b>	<b>TSI kanál</b>
<b>Y0</b>	PTC6	CH15	<b>X0</b>	PTB5	CH9
<b>Y1</b>	PTC7	CH16	<b>X1</b>	PTB4	CH8
<b>Y2</b>	PTA0	CH17	<b>X2</b>	PTD7	CH10
<b>Y3</b>	PTA1	CH18	<b>X3</b>	PTD6	CH7
<b>Y4</b>	PTE8	CH11	<b>X4</b>	PTD5	CH6
<b>Y5</b>	PTE3	CH24	<b>X5</b>	PTC1	CH23
<b>Y6</b>	PTE4	CH1	<b>X6</b>	PTC0	CH22
<b>Y7</b>	PTE5	CH0	<b>X7</b>	PTB3	CH21
			<b>X8</b>	PTB2	CH20

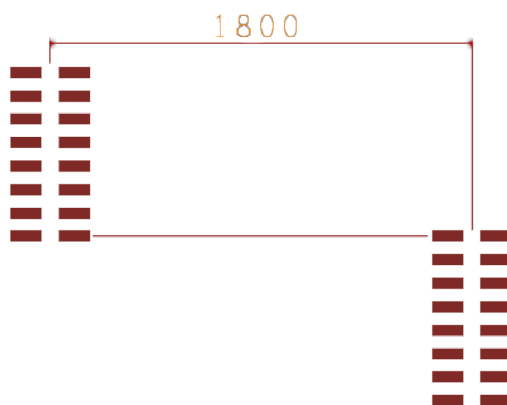
Kompletní schéma zapojení pro verzi touchpadu s KE15z + FRDM je k dispozici v elektronické příloze této práce.



### 3.4.2 Návrh plošného spoje touchpadu KE15z

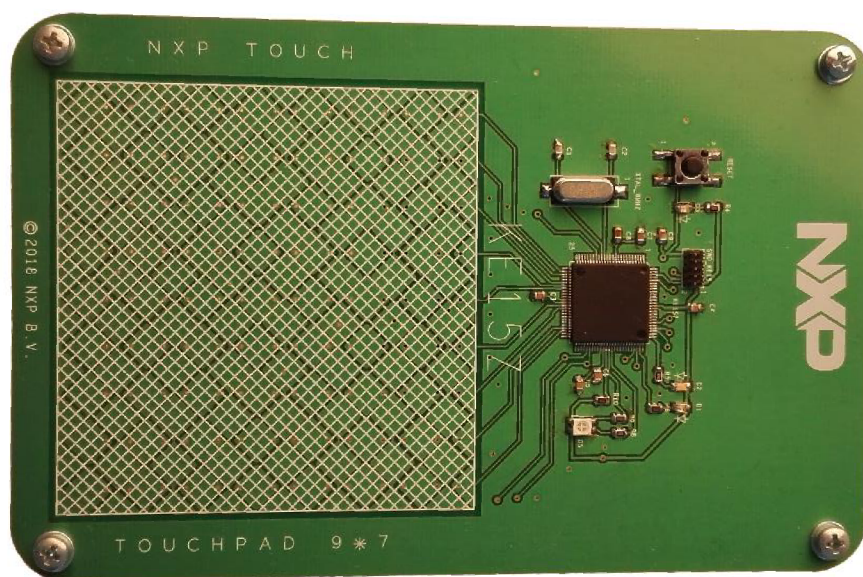
Při návrhu plošného spoje s možností připojení kitů platformy NXP Freedom ze spodní strany byl plošný spoj prodloužen tak, aby přípojovací konektory pro tyto kity nezasahovaly do aktivní oblasti touchpadu a zároveň byl kit skryt pod plošným spojem touchpadu. Díky tomuto prodloužení došlo k eliminaci problémů s překrýváním prokůvů svislých elektrod a pinů konektorů, jako tomu bylo u dceřiného modulu pro KE15z (viz kapitola 3.3.1).

Pro umístění 16 pinových konektorů na spodní stranu bylo nutné přesně dodržet rozměry, tyto rozměry byly získány z výrobních dat pro FRDM-KL25z, které jsou pro návrhový systém allegro volně dostupné na webových stránkách firmy NXP, viz záložka „Software & tools“ v [17]. Středové osy konektorů jsou od sebe vzdáleny v ose X 1800 mil. Jelikož mezera mezi 12 pinovým konektorem a 16 pinovým není 100 mil, ale pouze 50 mil, jsou první a poslední piny (1-2 a 15-16) konektorů osově zarovnané do shodné Y výšky. Rozmístění konektorů schematicky znázorňuje obrázek 33 dále:



Obrázek 33 Rozmístění Arduino kompatibilních konektorů

Veškeré součástky potřebné k funkci touchpadu jsou umístěny na horní straně plošného spoje, hotový modul touchpadu zobrazuje obrázek 34 dále. Ze spodní strany jsou umístěny pouze konektory pro připojení platformy NXP Freedom a dále je zde vyveden LPUART1 mikrokontroleru K15z pro připojení nástroje FreeMASTER (viz kapitola 5 dále).



Obrázek 34 Touchpad KE15z s možností připojení FRDM platformy

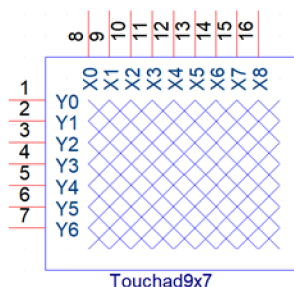
Kompletní data pro návrh plošného spoje jsou k dispozici v elektronické příloze této práce, data jsou vytvořena ve vývojovém prostředí Allegro 16.6. a Gerber souborech. Motiv plošného spoje je taktéž k dispozici v tištěné příloze č. 2. Osazovací plán vrstvy top je v příloze č. 3.

## 3.5 HID zařízení touchpad s KL25z

Pro vytvoření demonstračního zařízení s jedním mikrokontrolerem byl zvolen mikrokontroler KL25z. Tento mikrokontroler je vybaven USB periferií pro realizaci HID zařízení a také TSI periferií verze 4 s 16 dotykovými kanály.

### 3.5.1 Schéma zapojení touchpadu s KL25z

Původní modul touchpadu pro KE15z počítal s rozložením 9×8 elektrod, tedy využívá 17 pinů. Ovšem KL25z disponuje pouze 16 TSI kanály. Z tohoto důvodu musela být plocha touchpadu pro implementaci s KL25z zmenšena. Zmenšení bylo docíleno odstraněním elektrody X7 z původního motivu touchpadu. Vznikla upravená schematická značka, jak je ukázáno na obrázku 35 dále:



Obrázek 35 Schematická značka touchpadu 9×7

Zapojení elektrod k pinům mikrokontroleru je pro verzi s KL25z je shrnuto v tab. 5:

Tabulka 5 Zapojení touchpadu s KL25z

Elektroda	Port	TSI kanál	Elektroda	Port	TSI kanál
<b>Y0</b>	PTB1	CH6	<b>X0</b>	PTB2	CH7
<b>Y1</b>	PTB0	CH0	<b>X1</b>	PTB3	CH8
<b>Y2</b>	PTA2	CH3	<b>X2</b>	PTB16	CH9
<b>Y3</b>	PTA1	CH2	<b>X3</b>	PTB17	CH10
<b>Y4</b>	PTA4	CH5	<b>X4</b>	PTB18	CH11
<b>Y5</b>	PTC1	CH14	<b>X5</b>	PTB19	CH12
<b>Y6</b>	PTA2	CH15	<b>X6</b>	PTC0	CH13
			<b>X7</b>	PTA0	CH1
			<b>X8</b>	PTA3	CH4

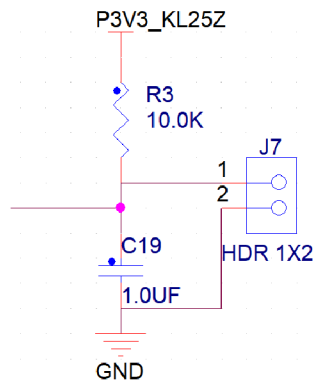
Při návrhu zapojení byl zjištěn trochu nelogický krok vývojářů čipu KL25z, kdy debugovací piny nejsou na dedigovaných GPIO pinech, ale na stejné piny jsou

vyvedeny i periferie. Pro návrh touchpadu je nutné využít všech 16 kanálů TSI periferie, z těchto 16 kanálů jsou dva společné s piny SWD, jak zobrazuje tabulka 6 dále:

Tabulka 6 PinMux SWD pinů [18]

Pin name	Default	Alt0	Alt1	Alt2	Alt3	Alt7
PTA0	SWD_CLK	TSIO_CH1	PTA0	-	TPM0_CH5	SWD_CLK
PTA3	SWD_DIO	TSIO_CH4	PTA3	I2C1_SCL	TPM0_CH0	SWD_DIO

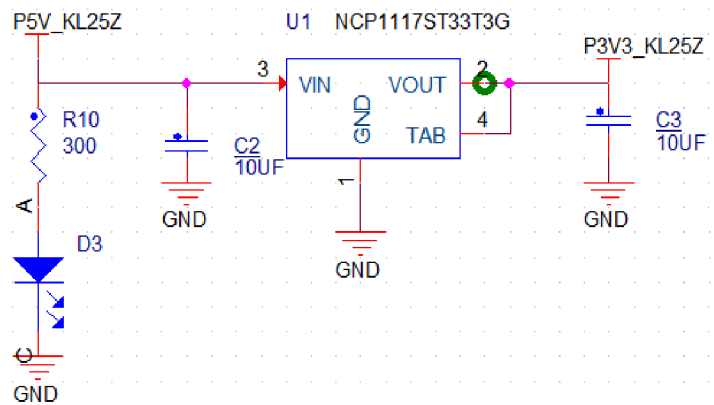
Z důvodu potřeby využití všech kanálů TSI periferie bylo nutné vyřešit přepínání mezi plným rozlišením touchpadu a možností debugování. Po resetu jsou piny nastaveny do režimu SWD i v případě, že byly předchozím programem přepnuty na TSI periferii, a tedy s opětovným naprogramováním mikrokontroleru není problém. Ovšem, jelikož se jedná o referenční zařízení, bylo nutné navrhnout řešení pro spolehlivé přepnutí touchpadu z provozního módu do ladicího. Jako řešení byl zvolen pomocný kontakt. Obvod zapojení pomocného kontaktu je zachycen na obrázku 36 dále.



Obrázek 36 Obvod pomocného kontaktu pro povolení ladění

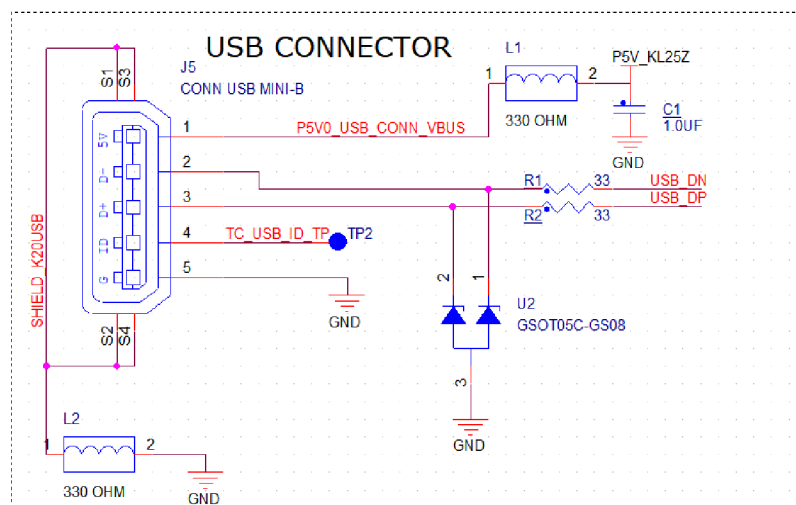
Tento pomocný kontakt je připojen k pinu PTD2. Po resetu je otestován a v případě jeho zkratování jsou piny při inicializaci ponechány v defaultním nastavení. Samotný touchpad bude zmenšen o elektrody X7 a X8. Tyto elektrody nejsou zapnuty a při inicializaci knihovny NTouch jsou vypuštěny také z popisujících struktur. Po opětovném resetu s vyjmutým jumperem J7 bude obnoveno plné rozlišení touchpadu. SWD konektor je ovšem s elektrodami fyzicky propojen a je tedy nutné při využití plného rozlišení zajistit odpojení externího debuggeru, který by svými pull up rezistory znemožnil využití těchto elektrod.

