České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra teorie obvodů

Bakalářská práce

Technické řešení standardizace síly vyvíjené vyšetřujícím na sonografickou sondu

A technical solution for standardizing the force exerted by the examiner on the sonographic probe

Autor: Margarita Tkachenko

Vedoucí práce: doc. Ing. Vratislav Fabián, Ph.D.

Studijní program: Lékařská elektronika a bioinformatika

Praha 2024

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Tkachenko** Jméno: **Margarita** Osobní číslo: **498985**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra teorie obvodů**

Studijní program: **Lékařská elektronika a bioinformatika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Technické řešení standardizace síly vyvíjené vyšetřujícím na sonografickou sondu

Název bakalářské práce anglicky:

A technical solution for standardizing the force exerted by the examiner on the sonographic probe

Pokyny pro vypracování:

Při běžném sonografickém vyšetření v klinické praxi není třeba sílu vyvíjenou v různých směrem na sonografickou sondu striktně standardizovat. Jiná situace může nastat v biomedicínském výzkumu. Cílem bakalářské práce je navržení konkrétního technického řešení, které zajistí standardizovanou sílu vyšetřujícího směrem kolmo k povrchu těla pacienta a také podélně ve směru povrchu těla pacienta. Účelem takového řešení je využití ve výzkumu ve fyzioterapii. Konkrétně využití při dynamickém hodnocení hluboké fasciální vrstvy, kdy je na základě standardizovaného sonografického záznamu měřen posun dvou vrstev hluboké fascie vůči sobě a reliabilita takového vyšetření je velmi závislá na konkrétním přítlaku/síle vyvinuté na sonografickou sondu ve dvou na sobě kolmých směrech.
1. Proveďte rešerši v oblasti technologií a metodik sonografických vyšetření svalových fascií.
2. Navrhněte technické řešení, které zajistí měření aplikace síly na sonografickou sondu. Do návrhu zahrňte přehled potřebných technologií a komponent pro realizaci řešení.
3. Proveďte realizace hardware (HW) prototypu, včetně dokumentace HW prototypu.
4. Otestujte navržené řešení a proveďte vyhodnocení výsledků a zhodnocení, jak prototyp přispívá k zlepšení standardizace přítlaku při sonografickém vyšetření svalových fascií a jeho potenciálního využití ve fyzioterapii.

Seznam doporučené literatury:

[1] DE SOUSA SOARES, Hélio Rafael. Ultrasound Assessment of Deep Fascia Sliding Mobility in Vivo. A Scoping Review. 2019. PhD Thesis. Instituto Politecnico do Porto (Portugal).
[2] STECCO, Carla. Functional atlas of the human fascial system. Elsevier Health Sciences, 2014.
[3] PAVAN, Piero G., et al. Painful connections: densification versus fibrosis of fascia. Current pain and headache reports, 2014, 18: 1-8.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

doc. Ing. Vratislav Fabián, Ph.D. katedra fyziky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **06.02.2024** Termín odevzdání bakalářské práce: **24.05.2024**

Platnost zadání bakalářské práce: **21.09.2025**

doc. Ing. Vratislav Fabián, Ph.D. podpis vedoucí(ho) práce	doc. Ing. Radoslav Bortel, Ph.D. podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Studentka bere na vědomí, že je povinna vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

. _____
Datum převzetí zadání

_____
Podpis studentky

Prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 24.05.2024 .....

Podpis autora práce

Poděkování

Ráda bych v této části vyjádřila své hluboké poděkování všem, kteří mi poskytli podporu a pomoc při tvorbě této bakalářské práce.

Nejprve bych ráda poděkovala doc. Ing. Vratislavu Fabiánovi, Ph.D. za jeho cenné rady, vedení a trpělivost během celého procesu vytváření této práce. Jeho odborné znalosti a vstřícnost mi byly neocenitelnou oporou.

Dále bych ráda poděkovala Mgr. Stanislavu Machačovi, Ph.D. a Mgr. Tereze Štěpánkové za jejich odborné rady a konzultace v oblasti fyzioterapie, které mi pomohly hlouběji porozumět tématu fascií.

Nemohu zapomenout ani na Antona Korbana, Ph.D., Ing. Kateřinu Indrovou a Mgr. Kiryla Vasiutoviche, jejichž pomoc a podpora byla klíčová při získávání potřebných informací a při praktickém řešení některých problémů.

Abstrakt

Práce je zaměřěna na realizaci technického řešení pro standardizaci síly aplikované na sonografickou sondu ve výzkumném kontextu fyzioterapie. Konkrétně se jedná o vyhodnocení pohybu dvou vrstev hluboké fascie. Práce poskytuje základní informace o fasciích, zkoumá diagnostické metody a identifikuje nedostatek standardizace parametrů při vyšetřování.

Pro dosažení možnosti porovnávání výsledků ultrazvukových vyšetření fascie je nezbytné zajistit standardizovaný tlak aplikovaný sonografickou sondou kolmo i podél povrchu těla pacienta. V této práci je navrženo technické řešení, které využívá senzory síly typu HONEYWELL FSG020WNPB a signál je vyhodnocován pomocí Arduino Nano. Implementace zahrnuje výběr komponent, zapojení, kalibraci a programovací část.

V závěru práce jsou diskutovány další možnosti zlepšení projektu. V současné době bylo zařízení otestováno částečně při vyšetření fascie, protože projekt je ve fázi vytváření adaptéru ke konkrétním ultrazvukovým sondám, což přesahuje rámec této bakalářské práce.

Klíčová slova: Fascie, měření síly, tlakový senzor

Abstract

The thesis is dedicated to implementation of technical solution for standardizing the force applied to a sonographic probe in the research context of physiotherapy, specifically for assessing the movement of two layers of deep fascia. The thesis provides basic information about fascia, explores diagnostic methods and identifies a lack of standardization of parameters in examinations.

To enable the comparison of results from ultrasound examinations of fascia, it is necessary to ensure standardized pressure applied perpendicularly and along the surface of the patient's body through the sonographic probe. This thesis proposes a technical solution utilizing HONEYWELL FSG020WNPB force sensors and evaluation by Arduino Nano. The implementation involves component selection, circuitry, calibration, and programming.

The conclusion of the thesis discusses further project improvement possibilities. Currently, the device has been partially tested in fascia examination because it is in the process of creating an adapter for specific ultrasound probes, which goes beyond the scope of this bachelor's thesis.

Keywords: Fascia, force measurement, pressure sensor

Seznam použitých zkratek

EEPROM - Electrically Erasable Programmable Read-Only Memory

SPI - Serial Peripheral Interface

FSR - Force-Sensing Resistor

Seznam tabulek

Tabulka 1: Současné diagnostické zobrazovací metody používané ke studiu struktury a funkce fasciálních tkání 15
Tabulka 2: Porovnaní senzorů síly. 20
Tabulka 3: Porovnaní desek Arduino 22
Tabulka 4: Ověření přestnosti kalibrace 33
Tabulka 5: Intervaly spolehlivosti 33

Seznam obrázků

Obrázek 1: Fascie v těle člověka. 12
Obrázek 2: Ultrazvuk stehna. 13
Obrázek 3: Návrh rozmístění senzorů na sondu 17
Obrázek 4: Návrh instalace senzorů. 18
Obrázek 5: Fotografie použitého displeje 23
Obrázek 6: Membránová klávesnice 4x4. 23
Obrázek 7: Miniaturní bzučák. 24
Obrázek 8: Kolíkové a dutinkové lišty 24
Obrázek 9: Zapojení senzoru FSR. 25
Obrázek 10: Proces kalibrace FSR senzorů 26
Obrázek 11: FSR senzor, převodní charakteristika napětí na sílu 27
Obrázek 12: FSR senzor, převodní charakteristika odporu na sílu 27
Obrázek 13: Wheatstoneův můstek. 29
Obrázek 14: Výsledné zapojení na montážní desku 29
Obrázek 15: Schéma výsledného zapojení. 30
Obrázek 16: Realizace výsledného zapojení 30
Obrázek 17: Příslušenství k realizovanému projektu. 31
Obrázek 18: Připojení senzoru 31
Obrázek 19: Kalibrace senzorů HONEYWELL FSG020WNPB 32
Obrázek 20: Kalibrační křivky senzorů HONEYWELL FSG020WNPB 32
Obrázek 21: Programování - použité knihovny 34
Obrázek 22: Programování - definované hodnoty 35
Obrázek 23: Programování - definované hodnoty 2 35
Obrázek 24: Programování - funkce “print” 36
Obrázek 25: Programování - funkce “print”, výsledek volání funkce 36
Obrázek 26: Programování - funkce “print”, volání funkce 36
Obrázek 27: Programování – funkce “error” a “print_set_the_range” 37
Obrázek 28: Programování - funkce “error” a “print_set_the_range”, výsledek volání funkce 37
Obrázek 29: Programování - funkce “set_the_range”, výsledek volaní funkce 38
Obrázek 30: Programování - funkce “set_the_range” 38
Obrázek 31: Programování - funkce “EepromIntWrite“ a “EepromIntRead” 39
Obrázek 32: Programování - funkce “EepromIntRead”, volání funkce 39
Obrázek 33: Programování - funkce “EepromIntWrite”, volání funkce 39
Obrázek 34: Programování - funkce “PlayTone” 40
Obrázek 35: Programování - funkce “PlayTone”, volání funkce 40

Obrázek 36: Programování - funkce “Setup” 40
Obrázek 37: Programování - část funkce “Loop” 41
Obrázek 38: Programování - část funkce “Loop” 2 42

Obsah

1. Úvod 11
2. Fascie 12
   2.1. Definice a základní funkce 12
   2.2. Typy fascií 13
   2.3. Poruchy a diagnostika 14
3. Návrh řešení 17
4. Realizace 19
   4.1. Použité součástky 19
      4.1.1. Senzory síly 19
      4.1.2. Arduino 21
      4.1.3. Displej 22
      4.1.4. Klávesnice 23
      4.1.5. Miniaturní bzučák 23
      4.1.6. Souhrn 24
   4.2. Zapojení 25
      4.2.1. Zapojení – senzory FSR 25
      4.2.2. Zapojení – senzory HONEYWELL FSG020WNPB 29
   4.3. Programovaní 34
      4.3.1. Knihovny 34
      4.3.2. Předem definované hodnoty 34
      4.3.3. Funkce 36
5. Závěr 43
6. Seznam literatury 44
7. Příloha č. 1: Celý kód programu. 45

1. Úvod

V oboru fyzioterapie se v posledních letech zvýšil zájem o fascie a jejich roli v pohybové funkci člověka. Fascie je jednou z nejdůležitějších součástí pohybového aparátu člověka. Fascie si můžeme představit jako trojrozměrnou síť, která obklopuje naše tělo, pokrývá povrchy orgánů a propojuje různé části těla. Jejich funkce je zásadní pro udržení stability, přenos síly a podporu pohybu.

Poruchy fascií mohou vést k různým problémům včetně bolesti a omezení pohybu, což ovlivňuje celkové zdraví člověka. Důvody poruch mohou být různé, jako například přirozené stárnutí, užívání léků, zranění nebo trauma. Zmenšení bolesti nebo zlepšení stavu fascie pomáhají různé techniky a metody, kterými se zabývají fyzioterapeuti. Ale před tím, než problém nějakým způsobem řešíme, je nutné ho nejdříve určit.

V případě vyšetření fascie se jedná o dva typy vyšetření: nezobrazovací a zobrazovací. První typ zahrnuje například manuální palpaci a biopsii. Druhý typ vyšetření používá ve většině případů ultrazvuk a různá rozšíření pro lepší zobrazení fascie. Problém nastává, pokud chceme výsledky vyšetření porovnávat mezi sebou. Je jasné, že porovnávat můžeme jen data, která byla nasbírána za stejných podmínek měření.

V sonografickém vyšetření fascií se často setkáváme s nedostatkem standardizace síly, kterou aplikuje vyšetřující na sonografickou sondu. V závislosti na této síle se může měnit výsledný tvar fascie na ultrazvukovém obraze. Zatím neexistuje komerční řešení, jak tuto sílu standardizovat, a proto je tento cíl hlavním zaměřením mé bakalářské práce.

2. Fascie

2.1. Definice a základní funkce

Termíny „fascie“ a „fasciální systém“ byly přesněji definovány na 4. mezinárodním kongresu Společnosti pro výzkum fascií v roce 2015. Nicméně různé oblasti medicíny mohou tyto pojmy interpretovat odlišně, což vyvolává diskuse 6. Zároveň je tato nejednotná terminologie přítomna i v odborné literatuře, což ztěžuje porozumění a porovnání výsledků jednotlivých výzkumů.

