

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra radioelektroniky

Konfigurovatelný nápojový automat

Jan Řehák

Vedoucí práce: doc. Ing. Stanislav Vitek, Ph.D.

Studijní program: Elektronika a komunikace

Květen 2024

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Řehák** Jméno: **Jan** Osobní číslo: **507648**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra radioelektroniky**
Studijní program: **Elektronika a komunikace**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Konfigurovatelný nápojový automat

Název bakalářské práce anglicky:

Configurable Beverage Machine

Pokyny pro vypracování:

Cílem bakalářské práce je návrh konfigurovatelného stavového automatu, který bude možné použít pro flexibilní realizaci nápojových a výdejových automatů. Při vypracování se řiďte následujícími pokyny:

- 1) proveďte rešerši existujících řešení pro výdejové automaty;
- 2) na základě studie navrhnete plně konfigurovatelný nápojový automat;
- 3) navržené řešení implementujte prostřednictvím vhodné embedded platformy;
- 4) implementované zařízení otestujte v reálném provozu a diskutujte případné nedostatky.

Seznam doporučené literatury:

- [1] WHITE, Elecia. Making Embedded Systems: Design Patterns for Great Software. " O'Reilly Media, Inc.", 2011.
- [2] MALÝ, Martin. Data, čipy, procesory: vlastní integrované obvody na koleni. CZ. NIC, zspo, 2020.
- [3] NOVELLO, Carmine. Mastering STM32 book. 2019.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

doc. Ing. Stanislav Vítek, Ph.D. katedra radioelektroniky FEL

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

Datum zadání bakalářské práce: **08.02.2024**

Termín odevzdání bakalářské práce: **24.05.2024**

Platnost zadání bakalářské práce: **21.09.2025**

doc. Ing. Stanislav Vítek, Ph.D.
podpis vedoucí(ho) práce

doc. Ing. Stanislav Vítek, Ph.D.
podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen 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 studenta

Poděkování

Děkuji vedoucímu práce doc. Ing. Stanislavu Vítkovi, Ph.D. za jeho odborné připomínky, rady a cenné konzultace při tvorbě této práce. Také bych rád poděkoval své partnerce a rodině za podporu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací. V Praze, 22. května 2024

Abstrakt

Tato práce popisuje existující řešení pro výdejové automaty, jejich možné dělení, a detailně přibližuje jejich fungování. Dále popisuje návrh konfigurovatelného nápojového automatu a řešerši, na základě které byl tento automat navrhnout. V Poslední části tato práce popisuje realizaci všech částí takového automatu a jeho detailní fungování.

Klíčová slova: konfigurovatelný nápojový automat, mikrokontrolér, Arduino Nano Every, 3D tisk, řízení stejnosměrných motorů, čerpadla

Vedoucí práce: doc. Ing. Stanislav Vítek, Ph.D.

Abstract

This work describes existing solutions for vending machine, their possible subdivision, and presents their operation in detail. It also describes the design of a configurable beverage machine and the research on the basis of which this dispenser was designed. In the last part, this paper describes the implementation of all parts of such a vending machine and its detailed functioning.

Keywords: configurable beverage machine, microcontroller, Arduino Nano Every, 3D printing, DC motor control, pumps

Title translation: Configurable Beverage Machine

Obsah

Úvod	1
1 Koncept výdejních automatů	3
1.1 Nápojové automaty	4
1.2 Potravinové automaty	5
1.3 Automaty na spotřební zboží	7
1.4 Netradiční řešení nápojových automatů	7
2 Návrh nápojového automatu	9
2.1 Čerpadla	9
2.2 Řízení čerpadel	11
2.2.1 H-můstek	12
2.2.2 Relé	12
2.3 Kontrola průtoku	12
2.4 Uživatelské rozhraní	14
2.5 Volba embedded platformy	14
2.6 Konfigurace	15
2.7 Napájení komponent	15
2.8 Zapojení modulů	16
2.9 Návrh 3D modelu	17
2.10 Návrh software	18
3 Realizace	19
3.1 Hardware	19
3.2 Software	21
3.2.1 Prvotní nastavení	21
3.2.2 Řídicí program	24
4 Nápojový automat	29
Závěr	33
Literatura	35

Obrázky

1.1 Nápojový automat typu post-mix [5]	4
1.2 Výdejový automat [9]	6
2.1 Membránové čerpadlo [14]	9
2.2 Peristaltické čerpadlo [15]	10
2.3 Princip fungování H-můstku [20]	12
2.4 Zapojení senzorů váhy [23]	13
2.5 Arduino Nano Every [25]	15
2.6 Blokové schéma zapojení napájení	16
2.7 Blokové schéma zapojení modulů	16
2.8 3D model v programu Autodesk Inventor	17
2.9 Vývojový diagram pro vývoj řídicího programu	18
3.1 Elektronický návrh desky s tlačítky	20
3.2 Elektronický návrh pro hlavní desku	20
3.3 Diagram stavového automatu ...	25
3.4 Zobrazení funkce <i>LCD_Print_Insert_Cup</i> na displeji	27
4.1 Přední část automatu	29
4.2 Umístění čerpadel	29
4.3 Zapojení čerpadel	30

Tabulky

3.1 Použité stavy stavového automatu	24
4.1 Použité materiály a součástky ..	31



Úvod

Výdejové automaty jsou dnes velmi rozšířeným způsobem prodeje různého zboží bez nutnosti přítomnosti obsluhy. Prvotní zmínky sahají do devatenáctého století. Od počátků své existence nabraly tyto automaty mnoho podob, první plně automatické vydávaly poštovní známky, později se do jejich nabídky přidaly další doplňky pro posílání dopisů jako jsou obálky, pohledy či papíry. V devatenáctém století se v Paříži objevily první nápojové automaty, které, nepřekvapivě na tehdejší dobu, vydávaly pivo, víno a tvrdý alkohol [1]. Dnes můžeme různé varianty výdejních automatů, ať už na potraviny, nápoje či spotřební zboží, nalézt prakticky kdekoli, v chodbách fakulty elektrotechnické, ve vestibulech metra, v čekárnách na úřadech nebo ve výrobních halách.

Právě výdej čerstvých či trvanlivých potravin, ale i nápojů v lahvích, plechovkách či do kelímků, drobné spotřební elektroniky a kancelářských potřeb patří v dnešní uspěchané době mezi velmi oblíbené a rozšířené. Popularitu se často těší i netradiční automaty, jako jsou robotičtí barmani nebo automaty na míchání koktejlů. Těmito netradičními koncepty společnosti často ukazují své vyspělé a moderní technologické možnosti či úplně naopak jak se tyto projekty dají udělat pomocí běžně dostupných materiálů a součástek, kdy zároveň poskytnou dokumentaci, tedy tzv. open-source.

Cílem této práce je implementace konfigurovatelného nápojového automatu, který bude sestaven pomocí běžně dostupných materiálů. Důraz je ovšem veden na softwarovou stránku práce, kdy řídicí program je navržen jako obecný stavový automat, který při svém startu zpracuje konfigurační soubor a dle něho upraví svůj chod. Práce také poskytne náhled na návrh takového projektu. Motivace tohoto projektu vychází také z mé maturitní práce na velmi podobné téma. První část této práce je spíše teoretická, zabývající se rešerší existujících řešení pro výdejové a nápojové automaty, návrhem nápojového automatu a argumentací výběru použitých komponent, zatímco druhá bude přibližovat jeho realizaci funkcionalitu a testování.

Kapitola 1

Koncept výdejních automatů

V této kapitole rozeberu koncept výdejních automatů, jejich dělení a jak tyto automaty fungují. Výdejní automat je automatizovaný stroj na výdej různého zboží, jako jsou potraviny, nápoje, tabákové výrobky apod. Výdej probíhá zpravidla po zaplacení v lokální měně, to však není podmínkou. Dnes jsou přijímány různé druhy plateb, mezi něž patří mince, bankovky či testovací žetony, nebo již stále častěji bezhotovostní platby. Tyto automaty se dají dělit dle různých kritérií, možným kritériem dělení je typ prodávaného zboží, a to na

- nápojové automaty,
- potravinové automaty,
- automaty na elektroniku, kancelářské potřeby nebo hračky,
- automaty na jízdenky a vstupenky.

V závislosti na typu prodejního automatu se také liší jeho součásti. Do nápojových automatů v komerční a nejčastěji rozšířené formě se dají zařadit řešení pro výdej tzv. post-mix nápojů, které nejčastěji najdeme ve fast-food restauracích. Toto řešení míchá nápoj během výdeje. Po volbě příchuti čerpá sirup současně se sycenou či nesycenou vodou přímo do kelímku [2]. Další možností výdeje, kdy je nápoj míchán těsně před výdejem jsou automaty na zpravidla horké nápoje, jako je káva nebo čaj. Tento typ automatu obvykle připravuje kávu instantní. Příprava instantní kávy spočívá smícháním sypké směsi s horkou vodou ve varné komoře, načež dojde k výdeji nápoje. Během tohoto procesu můžou být přidány i další ingredience jako je mléko nebo cukr [3].

Další možností výdeje je automat konstruovaný pro výdej potravin, ale i balených nápojů. Nejčastějším řešením tohoto typu výdeje je výdej ze zásuvek se spirálou. Každá zásuvka má několik pozic pro skladování předmětů pro výdej, kdy je každá pozice vybavena mechanismem umožňujícím uvolnění předmětu. Po uvolnění předmětu probíhá výdej zboží jedním ze dvou způsobů. První možností je pád do výdejního prostoru, druhým pak přesun zboží do výdejního prostoru pomocným mechanismem připomínajícím výtah. Konstrukce pro výdej potravin, zejména hotových jídel může ale také být na

principu matice zásuvek, kdy pro otevření zpřístupněna jen jedna zásuvka. Automaty pro výdej potravin a nápojů jsou ještě doplněny o chladicí mechanismus. Konstrukce pro výdej elektroniky, kancelářských potřeb nebo hraček jsou o chladicí mechanismus chudší, avšak jejich konstrukční řešení je velmi podobné [4].