Celé zařízení touchpad KL25z využívá přímého připojení po sběrnici USB a je taktéž touto sběrnici napájeno. Napájecí napětí KL25z je 3,3 V, ovšem sběrnice USB poskytuje 5 V napájecího napětí. Pro úpravu napájecího napětí je využito lineárního stabilizátoru, jeho zapojení zobrazuje obrázek 37 dále:



Obrázek 37 Zapojení lineárního stabilizátoru pro KL25z

Jelikož mikrokontroler KL25z disponuje přímou podporou USB, je jeho připojení k počítači jednoduché. Obvody pro USB sběrnici obsahují pouze USB mini-B konektor, přepěťovou ochranu a ochranné rezistory datových linek. Kompletní zapojení viz obrázek 38 níže:



Obrázek 38 Zapojení USB mini B konektoru pro touchpad [17]

Pro signalizační funkci je verze s mikrokontrolerem KL25z vybavena shodně s verzí KE15z dvěma klasickými LED diodami, jednou RGB diodou a čtyř pinovým HDR konektorem pro vyvedení sériové linky. Využití piny mikrokontroleru shrnuje následující tabulka:

Tabulka 7 Zapojení pinů s mikrokontrolerem KL25z

<b>Pin MCU</b>	<b>Reference ve schématu</b>	<b>Popis</b>
PTE29	D1	červená LED
PTE30	D4	Zelená LED
PTC9	D5	RGB – modrá led
PTD0	D5	RGB – zelená led
PTD1	D5	RGB – červená led
PTD2	J7	Pomocný kontakt
-	J8-1	GND
-	J8-2	VCC 3.3 V
PTD6	J8-2	UART0_RX
PTD7	J8-4	UART0_TX

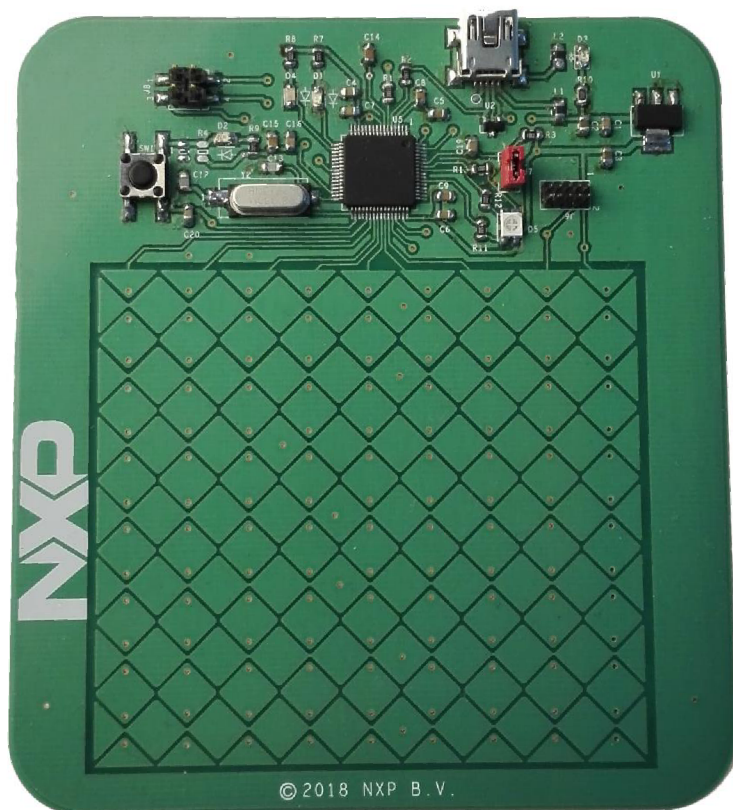
Zapojení jednotlivých elektrod k mikrokontroleru je v tabulce 5 výše. Verze touchpadu s KL25z je taktéž vybavena 10 pinovým SWD konektorem pro připojení externího debuggeru, který je pro naprogramování mikrokontroleru nezbytný. Zapojení pinů SWD konektoru je shodné se zapojením pro KE15z vyobrazeném na obrázku 31 výše.

Kompletní schéma zapojení je k dispozici pro vývojové prostředí OrCad 16.6 a v PDF v elektronické příloze této práce.

### 3.5.2 Návrh plošného spoje pro touchpad s KL25z

Před návrhem plošného spoje bylo nutné upravit „pouzdro“ touchpadu. Tato úprava spočívala v odebrání poslední vodorovné elektrody a úpravou zakončení svislých elektrod.

Plošný spoj je oproti verzi KE15z zmenšen o prostor potřebný pro překrytí platformy NXP Freedom. Rozměry plošného spoje jsou 97×107 mm. Výsledný plošný spoj ukazuje obrázek 39:



Obrázek 39 Touchpad s KL25z

Veškeré součástky jsou umístěny ve vrstvě TOP. Volná plocha je vylita plnou mědí využitou pro zemní plochu. Motiv strany top plošného spoje je k dispozici v příloze č. 4. Osazovací plán je k dispozici v příloze č. 5.

Strana bottom je využita pouze pro vedení propojovacích cest. Zbýlý prostor je využit pro zemní plochy, pouze prostor pod elektrodami touchpadu je vyšrafován křížovým motivem a v rozích připojen k zbylé zemní ploše. Strana bottom plošného spoje je v tištěné příloze č. 6. Kompletní výrobní podklady pro vývojové prostředí Allegro 16.6 a ve formátu Gerber je možné nalézt v elektronické příloze této práce.



## 4 NÁVRH SOFTWARE TOUCHPADU

Vývoj SW pro touchpad byl rozdělen do dvou větví. První větví byla úprava knihovního příkladu pro kompozitní USB HID zařízení a druhou větví implementace knihovny NTouch s pomocnými funkcemi pro funkcionalitu HID touchpadu.

### 4.1 Testování HID zařízení s FRDM-KL25z

Vývoj prototypového hardwaru pro touchpad byl uzpůsoben pro finální platformu KE15z, ovšem tento mikrokontroler nedisponuje USB periferií, z tohoto důvodu je pro prvotní otestování použita platforma jiná, a to FRDM-KL25z. Modul touchpadu pro FRDM-KE15z popsaný výše samozřejmě není svými vývody kompatibilní s FRDM-KL25z, nicméně pro vývoj USB komunikační části není toto omezení tak závažné, jelikož data o pozici, případně další povely, které bude ve výsledku generovat touchpad, je možné nasimulovat z jiných dostupných periferií.

Firma NXP Semiconductors dodává ke své rodině procesorů Kinetis soubor knihoven SDK. Aktuální verzi je možné stáhnout z webového nástroje MCUXpresso SDK z [19]. V těchto knihovních balíčcích je mimo jiné k dispozici sada demonstračních projektů pro USB komunikaci, není tedy nutné studovat od základů obsáhlou problematiku USB protokolu, ale je možné jako odrazový můstek využít těchto předpřipravených projektů.

#### 4.1.1 Volba typu USB zařízení

Jak bylo popsáno v kapitole 2, USB zařízení se dělí do tříd. Pro zařízení interagující s „pomalou“ periferií, v tomto případě myšleno s člověkem, slouží třída HID. V této třídě existují různá zařízení, logickou volbou pro touchpad je HID zařízení typu mouse – myš.

Knihovna NTouch umí v režimu snímání matrix rozpoznávání jak uživatelem zadaných gest, tak určitou sadu předpřipravených gest. Tato gesta jsou uživatelům známa z touchpadů notebooků. Jako možný příklad lze uvést scrollování, zvětšování a zmenšování obrazu, listování v položkách atd.

Pro scrollování by dostačoval typ myš. Kolečko je v základním popisu myši, ovšem pro implementaci listování v položkách je nutné simulovat stisknutí kláves. Například zmíněné zoomování je dostupné uživateli známou kombinací klávesy CTRL a pohybem kolečka na myši.

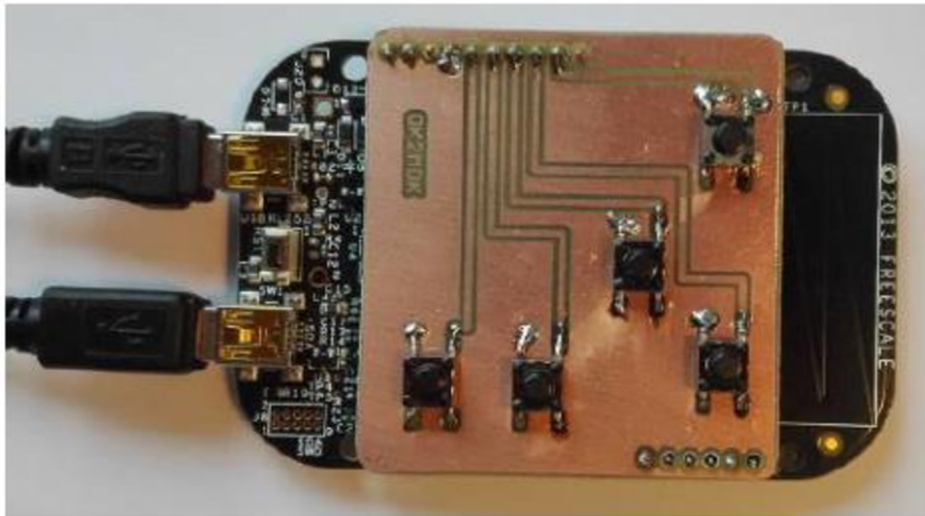
Z tohoto důvodu se musí touchpad hlásit systému jako kompozitní zařízení, které posílá nejen mouse reporty, ale v určitých chvílích taktéž klávesnicové reporty. Tento typ kombinované myši je ovšem relativně běžný, a tak i v knihovní sadě je možné najít demo aplikaci kompozitního zařízení v kombinaci myši a klávesnice.



#### 4.1.2 Prototypový testovací SW kompozitního HID zařízení

Pro otestování USB komunikační části byla zvolena platforma FRDM-KL25z. Bylo potřeba nasimulovat všechna vstupní data, která ve výsledném zařízení bude generovat touchpad. A to tedy XY pohyb myši, stisk tlačítek myši a další.

Za tímto účelem byla v první části projektu vytvořena jednoduchá rozšiřující destička pro FRDM-KL25z s pěti mikropínači, destička je na obrázku 40 dále. Pro definici pohybu kurzoru byl využit akcelerometr, který je na vývojové platformě k dispozici.



Obrázek 40 Testovací přípravek pro ladění USB

Tento prototypový modul byl využit v první části projektu pro testování USB a simuluje prezentační zařízení – kabelový prezentér. Pohyb kurzoru zajišťuje akcelerometr, miniaturní klávesnice s pěti tlačítky slouží pro zapnutí prezentace, pohyb mezi jednotlivými snímky, začernění prezentace a aktivaci ukazovátka na monitoru.

Tyto vlastnosti jsou klávesovými zkratkami a ve spojení s pohybem kurzoru tento testovací HW dobře pokrývá možnosti, které by měl výsledný touchpad umožňovat.

Celá aplikace pro otestování USB je postavena na demonstrační aplikaci z knihovního balíku SDK, konkrétně kompozitního zařízení myš a klávesnice. Pro její nasazení byl tento vzorový projekt očištěn od nadbytečného kódu, který je v tomto případě přítomen pro udržení univerzálnosti knihovních demo aplikací na široké škále kontrolérů.

Následně byla přidána obsluha akcelerometru a mikropínačů pro simulaci klávesnice. Výhodou volby FRDM-KL25z je, že takto předpřipravený kód bylo možné dále využít pro následnou implementaci na obou verzích touchpadu.

## 4.2 SW pro touchpad s KL25z

Verze touchpadu s mikrokontrolerem KL25z kombinuje funkce ze dvou mikrokontrolerů použitých ve verzi s mikrokontrolerem KE15z (viz dále kapitola 4.3). Na této verzi touchpadu probíhal hlavní vývoj a testování funkcionality touchpadu. Výsledná funkce byla poté přenesena na verzi touchpadu s KE15z.

### 4.2.1 Inicializace knihovny NTouch

Plug in modul touchpadu pro KE15z posloužil pro vývoj a testování maticového snímání elektrod – matrix control. Tento typ snímání byl z prvopočátku nasazen i na touchpad s KL25z. Ovšem při testování se ukázalo, že rozlišení, respektive jemnost pohybu, je při digitálním systému snímání příliš hrubá pro pohyb kurzoru myši po velké ploše počítačového monitoru. Pro realizaci základní funkce myši byl využit jiný princip snímání dostupný v knihovně, a to posuvník.

Knihovna NTouch má k dispozici dva typy posuvníku, digitální a analogový. Digitální posuvník se svou funkcí blíží principu maticového snímání – dává informaci pouze o poloze prstu uživatele na klávese a v poloze mezi dvěma klávesami. Kvůli této podobnosti by nedošlo ke kýženému zvýšení jemnosti pohybu kurzoru. Z důvodů tohoto požadavku byl nasazen posuvník analogový.