Obrázek 1: Fascie v těle člověka. Obrázek byl vygeneroval v aplikace Complete Anatomy

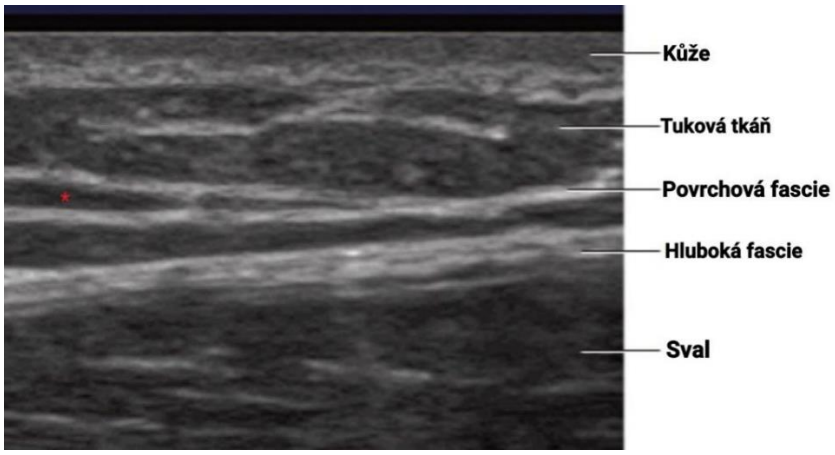
Každopádně fascie lze popsat pomocí jejich vlastností a hlavních funkcí. Národní zdravotnický informační portál České republiky definuje: fascii neboli povázku (*lat. fascia*) jako vazivovou membránu, která obaluje orgány, svaly nebo dokonce skupiny svalů a mnoho dalších tělních struktur.

Mezi hlavní funkce fascií patří přenos silového působení, podíl na motorické koordinaci, senzorické funkce (propiocepce a nocicepce), ochrana a fixace tělních struktur. Fascie jsou pasivní strukturou ve srovnání se svalovou tkání, mají klíčový vliv na tvar a pružnost svalů a orgánů. Umožňují také plynulý pohyb mezi svaly a dalšími tkáněmi, díky své kluzké povrchové vrstvě. Tím usnadňují pohyb, snižují tření a odolávají mechanickému tlaku. Fascie se skládají z kolagenních, elastických a retikulárních vláken. Uvnitř fascie se nacházejí buňky pojivové tkáně [2]. Na obrázku 1 je ilustrováno, jak fascie se šíří po celém těle ve formě trojrozměrné sítě.

2.2. Typy fascií

Hustota tkáně a směr uspořádání vláken má přímý vliv na vlastnosti fascie, jako jsou pevnost, pružnost a pohyblivost. Na základě umístění, funkce a vlastností lze rozlišit 4 typy fascií [2]:

1. **Povrchová fascie** (lat. fascia superficialis) - je nejsvrchnější vrstva pod kůží, součást podkoží. Kvůli nízkému obsahu kolagenních vláken, je velice pohyblivá, slouží k upevnění kůže [2]. Na Obrázku 2 je vyznačena povrchová fascie, která se rozděluje na dvě podvrstvy.
2. **Hluboká fascie** (lat. fascia profunda) – vrstva, která obklopuje a chrání jednotlivé svaly a celé svalové skupiny [2]. Vrstva je hustá a dobře organizovaná. Jak vyplývá z názvu, hluboká fascie je uložena hlouběji v těle než povrchová fascie a obsahuje více kolagenních vláken. Přenos silového působení je hlavní funkcí hluboké fascie [3]. Obrázek 2 zobrazuje hlubokou fascii.



Obrázek 2: Ultrazvuk stehna. Povrchové i hluboké fascie jsou dobře rozlišitelné. Povrchová fascie se rozděluje na dvě podvrstvy, které obalují tukovou tkáň (označeno na obrázku hvězdičkou). Obrázek byl převzat z [4]

3. **Viscerální fascie** (lat. fascia visceralis) - tenká vrstva, která vystýlá tělní dutiny, jako jsou hrudní, břišní a pánevní dutiny; obklopuje orgány, které se v nich nacházejí. Další

podstatná funkce této vrstvy kromě ochrany je fixace orgánů s možností fyziologického pohybu [3].

4. **Meningeální fascie** – vrstva, která poskytuje obal, podporu a ochranu pro nervová vlákna (axony) a jejich svazky [2].

Fascie jsou součástí většího funkčního celku označovaného jako “fasciální systém”. Tento systém zahrnuje i tukovou tkáň, neurovaskulární membrány, šlachy a další tělesné struktury 6. Hraje důležitou roli v rámci pohybového aparátu člověka, společně se svaly, vazy a dalšími strukturami se významně podílí na zajištění motorické koordinace a přenosu silového působení [3]. Například během svalové kontrakce hraje fascie roli pružné struktury, která pomáhá svalu vrátit se do své uvolněné (relaxované) polohy. Další podstatnou vlastností je, že se ve vrstvách fascie nacházejí receptory (proprioceptory), pomocí kterých vnímáme podněty a přenášíme informaci týkající se současné polohy těla a jeho pohybu [2].

2.3. Poruchy a diagnostika

Vlivem různých faktorů může docházet na úrovni fasciálních vrstev k určitým změnám, které mohou být fyziologické i patologické. Mezi faktory, které mohou vést k změnám ve fasciální tkáni, patří například opakované zatěžování svalů a kloubů, různé druhy fyzické aktivity nebo namáhavé pohyby, přirozený proces stárnutí, užívání určitých léků, zranění nebo trauma [2]. V důsledku patologických alterací fasciální tkáně může být přítomna bolest, snížení výkonnosti či omezení pohyblivosti. K ovlivnění patologických změn existuje řada terapeutických přístupů, jako jsou různé druhy fasciálních terapií, mezi které se řadí například koncept Fasciální manipulace dle Stecca[2].

Stav fascie ovlivňuje celkové zdraví člověka, a proto je důležité včas rozpoznat a léčit případné poruchy. Z toho vyplývá potřeba využití diagnostických nástrojů pro důkladné zkoumání fungování fasciálních tkání. Existují dvě skupiny metod vyšetření fascie: nezobrazovací, ke které patří biopsie, bioimpedance, manuální palpace, indentometrie; a zobrazovací, na které se soustředíme v rámci této práce [6]. Porovnání zobrazovacích metod je znázorněno v Tabulce 1.

Tabulka 1: Současné diagnostické zobrazovací metody používané ke studiu struktury a funkce fasciálních tkání. Tabulka byla převzata z [6] a doplněna údaji z [7]

Metody	**Cíl vyšetření**	**Výhody**	**Nevýhody**
Ultrazvukové zobrazování	Tloušťka vrstev, prodloužení šlach.	Umožňuje diagnostiku fibrotického ztluštění nebo odezvy na natažení šlachy/fascie během zatížení.	Obtížnost standardizace přesného pozorovacího úhlu.
Ultrazvukové zobrazování s korelačním softwarem	Relativní smykový pohyb sousedních vrstev.	Umožňuje diagnostiku adhezivních tkáňových spojení [1]	Chybějící standardy pro výběr oblastí vyšetření.
Statická ultrazvuková elastografie [2]	Ztuhlost.	Měření ve větší hloubce. Jednoduchost, široká dostupnost a nízké náklady.	Nedostatek standardizace. Neznalost velikosti síly deformace, což brání kvantitativnímu stanovení elastických vlastností tkáně. Častý výskyt artefaktů.
Dynamická ultrazvuková elastografie [3]	Ztuhlost.	Vysoké prostorové rozlišovací schopnosti. Přesný kvantitativní popis elastických vlastností tkáně.	Nedostatek standardizace. Větší technologická náročnost a s tím spojená vyšší cena.
Ultrasonografie v B-módu [4]	Struktura a mechanické/materiálové vlastnosti tkáně.	Aplikace v perspektivních studiích. Relativně levné.	Přesnost závisí na uživateli. Použitelnost je omezena především na povrchové struktury. Frekvence snímání je momentálně omezena.

[1] Těsné spojení na povrchu epitelu, které vytváří pruh pod jejich úrovní. Slouží jako stabilní mechanické spojení, které pevně spojuje buňky dohromady.
[2] Ultrazvuková zobrazovací elastografie založená na kompresi – metoda, která porovnává signály UZ před a po stlačení tkáně.
[3] Ultrazvuková zobrazovací elastografie střižných vln – hodnocení rychlosti šíření střižných vln (shear waves), které se vytvářejí jako odezva elastického odporu tkáně vůči tlakovým pulzům nebo mechanickým vibracím s nízkou frekvencí.
[4] Jas každého bodu odpovídá intenzitě elektrických signálů generovaných přijímacím měničem, což závisí na odrazu ultrazvukových vln. Čím větší odraz, tím světlejší je příslušný pixel obrazu.

Ve vědeckém výzkumu fascií je často využíváno několik metod, aby bylo možné detailněji prozkoumat jejich strukturu. Nicméně většina těchto metod trpí nedostatkem standardizace, což komplikuje srovnání výsledků a brání pokroku v této oblasti. Standardizace parametrů, jako je například síla, s jakou senzor tlačí na tělo pacienta při měření, by mohla zlepšit tuto situaci. Právě tento cíl si klade za úkol má bakalářská práce.

3. Návrh řešení

Jedním z možných řešení standardizace síly vyvíjené vyšetřujícím na sonografickou sondu je průběžné měření aplikované síly na sondu během vyšetření. Tento způsob umožňuje vyšetřujícímu reagovat na aktuální hodnoty a jednodušeji se přizpůsobit požadovanému přítlaku. Navržené řešení by mělo být snadno použitelné a dostatečně flexibilní pro přizpůsobení různým typům vyšetření a potřebám výzkumu.

Na základě rešerše bylo rozhodnuto využít senzory, které jsou schopné měřit sílu aplikovanou na jejich aktivní plochu. Pro zajištění přesnějšího měření a z důvodu, že sondy jsou standardně drženy určitým způsobem (třemi prsty), bylo rozhodnuto použít tři senzory s umístěním podle schématu zobrazeného na Obrázku 3. Toto uspořádání zajistí měření síly podél a kolmo k povrchu těla pacienta.

Obrázek 3: Návrh rozmístění senzorů na sondu

Vytváření držadla, viz Obrázek 4, na senzory přesahuje rámec této bakalářské práce. Návrh uvedený na Obrázku 4 byl vytvořen a pak realizován Filipem Křivohlavým (Ústav pro životní prostředí, Přírodovědecká fakulta, Univerzita Karlova).

Obrázek 4: Návrh instalace senzorů. Autor: Filip Křivohlavý

Pro provedení uvedeného řešení je nezbytné mít nejen senzory síly, ale také řídící a vyhodnocovací komponent – mikrokontrolér. Softwarové zpracování a grafické zobrazení nejsou příliš náročné, takže výběr výpočetního modulu není výrazně ovlivněn technickými požadavky.

Pro zobrazování naměřených hodnot síly je dobrou volbou použití dostatečně velkého displeje, který by zobrazoval aktuální hodnoty sil na jednotlivých senzorech a také předem určený povolený rozsah sil. Tento rozsah by měl být stanoven během prvních několika pokusů a poté by sloužil jako standardní rozsah sil. Předpokládáme, že standardní rozsah bude odlišný pro různé oblasti vyšetření, například pro záda a dolní končetiny. Do sestavy tak byla přidána klávesnice, pomocí které lze upravovat nastavené hranice pro různá vyšetření. Pro zajištění pohodlí vyšetřovaného bylo do designu přidáno zařízení, které bude generovat zvukový signál při překročení nastavené hranice. Tímto způsobem bude vyšetřující okamžitě upozorněn na změnu síly a bude moci včas zareagovat.