Konstrukční řešení vybraných typů přiblížím v následujících podkapitolách.

1.1 Nápojové automaty

Post-mix výdej

Automaty pro takzvaný post-mix výdej jsou vhodným řešením pro velké objemy výdejů stejných nápojů pro například již zmíněné restaurace nebo i bary. Mezi hlavní součásti tohoto systému patří zdroj vody, zdroj CO_2 ¹, systém na sycení nápoje, zdroj sirupu, chlazení a výdejový mechanismus. Tyto automaty mohou také být rozšířeny o výdejník ledu. Příklad takového automatu je na obrázku 1.1.

Zdroj vody je přímo z vodovodního řádu. Pro správný chod automatu jsou také stanoveny požadavky pro průtok a tlak vody. Tlak je následně regulován, aby došlo ke správnému nasycení nápoje, ten totiž nesmí převýšit tlak oxidu uhličitého. Zároveň nesmí být moc i nízký, je tedy možné použít i čerpadlo pro jeho zvýšení. Voda také podléhá přísným požadavkům, musí být pitná a nesmí obsahovat moc látek jako je železo nebo chlór, to může ovlivnit její chuť. To je řešeno filtry na vodu, ty pak musejí být periodicky měněny.

Oxid uhličitý má především dvě úlohy, tlakem pohání sirup do míchacího mechanismu a nasycuje vodu. Ten je uložen v natlakovaných nádobách o tlaku 800 psi. CO_2 při stlačení kapalný, tento tlak zajišťuje zachování jeho kapalného stavu. Plná nádoba obsahuje typicky 65% CO_2 v kapalně formě a 34% v plynné. Při snížení tlaku se oxid začne přeměňovat zpět do plynné formy, dokud se tlak neobnoví. Jelikož je tlak v nádobě přibližně dvacet sedmkrát vyšší, než například v pneumatikách² je třeba dodržet bezpečnostní prvky i zde. Nádoby musí být uloženy ve svislé poloze, okolí nesmí překročit předepsanou teplotu a jsou vybaveny regulátory tlaku a zařízením zvaným "burst disk." Ten při zvýšení tlaku v nádobě, například zvýšením okolní teploty, nad bezpečnou hodnotu postupně a bezpečně uvolní obsah nádoby.

Sirupy jsou uchovány v BIB (Bag In Box)³ zásobnících nebo v nerezových



Obrázek 1.1: Nápojový automat typu post-mix [5]

¹<https://www.britannica.com/science/carbon-dioxide>

²<https://www.portalridice.cz/clanek/tabulka-husteni-pneumatik>

³<https://www.diffpack.com/bag-box-bib/>

tancích. Jak již bylo naznačeno, sirup se dostává do výdejového mechanismu pomocí oxidu uhličitého, ten je v případě nerezových tanků přiveden přímo do tanku, podobně jako u piva⁴, odkud jej tlačí do výdejového mechanismu. V případě BIB zásobníků jsou z tohoto zásobníku připojeny na čerpadlo, které je poháněno CO₂, čerpadlo sirup přečerpá do komory, odkud je již oxidem uhličitým tlačěn do výdejového mechanismu.

V závislosti na zvoleném nápoji je pak voda před výdejem sycena oxidem uhličitým. Toto se děje v uzavřené nádobě, kam je načerpána voda, která se následně míchá s CO₂ a po jejím nasycení je čerpána dál, směrem k výdeji. Před výdejem je voda ještě ochlazená, to se děje jedním ze dvou způsobů. Prvním je ledová koupel, trubky s vodou vedou přes zásobník s kostkami ledu, ty jsou ledem ochlazeny, postupně se tak rozpouštějí a jsou vyráběny nové, voda na konci okruhu vytéká chladná. Druhým je mechanické chlazení, to spočívá v ponoření vodního okruhu, tedy trubek, do vody, která je chlazená chladicími cívkami.

Výdejový mechanismus je typicky spouštěn uživatelem, který si chce načerpat nápoj, ten sepne spínač, obvykle tlačítkem nebo páčkou, který aktivuje výdej nápoje, nyní se dohromady míchá z jedné trysky sycená či nesycená voda a sirup, odtud je tedy název post-mix. Další informace o automatech typu post-mix lze nalézt v tomto zdroji [6].

■ Automaty na horké nápoje

Nejčastějším zástupcem tohoto typu automatů jsou kávovary, tyto stroje ale často umí mimo instantní kávy připravit i čaj, čokoládu nebo instantní vývar [7]. Proces přípravy a výdeje kávy začíná volbou nápoje na vstupním rozhraní, to může být od matice tlačítek, kde má každé tlačítko přiřazené svůj nápoj, přes kombinaci tlačítek a displeje po dotykový displej.

Po zvolení nápoje a jeho zaplacení započne stroj proces výroby zvoleného nápoje. Nejdříve začne nahřívání vody, typicky pomocí varného tělesa nebo bojleru. Voda je obvykle čerpána přes filtr přímo z vodovodního řádu. Mechanismus na ohřev vody bývá jeden, může jich mít automat více v závislosti na umístění a poptávce.

Ohřátá voda je pak ve varné komoře smíchána s instantní směsí kávy, čaje, cukru nebo i mléka, dle předchozí volby nápoje. Tyto suroviny jsou uchovány v zásobnících vně automatu. Některé automaty nabízejí možnost přípravy nápoje do vlastního šálku, v tomto případě započne po smíchání surovin výdej, pokud tuto možnost konkrétní stroj nenabízí nebo uživatel svůj šálek nemá, automat mu jej vydá, a poté započne výdej [3][8].

■ 1.2 Potravinové automaty

Do tohoto typu automatů patří především dva zástupci. Prvním z nich je typ, který můžeme označit jako zásuvkový. Ten můžeme rozdělit dle vnitřního

⁴<https://www.pivoteka.cz/clanek/tlakovani-piva-tlacnym-plynem-co2-40>

uspořádání, jednou možností je, že se uvnitř nachází válcový mechanismus, který má několik samostatných kol, které se mohou otáčet, každé kolo má pak několik zásuvek na potraviny. Druhou možností vnitřního uspořádání je pevné rozmístění zásuvek, velmi podobné dnes populárním výdejním boxům⁵. Obě možnosti jsou principem velmi podobné, zákazník si vždy zvolí, ze které zásuvky chce své jídlo vydat a automatu mu po zaplacení konkrétní zásuvku zpřístupní.

Druhým zástupcem je mnohem rozšířenější typ, který můžeme označit jako spirálový. Ten má zboží uloženo také v zásuvkách, kde každá zásuvka má několik pozic, které mohou být i spojené do dvojic a každá pozice má vlastní výdejovou spirálu. Každá pozice či dvojice pozic, dle velikosti prodávajícího zboží má pak své kódové označení. Vstupní rozhraní tohoto automatu, i automatu zásuvkového je velmi podobné tomu na horké nápoje. Nejčastěji najdeme displej v kombinaci s tlačítky, kam uživatel zadá kód produktu, který chce, aby mu byl vydán a opět po zaplacení dojde k výdeji.



Obrázek 1.2: Výdejový automat [9]

Mimo řídicí jednotku je tento automat vybaven i další řadou ovládacích prvků svých součástí. Ten nejdůležitější je systém pro výdej produktu. Každá spirála má svůj motorek který ji otáčí, po příkazu na výdej od řídicí jednotky motorek otočí spirálou, načež dojde v uvolnění produktu do výdejního prostoru. Výdej je kontrolován senzory, pokud k němu nedojde, provede se pokus o výdej znovu, při dalším neúspěšném pokusu jsou vráceny peníze. Výdej produktu je buď pouze otočením spirály a volným pádem do prostoru pro výdej nebo se využívá dvouosý mechanismus podobný výtahu, který jej u své pozice nabere a do výdejního prostoru jej dopraví. Ten se využívá pro předměty náchylné na poškození pádem. Další důležitou součástí těchto automatů je chlazení, jelikož potraviny podléhají hygienickým normám určujícím jejich skladování před prodejem⁶ [4][10].

⁵<https://distribox.cz/>

⁶<https://eur-lex.europa.eu/legal-content/CS/LSU/?uri=CELEX:32004R0852>

1.3 Automaty na spotřební zboží

Automaty na elektroniku, kancelářské potřeby a hračky, a automaty na jízdenky a vstupenky jsem sloučil do jedné podkapitoly. První zmíněné jsou, jak již bylo nastíněno, svou konstrukcí velmi podobné spirálovým automatům na potraviny, jen často nevyžadují chlazení, nebudu se jim tedy podrobně věnovat, ale lehce přiblížím mechanické řešení automatů na jízdenky.

To je především díky svému typickému umístění odolné proti vnějším vlivům, uvnitř je ale mechanismus oproti již zmíněným řešením jednodušší. Mimo uživatelské rozhraní, kde si uživatel vybere, jakou jízdenku či vstupenku mají uvnitř tepelnou tiskárnu s ořezávačem a zásobník na papír. Ta pak dle pokynů řídicí jednotky vytiskne požadovaná data a po oříznutí uvolní do výdejního prostoru [11].

1.4 Netradiční řešení nápojových automatů

Nápojové automaty začínají ale nabývat i nových forem. Moderní restaurace a bary se snaží nalákat své zákazníky na lukrativní přípravy nápojů. K těmto netradičním způsobům přípravy nápojů patří robotičtí barmani nebo dnes i AI barmani [12]. Robotičtí barmani jsou realizováni s použitím průmyslových robotů nebo na principu, kdy je sklenice automaticky přesouvána pod jednotlivé ingredience, které jsou do ni pak vydány.

Méně tradičnímu řešení, připomínajícímu automaty post-mix, jeho návrhu a realizaci se budu věnovat v následující části této práce.