Analogový posuvník má svou parametrickou strukturu, viz dále:

```
struct nt_control_aslider aslider_params = {
    .range = 255, /*Maximum range for the ram->position value. */
    .insensitivity = 0, /* Set insensitivity to ignore changes less than 0.*/
};
```

Struktura umožňuje nastavit parametry aslider controlu, a to rozsah návratové hodnoty posuvníku a senzitivitu na změnu pozice. Vyhodnocování pozice probíhá neustále, hodnota necitlivosti pouze udává, nad jakou hodnotou bude vyvolána událost indikující pohyb posuvníku do hlavního programu.

Nastavení rozsahu posunu bylo zvoleno na maximální povolenou hodnotu, a to 255. (proměnná *range* je datového typu `uint8_t`). Tato maximální hodnota byla zvolena s ohledem na systém posunu kurzoru myši, kdy hodnota změny polohy je využita jako hodnota pro posun kurzoru myši.

Jelikož posun myši je 2D souřadnice, jsou na touchpad nasazeny dva analogové posuvníky. Jeden pro směr Y a druhý pro X. Z pohledu knihovny NTouch se jedná o dva nezávislé bloky. Samotný pohyb kurzoru je v HID reportu myši taktéž realizován dvěma nezávislými proměnnými.

V souboru *nt\_setup.c* jsou shrnuta veškerá nastavení pro dotykovou knihovnu. Pro každou z controls jednotek a také celý modul jsou složena pole s patřičnými elektrodami, ukázka kódu pro elektrody X je v kódu dále:

```
static const struct nt_electrode * const module_0_electrodes_x[] = {
    &electrode_X0, &electrode_X1, &electrode_X2, &electrode_X3,
    &electrode_X4, &electrode_X5, &electrode_X6, &electrode_X7,
    &electrode_X8, NULL};
```

Dále pro každý controls je nutné vytvořit popisnou strukturu. Příklad pro posuvník X je v následujícím kódu:

```
struct nt_control aslider_x =
{
    .interface = &nt_control_aslider_interface,
    .electrodes = module_0_electrodes_x,
    .control_params.aslider = &aslider_params,
};
```

Dojde tedy k základnímu naplnění informacemi o použitém rozhraní, použitých elektrodách a také k přiřazení struktury s nastavenými parametry pro posuvník.

Pro korektní uvedení do provozu je nutné ještě zaregistrovat funkce pro předávání informace o událostech jednotlivých posuvníků:

```
static void aslider_x_callback(const struct nt_control *control,
                              enum nt_control_aslider_event event,
                              uint32_t position);
```

Dále je po startu programu nutné inicializovat knihovnu NTouch, o tento úkol se stará funkce *NXPtouch\_init()*; v souboru *Touchpad\_KL25z\_aslider.c*, která inicializuje knihovnu, zapne všechny potřebné elektrody a také předá pointer na funkci *aslider\_callback* do patřičné obsluhy:

```
nt_control_enable(&aslider_x);
nt_control_aslider_register_callback(&aslider_x, &aslider_x_callback);
```

Veškerá nastavení pro dotykovou knihovnu jsou v souborech *nt\_setup.c* a *nt\_setup.h* dostupné v elektronické příloze této práce.

## 4.2.2 Řešení pro přepínání plné a „ladící“ velikosti touchpadu

Jak bylo zmíněno v kapitole 3.5.1, touchpad s mikrokontrolerem KL25z využívá veškeré dostupné piny TSI periferie, z důvodů umožnění plného ladění programu bylo nutné zajistit možnost přepínání mezi plným a redukovaným rozlišením. K indikaci, který mód je aktuálně (po resetu) požadován, slouží jumper J7 na plošném spoji, který je připojen k portu D, pinu 2. Pro snazší přehled jsou v programu nadefinována následující makra:

```
#define JUMPER_GPIO GPIO_D
#define JUMPER_PIN 2
#define DEBUG_DISABLE 0
#define DEBUG_ENABLE 1
```

Při inicializaci pinů na začátku funkce main dochází k detekci zapojení jumperu, jak popisuje následující kód:

```
    /*Debug jumper detect */
if(GPIO_ReadPinInput(JUMPER_GPIO, JUMPER_PIN) )
{
    jumper_on = DEBUG_DISABLE;
    BOARD_InitPins_DEBUG(); // Switch DEBUG pins to TSI!
} else {
    jumper_on = DEBUG_ENABLE;
}
set_sebug(jumper_on); /*Change settings of Ntouch structure*/
```

Pokud jumper není zapojen, dojde k zavolání dodatečné funkce BOARD\_InitPins\_DEBUG(), která přepne piny 0 a 3 portu A z defaultního nastavení na kanály TSI1 a TSI4.

Další fází je zavolání funkce set\_sebug(jumper\_on);, která zajistí naplnění struktur pro knihovnu NTouch korektními hodnotami:

```
void set_sebug(uint8_t debug_enable)
{
    if(debug_enable == DEBUG_ENABLE)
    {
        #if USE_MATRIX
            matrix_params.touchpad_size_horizontal = 7;
            matrix_params.touchpad_size_vertical   = 7;
            matrix_0.electrodes = module_0_electrodes_debug;
        #endif
        aslider_x.electrodes= module_0_electrodes_x_debug;
        tsi_module.electrodes = &module_0_electrodes_debug[0];
    }else
    {
        #if USE_MATRIX
            matrix_0.electrodes = module_0_electrodes;
            matrix_params.touchpad_size_horizontal = 9;
            matrix_params.touchpad_size_vertical   = 7;
        #endif
        aslider_x.electrodes= module_0_electrodes_x;
        tsi_module.electrodes = &module_0_electrodes[0];
    }
};
```

Jelikož zdrojové soubory pro touchpad mají sloužit jako demonstrační, je ve funkci zachována možnost nasadit na obsluhu touchpadu také control typu matrix. Jeho funkcionalita aktuálně není využita, ovšem pro jednoduchost využití je zde tento kód zachován a je možné jej pomocí makra *USE\_MATRIX* zapnout.

### 4.2.3 Realizace obsluhy pohybu myši

Základní funkcionalitou touchpadu je simulace funkce myši. Jak je možné znát z touchpadů pro notebooky, jedná se o pohyb kurzorem, levý klik, scrollování.

Výchozím bodem pro simulaci myši jsou výstupní návratové funkce z knihovny NTouch, a to pro každý posuvník zvlášť:

```
static void aslider_y_callback(const struct nt_control *control,
                              enum nt_control_aslider_event event,
                              uint32_t position)
```

Uspořádání elektrod počítá se situací, že při doteku jsou vždy ovlivněny jak řádkové, tak sloupcové elektrody, ovšem v režimu nasazení dvou nezávislých posuvníků je obsluhován každý směr pohybu samostatně. Analogový posuvník vrací tři typy událostí – prvotní dotek, pohyb a uvolnění všech elektrod. Těchto tří událostí je využito pro realizaci myši.

Ve funkci pro X a Y posuvník je interní statická proměnná udržující starší hodnotu pozice na posuvníku. Při vyvolání události „*NT\_ASLIDER\_ALL\_RELEASE*“ je do této statické proměnné uložena hodnota -1, čímž je indikováno, že předchozí pozice prstu není známa.

Při vyvolání události „*NT\_ASLIDER\_MOVEMENT*“ je nejdříve detekováno, zda je známa předchozí pozice prstu uživatele (tedy že  $x\_old > 0$ ). Principiální kód je dále:

```
case NT_ASLIDER_MOVEMENT:
    LED_TOGGLE(RGB_BLUE_GPIO, RGB_BLUE_PIN);

    if(!(x_old < 0)) /*Prevent invalid movement*/
    {
        hid_report.moveLR = (((int32_t)position)-x_old)*MULTIPLIER;
    }
    x_old = position;
    break;
```

Pokud je pozice prstu známa, je nejdříve otestována její diference, zda není mimo zvolenou mez, jelikož občas nastane situace s neplatnou detekcí doteku. Pokud tuto zvolenou mez posun nepřekročil, dojde k naplnění položky v globální struktuře `hid_report.moveLR` hodnotou diference mezi původní a novou pozicí, násobenou hodnotou definovanou makrem:

```
#define MULTIPLIER 2
```

Tato globální struktura je využita v souboru *mouse.c* ve funkci plnění HID reportu pro myš.

Obsluha pro události „*NT\_ASLIDER\_MOVEMENT*“ je komplexnější, v této hlavní události dochází k volání/obsluze také ostatních funkcionalit touchpadu, a to scrollování, zoomování a klikání myši. Popis viz dále.



#### 4.2.4 Realizace scrollování

Na touchpadech notebooků jsou známy dvě implementace scrollování. První verze je vyznačená část plochy na krajní, většinou pravé, straně touchpadu sloužící pro vertikální scrollování. Druhá verze využívá pro scrollování gesto, kdy je vyhodnocen dotek touchpadu dvěma prsty a následný pohyb v ose Y. V této diplomové práci byla zvolena druhá metoda, tedy metoda gesta.

Pro toto gesto je nutné mít možnost vyhodnotit pozici dvou doteků. Samotná komponenta analogového posuvníku nemá možnost určit pozici dvou prstů zároveň. U analogového posuvníku je tento dvojitý dotek nepovoleným stavem a knihovna umožňuje jeho detekci pomocí funkce.

```
uint32_t nt_control_aslider_get_invalid_position(const struct nt_control
*control);
```

Tato funkce vrací „1“ v případě, že došlo k „nepovolenému doteku“ – tedy v případě, že jsou detekovány doteky na více elektrodách, které jsou od sebe vzdáleny. V případě uspořádání touchpadu je tohoto využito pro detekci doteku dvěma prsty vedle sebe v ose X, což přesně odpovídá gestu známému pro scrollování.

Ve funkci pro obsluhu Y směru pohybu myši je přidána podmínka detekce dvojitého doteku na posuvníku pro detekování X pozice, jak zobrazuje následující kód:

```
if(nt_control_aslider_get_invalid_position(&aslider_x))
{ /*Scroll enable*/
hid_report.Scroll = -(((int32_t)position-y_old)/6;
LED_ON(USB_GREEN_GPIO,USB_GREEN_PIN);
LED_ON(USB_RED_GPIO,USB_RED_PIN);
LED_OFF(USB_BLUE_GPIO,USB_BLUE_PIN);
} else{ /*Normal MOVE*/
LED_TOGGLE(USB_RED_GPIO,USB_RED_PIN);
hid_report.moveUD= (position-y_old)*MULTIPLY;
}
```

Pokud je při detekci pohybu v ose Y detekován dvojitý dotek na posuvníku pro osu X, nedojde k posunu kurzoru myši, ale je naplněna položka struktury *hid\_report.Scroll*. Tato proměnná je potom v přerušení od USB naplněna do HID reportu.

Aby bylo scrollování přiměřeně rychlé, je rozdíl mezi aktuální a předchozí pozicí prstů dělen 6.

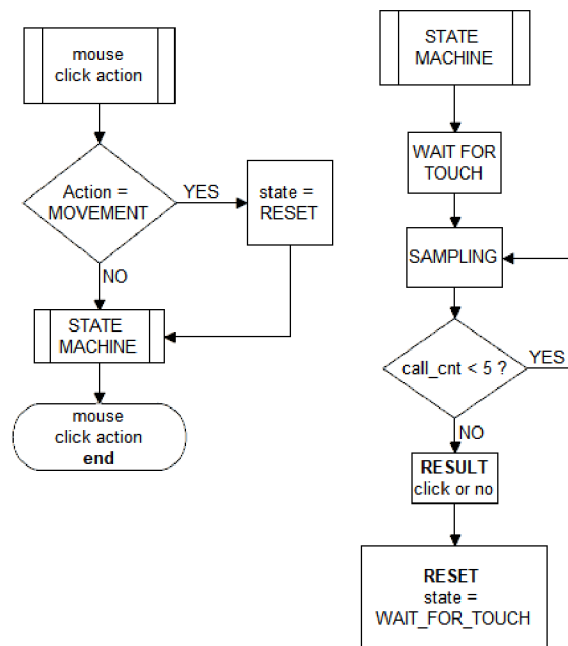
#### 4.2.5 Realizace stisku levého tlačítka

Levé tlačítko myši je na počítačových touchpadech realizováno krátkým dotekem. Stejná metoda detekce byla použita i při tvorbě tohoto referenčního touchpadu.