4. Realizace

4.1. Použité součástky

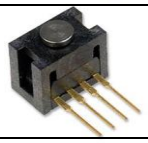
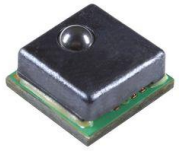
4.1.1. Senzory síly

Podstatná kritéria, která jsme zohlednili při výběru senzoru, jsou:

- Rozsah měřené síly
- Velikost senzoru a průměr aktivní plochy
- Budící napětí senzoru
- Cena
- Dostupnost u dodavatele

Na začátku práce jsme předpokládali maximální hodnotu síly převedenou na hmotnost - 2 kg, což bylo ověřeno pomocí senzoru při prvním pokusu, viz kapitola 4.2.1. Velikost senzoru je klíčový parametr, zejména s ohledem na instalaci na sonografickou sondu, která má omezený prostor pro umístění senzoru. Přesto je důležité, aby aktivní plocha senzoru byla dostatečně velká k přesnému měření síly, která na něj působí. Budící napětí musí být v rozmezí od 3,3 V do 5 V, což odpovídá výstupnímu napětí, které může poskytnout mikropočítač Arduino. Dostupnost u dodavatele byla ověřena v době zpracování práce, konkrétně od října 2023 do listopadu 2023. Na základě těchto kritérií byla vytvořena Tabulka 2.

Tabulka 2: Porovnaní senzorů síly. Byly využity informace včetně fotografií a cen získané z webové stránky cz.farnell.com. Technické specifikace pocházejí z příslušné dokumentace každého senzoru, která je k dispozici na uvedené stránce

Č.	**Název**	**Obrázek**	**Cena [Kč]**	**Rozsah měření [g]**	**Rozměry / průměr aktivní plochy [mm x mm x mm] / [mm]**	**Budící napětí [V]**
1.	HONEYWELL FSS020WNGB		3 115	0-2000	13,7 x 5,6 x 3,2/ 1,5	3,3-12,5
2.	HONEYWELL FSG020WNPB		4 334	0-2000	12,7 x 18 x 9/ 5,1	3,3-12,5
3.	HONEYWELL FSAGPDXX005LCSB5		5 179	30-2270	31,5 x 17,4 x 8,2/ 12,7	4,75-5,25
4.	HONEYWELL FMAMSDXX025WCSC3		838 [5]	0-2000	5 x 5 x 2,6/ 0,45	3-3,6
5.	OHMITE FSR05CE		208	30-5000	38,1 x 6,4 x 0,3/ 5,6	0-6

Za účelem ověření předpokládaného rozsahu měřené hodnoty síly a s ohledem na cenu byl vybrán senzor číslo 5 - OHMITE FSR05CE. Tento senzor nabízí největší rozsah měřených hodnot a zároveň je cenově nejvýhodnější. Po vyzkoušení a zjištění nevýhod daného senzoru, viz kapitola 4.2.1, byl vybrán jiný typ - HONEYWELL FSG020WNPB. Hlavními kritérii, na základě kterých bylo učiněno toto rozhodnutí, jsou průměr aktivní plochy a rozměry senzoru. U senzorů číslo 1 a 4 je aktivní plocha příliš malá a nevhodná k přímému tlaku prstů. Na druhou stranu senzor číslo 3 má příliš velké rozměry, což by při instalaci na sonografickou sondu způsobovalo potíže. Vybraný senzor HONEYWELL FSG020WNPB má své nevýhody jako je relativně vysoká cena ve srovnání s jinými senzory a typické budící napětí, které činí 10 V, což není kompatibilní s Arduinem. Nicméně hodnoty napětí v rozmezí 3,3-5 V, které jsou výstupem z Arduino, jsou v povoleném rozsahu budícího napětí senzoru.

[5] Cena za 1 kus, s minimálním objednáním 10 kusů.

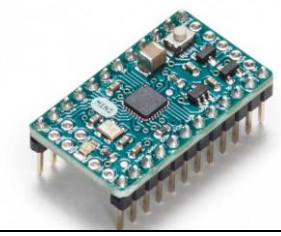
4.1.2. Arduino

Pro realizaci tohoto projektu byla zvolena populární vývojovou desku z rodiny Arduino. Při výběru hrála podstatnou roli nejen osobní zkušenost s touto deskou, ale také její nízká cena, vysoká dostupnost, příjemné vývojové prostředí a široká škála dostupných modelů.

V současné době existuje několik různých druhů Arduino desek, které se liší ve velikosti, počtu pinů a procesoru. Mezi nejznámější patří Arduino Mini, Arduino Nano, Arduino Micro, Arduino Uno a Arduino Mega 2560. Každá z těchto desek má své vlastní specifické vlastnosti a je vhodná pro různé typy projektů a aplikací, viz Tabulka 3.

Pro realizaci dané práce byla vybrána deska Arduino Nano. Desky Arduino Mini a Arduino UNO nemají dostatečný počet pinů pro realizaci projektu. Pro řešení absence USB konektoru na Desce Arduino Mini je nutné použít externí převodník, což vede ke zvětšení rozměrů modelu, pro tuto aplikaci je tedy nevhodné. Verze Arduino Micro a Arduino Mega 2560 nebyly zvoleny kvůli jejich větší velikosti a počtu pinů, které by zůstaly nevyužity ve finální verzi projektu.

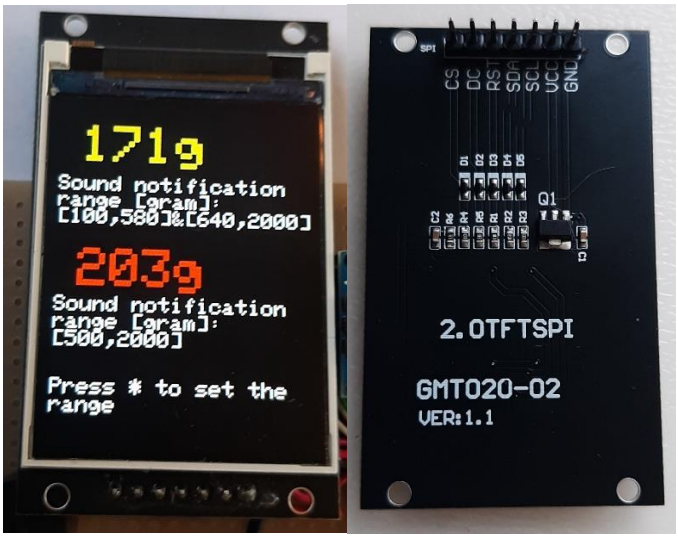
Tabulka 3: Porovnaní desek Arduino. Byly využity informace z oficiální webové stránky arduino.cc

Název	Obrázek	Vlastnosti
Arduino Mini		Mikrokontrolér ATmega328P Paměť 32 KB EEPROM 1 KB Digitální I/O piny: 14 Analogové vstupy: 6 Nejmenší deska, nemá USB převodník.
Arduino Nano		Mikrokontrolér ATmega328 Flash paměť 32KB EEPROM 1 KB Digitální I/O piny: 14 Analogové vstupy: 8 Je zmenšená deska Arduino UNO, které má více analogových pinů.
Arduino Micro		Mikrokontrolér ATmega32U4 Flash paměť 32KB EEPROM 1 KB Digitální I/O piny: 20 Analogové vstupy: 12 Může sloužit jako klávesnice nebo myš, posílat příkazy jako stisk kláves nebo pohyb myši.
Arduino UNO		Mikrokontrolér ATmega328P Flash paměť 32 KB EEPROM 1 KB Digitální I/O piny: 14 Analogové vstupy: 6 Je základní a nejpopulárnější verze celé série.
Arduino Mega 2560		Mikrokontrolér ATmega2560 Flash paměť 256 KB EEPROM 4 KB Digitální I/O piny: 54 Analogové vstupy: 16 Více pinů a větší výkon, vhodný pro rozsáhlejší projekty.

4.1.3. Displej

Pro zobrazení měřených hodnot byl vybrán 2,0‘‘ displej s rozlišením 320 x 240 barevných pixelů. Plný název displeje je GMT020-02 Ver1.1 2.0TFTSPI LCD. Displej využívá protokol rozhraní SPI a vyžaduje 7 pinů, včetně GND a Vcc. Maximální vstupní napětí nesmí překročit 3,3 V, což Arduino

Nano bez problémů umožňuje. Na Obrázku 5 jsou fotografie vybraného displeje zobrazené z přední a zadní strany.



Obrázek 5: Fotografie použitého displeje

4.1.4. Klávesnice

Pro nastavení povolených hranic síly byla nainstalována membránová klávesnice typu 4x4 s celkem 16 tlačítky, viz Obrázek 6. Klávesnice je vybavena 8 piny pro připojení. Výrobce uvádí, že klávesnice má odezvu 1 ms a zároveň nabízí životnost až 100 milionů stisků.

Obrázek 6: Membránová klávesnice 4x4. Převzato z webu dratek.cz, kde klávesnice byla koupena

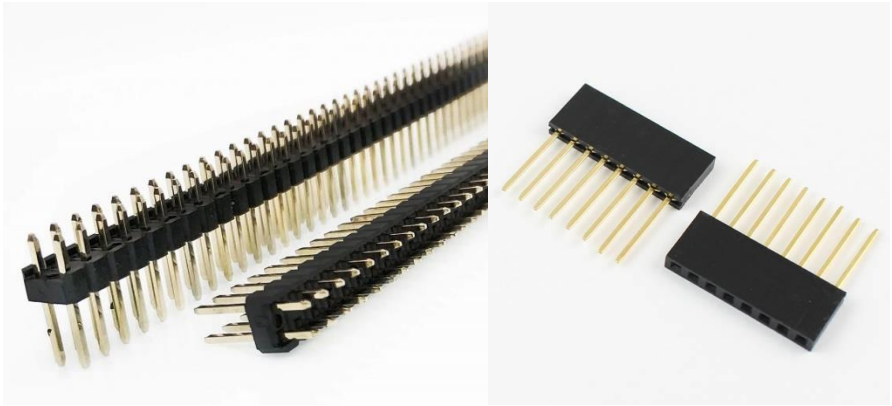
4.1.5. Miniaturní bzučák

Pro oznamování překročení nastavených hranic byl použit miniaturní bzučák ACS95. Bzučák má hlučnost 85 dB při napájení 5 V a generuje nepřerušovaný tón (Obrázek 7).

Obrázek 7: Miniaturní bzučák. Převzato z webu conrad.cz

4.1.6. Souhrn

Pro realizaci projektu byly použity 3 kusy senzorů síly FSG020WNPB, deska Arduino Nano, displej GMT020-02, membránová klávesnice 4x4 a miniaturní bzučák ACS95. Zapojení a programování uvedených součástek budou popsány v dalších kapitolách. Vzhledem k nízké ceně a možnému poškození součástek, jako jsou Arduino, displej a klávesnice, bylo rozhodnuto provést zapojení ne přímo na desce plošných spojů, ale použít dutinkovou a kolíkovou lištu. Díky tomuto zapojení je možné součástky snadno vyjmout a nahradit za stejný typ bez nutnosti pájení. Lišty jsou zobrazeny na Obrázku 8.

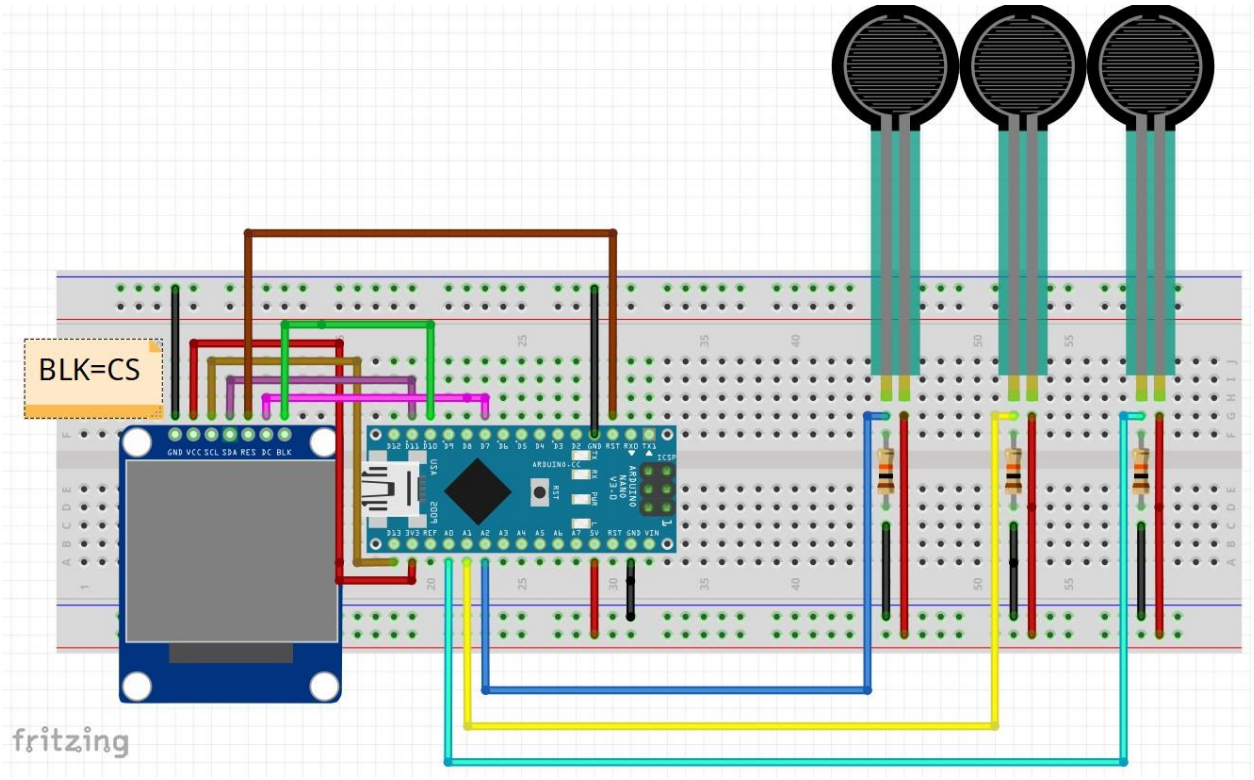


Obrázek 8: Kolíkové (vlevo) a dutinkové (vpravo) lišty. Převzato z webu dratek.cz