Kapitola 2

Návrh nápojového automatu

Hlavním úkolem je návrh konfigurovatelného nápojového automatu, tedy mimo vývoje řídicího programu, který bude důležitou součástí, je také volba součástek a materiálů, které budou na výrobu a testování zařízení použity. Je tedy potřeba provést rešerši vhodných možností pro řešení jednotlivých dílčích částí, a na jejím základě pak tyto dílčí části navrhnout. Jedná se tedy zejména o způsob přečerpání tekutin, způsob kontroly jejich přečerpání, uživatelské rozhraní a vhodnou embedded platformu.

2.1 Čerpadla

V této podkapitole se budu věnovat převážně čerpadlům používaným v potravinářském a nápojovém průmyslu, zejména i pro moji potřebu přepravovat tekutiny tak, aby byla zachována zdravotní nezávadnost. Čerpadla jsou využívána v různých provedeních a typech dle jejich účelu. Nejčastější výskyt v průmyslu mají dnes čerpadla odstředivá, objemová, membránová, zubová, dále také pístová, progresivní dutinová, bubnová, a v neposlední řadě také peristaltická. Informace pro tuto kapitolu jsem převzal ze zdroje [13].

Odstředivá čerpadla patří díky svým vlastnostem mezi nejčastější, jsou jednoduchá a účinná. Jejich základním principem je využití odstředivé síly, disk s lopatkami se uvnitř čerpadla otáčí a tím předává tekutině kinetickou energii. Tato radiální síla směrem ven pak urychluje kapalinu do difuzoru nebo jiného výstupu a přeměňuje se na sílu tlakovou.



Obrázek 2.1: Membránové čerpadlo [14]

Membránová čerpadla používají pro pohyb kapalin pohyblivou membránu, ta bývá vyrobena z pružných materiálů jako je pryž. Membrána je pak rozkmitána pohonným mechanismem, tato akce vytváří vakuum a to pak při expanzním pohybu nasává kapalinu do komory čerpadla a při kontrakčním pohybu ji pak z komory vypouští ven. Tento typ čerpadla dokáže pracovat s velkou škálou kapalin jako jsou korozivní nebo obsahující

pevné částice. Jsou vhodná také na zpracování chemikálií nebo tekutin s vysokou viskozitou.

Objemová čerpadla zachycují a přečerpávají pevné množství kapaliny ze sacího do výtlačného hrdla. Vzhledem k tomuto principu jsou velmi přesná a s konstantním průtokem. Tyto vlastnosti zajišťuje také jejich konstrukce díky použitým mechanismům jako jsou písty, membrány nebo laloky. Čerpadlo vytváří pro přepravu tekutiny podtlak. Klíčovou vlastností tohoto druhu je vysoký tlak a přenos kapalin s vysokou viskozitou.

Zubová nebo také převodová čerpadla čerpají jsou používána pro svoji efektivitu při čerpání při relativně vysokém tlaku. Používají obvykle dvě ozubená kola, a to hnací a hnané. Hnací kolo oběma koly otáčí a tím je kapalina nasávána do prostoru mezi zuby kol na vstupní straně a poté vytlačována na výstupní straně. Tento mechanismus zajišťuje konzistentní průtok. Jsou vhodná pro husté tekutiny jako jsou oleje, barvy nebo sirupy. Nevhodné jsou naopak pro abrazivní tekutiny způsobující opotřebení a poškození částí mechanismu. Jsou snadno a levně udržovatelná s kompaktními rozměry.

Šroubová čerpadla kombinují prvky objemových a zubových. Za použití jednoho či více šroubů ve válcové dutině vedou kapalinu po jejich délce, dosahují tímto mechanismem velmi stálého a konzistentního průtoku. Jejich použití je poměrně univerzální, mohou čerpat tekutiny s různou viskozitou nebo také látky ve smíšeném skupenství, jako je směs plynu a kapaliny.

Pístová čerpadla operují pomocí jednoho či více pístů. Základním principem je cyklus zdvihu pístu, nejdříve je proces nazvaný sací zdvih, kdy píst vytváří podtlak, čímž otevírá vstupní ventil a nasává kapalinu, následuje proces zvaný výtlačný zdvih, kdy se pohybuje opačným směrem, uzavírá vstupní ventil, otevírá výstupní a vytlačuje kapalinu. Důležitou vlastností tohoto typu čerpadla je vytvoření vysokého a přesného tlaku na výstupu.

Progresivní dutinová svým vzhledem i principem připomínají čerpadla šroubová. Konstrukce umožňuje jemné čerpání, což je vyžadováno při přesném dávkování nebo při manipulaci s kapalinami citlivými na smyk. Základem mechanismu je šroubovitý rotor, který uvnitř dvojitého šroubovitého statoru vytváří dutiny svým excentrickým pohybem.

Bubnová čerpadla jsou určena k přečerpání kapalin z nádob jako jsou sudy nebo barely. Vkládají se do otvoru nádoby, a pomocí dlouhé trubice dosahující až na dno není třeba nádobou pro přenos obsahu nijak hýbat. Tento typ pracuje velmi podobně jako čerpadla odstředivá, jen bývají osazena mimo elektrický pohon také pohonem ručním.



Obrázek 2.2: Peristaltické čerpadlo [15]

Peristaltická čerpadla jsou svým principem od výše zmíněných vzdálena. Přenos média funguje na základě stlačování a uvolňování pružné hadičky umístěné uvnitř kruhové hlavy čerpadla. Hadička je stlačo-

vána pohybem válečků umístěných na rotoru pohonného motoru, tyto válečky hadičku stlačují právě proti kruhové hlavě. Tímto pohybem se v hadičce vytváří podtlak, který pak nasává tekutinu, ta je vytlačována vpřed, čímž vzniká peristaltický pohyb.

Jednou z hlavních výhod je absence dotyku tekutiny s žádnou pohyblivou částí čerpadla a relativně nízká cena. Tento způsob čerpání je také velmi přesný s minimální potřebou údržby, zpravidla stačí vyměnit pouze flexibilní hadici. Zejména pro jednoduchou údržbu a absenci dotyku kapaliny s pohyblivými částmi. Peristaltické čerpadlo je vidět na obrázku 2.2

2.2 Řízení čerpadel

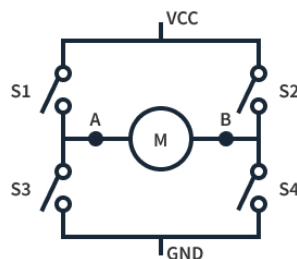
V závislosti na využití a výkonu čerpadel, se používají různé druhy pohonů. Nejčastějším způsobem je využití střídavého nebo stejnosměrného motoru s pomocným mechanismem jako je řemenový pohon nebo převodovka. Průmyslové aplikace používají výkonné střídavé motory, které rozdělujeme na synchronní a asynchronní. Tyto elektromotory jsou tvořeny dvěma hlavními částmi, těmi jsou rotor a stator. Synchronní motor pracuje na střídavý proud, jeho rotor se otáčí synchronně s točivým polem statoru. Na cívkách statoru je přiváděn střídavý proud, který pak roztočí stator tvořený buď cívkami se statickým elektromagnetickým polem nebo permanentními magnety. Asynchronní motor, známý také jako indukční je založen na principu elektrické indukce, kdy po přivedení proudu na stator se vytváří točivé elektromagnetické pole, které indukuje proud na vinutí rotoru a tím si rotor vytváří vlastní magnetické pole, které interaguje s polem statoru [16].

Časté komerční použití vhodnější pro pohon peristaltických čerpadel ale nabízejí motory stejnosměrné, ty svými rozměry bývají kompaktnější a pro moji aplikaci výkonnostně dostačující. Častými typy pro komerční použití jsou motory krokové, servo, kartáčové a bezkartáčové. Krokové motory jsou vhodné pro aplikace vyžadující vysoký kroutící moment při pomalém pohybu. Pro svůj pohyb potřebuje řídicí elektroniku, která mu pomocí postupného napájení pólových dvojic určí přesnou polohu [17]. Servo motor umožňuje přesnou kontrolu úhlu, rychlosti a točivého momentu, pro kontrolu svého chodu využívá zpětné vazby v podobě enkodéru, což mu umožňuje vysokou preciznost [18]. Kartáčový motor má komutátor a kartáče které přepínají vinutí rotoru. To umožňuje změnu proudu a jejich pomocí vytváří magnetické pole. Otáčky jsou závislé na napětí a zátěži. Bezkartáčový motor (BLDC) nemá ani komutátor ani kartáče ale využívá místo nich střídač k vytváření pulzů do cívek statoru, má vyšší životnost, je tišší a vyžaduje méně údržby [19].

V následujících dvou podkapitolách rozeberu možnosti řízení stejnosměrných motorů.

2.2.1 H-můstek

H-můstek nabízí ovládání chodu motoru, a to jeho směr a rychlost. Stejnoseměrné motory se otáčejí po přivedení napětí na svorky motoru, po otočení polarit napájení se změjí směr otáčení motoru. Obvod, který H-můstky používají je zjednodušeně ilustrován na 2.3 Spínače označené S1-S4 reprezentují tranzistory, které mají zpravidla



Obrázek 2.3: Princip fungování H-můstku [20]

také svůj řídicí obvod. Pro roztočení motoru se sepnou spínače vždy diagonálně, tedy S1 a S4 obsluhují jeden směr otáčení a S2 a S3 směr druhý. Pomocí pulzní šířkové modulace (PWM) lze řídit i jeho rychlost. Tato modulace spočívá v cyklickém pulzování signálu, který ovládá motor, pro plnou rychlost se nastaví PWM na 100%, zatímco na poloviční na 50%. Nevýhodou tohoto typu ovládání chodu motoru je omezení protékajícího proudu přes řídicí tranzistory [20].