Jednoduchou metodou je nastavení horní hranice délky stisku touchpadu pro generování stisku tlačítka. Ovšem jelikož stejně jako mechanické tlačítko i dotykové snímání může vykazovat falešné uvolnění elektrod, je nutné zavést jisté ošetření zákmitů tlačítek. Pro jednoduchost implementace byla zvolena metoda využívajícího posuvního registru.

Pro detekci tlačítka slouží funkce *mouse\_click\_actions()*, která je periodicky volána z hlavní smyčky programu pokaždé, když jsou známa nová data z knihovny NTouch. Mimo toto základní časovací volání dochází k volání této funkce také z obsluhy posuvníku pro směr X z jeho události „*NT\_ASLIDER\_INITIAL\_TOUCH*“ a „*NT\_ASLIDER\_MOVEMENT*“. V těchto případech dojde k zavolání funkce „*mouse\_click\_actions()*“ s parametrem aktuální události na posuvníku. V této funkci dojde k nastavení prvního bitu posuvného registru a stavový automat přechází s dalším voláním z hlavní smyčky do stavu *SAMPLING*, kde dochází díky periodickému volání z hlavní smyčky ke vzorkování stavu doteku na posuvníku X. Viz zjednodušený vývojový diagram na obrázku 41 dále.

Po 5 vzorcích přechází stavový automat do stavu *RESULT* kde je vyhodnoceno, zdali jsou poslední tři vzorky nulové, tedy že posuvník X je uvolněn více jak cca 90 ms. Pokud ano, dojde k stisku tlačítka myši a funkce zavolá sebe sama, aby stavový automat ihned přešel do stavu *RESET*, kde dojde k vyresetování sekvence a čekání na první dotek posuvníku X.



Obrázek 41 Vývojový diagram funkce pro detekci kliku myši

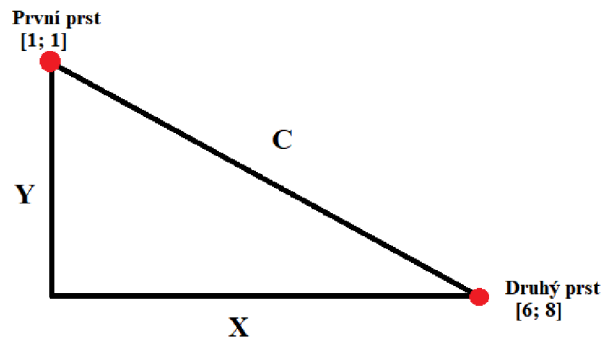
## 4.2.6 Realizace zoomovacího gesta

Pro zoomování je z dotykových zařízení známé gesto se vzájemným přibližováním, respektive oddalováním dvou prstů pro přiblížení, respektive oddálení prvku na monitoru. Pro scrollování postačilo využití detekce dvojnásobného doteku na posuvníku pro směr X, ovšem pro gesto zoomování je třeba přesně detekovat polohu těchto doteků a směr jejich vzájemného pohybu.

### Zoomovací gesto při detekci pozice dvou prstů

V případě využití algoritmu matrix je možné implementovat zoomování pomocí dvou prstů, jelikož matrix control dokáže poskytnout přesnou polohu pro dva prsty.

S výhodou je v tomto algoritmu využito pravoúhlého trojúhelníka a Pythagorovy věty, uspořádání prstů je na obrázku 42 dále.



Obrázek 42 Realizace dvouprstového zoomování

Ve funkci pro události matrix control jsou využity dvě události, a to „*NT\_MATRIX\_MOVEMENT*“ a „*NT\_MATRIX\_RELEASE*“. Kompletní obsluhu zoomování zobrazuje následující kód:

```
case NT_MATRIX_MOVEMENT :  
  
    if ( finger_position[1].x )  
    { // rectangular triangle  $c^2 = dx * dx + dY*dY$ ;  
      dx = finger_position[0].x - finger_position[1].x;  
      dY = finger_position[0].y - finger_position[1].y;  
      c2 = dY*dY + dx*dx;  
  
      if (c_old != -1)  
      {  
        LED_TOGGLE( RGB_GREEN_GPIO, RGB_GREEN_PIN );  
        hid_report.Scrool = (((int32_t)c2) - c2_old);  
        hid_report.keyboard_modifier_button = MODIFIERKEYS_RIGHT_CTRL;  
      }  
      c2_old = c2;  
    }  
    break;
```



```
case NT_MATRIX_RELEASE :
    c2_old = -1;
    break;
```

Jak je patrné z výše popsaného kódu, jedná se o periodické porovnávání přepony myšleného pravoúhlého trojúhelníka z obrázku 42 výše s její předchozí velikostí. Jelikož mikrokontroler KE15z nemá aritmetickologickou jednotku, je vynecháno zpětné odmocnění této hodnoty.

Rozdíl druhých mocnin přepon je následně naplněn do struktury hid\_report pro posun kolečka myši, jehož pohyb za současného stisknutí klávesy CTRL zajistí zoomování.

Tento přístup není v práci využit, v kódu zůstává pro demonstrační účely.

### Zoomovací gesto pro analogový posuvník

Jelikož analogový posuvník není schopen určit přesnou polohu dvou prstů, bylo nutné využít jiný přístup, kde pro jednoduchou implementaci zoomování jedním prstem byla inspirací implementace scrollování na některých touchpadech, kde se využívá krajní pozice. Pro zoomování na demonstračním touchpadu je využito podobného principu, ale na spodní hraně touchpadu, kdy pohybem v ose X dochází ke zvětšování/zmenšování obrazu, celá obsluha je tedy implementována v návratové funkci posuvníku X.

Pro realizaci tohoto gesta bylo nutné při doteku touchpadu detekovat, zdali došlo k doteku ve vybrané oblasti. Při vyvolání události „*NT\_ASLIDER\_INITIAL\_TOUCH*“ ještě není k dispozici spolehlivá informace o aktuální pozici posuvníku, je proto využita funkce pro detekci stavu elektrody:

```
int32_t nt_control_get_touch_button(const struct nt_control *control,
                                   uint32_t index);
```

Tato funkce při korektním volání vrací informaci o poslední dotknuté elektrodě, konkrétně její pořadové číslo. V případě, že je patřičný control bez dotknuté elektrody, vrátí „-1“. Touchpad s KL25z má v ose Y 7 elektrod, maximální návratovou hodnotou je tedy 6, pro snazší dosažení zoomovací funkce je nastaveno, že prvotní dotek musí být na sedmé elektrodě v ose Y. Tuto informaci zajišťuje výše zmíněná funkce, která je volána při každém vyvolání události na posuvníku.

Pokud je při události „*NT\_ASLIDER\_INITIAL\_TOUCH*“ na posuvníku X aktuálně dotknuta elektroda 7 na posuvníku Y, dojde k nastavení návěští „*zoom\_enable*“:

```
if (last_touched_electrode >5)/*If first touch is to ZOOM ZONE */
{
    zoom_enable = 1;
}else
{
    zoom_enable = 0;
}
```

Toto návěstí je následně testováno při události „*NT\_ASLIDER\_MOVEMENT*“ posuvníku X, jak ukazuje detail kódu dále:

```
if((last_touched_electrode > 5)&& zoom_enable)
{ /*If Yslider electrode 6 is currently touched = > ZOOM*/
  LED_TOGGLE(LED_GREEN_GPIO,LED_GREEN_PIN);
  hid_report.Scrroll = (((int32_t)position)-x_old)/6;
  hid_report.keyboard_modifier_button = MODIFIERKEYS_RIGHT_CTRL;

}else{/*If finger leaves ZOOM area, turn zoom off until next first touch*/
  hid_report.moveLR = (((int32_t)position)-x_old)*MULTIPLY;
  zoom_enable = 0;
}
```

Díky využití popsaného návěstí je možné pro pohyb kurzoru využít celou plochu touchpadu a pouze při prvotním doteku na spodní hranu dojde k aktivaci zoomovacího gesta. Stejně tak při opuštění spodní části touchpadu dojde k vypnutí tohoto gesta, a tak není nutné po přiblížení obsahu uvolnit touchpad. Pro obsluhu zoomování byla využita zkratka stisknuté klávesy CTRL se současným pohybem kolečka myši.

### 4.3 SW pro touchpad s KE15z + FRDM KL25z

Verze touchpadu s možností připojení platformy NXP Freedom obsahuje dvě nezávislé HW i SW části. Dceřiný modul platformy FRDM-KL25z obsluhuje pouze USB-HID a část s mikrokontrolerem KE15z komunikuje skrze sériové rozhraní UART. Veškerou funkcionalitu touchpadu tedy generuje mikrokontroler KE15z, do dceřiného FRDM modulu posílá pouze informace, které se mají naplít do USB reportů.

#### 4.3.1 SW pro KE15z

Jelikož software pro mikrokontroler KE15z obsahuje kompletní funkcionalitu demonstračního touchpadu, jaká je popsána v kapitole 4.2. výše pro verzi touchpadu s KL25z, jsou v této kapitole popsány pouze rozdíly obou obslužných programů.

Rozdíly jsou v zapojení elektrod a také v inicializaci knihovny NTouch, kdy Kinetis KE15z obsahuje periférii TSIv5, zatímco KL25z disponuje verzí TSIv4. Oproti verzi touchpadu s KL25z má KE15z touchpad ponecháno 9×8 elektrod, jelikož KE15z disponuje 25 TSI kanály. Ze stejného důvodu odpadá nutnost rozlišování plného a redukovaného rozměru touchpadu, a tedy obsluha jumperu zde není.

Zatímco verze touchpadu s KL25z využívala přímého USB připojení k počítači, verze s KE15z musí svá data posílat do dceřiného FRDM modulu pomocí sériové linky.

USB obsluha si vyčítá data z globální struktury a po vyčtení je zneplatní, respektive nastaví nulovou akci. Pro zachování tohoto přístupu bylo nutné zajistit synchronizaci odesílání dat přes sériovou linku s jejich zpracováním knihovnou NTouch. V hlavní smyčce programu je využito návratové hodnoty funkce *nt\_task()*; :

```

result = nt_task();
if(result == NT_SUCCESS)
{ // send to KL25z every new data
  mouse_click_actions(PERIODIC);

  printf("?L%04dU%04dS%04dB%03dK%03d@",
hid_report.moveLR,hid_report.moveUD,hid_report.Scrool,hid_report.mouse_button
,hid_report.keyboard_modifier_button);
/* after send, clear all data */
  hid_report.moveLR = STOP;
  hid_report.moveUD = STOP;
  hid_report.Scrool = STOP;
  hid_report.mouse_button = STOP;
  hid_report.keyboard_modifier_button= STOP;
}

```

Díky tomuto uspořádání dochází nejen k synchronizaci vzorkování posuvného registru pro detekci kliku tlačítka myši, ale hlavně odeslání platných dat v HID reportu vždy pouze jednou, což zaručí správnou funkci USB programu v FRDM-KL25z.

Zbývá obsluha touchpadu, jako posuvníky a funkce pro detekci stisku tlačítka myši, je shodná s verzí touchpadu s mikrokontrolerem KL25z.

Kompletní zdrojový projekt pro KE15z je k dispozici v elektronické příloze této práce.

### 4.3.2 SW pro FRDM-KL25z

Vývojový kit FRDM-KL25z má za úkol pouze přijímat pokyny od KE15z a tyto pokyny posílat jako USB HID reporty do počítače.