4.2. Zapojení

4.2.1. Zapojení – senzory FSR

Při prvním zapojení jsme využili senzory typu FSR, jak jsme již zmínili dříve. Důvody použití tohoto typu jsou jednoduchý základní princip fungování, přijatelná cena a dostatečně široký rozsah měření (30-5000 g). Avšak vzhledem k omezené přesnosti detekce tlaku se nedoporučuje tento senzor používat v případech, kde je vyžadováno precizní měření tlakových hodnot. Nicméně toto zapojení je nezbytné pro určení přibližného rozsahu hodnot, který budeme měřit při vyšetření fascií. Navržené a realizované zapojení je znázorněno na Obrázku 9.

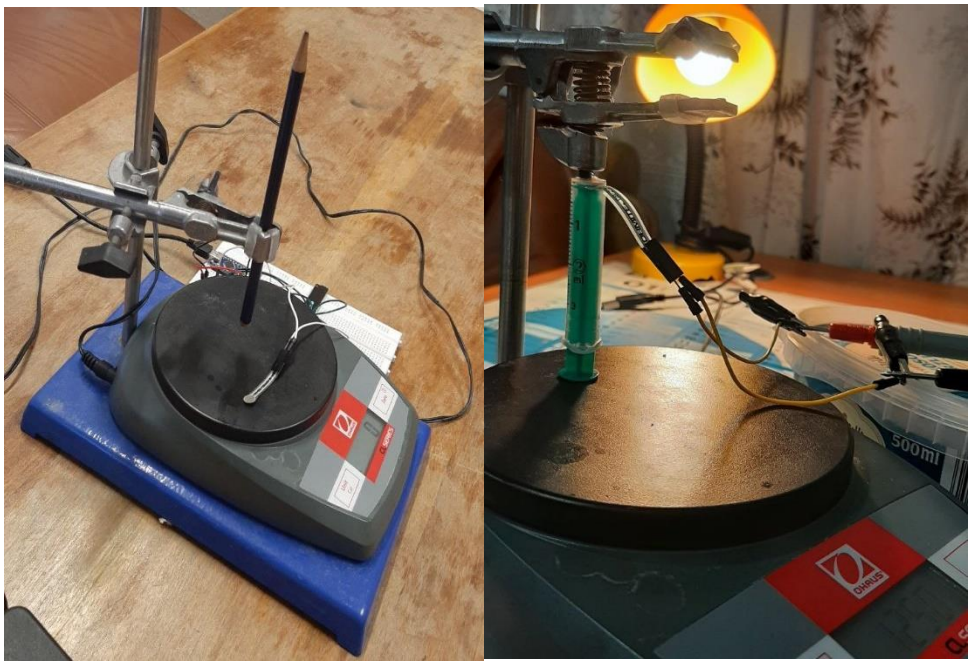


Obrázek 9: Zapojení senzoru FSR. Zapojení bylo vytvařeno v aplikace fritzing

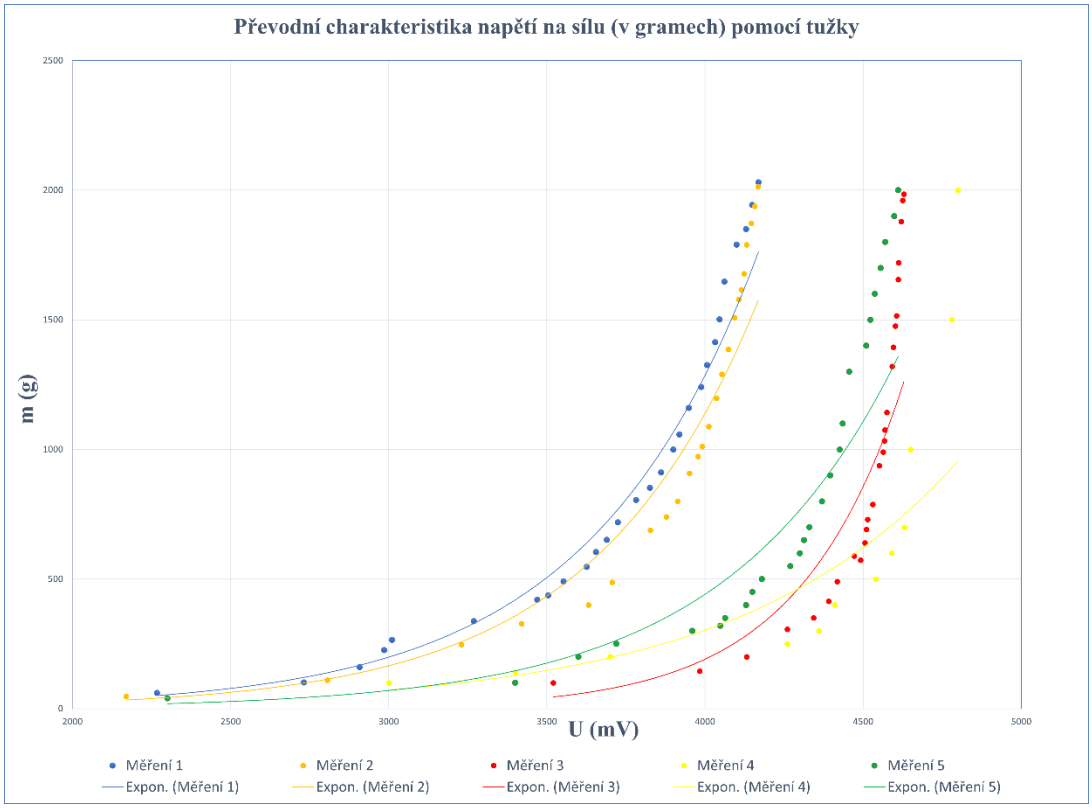
Flexibilní tlakový senzor (FSR) je ultra-tenký rezistivní senzor, který reaguje na tlak v citlivé oblasti. Tlak snižuje odpor senzoru. Tato změna odporu vede ke změně výstupního napětí: čím větší tlak, tím vyšší napětí. Závislost výstupního napětí na odporu FSR senzoru není lineární, což je vidět ze Vzorce 1, kde V_0 je výstupní napětíí, V_{cc} – vstupní napětí, R se rovná 10 kΩ, R_{FSR} – odpor FSR senzoru. Tento vzorec platí pro každý zapojený senzor.

$$V_0 = V_{cc} \frac{R}{R+R_{FSR}} \quad (1)$$

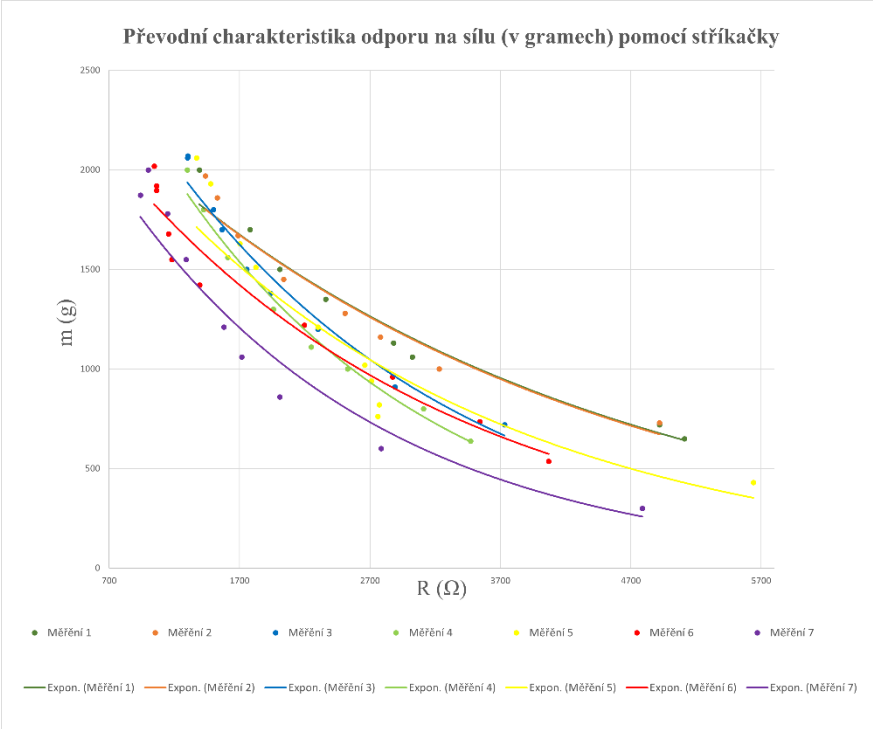
Kvůli tomu převodní charakteristika napětí/odpor R_{FSR} na sílu (v gramech) nebude lineární, což bylo ověřeno během kalibrace. Proces kalibrace byl proveden dvěma způsoby: pomocí tužky s plochým koncem a prostřednictvím stříkačky s dírou pro senzor. Zvolili jsme tyto netradiční metody, protože standardní kalibrace pomocí závaží není možná, senzor FSR nereaguje, když na něj položíme předmět s určitou hmotností. Stříkačka byla použita z důvodu potřeby rozložení tlaku na celou plochu senzoru. V prvním případě jsme přímo měřili napětí pomocí Arduino. Ve druhém typu kalibrace jsme sledovali odpor senzoru pomocí multimetru a následně ho přepočítali na napětí pomocí Vzorce 1. Oba typy kalibrace byly provedeny za použití laboratorního stojanu a váhy s kapacitou do 2 kg. Proces kalibrace a získané převodní křivky jsou zobrazeny na obrázcích 10, 11 a 12.



Obrázek 10: Proces kalibrace FSR senzorů



Obrázek 11: FSR senzor, převodní charakteristika napětí na sílu (v gramech) pomocí tužky. Body jsou proloženy pomocí exponenciálního polynomu.



Obrázek 12: FSR senzor, převodní charakteristika odporu na sílu (v gramech) pomocí stříkačky. Body jsou proloženy pomocí exponenciálního polynomu.

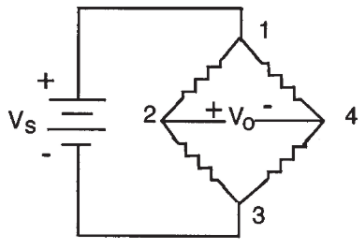
Z převodních křivek obou typů kalibrace je patrné, že senzor typu FSR není přesný, a že hodnoty měření se od sebe hodně liší. Na Obrázku 11 jsme pozorovali výrazný posun křivek doprava, který byl způsoben změnou tužky. Byly použity dvě tužky různého průměru, které měly menší průměr než aktivní plocha senzoru. Tato vlastnost senzoru měnit svůj odpor v závislosti na místě nebo ploše aplikované síly, není vhodná, zejména pokud má senzor sloužit pro měření tlaku prsty.

Druhý typ kalibrace byl přesnější, avšak příliš idealizovaný, neboť tlak pístu stříkačky nepůsobí stejně jako tlak aplikovaný lidským prstem. Nicméně, tento senzor byl vybrán jako nástroj pro přibližné měření, a proto jsme využili průměr získaný z křivek z druhé kalibrace.