2.2.2 Relé

Dalším způsobem řízení stejnosměrných motorů je relé. Relé je elektrotechnická součástka, která obsahuje elektromagneticky ovládané kontakty a funguje jako spínač. Uvnitř obsahuje cívku, fungující jako elektromagnet, magnetický obvod a kotvu. Proud přivedený do cívky vyvolá magnetické pole, které přitáhne kotvu, a ta spínač sepne, či rozezne [21]. Hlavní výhodou od H-můstek je maximální spínaný proud a napětí. Jelikož pro svoji aplikaci nebudu potřebovat měnit rychlost motoru, je pro mě relé lepší volbou, využívat budu osmikanálový relé modul s optočlenem.

2.3 Kontrola průtoku

Na měření průtoku kapalin jsou používány různé typy senzorů či zařízení. Tato zařízení fungují na různých principech, jako je ultrazvukový průtokoměr, indukční nebo lopatkový. Ultrazvukové průtokoměry využívají Dopplerova principu¹. Přístroj vysílá do potrubí ultrazvukové vlnění s konstantním kmitočtem a přijímá vlnu odraženou od částic či bublin v tekoucí kapalině, rychlost je pak úměrná rozdílu frekvencí.

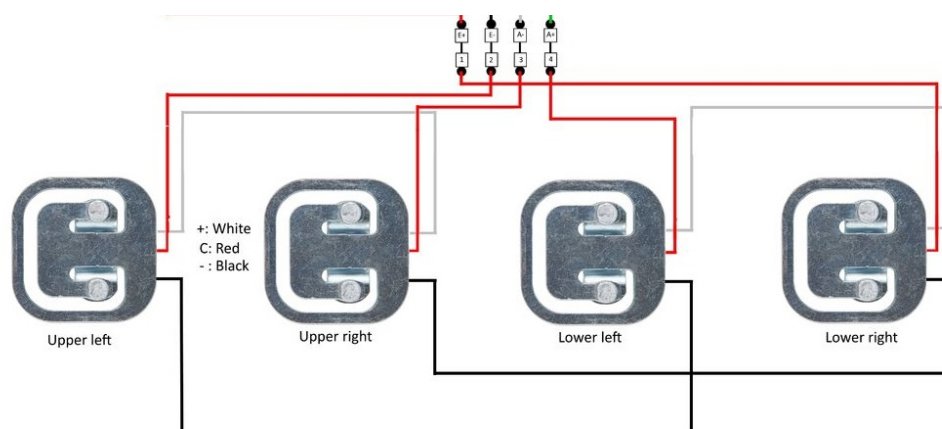
Indukční průtokoměry využívají Faradayova zákona elektromagnetické indukce. Pohybem tekutiny (v definici pohybem vodiče) se v homogenním magnetickém poli indukuje elektrické napětí. To je pak přímo úměrné indukci

¹<https://edu.techmania.cz/cs/encyklopedie/fyzika/akustika/doppleruv-jev>

magnetického pole, délce potrubí, na které měříme, a rychlosti proudění. Společně s ultrazvukovým jsou oba tyto průtokoměry bezdotykové.

Lopátkové průtokoměry se umísťují přímo do cesty protékající kapaliny, ta roztočí volně otočný rotor s lopatkami, který se vlivem proudu otáčí. Otáčky jsou úměrné rychlosti kapaliny a snímají se Hallovým senzorem. Tento senzor je z výše uvedených finančně nejdostupnější i pro mé použití nevhodnější. Jeho největší nevýhodou je náchylnost na usazování nečistot v pohyblivých částech, což vzhledem k použití nápojů s vyšším obsahem cukru je zároveň nevýhodou podstatnou, jelikož by časem byla ovlivněna přesnost měření [22].

Pro použití nápojového automatu je možné také kontrolovat již přečerpanou kapalinu, jejíž tok lze díky vlastnostem peristaltických čerpadel poměrně rychle zastavit a tím lze docílit relativně přesné kontroly přečerpání tekutin. Tato kontrola lze provádět za použití měření váhy přečerpané kapaliny. Tato možnost elegantně řeší podstatnou nevýhodu lopátkových průtokoměrů a je cenově dostupnější než průtokoměry ultrazvukové nebo indukční.



Obrázek 2.4: Zapojení senzorů váhy [23]

Pro měření načerpané tekutiny budu tedy používat kombinaci 4 senzorů váhy v zapojení znázorněném na obrázku 2.4. Tyto senzory jsou připojené do AD převodníku², který pak bude připojen do řídicí jednotky automatu. Váhové senzory fungují na principu tenzometrického můstku, váha se vyhodnocuje na základě výstupního napětí U_{out} obvodu, to se mění dle odporů jednotlivých senzorů. Změnu odporu každého ze senzorů způsobí změna váhy, která na ně rozprostřeně působí. R_1 až R_4 znázorňují odpory jednotlivých senzorů, U_{in} pak vstupní napětí.

$$U_{out} = \left(\frac{R_3}{R_3 + R_4} + \frac{R_2}{R_1 + R_2} \right) \cdot U_{in} \quad (2.1)$$

Výstupní napětí je vyhodnoceno v AD převodníku, ten pak dává digitální hodnotu řídicí jednotce, ta je třeba násobit kalibračním koeficientem, to rozeberu podrobněji v Kapitole 3.

²<https://physics.mff.cuni.cz/kfpp/skripta/elektronika/kap8/pevodnky.html>

2.4 Uživatelské rozhraní

Výdejové automaty mají obvykle jako vstupní periférii tlačítka, displej či jejich kombinaci. Varianta pouze tlačítek není pro tento projekt příliš vhodná, jelikož bude automat konfigurovatelný, různé konfigurace by znamenaly nutnost změny například popisků u tlačítek. Vhodnou variantou se nabízí dotykový displej, jehož výhodou je, že při načtení různých konfigurací nemusejí být provedeny mechanické změny, ale stačí změny softwarové, avšak nevýhodou je jeho cena a náročnost na výpočetní výkon. Cenově přívětivější je komerčně nejvyužívanější kombinace tlačítek a displeje.

Pro automat volím LCD displej se čtyřmi řádky po dvaceti zobrazovacích pozicích, je cenově dostupný s dostatečnou zobrazovací kapacitou. Bude doplněn o 4 tlačítka, což bude pro jednoduchou volbu zcela dostačující. Displej je vybaven modulem pro komunikaci po sběrnici I²C³. Tlačítka pak budou klasické spínače.

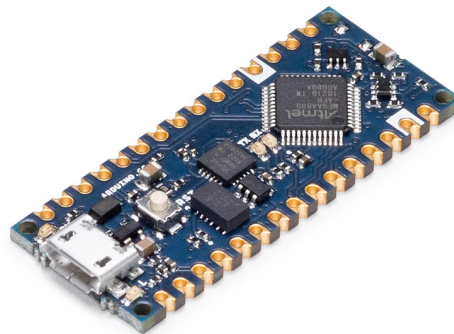
2.5 Volba embedded platformy

Pro řídicí jednotku automatu je třeba vhodně zvolit embedded platformu. Mezi populární volbu patří Arduino, Raspberry Pi, ESP nebo STM. Mezi důležité parametry pro výběr patří výpočetní výkon, paměť, vstupně-výstupní piny, podpora programovacích jazyků, ale také i komunita věnující se dané desce, dostupné knihovny nebo dostupná podpora. Tyto desky nabízí výkony od 16 MHz až po 480 Mhz, počty pinů od 17 po 76 pinů, ale i vysoké rozsahy paměti Flash a SRAM [24]. Jako řídicí jednotku jsem vybral Arduino Nano Every[25], je to velmi dostupná a rozšířená deska, která má vysoký počet kompatibilních modulů, knihoven a širokou komunitu. Oproti její předchozí verzi Arduino Nano má rychlejší procesor, větší flash paměť a především větší SRAM, což umožňuje čtení větších souborů. Její důležité parametry uvádím v následujícím seznamu.

- Mikrokontrolér ATmega4809
- Operující napětí 5V
- Flash paměť 48 KB
- SRAM 6 KB
- Taktovací frekvence 20 MHz
- 22 I/O pinů (14 digitálních a 8 analogových)
- 20 mA proud na I/O pin (50 mA pro 3,3 V pin)
- Vstupní napětí 7-21 V
- I²C, SPI, UART komunikace

³<https://learn.sparkfun.com/tutorials/i2c/all>,

Tato deska stojí 12,50 € (přibližně 320 Kč)[25], verze s napájenými headery 13,50 € (přibližně 340 Kč) [26]. Pro její programování lze použít volně dostupné Arduino IDE [27], přímo od společnosti Arduino nebo například Visual Studio Code s rozšířením Platformio [28]. Obě tyto vývojové prostředí poskytují pomoc při vývoji od importu knihoven po kompilaci, nahrání kódu a využití seriového monitoru.