Pro posílání dat z KE15z do FRDM-KL25z se využívá paket, který má vždy následující strukturu:

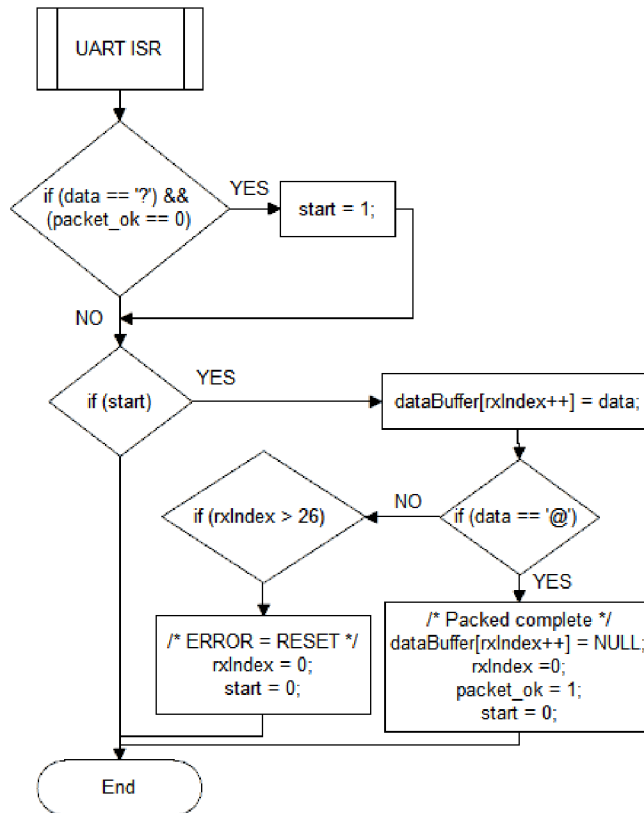
“?L±xxxU±xxxS±xxxBxxxKxxx@”

Kde:

- ? – synchronizační znak paketu
- L – znaménková int. hodnota pro pohyb kurzoru ve směru LR
- U – znaménková int. hodnota pro pohyb kurzoru ve směru UD
- S – znaménková int. hodnota pro pohyb kolečka myši
- B – číslo tlačítka myši, využito pro levý klik
- K – prostor pro poslání klávesové zkratky, využito pro CTRL
- @ – ukončující znak paketu

V přerušení od UART1 program čeká na příjem startovacího znaku paketu „?“, po jeho detekování začne ukládání přijatých dat, dokud není detekován znak „@“, případně pokud dojde k chybě, a tím pádem k přetečení povoleného počtu přijatých

znaků, je paket zahozen a čeká se na další příchozí. Celý postup prezentuje vývojový diagram na obrázku 43 dále:



Obrázek 43 Vývojový diagram příjmu paketu od KE15z

Po korektním přijetí paketu je nastavena proměnná packet\_ok, která předává informaci o korektních vstupních datech.

USB část, respektive její zdrojové soubory, je naprosto shodná s verzí touchpadu s mikrokontrolerem KL25z. Ve verzi touchpadu s KL25z jsou data do HID reportu plněna z globální struktury:

```

volatile struct hid_composite_t hid_report = {
    .moveUD = STOP,
    .moveLR = STOP,
    .Scrool = STOP,
    .mouse_button = STOP,
    .keyboard_button = STOP,
    .keyboard_modifier_button = STOP,
    .mouse_complete = STOP,
    .keyboard_complete = STOP
};
  
```

A tedy pro implementaci stejné USB obsluhy pro FRDM-KL25z postačí tyto data naplnit z přijatého paketu.

Pro rozřídění dat do globální struktury pro HID report je využita funkce *sscanf()*, která naplní jednotlivé proměnné z přijatého paketu:

```
/* Process data only when correct packet was received */
if (packet_ok == 1)
{
    result = sscanf(dataBuffer, "?L%4ldU%4ldS%4ldB%3huK%3hu@",
        &hid_report.moveLR, &hid_report.moveUD, &hid_report.Scrool,
        &hid_report.mouse_button, &hid_report.keyboard_modifier_button);
    packet_ok = 0;
}
```

Po rozřídění dat z paketu je nastavena proměnná *packet\_ok* do 0, což zajistí, že nedojde k příjmu dalšího paketu před dokončením zpracování předchozího.

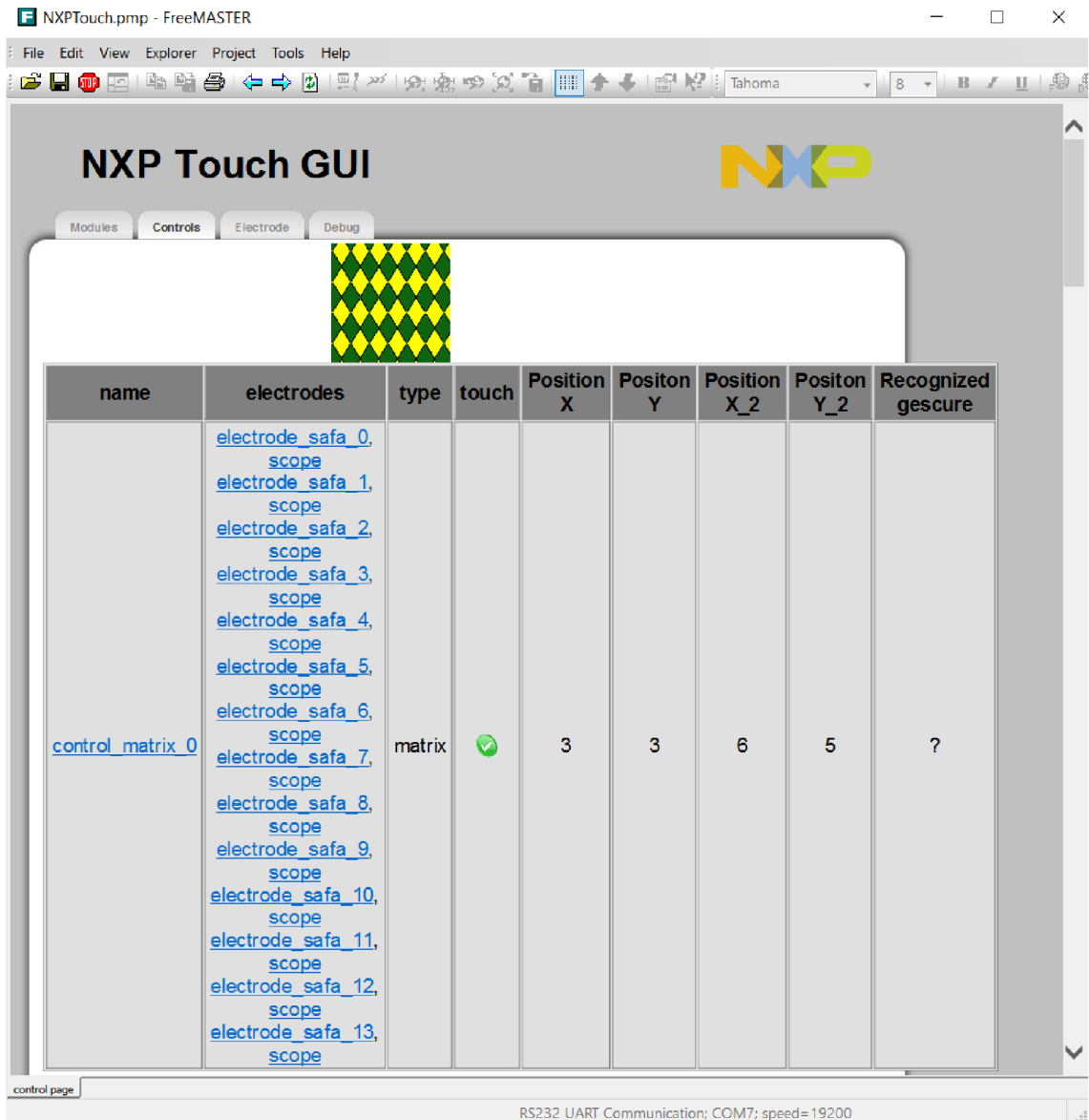
Tento výše zmíněný kód je zároveň jedinou náplní hlavní smyčky programu pro FRDM-KL25z.

## 5 NÁSTROJ FREEMASTER

Pro testování a ladění embedded aplikací existuje celá řada nástrojů, firma NXP používá pro své aplikace vlastní nástroj – FreeMASTER, jedná se o nástroj, který umožňuje vyčítat v reálném čase obsah paměti procesoru, tedy konkrétní proměnné, kromě tohoto umožňuje celou řadu vizualizací. Ty jsou k dispozici díky podpoře hypertextových stránek, kdy se vizualizace provádí pomocí java scriptu na straně počítače. FreeMASTER je volně ke stažení ze stránek NXP viz [20].

Tento nástroj je taktéž používán pro ladění aplikací s použitím knihovny NTouch. Demo aplikace v sobě obsahuje definiční soubory – TSA, které je FreeMASTER schopen číst pomocí virtuální sériové linky. Pro vizualizaci jsou spolu s demo aplikací dodávány taktéž zdrojové soubory pro FreeMASTER.

Jelikož byla funkcionality matrix control do knihovny nově dodávána, tyto soubory neobsahovaly možnost zobrazit informace z tohoto modulu. Z tohoto důvodu bylo nutné přidat definiční javascriptový soubor. Strukturou nový soubor vychází z ostatních modulů, pouze došlo k jeho předefinování pro čtení korektních dat. Spolu s úpravou javascriptových souborů musí korespondovat úprava TSA souborů v demonstračním projektu. Výsledná záložka controls pro matrix control aplikaci je zobrazena na obrázku 44 níže.



Obrázek 44 Záložka Controls ve FreeMASTERu pro MATRIX

Při návrhu bylo rozhodnuto, že bude obsahovat návěští, zda má některá elektroda detekován dotek, informaci o pozici prstů uživatele, a při uvolnění elektrod taktéž zda bylo rozpoznáno některé z nahraných gest (gesta rozpoznává mikrokontroler na základě knihovných funkcí).

První tři sloupce tabulky byly již předpřipraveny z jiných modulů. Pro získání proměnných s uloženou aktuální pozicí prstu byly napsány funkce v javascriptu pro vyčtení pozice X, Y a rozpoznávaného gesta. Příklad funkce pro Y pozici je vidět dále:

```
NTControl_matrix.prototype.GetPosY= function()
{
    if(!pcm_read_var(this._symbol + "->data.actual_position[0].y", 2))
        return 0;

    return pcm.LastVariable_vValue;
}
```

Tato funkce využívá již hotových funkcí, pouze se odkazuje na proměnnou `actual_position[0].y` s délkou 2. Tímto způsobem bylo nutné přistoupit ke každé zobrazené proměnné. Samozřejmě bylo nutné vytvořit na obrázku 44 zobrazenou tabulku, tato část je již definována pomocí HTML, ovšem její generování je opět provedeno pomocí javascriptu. Kód pro naplnění tabulky je demonstrován níže:

```
NTControl_matrix.prototype.GetGuiTag = function()
{
    var str = "";
    str = "<div ><center><img src=\"images/matrix.png\" width=120px
height=120px></center></div>";
    return str;
}

NTControl.prototype.GetGuiTags = function()
{
    var str = "";
    str = "<td><img id=" + this._gui_id0 + "Touch src=
\"images/icn_yes_hidden.png\" style=\"height:24px;width:24px;\"></td>"+
        "<td><span id=" + this._gui_id0 + "PosX?</span></td>"+
        "<td><span id=" + this._gui_id0 + "PosY?</span></td>"+
        "<td><span id=" + this._gui_id0 + "Gestcure?</span></td>";
    return str;
}
```

Celý javascriptový soubor je k dispozici v elektronické příloze této diplomové práce v rámci zdrojových kódů pro FreeMASTER.

## 5.1 Použití nástroje FreeMASTER

V knihovním balíku NTouch je dodáván také kompletní projekt pro ladění v nástroji FreeMASTER. Pro uživatelské použití je zásadní pouze soubor `NXPTouch`.

Při otevření souboru `NXPTouch.pmp` je nutné zkontrolovat nastavení vstupního COM portu ke kterému je připojen touchpad. Touchpad posílá data rychlostí 19200 Bd. Po tomto základním nastavení, je nutné kliknout na ikonku „stop“ která spustí/zastaví komunikaci. Po spuštění komunikace, si FreeMASTER vyčte z TSA tabulek potřebné údaje a zobrazí úvodní obrazovku se záložky „modules“, tato obrazovka pro Touchpad s KL25z je zobrazena v příloze č. 7.

Je možné na ní vidět signály z elektrod a jejich jednotlivé stavy.

V příloze č. 8. je zachycen stav záložky „Controls“ při scrollovacím gestu. Kdy je korektně detekován neplatný dotek na slideru X. Oba posuvníky mají dotek detekován a známou pozici prstu.



### 5.1.1 FreeMASTER na straně mikrokontroléru

Pro korektní funkci FreeMASTERu je nutné v aplikaci korektně inicializovat jednotku UART, toto nastavení musí korespondovat s nastavením knihovny pro FreeMASTER a to konkrétně v souboru `freemaster_cfg.h`, kde je definována rychlost a bázeová adresa jednotky UART. O inicializaci FMSTR a UART jednotky se stará funkce „`init_freemaster_uart(void)`“, která je dostupná v zdrojovém souboru „`Touchpad_KL25z_aslider.c`“.