Díky tomuto senzoru jsme byli schopni provést testovací měření, během kterého jsme zjistili, že maximální hodnota síly při měření tlaku na fascie dolních končetin je přibližně 1400 g. Na základě těchto zjištění jsme zvolili senzor typu HONEYWELL FSG020WNPB, který měří v rozsahu od 0 do 2000 g, abychom měli určitou rezervu v případě potřeby.

4.2.2. Zapojení – senzory HONEYWELL FSG020WNPB

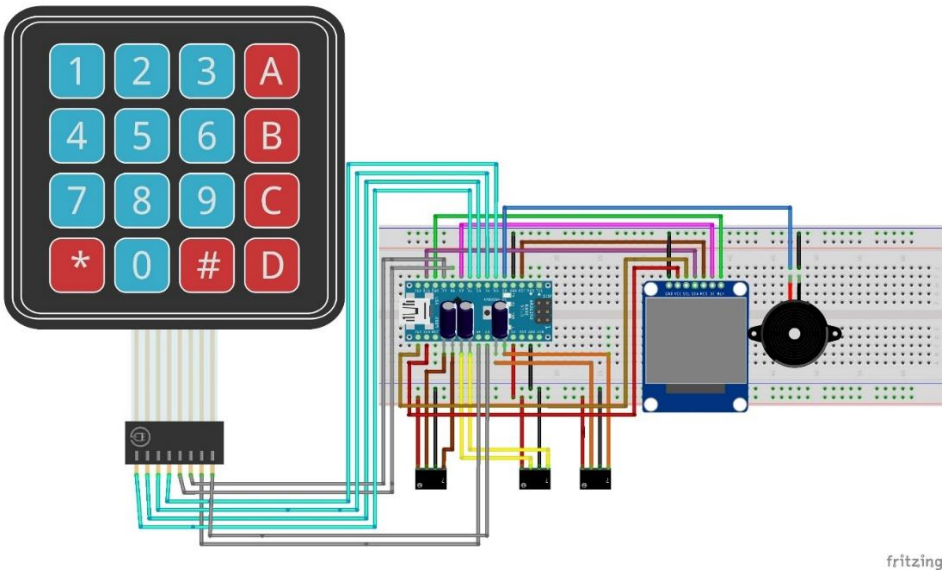
Druhé zapojení bylo realizováno se senzory typu HONEYWELL FSG020WNPB, které mají strukturu Wheatstoneova můstku, jak je zobrazeno na Obrázku 13.



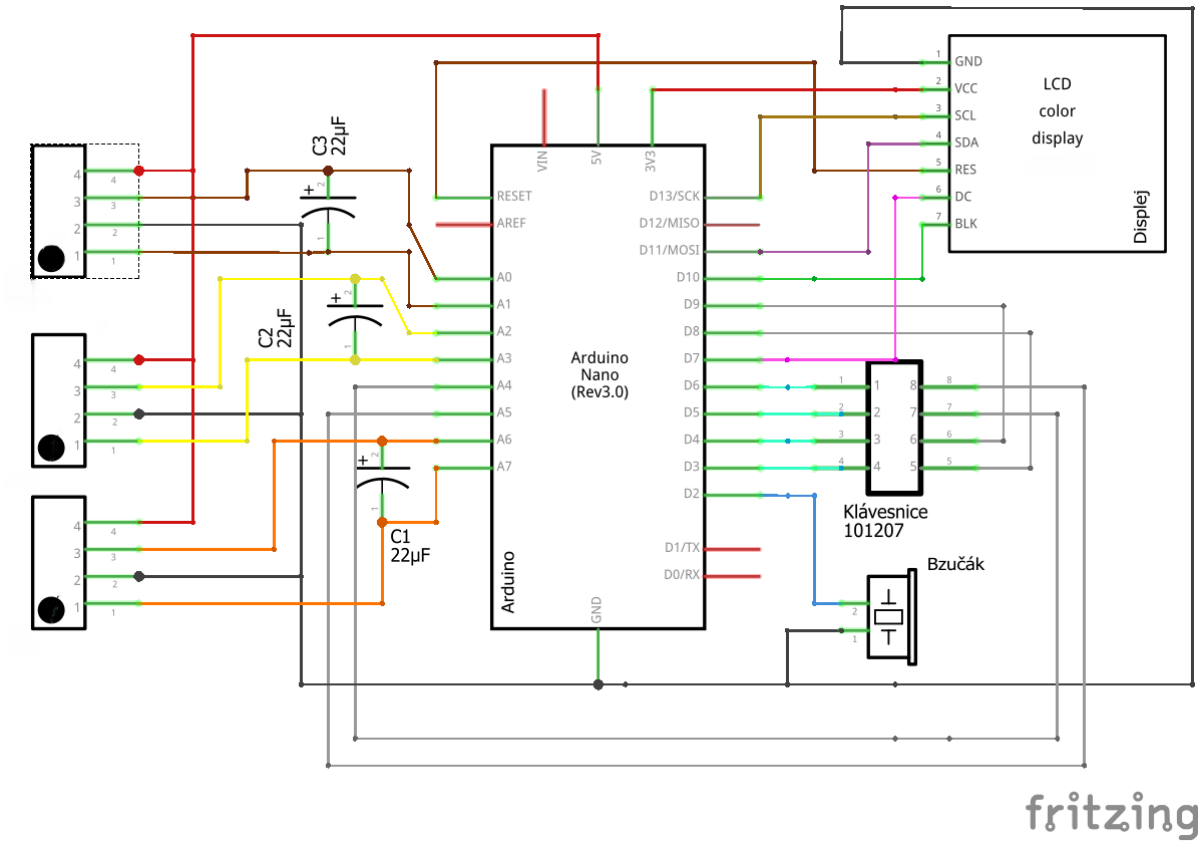
Obrázek 13: Wheatstoneův můstek. Obrázek převzat z návodu senzoru

Z obrázku je zřejmé, že výstupní napětí V_0 je mezi piny 2 a 4. Pro měření tohoto napětí jsme sledovali hodnotu na každém pinu zvlášť pomocí analogových pinů Arduino a následně tyto hodnoty odečetli od sebe. K eliminaci nežádoucího šumu na vyšších frekvencích jsme použili kondenzátory o velikosti 22 μF, čímž jsme vytvořili RC článek – filtr dolní propusti.

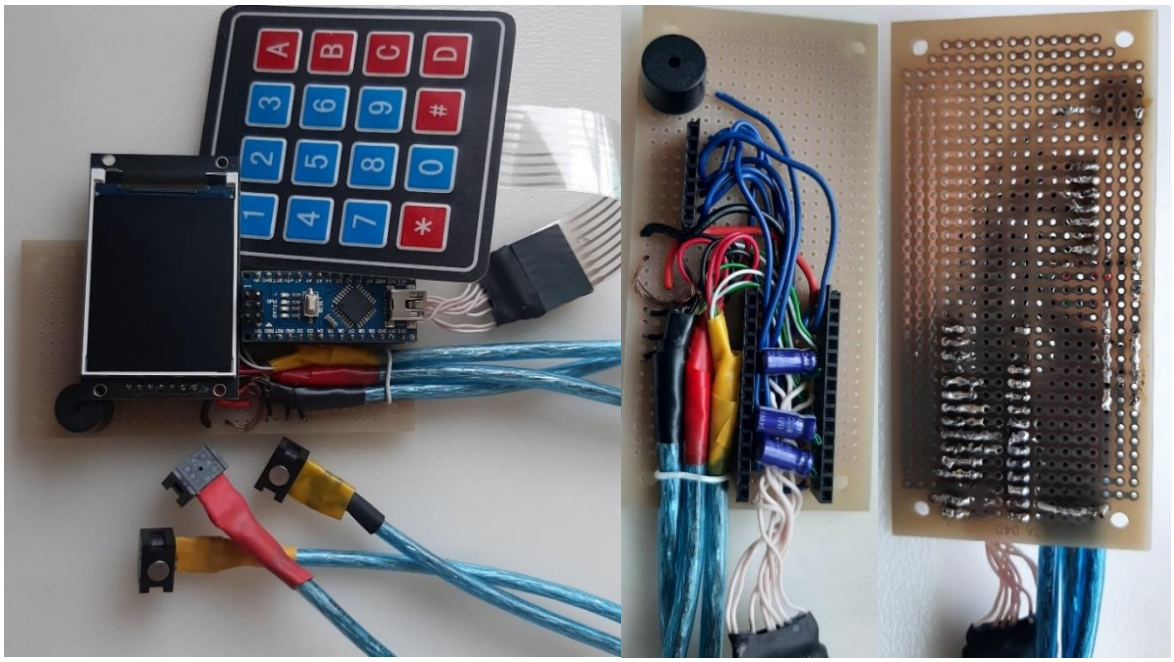
Výsledné zapojení je zobrazeno na Obrázcích 14 a 15. Pro lepší a stabilnější připevnění jsme využili desku plošných spojů a metodu pájení součástek, jak je patrné na Obrázku 16. Na Obrázku 17 jsou zobrazena některá z příslušenství vytvořená pomocí 3D tiskárny Filipem Křivohlavým z Ústavu pro životní prostředí na Přírodovědecké fakultě Univerzity Karlovy.



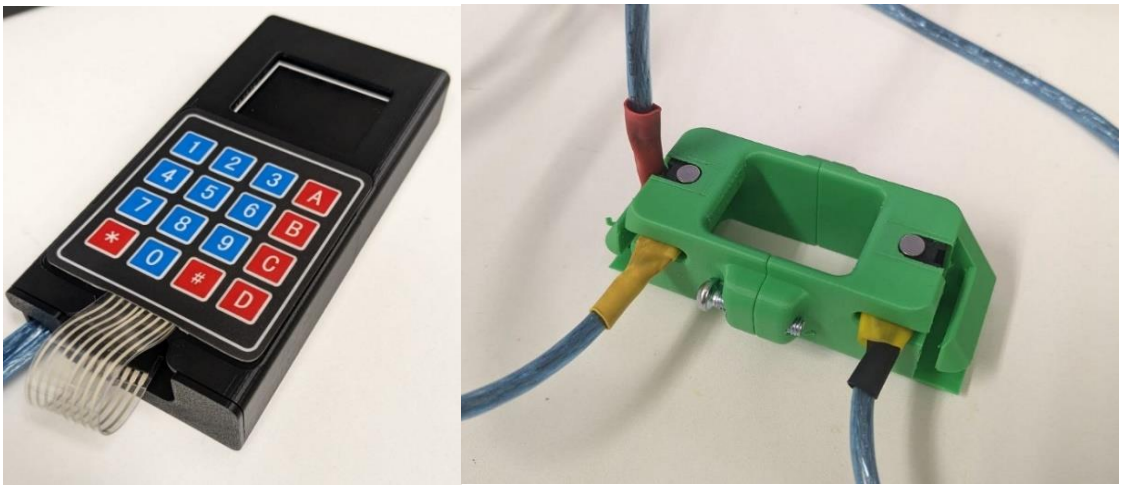
Obrázek 14: Výsledné zapojení na montážní desku. Zapojení bylo vytvořeno v aplikaci fritzing



Obrázek 15: Schéma výsledného zapojení. Zapojení bylo vytvořeno v aplikaci fritzing

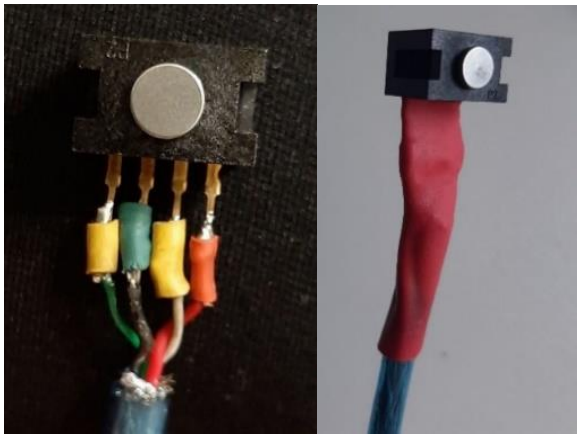


Obrázek 16: Realizace výsledného zapojení



Obrázek 17: Příslušenství k realizovanému projektu. Pouzdro (vlevo) a držadlo na sondu (vpravo)