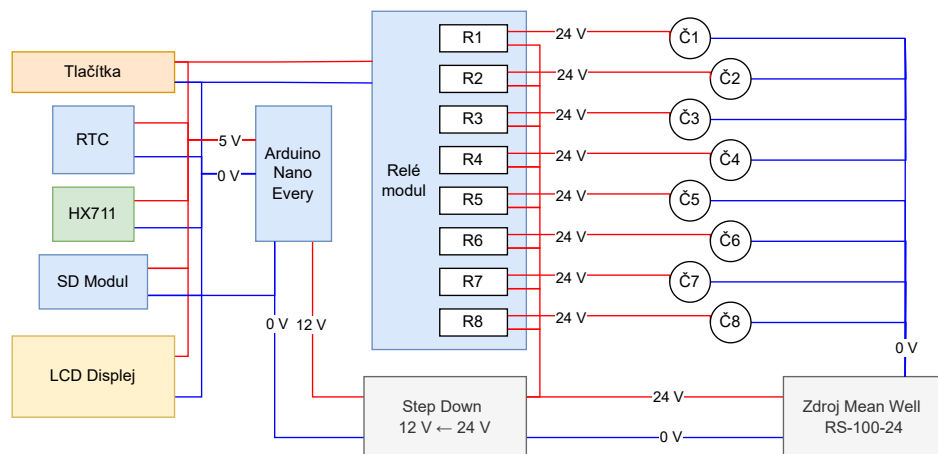
Obrázek 2.5: Arduino Nano Every [25]

2.6 Konfigurace

Řídicí software bude koncipován tak, aby byl do určité míry univerzální a především konfigurovatelný. Pro to je potřeba najít způsob, kterým se automatu bude tato konfigurace dát předat bez nutnosti přehrávání řídicího kódu. Pro tuto možnost je možné použít různé způsoby, pro moji aplikaci volím možnost předání informací na paměťové kartě. Je tedy potřeba vybrat způsob čtení paměťových karet přímo Arduinem, na to jsem vybral modul pro čtení SD karet s komunikací po sběrnici SPI. Tato možnost nevyžaduje pro nahrání nové konfigurace přítomnost žádného dalšího zařízení mimo modulu a samotného Arduina. Zároveň zajišťuje volnou volbu formátu konfigurace. Tato akce není ani nijak výpočetně náročná, jelikož se bude konfigurace načítat pouze při startu stroje.

2.7 Napájení komponent

Jelikož mám k dispozici zdroj Mean Well RS-100-24 [29], použiji ho pro napájení. Zdroj má výkon 100 W, což je při maximálním odběru čerpadel 800 mA zcela dostačující i pro běh několika najednou. Zdroj má výstupní napětí 24 V, tím budou čerpadla napájena, dále je použit pro napájení Arduina step-down měnič napětí XL4016 pro změnu napětí na 12V. Arduino pak disponuje operujícím napětím 5 V, které napájí ostatní připojené moduly.

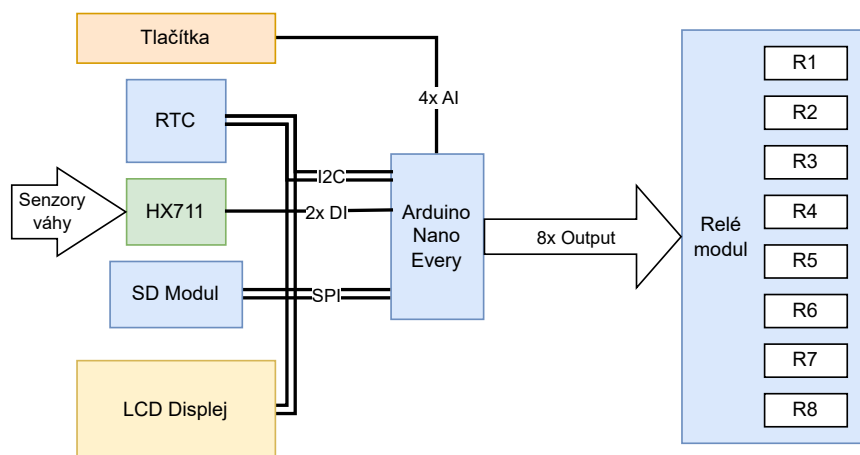


Obrázek 2.6: Blokové schéma zapojení napájení

2.8 Zapojení modulů

Pro zapojení modulů je vhodné navrhnout a vyrobit jednoduchou desku plošných spojů. Tato možnost zajistí znovupoužitelnost či jednoduchou výměnu použitých modulů. Zajistí také modularitu při budoucích úpravách stroje, kdy při změně některé z částí stroje a nutnosti výměny jednoho nebo více dílů bude nutné pouze upravit zapojení. Pro tlačítka je, vzhledem k jejich budoucímu umístění, také vhodné navrhnout desku plošných spojů a navrhnout způsob jejich připojení na hlavní desku, odkud budou signály vedeny přímo do Arduina.

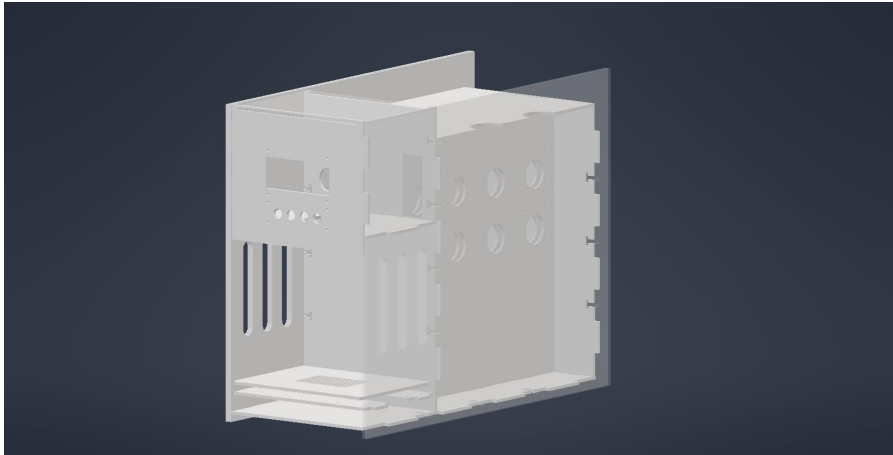
Pro návrh obou desek plošných spojů jsem využil open source program KiCad. Tento program je široce uživatelsky oblíbený, konkuruje komerčním programům jako je například Autodesk EAGLE a je navíc zdarma.



Obrázek 2.7: Blokové schéma zapojení modulů

2.9 Návrh 3D modelu

Pro návrh digitálního modelu těla zařízení jsem použil program Autodesk Inventor, který je přístupný zdarma pod studentskou licenci [30]. Je to také program využívaný v praxi, s širokou komunitou uživatelů a komplexními možnostmi, z nichž jsem využil například detektor kolizí k ověření správnosti navržené jednotlivých dílů. Navrhl jsem ho tak, aby sestavení a údržba byly co nejjednodušší a nevyžadovaly úplné rozebrání stroje.

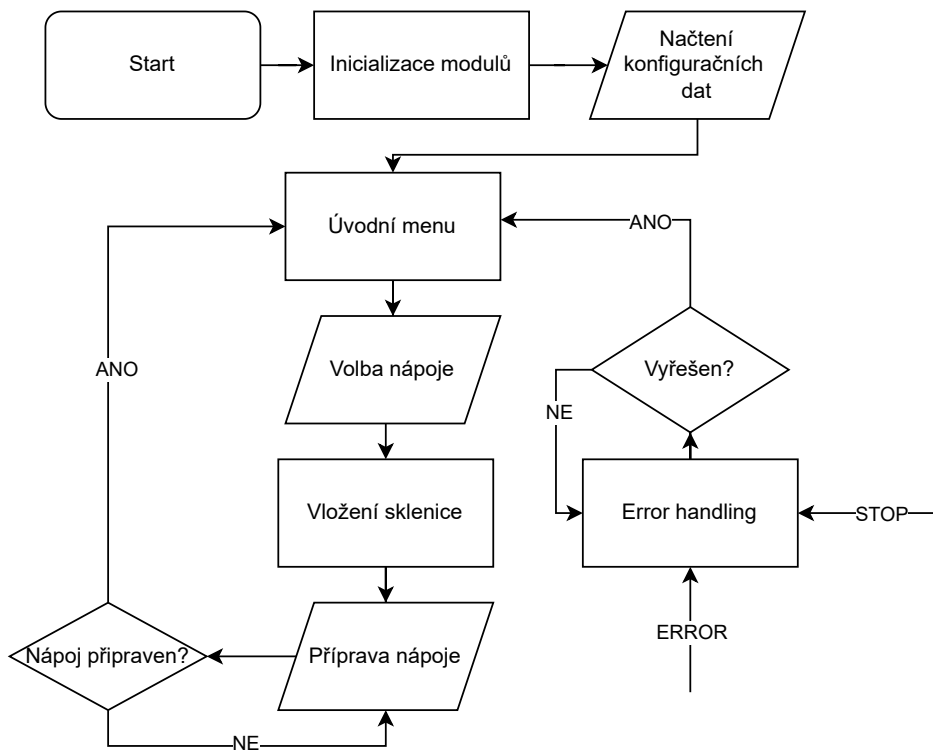


Obrázek 2.8: 3D model v programu Autodesk Inventor

Model je navržen tak, že po sestavení jsou boční díly uchyceny pomocí šroubů M5x20 a uchyceny čtyřhrannou maticí M5. Jednotlivé díly jsou pak uspořádány tak, že při odmontování jedné z bočnic například pro údržbu se zbytek stroje nerozloží. Na středovém panelu jsou uchycena čerpadla pomocí šroubů M3x20 a matic M3. Víko pak má prostor pro hadičky na přívod tekutin pro nápoje. Na druhé straně panelu bude umístěna elektronika, za displejem je pak vyústění hadiček.

2.10 Návrh software

Software je důležitou součástí tohoto projektu. Je koncipován tak, aby bylo jej bylo možné konfigurovat. Při startu programu načte z paměťové karty konfigurační soubor ve formátu json. Ten obsahuje počet čerpadel, nápoj, který každé čerpadlo má a možné kombinace koktejlů společně s jejich názvy nebo informací o poměrech míchání jednotlivých koktejlů. Tato informace je pak zpracována a na jejím základě je upraven chod automatu, zejména nabízené nápoje a koktejly. Automat bude řízen obecným stavovým automatem, který obsahuje obecné stavy. To zahrnuje stavy pro zobrazení menu, nabídku nápojů, volbu, respektive vložení sklenice, přípravu nápojů, zastavení, error handling nebo cyklus pro čištění. Vývojový diagram, ze kterého pak samotný vývoj bude vycházet je na obrázku 2.9.



Obrázek 2.9: Vývojový diagram pro vývoj řídicího programu

Kapitola 3

Realizace

Tato kapitola je zaměřena na realizaci jak fyzické, tak programové stránky projektu. V první podkapitole se zaměřím se na použité zdroje, materiály a výrobu jednotlivých komponent. Ve druhé pak na vývoj a podrobné fungování řídicího programu použitých modulů a limitací při jeho tvorbě.