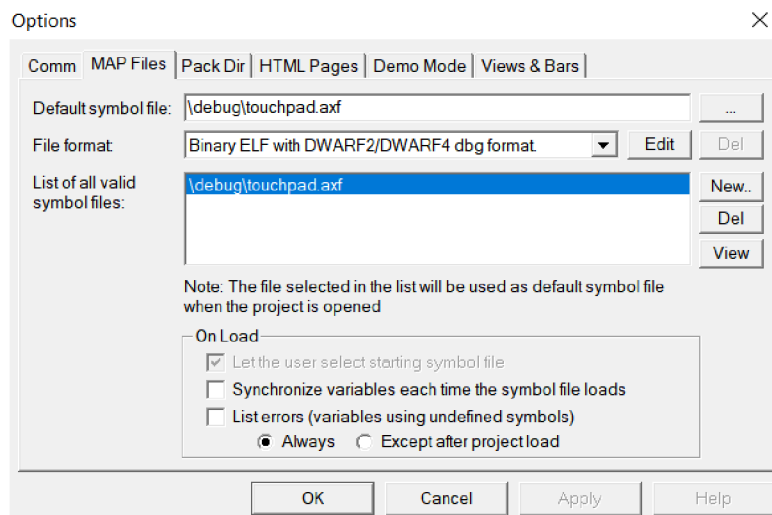
Kromě této inicializace, je nutné do hlavní smyčky umístit volání funkce „`FMSTR_Poll()`“ a do přerušeni od jednotky UART vložit volání obslužné funkce knihovny FreeMASTER:

```
void BOARD_UART_IRQ_HANDLER(void)
{
    FMSTR_Isr();
}
```

Zdrojové soubory pro referenční zařízení touchpad využívají předdefinovaných TSA tabulek pro knihovnu NTouch. Nicméně pro běžné použití existuje také jednodušší cesta využití nástroje FMSTR.

Pro implementaci je nutné v PC aplikaci FreeMASTER vložit výstupní soubor kompilovaného projektu (například `*.AXF`), díky kterému FreeMASTER identifikuje adresy jednotlivých proměnných a tyto proměnné je možné zobrazit.

Na straně mikrokontroleru je poté nutné pouze nainicializovat knihovnu FreeMASTER, s vypnutou možností využití TSA tabulek. Detail nabídky pro vložení výstupního souboru projektu je na obrázku 45 dále:



Obrázek 45 Nastavení výstupního souboru aplikace ve FMSTR

## 6 ZÁVĚR

V této diplomové práci byly prozkoumány možnosti návrhu touchpadu s pomocí technologie kapacitního snímání s využitím možnosti dotykové knihovny NXP Touch. Na základě tohoto šetření byl v první fázi práce navržen modul touchpadu pro vývojovou platformu FRDM-KE15z.

Pro tento modul, který sloužil pro potřeby vývoje funkcionality matrix control pro knihovnu NXP Touch, byly dále upraveny zdrojové soubory pro ladicí prostředí FreeMASTER a také odpovídající TSA soubory v demonstračním projektu knihovny pro zpřístupnění možnosti sledování parametrů touchpadu při ladění.

Dále byla v první části projektu na platformě FRDM-KL25z otestována implementace USB HID knihovny z knihovního balíku SDK2.2, jakožto kompozitního HID zařízení v kombinaci myši a klávesnice.

Na základě předchozích testů, byly navrženy dvě verze referenčního zařízení touchpad, a to verze s využitím mikrokontroleru Kinetis KL25z s přímou podporou USB a verze s mikrokontrolerem Kinetis KE15z, která vyžaduje ze spodní strany připojit platformu NXP Freedom pro podporu USB.

Při implementaci HID touchpadu bylo zjištěno, že digitální maticové snímání s využitím funkcionality matrix z knihovny NXP Touch není dostatečně jemné. Pro zlepšení této vlastnosti počítačového touchpadu byl na výsledné touchpady nasazen algoritmus posuvníku. Ovšem ukázalo se, že pro posuvník by bylo vhodnější navrhnout menší plochu elektrod, pro současný dotyk více řádkových/sloupcových elektrod. Což je požadavek, který jde proti požadavku platnému při využití maticového snímání, kde je požadavek pouze na společný dotyk na jedné X a jedné Y elektrodě.

Oba výsledné touchpady umožňují posun kurzoru myši, klik levého tlačítka myši, scrollování a zoomování. Použitá technologie vlastní kapacity není příliš vhodná pro implementaci ryziho počítačového touchpadu, nicméně bylo v této práci demonstrováno, že je možné s pomocí knihovny NXP Touch navrhnout a využívat touchpad pro různá embedded zařízení.

Výsledná navržená zařízení posloužila pro testování algoritmů maticového snímání při jeho vývoji a budou dále využita pro interní demonstrační účely knihovny NXP Touch na různých demonstračních akcích.

## LITERATURA

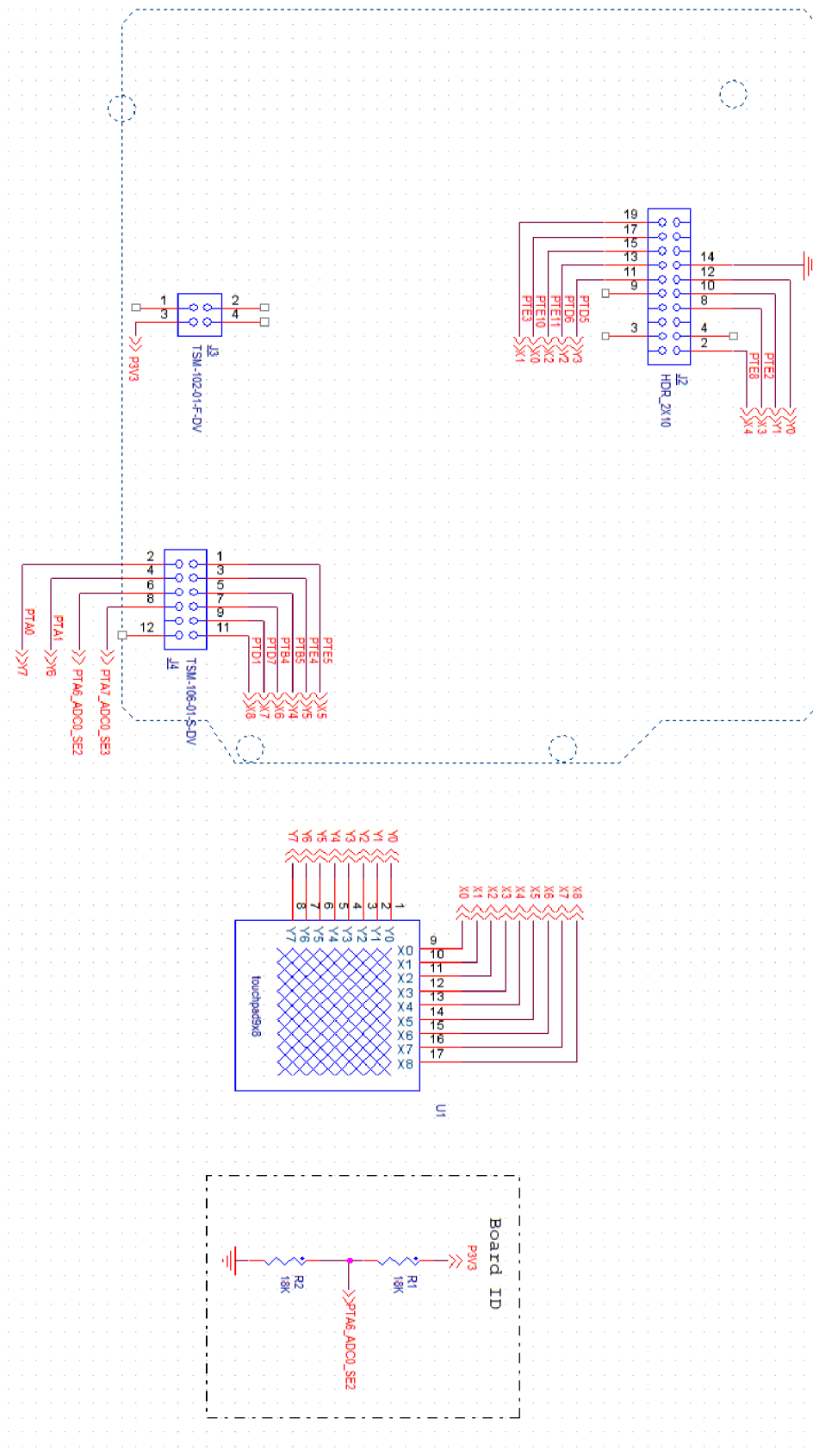
- [1] *NXP Touch Library Reference Manual* [online]. [cit. 2017-12-05]. Dostupné z: <https://www.nxp.com/docs/en/reference-manual/NT20RM.pdf>
- [2] *Parallel plate capacitor* [online]. In: . [cit. 2017-12-05]. Dostupné z: [https://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Parallel\\_plate\\_capacitor.svg/250px-Parallel\\_plate\\_capacitor.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Parallel_plate_capacitor.svg/250px-Parallel_plate_capacitor.svg.png)
- [3] COMERFORD, Richard. *Proximity sensing solution* [online]. 2013-09-19 [cit. 2018-04-13]. Dostupné z: <https://www.digikey.com/en/articles/techzone/2013/sep/proximity-sensing-solutions-part-1-capacitive-sensors>
- [4] MADAAN, Pushek a Priyadeep KAUR. *CAPACITIVE SENSING MADE EASY, Part 1*. [online]. April 2012 [cit. 2018-04-13]. Dostupné z: <https://www.digikey.com/en/articles/techzone/2013/sep/proximity-sensing-solutions-part-1-capacitive-sensors>
- [5] *KE15Z Touch sensing interface* [online]. Dec. 2016 [cit. 2018-04-13]. Dostupné z: <https://www.nxp.com/doc/KE15ZTSIUG>
- [6] GU, Holly a Chris STERZIK. *Capacitive Touch Hardware Design Guide, Texas Instruments* [online]. Nov. 2015 [cit. 2018-04-13]. Dostupné z: [www.ti.com/lit/an/slaa576a/slaa576a.pdf](http://www.ti.com/lit/an/slaa576a/slaa576a.pdf)
- [7] *Writing Touch Sensing Software Using TSI Module* [online]. [cit. 2017-12-05]. Dostupné z: <https://www.nxp.com/docs/en/application-note/AN4330.pdf>
- [8] GALDA, Michael. *NXP Touch LIBRARY v2.0*. Roznov p. Radh. 2017.
- [9] *E-field Basics Sensing Principle* [online]. [cit. 2018-04-13]. Dostupné z: <https://www.microchip.com/design-centers/capacitive-touch-sensing/gestic-technology/technology/sensing-method>
- [10] *Designing Touch Sensing Electrodes* [online]. [cit. 2017-12-05]. Dostupné z: [https://cache.freescale.com/files/sensors/doc/app\\_note/AN3863.pdf](https://cache.freescale.com/files/sensors/doc/app_note/AN3863.pdf)
- [11] Web knihovny NXT Touch, Dostupné z: <https://www.nxp.com/products/sensors/other-sensors/touch-sensors/nxp-touch-software:FREESCALE-TOUCH-SOFTWARE>
- [12] *USB Defined Class Codes* [online]. [cit. 2017-12-07]. Dostupné z: [http://www.usb.org/developers/defined\\_class](http://www.usb.org/developers/defined_class)
- [13] AXELSON, Jan. *USB Complete: The Developer's Guide*. 4. Madison: Lakeview Research, 2009. ISBN 13 978-1-931448-08-6.
- [14] Obrázek TWRPI Dostupné z : [https://www.nxp.com/assets/images/en/dev-board-image/TWRPI-TOUCH-STR\\_BD\\_2.jpg](https://www.nxp.com/assets/images/en/dev-board-image/TWRPI-TOUCH-STR_BD_2.jpg)
- [15] Obrázek FRDM KL25z Dostupné z : [https://www.nxp.com/assets/images/en/dev-board-image/FRDM-KL25Z\\_BD.jpg](https://www.nxp.com/assets/images/en/dev-board-image/FRDM-KL25Z_BD.jpg)

- [16] Obrázek FRDM KE15z Dostupné z :  
<https://www.nxp.com/assets/images/en/dev-board-image/FRDM-Touch-BOARD.jpg>
- [17] *FRDM-KL25Z overview* [online]. Dostupné z:  
<https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/kinetis-cortex-m-mcus/l-seriesultra-low-powerm0-plus/freedom-development-platform-for-kinetis-kl14-kl15-kl24-kl25-mcus:FRDM-KL25Z>
- [18] *KL25 Sub-Family Reference Manual* [online]. September 2012 [cit. 2018-04-13]. Dostupné z: <https://www.nxp.com/docs/en/reference-manual/KL25P80M48SF0RM.pdf>
- [19] *MCUXpresso SDK Builder* [online]. Dostupné z:  
<https://mcuxpresso.nxp.com/en/welcome>
- [20] Nástroj FreeMASTER Ke stažení z: <https://www.nxp.com/products/power-management/wireless-power/freemaster-run-time-debugging-tool:FREEMASTER>

# SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

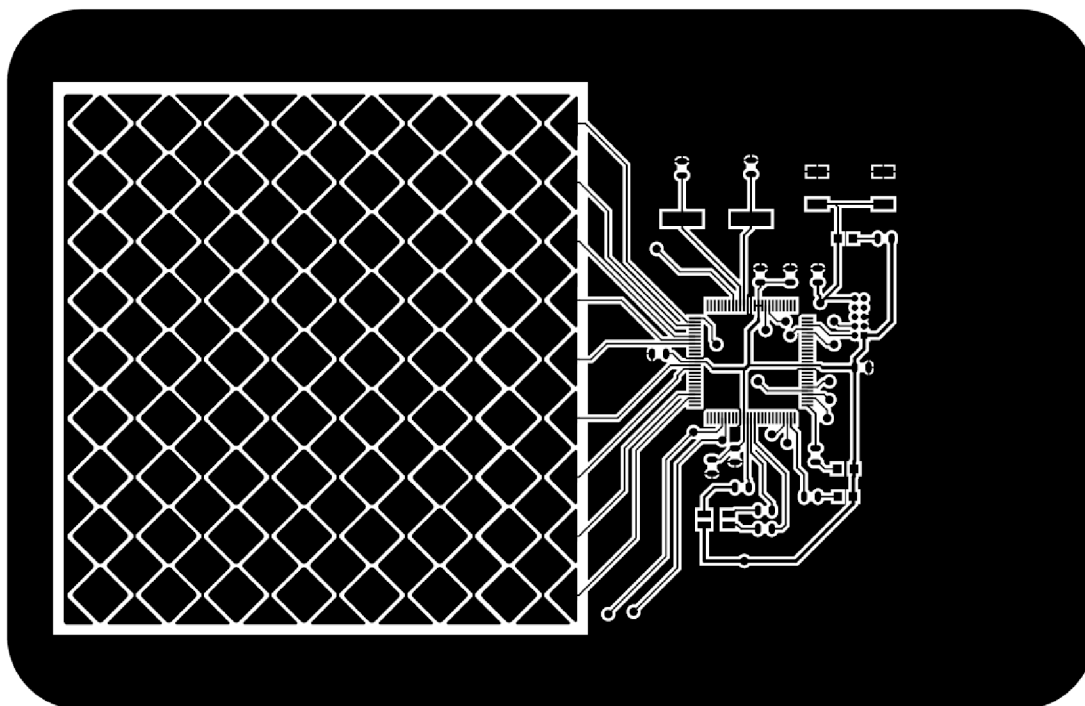
$C$	kapacita [F]
$A$	plocha elektrod [ $m^2$ ]
$D$	vzdálenost mezi elektrodami [m]
$k$	dielektrická konstanta materiálu mezi elektrodami
$\epsilon_0$	permitivita vakua ( $8,854 \cdot 10^{-12}$ F/m)
USB	Universal serial bus = univerzální seriová sběrnice
HID	Human Interface Device – zařízení pro ovládání člověkem
NXP	Zkratka společnosti NXP Semiconductors
MP3	Kodek pro komprimaci hudby
BT	BlueTooth komunikační platforma
GPIO	Genera Purpose Input Output – Vstupně/výstupní porty pro obecné použití
TSS	Touch Sensing Software – již neaktuální knihovna pro kapacitní snímání
NTouch	Aktuální knihovna pro kapacitní snímání od firmy NXP. Dostupná z [6]
TSI	Touch Sense Input – periferie pro přímou implementaci kapacitního snímání
PIT	Periodický čítač generující přerušení
FRDM	Platforma vývojových kitů firmy NXP
AD	Analogově-digitální převodník
SDK	Software Development Kit – sada knihoven pro rodinu mikrokontrolerů Kinetis od firmy NXP
FreeMASTER	Run-time ladící prostředí od firmy NXP. Dostupné z [12]

# Příloha 1 Schéma zapojení modulu pro KE15z

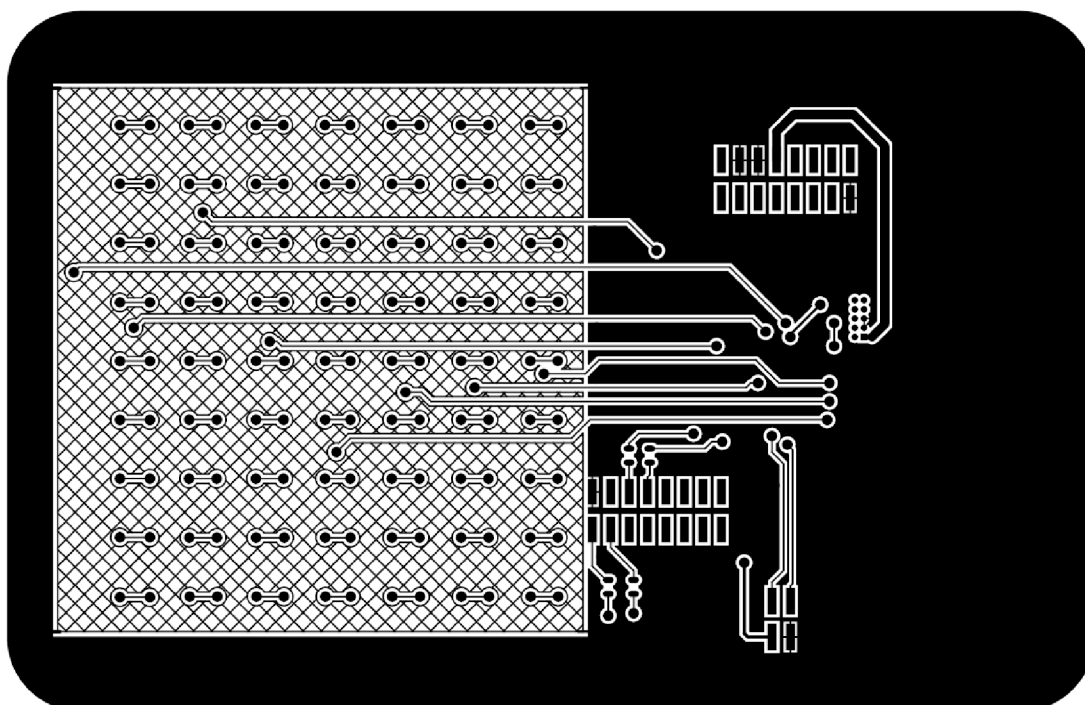


## Příloha 2 Motiv plošného spoje touchpadu s KE15z

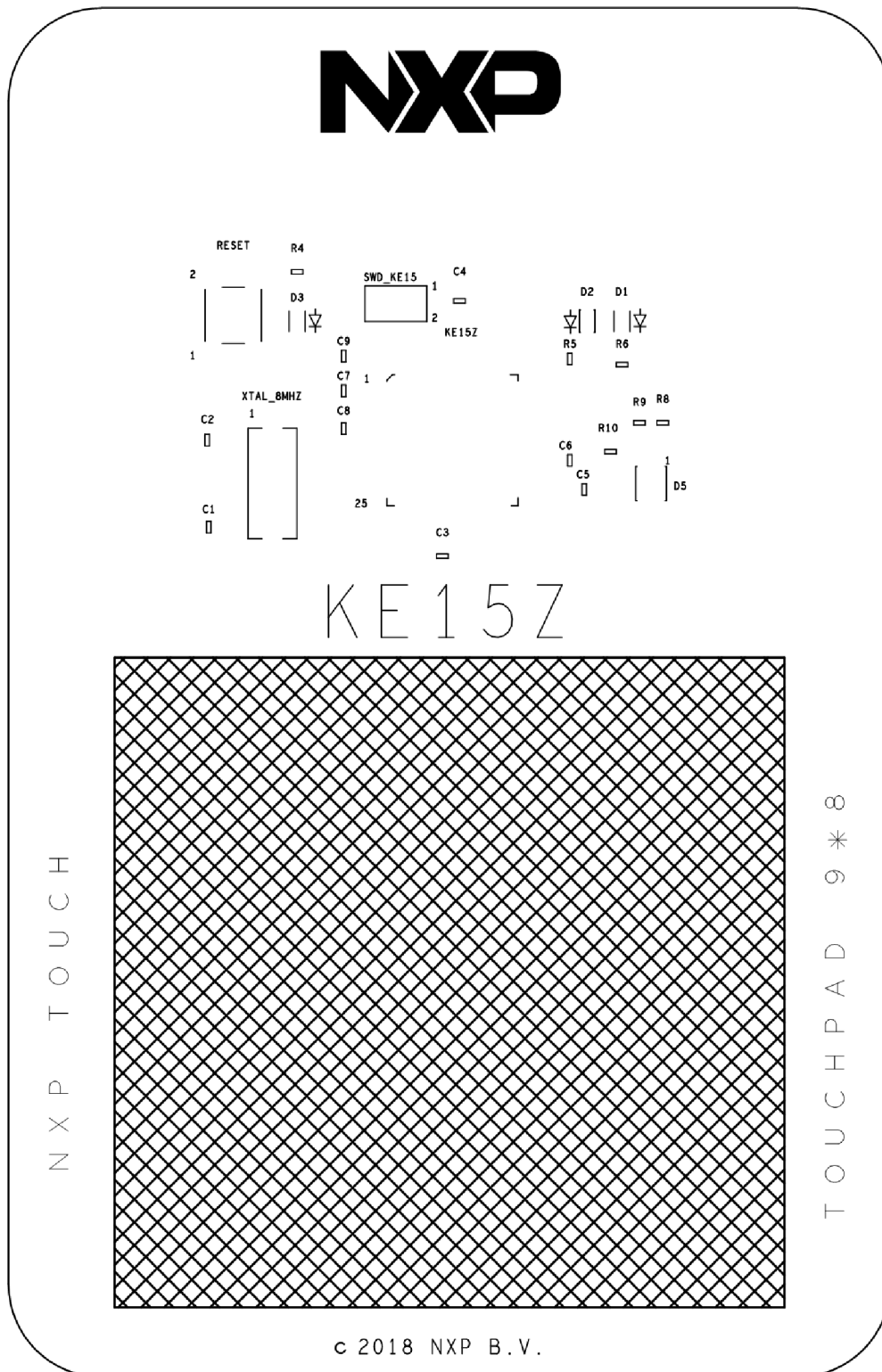
TOP



BOTTOM:



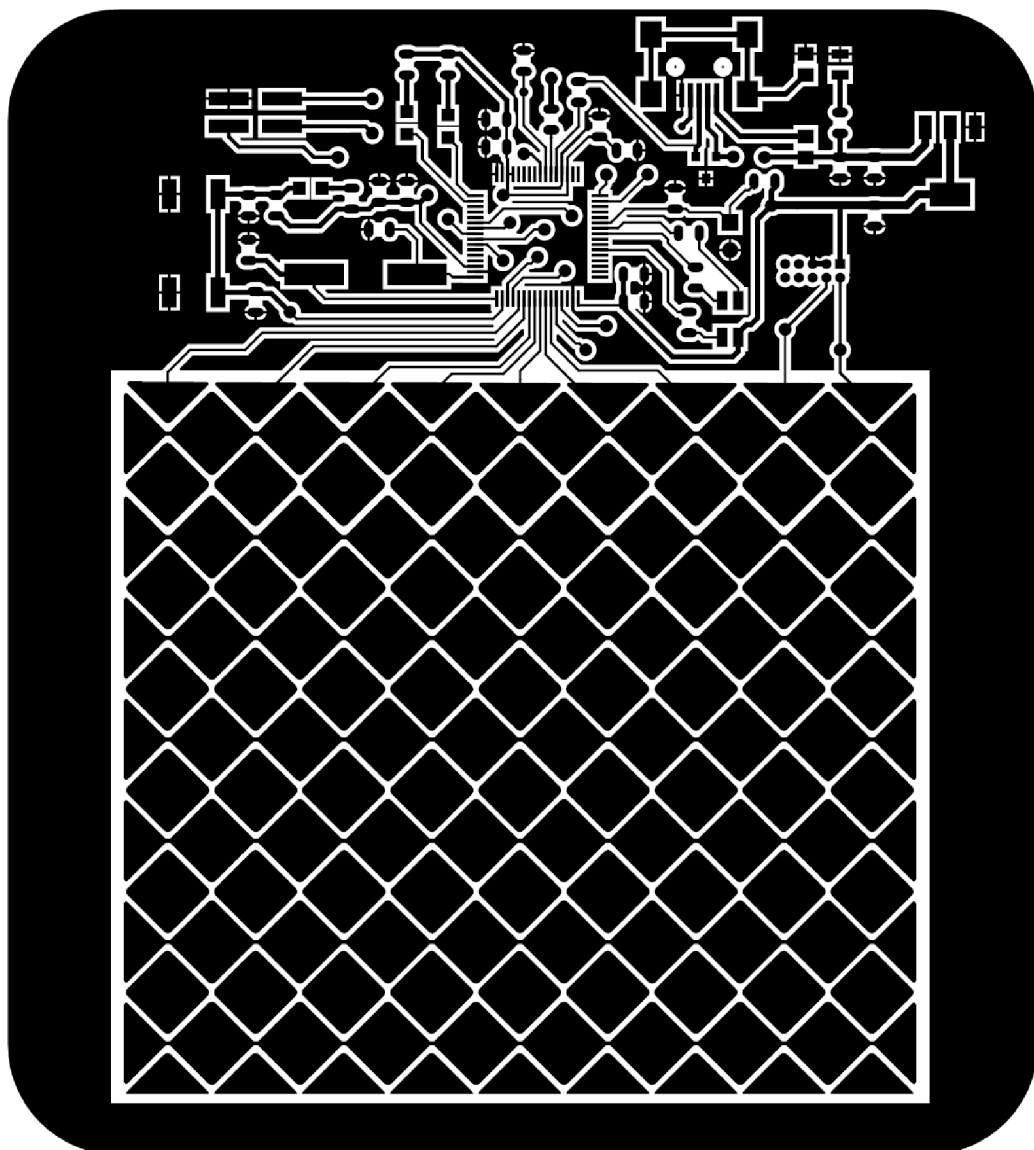
### Příloha 3 Osazovací plán touchpadu s KE15z