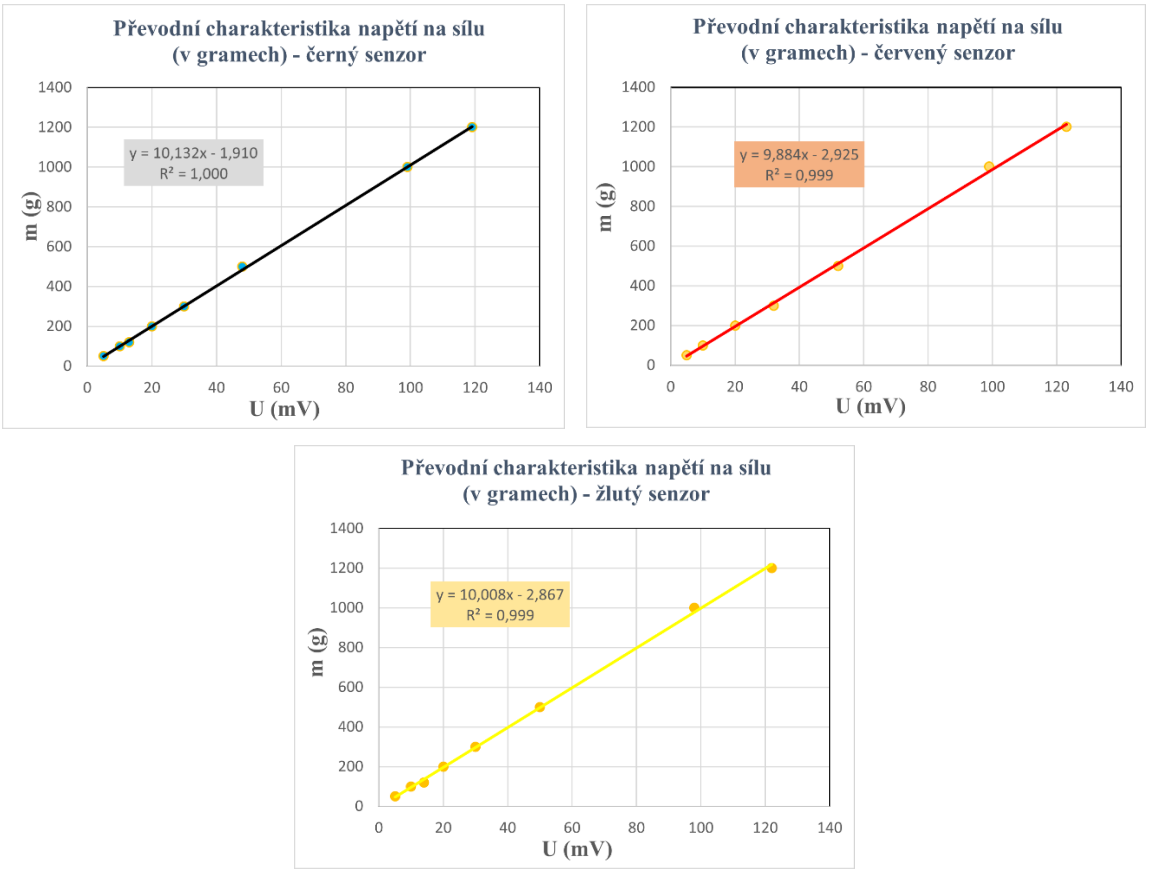
Pro překrytí a izolaci každého pinu senzoru byly použity odolné smršťující trubice, jak je patrné na Obrázku 18. Senzory, které budou snímat sílu kolmo k ploše kůže pacienta, byly označeny žlutou barvou, zatímco ten, který bude snímat sílu vodorovně, byl označen červenou. Stejně tak budou označeny informace na displeji, aby bylo jasné, k čemu patří vztahy.



Obrázek 18: Připojení senzoru

Kalibrace senzorů byla provedena pomocí sady závaží od 50 g do 1,2 kg, jak je znázorněno na Obrázku 18. Kalibrační křivky vykazují lineární charakter, přičemž výsledné přímky byly použity jako konstanty pro převod naměřených napěťových hodnot na odpovídající hmotnosti. Tyto křivky jsou prezentovány na Obrázku 20.

Obrázek 19: Kalibrace senzorů HONEYWELL FSG020WNPB



Obrázek 20: Kalibrační křivky senzorů HONEYWELL FSG020WNPB

Pro vyhodnocení chyby bylo provedeno 10 měření dvou různých hodnot pro každý senzor zvlášť. Na základě těchto údajů jsme vypočítali střední hodnotu, relativní chybu, nejistotu typu A a interval spolehlivosti. Hodnoty byly vybrány tak, aby se nepřekrývaly s kalibračními body a aby zatížení stabilně leželo na ploše senzoru. Střední hodnoty $\bar{x}$, směrodatné odchylky s, nejistotu typu A u_A, relativní chyby δ (spočítané podle Vzorce 2) jsou znázorněny v Tabulce 4.

$$\delta = \frac{x_{realná} - x_{změřená}}{x_{realná}} \cdot 100\% \quad (2)$$

Tabulka 4: Ověření přestnosti kalibrace

Senzor	Měřená veličina – 150 g				Měřená veličina – 600 g			
	$\bar{x}$ [g]	s [g]	δ [%]	$u_A[g]$	$\bar{x}$ [g]	s [g]	δ [%]	$u_A[g]$
Červený	154,8	8,9	3,2	2,8	601,9	18,5	0,3	5,8
Černý	139,0	12,9	-7,3	4,1	579,6	24,9	-3,4	7,8
Žlutý	138,2	11,4	-7,9	3,6	564,6	20,9	-5,9	6,6

Z relativně malého souboru dat (N = 10) jsme vypočítali interval spolehlivosti pomocí Studentova t-rozdělení. Výsledný interval je určen podle Vzorce 3, kde *N* je počet naměřených hodnot, *t* je tabulkové t-kritérium pro konfidenční úroveň 95 %, *s* je směrodatná odchylka a $\bar{x}$ je průměr naměřených hodnot. Intervaly spolehlivosti pro každý senzor a naměřenou hodnotu jsou zobrazené v Tabulce 5.

$$\left[\bar{x} - t \cdot \frac{s}{\sqrt{N}}, \bar{x} + t \cdot \frac{s}{\sqrt{N}}\right] \quad (3)$$

Tabulka 5: Intervaly spolehlivosti

Senzor	Měřená veličina – 150 g		Měřená veličina – 600 g	
	Interval spolehlivosti	Odchylka [%]	Interval spolehlivosti	Odchylka [%]
Červený	154 ± 6 [g]	4	601 ± 13 [g]	2
Černý	139 ± 9 [g]	6	579 ± 17 [g]	3
Žlutý	138 ± 8 [g]	5	564 ± 15 [g]	2

Z tabulek 4 a 5 je patrné, že černý a žlutý senzor mají hodnoty snížené ve srovnání s reálnou hodnotou. Proto bylo rozhodnuto zvětšit kalibrační konstanty, které používáme pro převod hodnot napětí na veličinu hmotnosti. Také je zřejmé, že se zvětšením hmotnosti se zlepšuje přesnost u každého senzoru.

4.3. Programovaní

4.3.1. Knihovny

```
1   #include <Adafruit_GFX.h>    // Core graphics library
2   #include <Adafruit_ST7789.h> // Hardware-specific library for ST7789
3   #include <SPI.h>
4   #include <Keypad.h>
5   #include <EEPROM.h>
```

Obrázek 21: Programování - použité knihovny

V projektu jsme využili několik klíčových knihoven, které zajišťují správnou funkčnost a interakci s periferiemi zařízení. Na Obrázku 21 jsou zobrazeny následující knihovny:

1. Adafruit_GFX.h: Tato knihovna poskytuje základní nástroje pro práci s grafikou. Její použití je klíčové pro vykreslování grafických prvků na displeji.

2. Adafruit_ST7789.h: Jedná se o knihovnu specificky navrženou pro práci s konkrétním typem displeje, který používáme v našem projektu. Zajišťuje komunikaci a ovládání tohoto displeje z mikrokontroléru.

3. SPI.h: Knihovna SPI je nezbytná pro komunikaci se zařízeními připojenými pomocí sériového periferního rozhraní (SPI [6]). V našem případě se jedná o komunikaci s periferiemi jako je EEPROM nebo samotný displej.

4. Keypad.h: Tato knihovna usnadňuje práci s klávesnicí. Pomocí ní můžeme detekovat stisknuté klávesy a reagovat na uživatelův vstup.

5. EEPROM.h: Knihovna EEPROM umožňuje práci s pamětí typu EEPROM, což je „Elektricky Vymazatelná Paměť Pouze pro Čtení“ [7], která si ukládá data i po vypnutí napájení. V našem projektu je využívána k ukládání nastavených hodnot - hranic síly při restartování zařízení.

4.3.2. Předem definované hodnoty

V další části programu jsou definovány piny, viz Obrázek 22, které jsou použity pro připojení displeje a klávesnice. V řádcích 7 až 10 jsou specifikovány piny pro řízení displeje. Tyto piny jsou klíčové pro správnou komunikaci s displejem a zajišťují zobrazení informací.

Na řádku 12 je uvedena hodnota referenčního napětí, která je důležitá pro správné čtení hodnot výstupního napětí na senzorech. Pro Arduina se obvykle používá referenční napětí 5 V, což odpovídá hodnotě 5000 mV.

[6] Serial Peripheral Interface.

[7] Electrically Erasable Programmable Read-Only Memory.

Následně program pokračuje v nastavení hodnot pro klávesnici typu 4x4. V řádcích 16 až 22 jsou interpretovány jednotlivá tlačítka na symboly, které chceme použít. Tento krok je důležitý pro pochopení vstupů od uživatele a jejich správnou interpretaci v programu. Poté následují definice použitých pinů pro sloupce a řádky klávesnice. Tyto piny umožňují detekci stisknutých tlačítek.

```
7   #define TFT_CS 10
8   #define TFT_RST -1
9   #define TFT_DC 7
10  Adafruit_ST7789 tft = Adafruit_ST7789(TFT_CS, TFT_DC, TFT_RST);
11
12  #define Vref 5000
13  #define ranks  4
14  #define columns  4
15
16  char symbols[ranks][columns] =
17  {
18    {'1','2','3','A'},
19    {'4','5','6','B'},
20    {'7','8','9','C'},
21    {'*','0','#','D'}
22  };
23  byte ranks_pins[ranks] = {3,4,5,6};
24  byte columns_pins[columns] = {8,9,18,19};
25  Keypad keyboard = Keypad(makeKeymap(symbols), ranks_pins, columns_pins, ranks, columns);
```

Obrázek 22: Programování - definované hodnoty

Na řádcích 27 až 41 na Obrázku 23 jsou definovány různé parametry a proměnné, které budou použity během programu. Mezi tyto předdefinované hodnoty patří definice pinů, které slouží k připojení senzorů (ř. 32 až 37) a bzučáku (ř. 41).

```
27  char symbol;
28  int result;
29  int A, B, C, D;
30
31  int i = 0;
32  int Pin0 = A0;  //black
33  int Pin1 = A1;  //black
34  int Pin2 = A2;  //yellow
35  int Pin3 = A3;  //yellow
36  int Pin4 = A6;  //red
37  int Pin5 = A7;  //red
38  double Voltage0, Voltage1, Voltage2;
39  double Voltage3, Voltage4, Voltage5;
40  double Mass0, Mass1, Mass2, Mass;
41  int speakerPin = 2;
```

Obrázek 23: Programování - definované hodnoty 2

4.3.3. Funkce

- **Print**

Na řádcích 53 až 57 je definována funkce "print", jak je naznačeno na Obrázku 24. Tato funkce slouží k zobrazení informací na displeji. Jejími vstupy jsou souřadnice počátku psaní (x, y), barva textu a velikost znaků. Po volání této funkce v hlavní funkci "loop" je potřeba specifikovat informace, které chceme zobrazit na displeji.

```
53  void print(int x, int y, char color, int size) {
54    tft.setCursor(x,y);
55    tft.setTextColor(color, ST77XX_BLACK);
56    tft.setTextSize(size);
57  }
```

Obrázek 24: Programování - funkce "print"

Obrázek 25: Programování - funkce "print", výsledek volání funkce

Obrázek 25 poskytuje ukázku výsledku volání funkce, zatímco programování posledních tří řádků je zobrazeno na Obrázku 26, kde D má hodnotu 500. Barva symbolů zobrazených žlutou a červenou barvou odpovídá barvě senzorů. Žlutá barva je přiřazena k měření hodnoty součtu dvou žlutých senzorů, zatímco červená barva je použita pro senzor identifikovaný červenou páskou.

```
228   print(0, 185, ST77XX_WHITE, 2);
229   tft.println("Sound notification  range [gram]:");
230   tft.print("[");
231   tft.print(D);
232   tft.println(",2000]");
233
234   print(0, 260, ST77XX_WHITE, 2);
235   tft.println("Press * to set the  range");
```