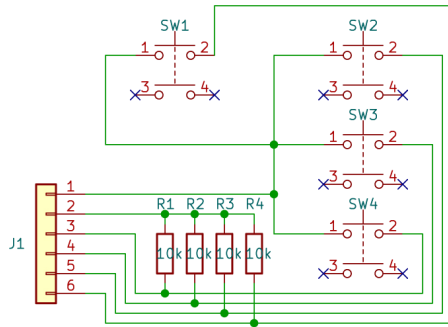
3.1 Hardware

Fyzická stránka automatu se skládá z několika dílčích částí, schránka stroje, umístění senzorů váhy a prostoru pro výdej nápoje, vyústění tekutin, středový panel pro umístění čerpadel a panelu pro osazení desky plošných spojů a modulů. Hlavním použitým materiálem je plexisklo, z něhož jsem vyrobil schránku stroje společně se středovým panelem, většinu dílů prostoru pro výdej nápoje i panel pro DSP a moduly. Díly jsem po zhotovení 3D modelu pro ověření správnosti navržených rozměrů vyřezal na laseru. Schránku automatu drží pohromadě především již zmíněné šrouby M5x20 v kombinaci se čtyřhrannou matkou. Čerpadla jsou do středového panelu uchycena pomocí šroubů M3x20 v kombinaci s matkou M3, avšak použití delších šroubů by bylo pro sestavení přívětivější. Přívod napájení je na zadní straně stroje, to je vyřešeno pomocí panelové euro vidlice s pojistkou v kombinaci s kolébkovým vypínačem, který umožňuje stroj, respektive zdroj vypnout bez nutnosti odpojení kableu. Čerpadla jsou pak připojena na napájení za použití faston konektorů, zemnicí vodič napájení je přiveden na terminál, odkud je spojen se zdrojem. Napětí je pak rozvedeno přes jednotlivá relé ke každému z čerpadel. Do nich jsou pak ze sací strany přivedeny silikonové hadičky od jednotlivých nápojů, výtláčná strana čerpadel je pak přivedena nad prostor pro výdej, kde jsem vymodeloval a na 3D tiskárně vytiskl díl pro vyústění, ten zdrží hadičky na místě. Pomocí 3D tisku jsem pak také vytiskl držáky na senzory váhy.

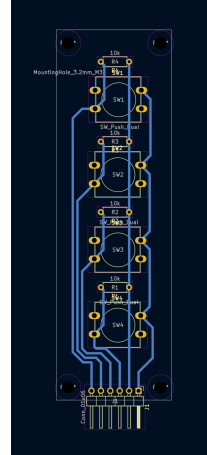
Desky plošných spojů jsem navrhl v programu KiCad a zhotovil na frézce Protomat E44 poskytnutou v IoT Labu katedry radioelektroniky¹. První deska plošných spojů je pro tlačítka, ty jsou připojeny v konfiguraci pull-up, tedy piny Arduina jsou připojeny na napětí 5 V a při stisku se uzemní a napětí klesne na 0 V. Tuto konfiguraci tlačítek jsem zvolil jelikož pro připojení tlačítek

¹<https://iot.fel.cvut.cz/cz/equipment/protomat/>

k Arduino využívám analogové piny a z testovaných možností jsem ho shledal jako nejspolehlivější proti falešnému stisknutí nebo falešnému vyhodnocení stisku několika tlačítek při stisku jednoho.



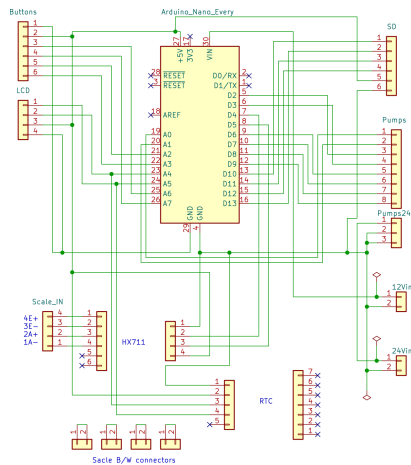
(a) : Schéma DPS



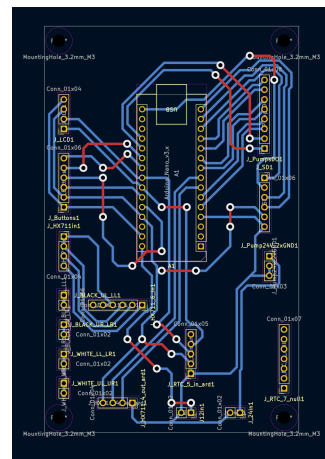
(b) : Rozvržení DPS

Obrázek 3.1: Elektronický návrh desky s tlačítky

Druhá deska plošných spojů slouží jako rozbočovač pro jednotlivé moduly. Přímou na desce je usazeno Arduino Nano Every společně s modulem reálného času a AD převodníkem HX711. na desce jsou pak osazeny šroubovací svorkovnice pro přívod napájení pro Arduino, ze kterého jsou pak napájeny moduly a zároveň také napájení pro osmikanálový relé modul. Tyto svorkovnice jsou použity i pro připojení senzorů váhy a jejich vzájemné propojení dle 2.4. Dále je deska osazena DuPont kolíkovými lištami pro připojení relé modulu, tlačítek, LCD displeje a SD modulu.



(a) : Schéma DPS



(b) : Rozvržení DPS

Obrázek 3.2: Elektronický návrh pro hlavní desku

3.2 Software

Vývoj programů pro Arduino je psaný ve variantně jazyka C++, která je uzpůsobena potřebám této platformy. Je velmi jednoduchý na porozumění a dobře čitelný. Jedním z hlavních rozdílů od C++ je práce s pamětí, Arduina mají výrazně menší paměť, než moderní počítače, a je s ní tak potřeba zacházet. Dalším rozdílem je, že zde není hlavní funkce `main()`, ta je nahrazena dvěma hlavními funkcemi, a to `setup()` a `loop()`. Funkce `setup()` je zavolána pouze při spuštění programu a obsahuje prvotní nastavení mikrokontroléru. Toto nastavení zpravidla obsahuje nastavení baudové rychlosti nebo inicializaci modulů. Funkce `loop()` je pak volána opakovaně po dobu běhu zařízení, ta obsahuje již řídicí program a implementovanou funkcionalitu. Dále lze definovat i uživatelské funkce, ty napomáhají čitelnosti kódu některé použité v této kapitole přiblížím.

Software pro tento projekt je navržen jako obecný stavový automat, který při svém startu načte konfigurační soubor. Na základě dat tohoto souboru jsou pak nabízeny nápoje a koktejly. Vývoj a funkcionalitu řídicího programu v této kapitole rozeberu.

3.2.1 Prvotní nastavení

Abych mohl začít s vývojem, bylo nejdříve potřeba si navrhnout strukturu konfiguračního souboru, tedy co bude konfigurovatelný parametr. To bude totiž jednou hlavních položek při volání funkce `setup()`. Jako formát jsem zvolil *JSON*, je to flexibilní, široce využívaný způsob předávání strukturovaných dat a je snadno rozebiratelný. Na jeho zpracování existuje velká řada knihoven, pro svoji aplikaci jsem využil knihovnu *ArduinoJson.h*. Struktura konfiguračního souboru je vidět na následující ukázce.

```

1 {
2   "pumps": [
3     {"id": 1, "drink": "Coca cola", "alcohol": 0.0},
4     {"id": 2, "drink": "Rum", "alcohol": 40.0},
5     {"id": 3, "drink": "Tonic", "alcohol": 0.0},
6     {"id": 4, "drink": "Gin", "alcohol": 40.0},
7     {"id": 5, "drink": "Vodka", "alcohol": 40.0},
8     {"id": 6, "drink": "Sparkling water", "alcohol": 0.0},
9     {"id": 7, "drink": "Prosecco", "alcohol": 12.0},
10    {"id": 8, "drink": "Aperol", "alcohol": 11.0}
11  ],
12  "combinations": [
13    {"name": "Rum and Coke", "pumps": [1,2], "amount": [200, 50], "ingrd":
14      ↪ 2},
15    {"name": "GT", "pumps": [3,4], "amount": [200, 50], "ingrd": 2},
16    {"name": "Vodka Soda", "pumps": [5,6], "amount": [50, 200], "ingrd": 2}
17    ↪ ,
18    {"name": "Aperol Spritz", "pumps": [6,7,8], "amount": [50, 150, 100],
19    ↪ "ingrd": 3},
20    {"name": "Vodka Tonic", "pumps": [5,3], "amount": [50, 200], "ingrd": 2}
21    ↪ }
22  ]
23 }

```

Ukázka kódu 3.1: Konfigurační soubor `config.json`

Pole `"pumps"` obsahuje tři položky, `"id"` reprezentuje číslo konkrétního čerpadla, `"drink"` pak jaký nápoj je k němu přiřazen a `"alcohol"` obsah alkoholu, dle něho se pak mění nabídka při čerpání samostatných nápojů. Pole `"combinations"` má položky čtyři, `"name"` nese název koktejlu, `"pumps"` nese čísla přiřazených čerpadel, `"amount"` má pak informaci o poměru jednotlivých nápojů, respektive kolik kterého nápoje je v daném koktejlu a `"ingrd"` reprezentuje počet ingrediencí pro daný koktejl. Správnost formátu není v kódu nijak ověřována, z hlediska uživatelské přívětivosti by optimálním řešením bylo vytvoření aplikace, která by umožnila snadné vkládání nastavení a zajistila správný formát.