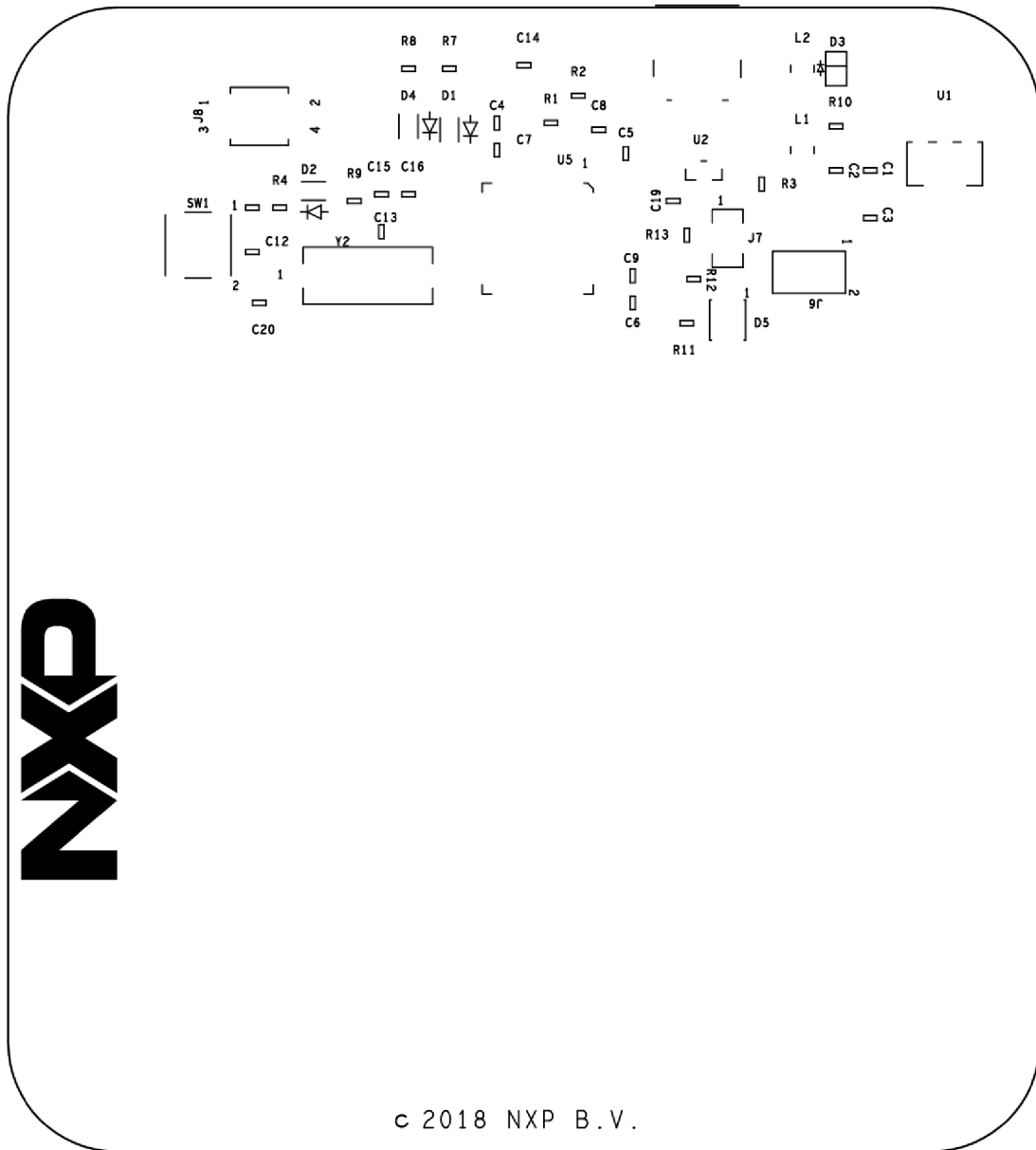
Výrobní data jsou k dispozici v elektronické příloze této práce ve formátu Gerber RS-274



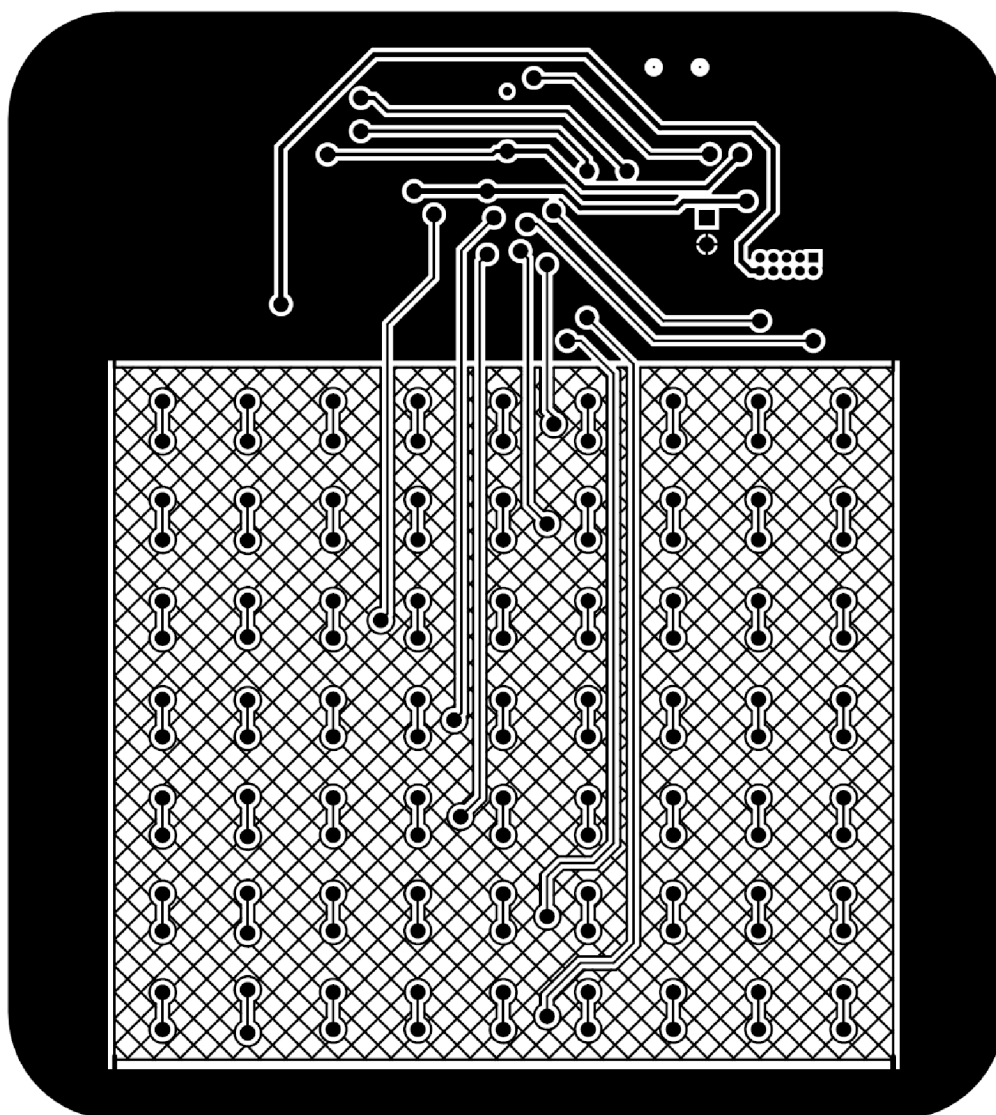
**Příloha 4 Motiv TOP plošného spoje touchpadu s KL25z**



## Příloha 5 Osazovací plán touchpadu s KL25z



## Příloha 6 Strana BOTTOM motivu plošného spoje KL25z



Výrobní data jsou k dispozici v elektronické příloze této práce ve formátu Gerber RS-274

## Příloha 7 Záložka Modules ve FreeMASTERu

NXPTouch.pmp - FreeMASTER

File Edit View Explorer Project Tools Help

Tahoma

### NXP Touch GUI

Modules Controls Ele ctrode Debug

**tsi detected . Module name: module\_tsi\_0.**

Electrode name	Keydetector	Raw Cnt	Baseline Cnt	signal	Touch	flags
<a href="#">electrode_safa_0_scope</a>	safa	8973	8971	8969		16
<a href="#">electrode_safa_1_scope</a>	safa	8753	8751	8750		16
<a href="#">electrode_safa_2_scope</a>	safa	8475	8477	8475		16
<a href="#">electrode_safa_3_scope</a>	safa	8345	8345	8343		16
<a href="#">electrode_safa_4_scope</a>	safa	8252	8250	8249		16
<a href="#">electrode_safa_5_scope</a>	safa	8270	8268	8267		16
<a href="#">electrode_safa_6_scope</a>	safa	8446	8440	8438		16
<a href="#">electrode_safa_7_scope</a>	safa	9439	9437	9436		16
<a href="#">electrode_safa_8_scope</a>	safa	9516	9513	9512		16
<a href="#">electrode_safa_9_scope</a>	safa	11236	11237	11236		16
<a href="#">electrode_safa_10_scope</a>	safa	10197	10196	10195		16
<a href="#">electrode_safa_11_scope</a>	safa	9324	9324	9324		16
<a href="#">electrode_safa_12_scope</a>	safa	9580	9579	9579		16
<a href="#">electrode_safa_13_scope</a>	safa	9625	9624	9623		16
<a href="#">electrode_safa_14_scope</a>	safa	9948	9948	9944		16
<a href="#">electrode_safa_15_scope</a>	safa	10377	10376	10376		16

control page

Ready RS232 UART Communication; COM7; speed=19200

## Příloha 8 Záložka Controls ve FreeMASTERu


NXPTouch.pmp - FreeMASTER

File Edit View Explorer Project Tools Help

Tahoma

### NXP Touch GUI

Modules Controls Electrode Debug



control\_aslider\_0

name	electrodes	type	position	range	touch	movement	direction	invalid touch
control_aslider_0	<a href="#">electrode_safa_0.scope</a>	aslider	139	255	✔	↔	↻	⚠
	<a href="#">electrode_safa_1.scope</a>							
	<a href="#">electrode_safa_2.scope</a>							
	<a href="#">electrode_safa_3.scope</a>							
	<a href="#">electrode_safa_4.scope</a>							
	<a href="#">electrode_safa_5.scope</a>							
	<a href="#">electrode_safa_6.scope</a>							
	<a href="#">electrode_safa_7.scope</a>							
	<a href="#">electrode_safa_8.scope</a>							

control\_aslider\_1

name	electrodes	type	position	range	touch	movement	direction	invalid touch
control_aslider_1	<a href="#">electrode_safa_9.scope</a>	aslider	33	255	✔	↔	↻	⚠
	<a href="#">electrode_safa_10.scope</a>							
	<a href="#">electrode_safa_11.scope</a>							
	<a href="#">electrode_safa_12.scope</a>							
	<a href="#">electrode_safa_13.scope</a>							
	<a href="#">electrode_safa_14.scope</a>							
	<a href="#">electrode_safa_15.scope</a>							

control page

RS232 UART Communication; COM7; speed=19200