Obrázek 26: Programování - funkce "print", volání funkce

- **Error a print_set_the_range**

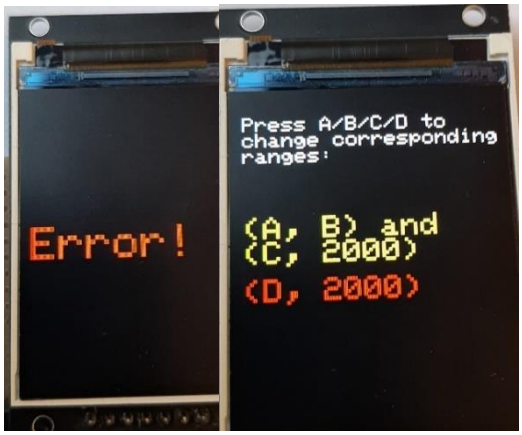
Na Obrázku 27 jsou uvedeny dvě podobné funkce: "error" a "print_set_the_range". Tyto funkce využívají předchozí funkci "print" a jsou určeny pro zobrazení informací při nastavení hranic uživatele.

```
73  void error(void) {
74    tft.fillScreen(ST77XX_BLACK);
75    print(0, 150, ST77XX_RED, 5);
76    tft.setTextColor(ST77XX_RED, ST77XX_BLACK);
77    tft.println("Error!");
78    delay(500);
79  }
80
81  void print_set_the_range(void) {
82    tft.fillScreen(ST77XX_BLACK);
83    print(0, 20, ST77XX_WHITE, 2);
84    tft.println("Press A/B/C/D to    change correspondingranges:");
85
86    print(0, 120, ST77XX_YELLOW, 3);
87    tft.println("(A, B) and   (C, 2000)");
88
89    print(0, 180, ST77XX_RED, 3);
90    tft.setTextColor(ST77XX_RED, ST77XX_BLACK);
91    tft.println("(D, 2000)");
92  }
```

Obrázek 27: Programování – funkce "error" a "print_set_the_range"

Funkce "error", jak název naznačuje, je vyvolána při zadání neplatného vstupu. Na displeji se zobrazí červený nadpis, jak je patrné na Obrázku 28 vlevo. Tato funkce je užitečná pro označení chybových stavů a upozornění uživatele na nesprávný vstup.

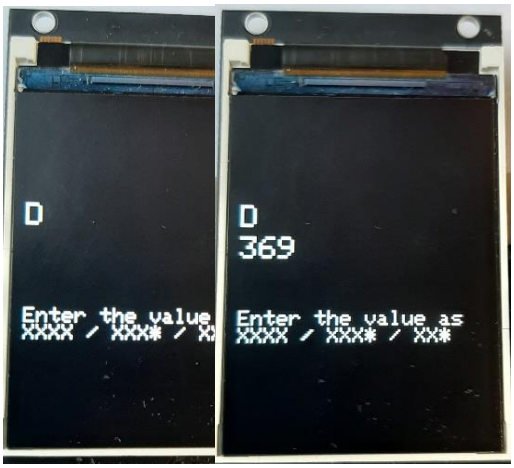
Druhá funkce, "print_set_the_range", slouží k zobrazení pokynů pro uživatele pro nastavení hranic síly. Výsledek volání této funkce je znázorněn na Obrázku 28 vpravo. Tato funkce poskytuje uživateli jasné instrukce a usnadňuje proces nastavení hranic síly podle jeho potřeb. Barva symbolů jako před tím odkazuje na příslušnost k určitým senzorům.



Obrázek 28: Programování - funkce "error" a "print_set_the_range", výsledek volání funkce

- **Set_the_range**

Funkce "set_the_range" je vyvolána, pokud uživatel reaguje na předchozí stav displeje stisknutím jednoho z písmen A-D. Pokud je vybrán jiný symbol, zobrazí se chybová hláška na displeji, nastavení hranic síly bude ukončeno a žádné změny nebudou uloženy. V případě správného vstupu se na displeji zobrazí vybraný symbol a postupně se zobrazí i číslo, které uživatel zadá na klávesnici, jak je ukázáno na Obrázku 29. Maximální počet symbolů je 4, pokud uživatel chce zadat 3 místné číslo nebo menší, výběr musí být ukončen stisknutím symbolu hvězdičky.



Obrázek 29: Programování - funkce "set_the_range", výsledek volaní funkce

Vstup je na začátku považován za typ "char", ale poté je převeden na typ "int", což je patrné na řádku 109, Obrázek 30. Postupně je vstup od uživatele ukládán do proměnné "result", která je i návratovou hodnotou funkce.

```
94  int set_the_range(void) {
95    print(0, 200, ST77XX_WHITE, 2);
96    tft.println("Enter the value as  XXXX / XXX* / XX*");
97    int counter = 0;
98    result = 0;
99    char symbol1 = ' ';
100   for (int i = 0; i < 4; i++) {
101     symbol1 = keyboard.waitForKey();
102     if (symbol1 == '*'){
103       break;
104     }
105     else if(symbol1 == 'A' | symbol1 == 'B' | symbol1 == 'C' | symbol1 == 'D' | symbol1 == '#'){
106       result = 2000;
107       break;
108     }
109     counter = int(symbol1 - '0');
110     result = result*10 + counter;
111     print(0, 130, ST77XX_WHITE, 3);
112     tft.println(result);
113   }
114   return result;
115 }
```

Obrázek 30: Programování - funkce "set_the_range"

- **EepromIntWrite a EepromIntRead**

```
59  void eepromIntWrite(int address, int value){
60   byte byte1 = value;
61   byte byte2 = value >> 8;
62   EEPROM.write(address, byte1);
63   EEPROM.write(address + 1, byte2);
64  }
65
66  int eepromIntRead(int address){
67   int value = EEPROM.read(address) + ((EEPROM.read(address+1)) << 8) ;
68   return value;
69  }
```

Obrázek 31: Programování - funkce "EepromIntWrite" a "EepromIntRead"

Knihovna EEPROM a dvě uvedené funkce na Obrázku 31 slouží k ukládání nastavených hranic do paměti Arduina a k jejich přečtení po restartu zařízení. Vzhledem k možnosti ukládat čísla až do hodnoty 1999 je při ukládání nebo čtení čísla nezbytné použít bitový posun.

Řádky 123 až 126 na Obrázku 32 jsou součástí funkce "setup", která je vyvolána při startu Arduina. Tyto řádky načtou hodnoty z uvedených adres a uloží je do proměnných A až D. Na Obrázku 33 je zobrazeno ukládání nové hodnoty od uživatele do proměnné A a následné zapsání této hodnoty na adresu, odkud je poté načtena při restartu programu.

```
123  A = eepromIntRead(0);
124  B = eepromIntRead(10);
125  C = eepromIntRead(20);
126  D = eepromIntRead(30);
```

Obrázek 32: Programování - funkce "EepromIntRead", volání funkce

```
141          case 'A':
142            if (result < B){
143              A = result;
144              eepromIntWrite(0, A);
145            }
146            else{error();}
147            break;
```

Obrázek 33: Programování - funkce "EepromIntWrite", volání funkce

- **PlayTone**

```
43  void playTone(int tone, int duration){
44    for (long i = 0; i < duration * 1000L; i += tone * 2)
45    {
46      digitalWrite(speakerPin, HIGH);
47      delayMicroseconds(tone);
48      digitalWrite(speakerPin, LOW);
49      delayMicroseconds(tone);
50    }
51  }
```

Obrázek 34: Programování - funkce "PlayTone"

Funkce PlayTone slouží k vytvoření zvukového signálu pomocí miniaturního bzučáku. Tato funkce, která je uvedena na Obrázku 34, byla převzata z oficiálních stránek Arduina.

```
237     Mass = Mass0 + Mass1;
238     if((Mass > A & Mass < B) || Mass > C){
239       playTone(4000,500);
240     }
241     if(Mass2 > D){
242       playTone(500,500);
243     }
```

Obrázek 35: Programování - funkce "PlayTone", volání funkce

Na Obrázku 35 jsou zobrazeny řádky, které vyvolávají zvukový signál s různě nastaveným tónem v závislosti na senzoru, kde naměřené hodnoty byly překročeny.

- **Setup**

```
117 void setup(void) {
118   Serial.begin(9600);
119   tft.init(240, 320); // Init ST7789 320x240
120   tft.fillScreen(ST77XX_BLACK);
121   pinMode(speakerPin, OUTPUT);
122
123   A = eepromIntRead(0);
124   B = eepromIntRead(10);
125   C = eepromIntRead(20);
126   D = eepromIntRead(30);
127 }
```

Obrázek 36: Programování - funkce "Setup"

Standardní funkce "setup" je volána automaticky při startu Arduina. Na Obrázku 36 jsou zobrazeny příkazy, které se provedou jako první. Na řádku 118 je příkaz Serial.begin(), který

nastavuje rychlost přenosu dat v bitech za sekundu pro sériový přenos dat [8]. Následující řádek nastavuje počet pixelů použitých na displeji. Řádek 120 vymaže všechny informace z displeje, které by mohly zůstat z předchozího použití zařízení. Poté je pin použitý pro bzučák nastaven do režimu "output", abychom mohli slyšet alarmy. Poslední čtyři řádky již byly popsány výše.

- **Loop**

Poslední, ale nejdůležitější funkcí je funkce "loop". Je to standardní funkce Arduina, která se vyznačuje tím, že se neustále opakuje. V tomto případě je ve funkci "loop" celkem 115 řádků, v textu níže se zaměřím pouze na významné řádky. Celou funkce tvoří Přílohu č. 1.

V první části funkce jsou použity dva příkazy z knihovny "Keypad.h" (ř. 131 a ř. 134, Obrázek 37), které se liší tím, že první příkaz se pokusí načíst hodnotu z klávesnice. V případě, že klávesnice nebyla zmáčknuta, do proměnné "symbol" nic nebude uloženo, program pokračuje dále, ale příkaz na řádku 134 zastaví běh programu a vyčkává na vstup od uživatele. V dalších řádcích je popsána kontrola a ukládání vstupu uživatele v závislosti na tom, která klávesa byla zmáčknuta.

```
131   symbol = keyboard.getKey();
132   if (symbol == '*'){
133     print_set_the_range();
134     symbol = keyboard.waitForKey();
```

Obrázek 37: Programování - část funkce "Loop"

Na Obrázku 38 je naprogramováno zpracování výstupního napětí ze senzoru. Pro odstranění šumu z načítaných signálů používáme průměrování posledních 100 naměřených signálů. Tato veličina byla zjištěna jako optimální během testování: rychlost změn hodnot není příliš nízká, aby to uživatel zaznamenal, ani příliš vysoká, aby případný šum výrazně ovlivnil měření. Hodnoty napětí čteme standardní funkcí "analogRead" a poté přemapujeme číslo z rozsahu [0, 1023] do rozsahu [0, 5000], čímž získáme veličiny napětí. Řádky 195 až 197 převedou naměřené a zprůměrované hodnoty napětí na hmotnost v gramech podle kalibračních křivek, což usnadní pochopení uživateli.

[8] Podle oficiálního webu Arduino.

```
177     i++;
178
179     Voltage0 += map(analogRead(Pin0), 0, 1023, 0, Vref); //black
180     Voltage1 += map(analogRead(Pin1), 0, 1023, 0, Vref); //black
181     Voltage2 += map(analogRead(Pin2), 0, 1023, 0, Vref); //yellow
182     Voltage3 += map(analogRead(Pin3), 0, 1023, 0, Vref); //yellow
183     Voltage4 += map(analogRead(Pin4), 0, 1023, 0, Vref); //red
184     Voltage5 += map(analogRead(Pin5), 0, 1023, 0, Vref); //red
185     if (i == 100) {
186       Voltage0 = Voltage0/i;
187       Voltage1 = Voltage1/i;
188       Voltage2 = Voltage2/i;
189       Voltage3 = Voltage3/i;
190       Voltage4 = Voltage4/i;
191       Voltage5 = Voltage5/i;
192
193       i = 0;
194
195       Mass0 = 10.6*(Voltage0 - Voltage1) - 1.9; //black
196       Mass1 = 10.8*(Voltage2 - Voltage3) - 2.8; //yellow
197       Mass2 = 9.88*(Voltage4 - Voltage5) - 2.9; //red
```

Obrázek 38: Programování - část funkce "Loop" 2

Poté probíhá proces vykreslování hodnot na displej a kontrola podmínek překročení hranic. Po dokončení těchto kroků začne program znovu běžet od prvního řádku funkce "loop".