Konfigurační soubor je uložen na paměťovou kartu, ta je načtena pomocí modulu komunikujícím přes sběrnici SPI. Před načtením a zpracováním konfiguračního souboru je tedy potřeba modul inicializovat. V následující ukázce kódu je úryvek, který ověřuje otevření sériové komunikace a otevření portu pro SD modul, pokud k tomuto nedojde, nedojde ani ke spuštění aplikace.

```

1 Serial.print("Initializing SD card...");
2 if (!SD.begin(10)) {
3   Serial.println("initialization failed!");
4   while (1);
5 }else{
6   Serial.println("initialization done.");
7 }

```

Ukázka kódu 3.2: Ukázka inicializace SD modulu

Poté je možné otevřít a zpracovat konfigurační soubor. Ten, přesto, že je navržen jako *JSON*, na paměťovou kartu je uložen ve formátu *TXT*. Textový

formát je pro Arduino pametově méně náročný a rychlejší na zpracování. Pro načtení a zpracování dat jsem napsal funkci `LoadConfigFromSD()`. Zde využívám knihovny `SD.h` a `SPI.h`, nejdříve je potřeba soubor otevřít, což při neúspěchu opět zamezí dalšímu pokračování programu. Po otevření souboru jsou data zpracována, úryvek zpracování pole `"pumps"` je v následujícím úryvku kódu.

```

1 DynamicJsonDocument doc(1024);
2 DeserializationError error = deserializeJson(doc, configFile);
3 if (error) {
4     Serial.print(F("deserializeJson() failed: "));
5     Serial.println(error.f_str());
6     return;
7 }
8
9 JSONArray pumps = doc["pumps"];
10 for (int i = 0; i < pumps.size(); i++) {
11     Drinks[i].id = pumps[i]["id"];
12     Drinks[i].name = pumps[i]["drink"].as<String>();
13     Drinks[i].alcohol = pumps[i]["alcohol"];
14 }

```

Ukázka kódu 3.3: Zpracování konfiguračních dat

Knihovny pro Arduino jsou psané pro co nejjednodušší použití, nejčastější syntax je, že se pomocí klíčového slova knihovny deklaruje instance třídy, kterou knihovna nabízí a pak se volají její funkce. Tedy pro knihovnu `RTClib.h` je proměnná inicializována jako `RTC_DS1307 rtc` a funkce `begin()` pro inicializaci modulu reálného času bude zavolána v syntaxu `rtc.begin()`. Dále je pomocí funkce `rtc.adjust(TIME)` nastaven čas modulu. Pro LCD displej využívám knihovnu `LiquidCrystal_I2C.h`, ten je deklarován pomocí `LiquidCrystal_I2C lcd(0x27, 20, 4)`, argumenty jsou postupně adresa na sběrnici I²C, počet znaků v řádku a počet řádků. Inicializován je pomocí funkce `init()`. Pro uživatelské menu pak pomocí funkce `createChar(0, checkMark)` vytvářím vlastní znaky, první argument je číslo vlastního znaku a druhý jeho název. Konkrétně deklarace znaku `checkMark` je v následující ukázce.

```

1 byte checkMark[8] = {
2     B00000,
3     B00001,
4     B00011,
5     B10110,
6     B11100,
7     B01000,
8     B00000,
9     B00000
10 };

```

Ukázka kódu 3.4: Deklarace znaku `checkMark`

Dalším modulem, který je potřeba při startu nastavit je váha. Na práci s ní používám knihovnu `HX711.h` deklaruji proměnnou `HX711 scale`. Po deklaraci proměnné reprezentující váhu je pak pomocí funkce `begin(DoutPin, SckPin)` váha inicializována. Váhu je také potřeba zkalibrovat, na to se využívá tři funkcí, nejdříve se zavolá funkce `set_scale()` bez argumentu, pak se

zavoláním funkce `tare()` váha vynuluje a po položení předmětu se známou hmotností se pomocí funkce `get_units(10)` odečte hodnota číslo v argumentu funkce vyjadřuje z kolika měření je vrácen průměr. Tuto hodnotu pak vydělím hmotností použitého předmětu, čímž získáme kalibrační parametr pro funkci `set_scale()`. Tento postup jsem pro získání mého kalibračního parametru opakovat s několika předměty se známou hmotností, získané parametry zprůměroval, a došel pak k parametru, který se do funkce vloží jako argument, tedy `set_sacle(22.f)`.

Posledním, co je při startu potřeba nastavit jsou piny, to je provedeno pomocí zavolání funkce `pinMode(pin, mode)`, kde argument `pin` reprezentuje nastavovaný pin a argument `mode` nastavovaný mód, tedy input nebo output.

3.2.2 Řídicí program

V této podkapitole rozeberu detailněji chod řídicího programu, diagram jeho stavového automatu je vidět na obrázku 3.3 na následující straně. Rozeberu také některé použité funkce a přiblížím rozhodovací procesy pro přechody mezi stavy. Zároveň přiblížím použité proměnné a struktury, včetně enumerátorů pro stavový automat, ty jsou vidět v tabulce 3.1

Tabulka 3.1: Použité stavy stavového automatu

Enumerátor	Hodnota
STATE_WAIT_FOR_CMD	0
STATE_IDLE	10
STATE_DRINK_SELECTION	20
STATE_COCTAIL_SELECTION	30
STATE_DRINK_CUP_SELECTION	40
STATE_DRINK_CUP_INSERT	50
STATE_COCTAIL_CUP_INSERT	60
STATE_DRINK_PREPARE	70
STATE_COCTAIL_PREPARE	80
STATE_STOP	90
STATE_WASH	100
STATE_AFTER_STOP	110
STATE_DONE	120
STATE_ERROR	4000

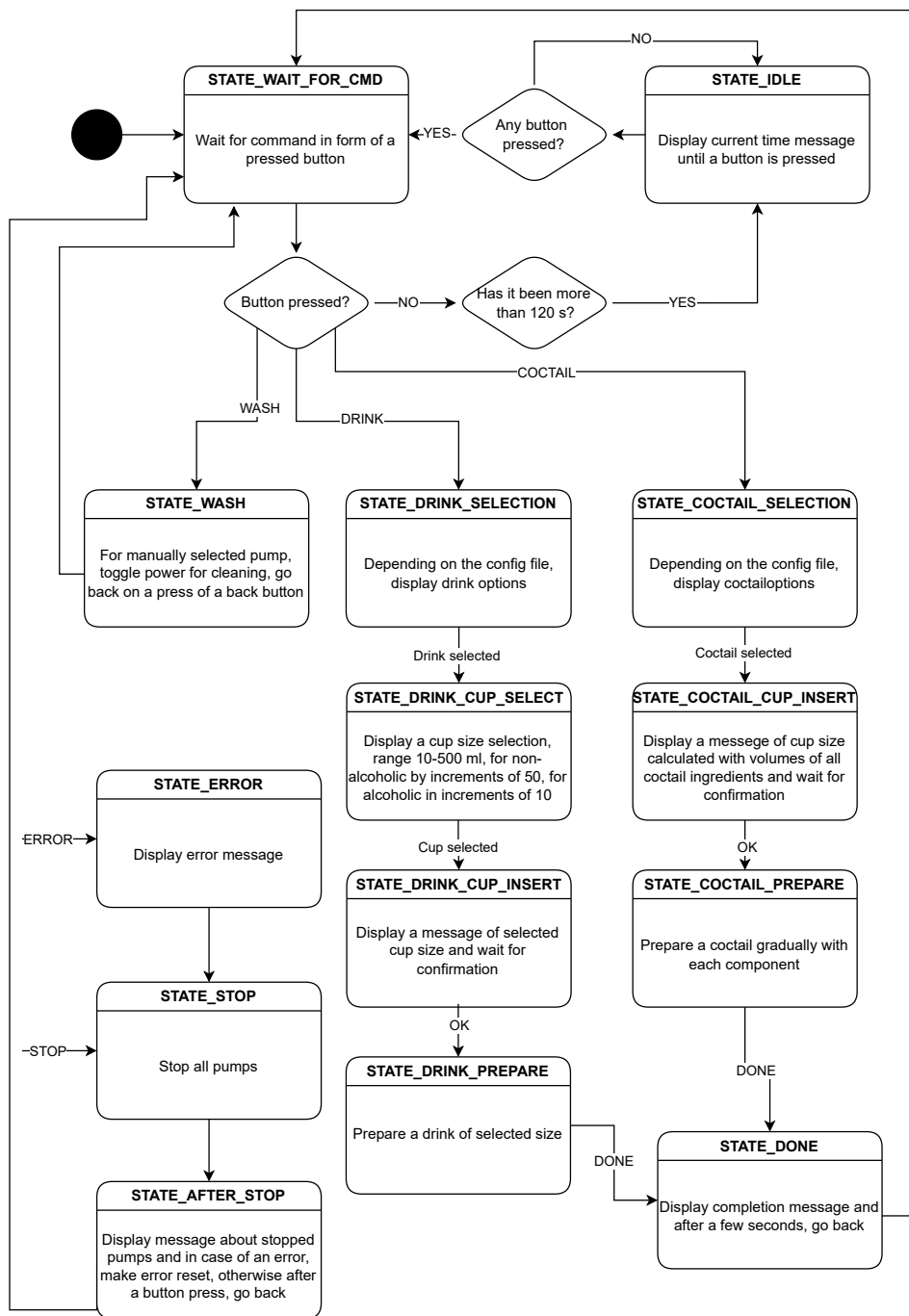
Na následující ukázce 3.5 je pak použita struktura pro řízení stavového automatu, pointerová logika je zde použita pro její rychlejší přístup k paměti, je tedy vhodná pro real-time aplikace a přispívá k bezpečnosti systému.

```

1 struct MachCtrlStruct {
2     MachineStateEnum State;
3 };
4
5 MachCtrlStruct MachineCtrl;
6 MachCtrlStruct* pInts = &MachineCtrl;