5. Závěr

V závěru této bakalářské práce je nutné zdůraznit, že problematika stavu fascií a jejich význam v pohybové funkci člověka představuje klíčové téma v oboru fyzioterapie a lékařství obecně. Fascie představují důležitou součást pohybového aparátu, a proto je nezbytné porozumět jejich struktuře, funkcím a možným poruchám. Jedním z hlavních překážek v pokroku v této oblasti je nedostatek standardizace při vyšetření fascií pomocí ultrazvuku. Tento nedostatek brání v dosažení konzistentních výsledků a přesné interpretaci ultrazvukových snímků.

Tato práce se zaměřuje na řešení problému nedostatku standardizace síly vyvíjené vyšetřujícím při sonografickém vyšetření fascií. Pro tento účel byl vyvinut systém s třemi senzory síly a mikropočítačem Arduino, který je určen k přímé instalaci na sonografickou sondu pomocí speciálně navrženého držadla (vývoj držadla přesahuje rámec této bakalářské práce). Držadlo je navrženo tak, aby přesně odpovídalo tvaru sondy a umožnilo provádět přesná měření. I když v době psaní této práce existuje pouze nedokonalý prototyp držadla, který neumožnil provádět rozsáhlou sadu měření přímo během vyšetření, celý vývoj byl konzultován s fyzioterapeuty z 2. lékařské fakulty Univerzity Karlovy a přizpůsoben jejich požadavkům. Byla implementována možnost nastavení hranic síly, které se mohou měnit v závislosti na místě vyšetření fascií. Formát těchto hranic a zvukový signál, který se objeví při překročení hranic, byly také instalovány pro pohodlí vyšetřujícího.

Jedním z možných vylepšení projektu by mohlo spočívat v optimalizaci zapojení senzorů síly typu HONEYWELL FSG020WNPB prostřednictvím instalace diferenčního zesilovače. Tím by se minimalizoval vliv souhlasného rušení a zvýšila přesnost měření. Tato úprava by mohla přinést zlepšení stability a spolehlivosti výsledků. Kromě toho je pravděpodobné, že během testování projektu vzniknou další oblasti, které je třeba zdokonalit. To může zahrnovat například implementaci dodatečných funkcí pro snadnější ovládání a nastavení. Z tohoto důvodu je navržená verze v této práci považována za první krok směrem k vylepšení, který bude následován dalšími iteracemi a úpravami.

Závěrem lze tedy říci, že standardizace síly při sonografickém vyšetření fascií je nezbytným krokem k dosažení přesných a spolehlivých výsledků. Je třeba podporovat další výzkum v této oblasti a hledat inovativní přístupy ke standardizaci vyšetřovacích postupů, aby bylo možné lépe porozumět stavu fascií a optimalizovat terapeutické postupy v oblasti fyzioterapie.

6. Seznam literatury

[1] S. Adstrum, G. Hedley, R. Schleip, C. Stecco, C.A. Yucesoy (2017). *Defining the fascial systém.* Online. Journal of Bodywork and Movement Therapies. Volume 21, Issue 1, Pages 173-177, ISSN 1360-8592. Dostupné z: https://doi.org/10.1016/j.jbmt.2016.11.003.

[2] Národní zdravotnický informační portál (cit. 06.04.2024). *Fascie a fasciální systém*. Online. Praha: Ministerstvo zdravotnictví ČR a Ústav zdravotnických informací a statistiky ČR. ISSN 2695-0340. Dostupné z: https://www.nzip.cz/clanek/1213-fascie-a-fascialni-system.

[3] D. Lesondak (2018). *Fascia: What it is and Why it Matters*. Online. Publisher: Handspring Publishing, ISBN: 1909141550.

[4] C. Stecco, et al (2015). *Functional Atlas of the Human Fascial Systém*. ISBN 978-0-7020-4430- 4.

[5] P. Pavan, A. Stecco, R.Stern (2014). *Painful Connections: Densification Versus Fibrosis of Fascia*. Online. Current pain and headache reports. Volume 18, Issue 8, Page 441. Dostupné z: https://doi.org/10.1007/s11916-014-0441-4.

[6] M. Zügel, C. N Maganaris, J. Wilke, K. Jurkat-Rott, et al. (2018) *Fascial tissue research in sports medicine: from molecules to tissue adaptation, injury and diagnostics: consensus statement.* Online. *British Journal of Sports Medicine*. Volume 52, Issue 23, Page 1497. Dostupné z: https://doi.org/10.1136/bjsports-2018-099308.

[7] M. Sedlár, E. Staffa, V. Mornstein (2014). *Zobrazovací metody využívající neionizující záření.* Online. Masarykova univerzita. ISBN 978-80-210-7156-8.

[8] Chat GPT verze 3.5. Dostupné z: https://chatgpt.com

7. Příloha č. 1: Celý kód programu.

```
#include <Adafruit_GFX.h>    // Core graphics library
#include <Adafruit_ST7789.h> // Hardware-specific library for ST7789
#include <SPI.h>
#include <Keypad.h>
#include <EEPROM.h>

#define TFT_CS 10
#define TFT_RST -1
#define TFT_DC 7
Adafruit_ST7789 tft = Adafruit_ST7789(TFT_CS, TFT_DC, TFT_RST);

#define Vref 5000
#define ranks  4
#define columns  4

char symbols[ranks][columns] =
{
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte ranks_pins[ranks] = {3,4,5,6};
byte columns_pins[columns] = {8,9,18,19};
Keypad keyboard = Keypad(makeKeymap(symbols), ranks_pins, columns_pins, ranks, columns);

char symbol;
int result;
int A, B, C, D;
int i = 0;
int Pin0 = A0;  //black
int Pin1 = A1;  //black
int Pin2 = A2;  //yellow
int Pin3 = A3;  //yellow
int Pin4 = A6;  //red
int Pin5 = A7;  //red
double Voltage0, Voltage1, Voltage2;
double Voltage3, Voltage4, Voltage5;
double Mass0, Mass1, Mass2, Mass;
int speakerPin = 2;

void playTone(int tone, int duration){
  for (long i = 0; i < duration * 1000L; i += tone * 2)
  {
    digitalWrite(speakerPin, HIGH);
    delayMicroseconds(tone);
    digitalWrite(speakerPin, LOW);
    delayMicroseconds(tone);
  }
}

void print(int x, int y, char color, int size) {
  tft.setCursor(x,y);
  tft.setTextColor(color, ST77XX_BLACK);
  tft.setTextSize(size);
}
```

```
58
59  void eepromIntWrite(int address, int value)
60  {
61   byte byte1 = value;
62   byte byte2 = value >> 8;
63   EEPROM.write(address, byte1);
64   EEPROM.write(address + 1, byte2);
65  }
66
67  int eepromIntRead(int address)
68  {
69   int value = EEPROM.read(address) + ((EEPROM.read(address+1)) << 8) ;
70   return value;
71  }
72
73  void error(void) {
74    tft.fillScreen(ST77XX_BLACK);
75    print(0, 150, ST77XX_RED, 5);
76    tft.setTextColor(ST77XX_RED, ST77XX_BLACK);
77    tft.println("Error!");
78    delay(500);
79  }
80
81   void print_set_the_range(void) {
82     tft.fillScreen(ST77XX_BLACK);
83     print(0, 20, ST77XX_WHITE, 2);
84     tft.println("Press A/B/C/D to    change correspondingranges:");
85
86     print(0, 120, ST77XX_YELLOW, 3);
87     tft.println("(A, B) and   (C, 2000)");
88
89     print(0, 180, ST77XX_RED, 3);
90     tft.setTextColor(ST77XX_RED, ST77XX_BLACK);
91     tft.println("(D, 2000)");
92   }
93
94   int set_the_range(void) {
95     print(0, 200, ST77XX_WHITE, 2);
96     tft.println("Enter the value as  XXXX / XXX* / XX*");
97     int counter = 0;
98     result = 0;
99     char symbol1 = ' ';
100    for (int i = 0; i < 4; i++) {
101      symbol1 = keyboard.waitForKey();
102      if (symbol1 == '*'){
103        break;
104      }
105      else if(symbol1 == 'A' | symbol1 == 'B' | symbol1 == 'C' | symbol1 == 'D' | symbol1 == '#'){
106        result = 2000;
107        break;
108      }
109      counter = int(symbol1 - '0');
110      result = result*10 + counter;
111      print(0, 130, ST77XX_WHITE, 3);
112      tft.println(result);
113    }
114    return result;
115  }
116
```

```
void setup(void) {
  Serial.begin(9600);
  tft.init(240, 320); // Init ST7789 320x240
  tft.fillScreen(ST77XX_BLACK);
  pinMode(speakerPin, OUTPUT);

  A = eepromIntRead(0);
  B = eepromIntRead(10);
  C = eepromIntRead(20);
  D = eepromIntRead(30);
}

void loop() {

  symbol = keyboard.getKey();
  if (symbol == '*'){
    print_set_the_range();
    symbol = keyboard.waitForKey();
    tft.fillScreen(ST77XX_BLACK);
    print(0, 100, ST77XX_WHITE, 3);
    tft.println(symbol);
    if (symbol == 'A' | symbol == 'B' | symbol == 'C' | symbol == 'D'){
      result = set_the_range();
      switch(symbol){
        case 'A':
          if (result < B){
            A = result;
            eepromIntWrite(0, A);
          }
          else{error();}
          break;
        case 'B':
          if (result < C & result > A){
            B = result;
            eepromIntWrite(10, B);
          }
          else{error();}
          break;
        case 'C':
          if (result < 2000 & result > B){
            C = result;
            eepromIntWrite(20, C);
          }
          else{error();}
          break;
        case 'D':
          if (result < 2000){
            D = result;
            eepromIntWrite(30, D);
          }
          else{error();}
          break;
        default:
          break;
      }
    }
    else{error();}
    tft.fillScreen(ST77XX_BLACK);
  }

```

```
  i++;

  Voltage0 += map(analogRead(Pin0), 0, 1023, 0, Vref); //black
  Voltage1 += map(analogRead(Pin1), 0, 1023, 0, Vref); //black
  Voltage2 += map(analogRead(Pin2), 0, 1023, 0, Vref); //yellow
  Voltage3 += map(analogRead(Pin3), 0, 1023, 0, Vref); //yellow
  Voltage4 += map(analogRead(Pin4), 0, 1023, 0, Vref); //red
  Voltage5 += map(analogRead(Pin5), 0, 1023, 0, Vref); //red
  if (i == 100) {
    Voltage0 = Voltage0/i;
    Voltage1 = Voltage1/i;
    Voltage2 = Voltage2/i;
    Voltage3 = Voltage3/i;
    Voltage4 = Voltage4/i;
    Voltage5 = Voltage5/i;

    i = 0;

    Mass0 = 10.6*(Voltage0 - Voltage1) - 1.9; //black
    Mass1 = 10.8*(Voltage2 - Voltage3) - 2.8; //yellow
    Mass2 = 9.88*(Voltage4 - Voltage5) - 2.9; //red

    if(Mass0 < 0){
      Mass0 = 0;
    }
    if (Mass1 < 0){
      Mass1 = 0;
    }
    if (Mass2 < 0){
      Mass2 = 0;
    }

     print(20, 20, ST77XX_YELLOW, 5);
     tft.print(int(Mass0+Mass1));
     tft.println("g  ");

     print(20, 140, ST77XX_RED, 5);
     tft.setTextColor(ST77XX_RED, ST77XX_BLACK);
     tft.print(int(Mass2));
     tft.println("g  ");

     print(0, 70, ST77XX_WHITE, 2);
     tft.println("Sound notification  range [gram]:");
     tft.print("[");
     tft.print(A);
     tft.print(",");
     tft.print(B);
     tft.print("]&[");
     tft.print(C);
     tft.println(",2000]");

     print(0, 185, ST77XX_WHITE, 2);
     tft.println("Sound notification  range [gram]:");
     tft.print("[");
     tft.print(D);
     tft.println(",2000]");

     print(0, 260, ST77XX_WHITE, 2);
     tft.println("Press * to set the  range");

     Mass = Mass0 + Mass1;
     if((Mass > A & Mass < B) || Mass > C){
       playTone(4000,500);
     }
     if(Mass2 > D){
       playTone(500,500);
     }

   }

}
```