```

Ukázka kódu 3.5: Ovládací struktura



Obrázek 3.3: Diagram stavového automatu

Dalšími dvěma důležitými strukturami jsou struktury pro nápoje a pro koktejly, do kterých se ukládají data z načtené konfigurace. Jsou pak deklarovány jako pole, které ovšem nejsou dynamicky alokovány, ale jsou definovány maximálním počtem nápojů respektive koktejlů. Tento přístup jsem zvolil hlavně pro bezpečnost práce s pamětí, která je při real-time aplikacích důležitá. Struktura zobrazená na ukázce 3.6 je struktura použitá pro ukládání dat při chodu automatu, jako je volba nápoje respektive koktejlu, volba objemu sklenice, nebo volba čerpadla. Záměrně používám zde, i v celé aplikaci proměnné typu *byte*, pro úsporu paměti oproti běžnému integeru. Proměnná typu *word* je kratší zápis proměnné typu *unsigned int*. Zároveň obě tyto proměnné nabývají pouze kladných hodnot, jelikož vzhledem k jejich účelu není záporných hodnot potřeba.

```

1 struct DrinkSelectionStruct{
2     byte drink_count;
3     byte combination_count;
4     byte selected_drink_index;
5     byte selected_combination_index;
6     word volume;
7     byte volume_iteration;
8     byte pump_wash_index;
9     unsigned long done_millis;
10 };

```

Ukázka kódu 3.6: Struktura pro data při běhu automatu

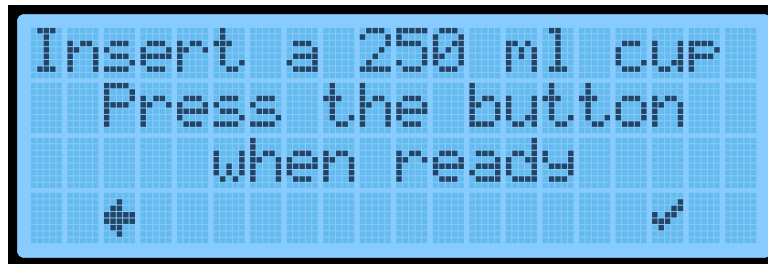
Než rozeberu chod stavového automatu, přiblížím některé použité funkce, za zmínku stojí nejčtenější typ použité funkce, a to zobrazování informací na displej, jako první ukázkou 3.7 uvádím funkci, která zobrazí informaci pro uživatele, jaký objem sklenice má vložit. Jako argumenty funkce posílám jaký displej bude použit pro vykonání funkce a parametr *volume*, který reprezentuje zvolený objem sklenice respektive pro koktejly celkový objem koktejlu. Na řádce 13 z ukázky volám funkci *LCD_Print_Selection_Button_Bar(...)*, ta vyobrazuje navigační menu na spodní řádek, argumenty mimo *lcd* určují, který ze znaků se má vypast, prostřední dva jsou pro navigační šipky, které na tomto menu ovšem nejsou potřeba. Dva krajní argumenty zobrazují tlačítka zpět a potvrdit, což lze vidět na obrázku 3.4.

```

1 void LCD_Print_Insert_Cup(LiquidCrystal_I2C lcd, word volume){
2     lcd.clear();
3     lcd.setCursor(0, 0);
4     lcd.print("Insert a ");
5     lcd.setCursor(9, 0);
6     lcd.print(volume);
7     lcd.setCursor(13, 0);
8     lcd.print("ml cup");
9     lcd.setCursor(2, 1);
10    lcd.print("Press the button");
11    lcd.setCursor(5, 2);
12    lcd.print("when ready");
13    LCD_Print_Selection_Button_Bar(lcd, true, false, false, true);
14 }

```

Ukázka kódu 3.7: Funkce na zobrazení pro LCD displej



Obrázek 3.4: Zobrazení funkce `LCD_Print_Insert_Cup` na displeji

Druhou funkci, kterou bych rád zmínil je funkce na detekování stisku tlačítka, ta jsou připojena na analogové vstupy a dle toho s nimi také pracují. Tento způsob je lehce komplikovanější, než kdybych s nimi pracoval jako s digitálními vstupy. Funkce, zobrazena na ukázce 3.8 bere jako vstupní argument jeden z pinů, na který jsou namapovaná tlačítka. Jelikož jsou tlačítka připojená v konfiguraci pull-up, je `lastButtonState` nastaven na `HIGH`, funkce tedy detekuje náběžnou hranu stisku, což znamená stav `LOW`. Konstantu `DEBOUNCE_DELAY` jsem po několika testech nastavil na 50 ms, ta zajišťuje správné vyhodnocení několika stisků za sebou. Funkce poté přečte hodnotu z analogového pinu a vyhodnotí stanovené podmínky, pokud ke stisku došlo, vrací `true`, pokud ne, vrací `false`.

```

1 bool detectRisingEdge(int pin) {
2     static int lastButtonState = HIGH;
3     static unsigned long lastDebounceTime = 0;
4     int buttonState = analogRead(pin);
5
6     if (buttonState < 100 && lastButtonState == HIGH && millis() -
7         lastDebounceTime > DEBOUNCE_DELAY) {
8         lastDebounceTime = millis();
9         lastButtonState = LOW;
10        return true;
11    }
12    if (buttonState >= 100) {
13        lastButtonState = HIGH;
14    }
15    return false;
16 }
```

Ukázka kódu 3.8: Funkce na detekování stisku tlačítka

Výchozím stavem stavového automatu je `STATE_WAIT_FOR_CMD`, tedy stav, kdy automat čeká na vstup od uživatele, ten je po delší neaktivitě překlopen do stavu `STATE_IDLE`, kde je zobrazena informace o aktuálním datu a času. Odtud se stiskem jakéhokoliv tlačítka zpět dostaneme do výchozího stavu, zde jsou zobrazeny dvě volby, na koktejly a na nápoje. Při volbě nápoje je automat překlopen do stavu `STATE_DRINK_SELECTION`, kde je zobrazena načtená volba nápojů, odtud po zvolení nápoje se přechází do stavu volby velikosti nápoje. Následuje stav `STATE_DRINK_CUP_INSERT`, kdy je pro potvrzení zobrazen zvolený objem a uživatel je vyzván vložit odpovídající sklenici, tedy volá se funkce zobrazená na ukázce 3.7. Tento stav je velmi podobný stavu `STATE_COCTAIL_CUP_INSERT`, jsou ale rozděleny

do dvou různých stavů kvůli čitelnosti a modularitě programu, již zmíněné stavy pracují převážně s funkcemi na detekci stisku tlačítek a zobrazování na displej.

Po potvrzení vložení sklenice již dochází k obsluze modulů, nejdříve je váha probuzena a vynulována, poté je automat překlopen do stavu *STATE_DRINK_PREPARE*, kde již dochází k přípravě nápoje. Větev chodu, kdy je zvolen koktejl se liší, mimo stavu na přípravu, pouze v absenci volby velikosti nápoje, jelikož je velikost nápoje stanovena konfigurací. Stav pro přípravu koktejlu, tedy *STATE_COCTAIL_PREPARE* je zobrazen na ukázce 3.9.

```

1 case STATE_COCTAIL_PREPARE:
2
3     for(size_t i = 0; i < Combinations[DrinkSelection.
4         selected_combination_index].ingredients; i++){
5         while (scale.get_units(10) < Combinations[DrinkSelection.
6             selected_combination_index].amount[i]){
7             pumpOn(Combinations[DrinkSelection.selected_combination_index].pumps[
8                 i]-1);
9             Serial.println(scale.get_units(10));
10
11             if(stopButtonPressed()){
12                 pInts->State = STATE_STOP;
13                 break;
14             }
15         }
16         scale.tare();
17         pumpOff(Combinations[DrinkSelection.selected_combination_index].pumps[i
18             ]-1);
19         Serial.println(scale.get_units(10));
20     }
21     scale.power_down();
22     LCD_Print_Done(lcd);
23     pInts->State = STATE_DONE;
24     break;

```

Ukázka kódu 3.9: Stav *STATE_COCTAIL_PREPARE*

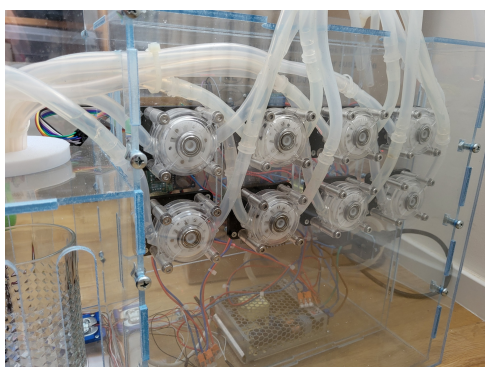
Na konci procesu přípravy nápoje, kdy se postupně pro každou ingredienci koktejlu vypíná a zapíná odpovídající čerpadlo, se následně vypne váha a automat je pak překlopen do stavu *STATE_DONE*. V tomto stavu se zobrazí pokyn k odebrání nápoje a po krátkém čaovém úseku se automat vrátí zpět do výchozího stavu.

Stav *STATE_WASH* je přístupný tlačítkem s výchozího menu a umožňuje čištění manuálním zapnutím a vypnutím jednotlivých čerpadel, možno i několika najednou. Stav pro řešení chyb zobrazí informaci o chybě a pokyn k restartu, odkud je okamžitě překlopen do stavu *STATE_STOP*, který zastaví všechna čerpadla.

Kapitola 4

Nápojový automat

V této kapitole shrnu funkci a ovládní automatu, jeho použité prvky, dále také testování funkčnosti a zjištěné nedostatky. Konfigurovatelný nápojový automat byl sestaven z mikrokontroléru Arduino Nano Every, osmi peristaltických čerpadel, osmikanálového relé modulu, senzorů váhy a AD převodníku HX711, RTC modulu, LCD displeje, tlačítek a zdroje napájení. Automat se zapíná, po připojení síťového kabelu, pomocí kolébkového spínače na zadní straně. Po zapnutí probíhá několik vteřin inicializace nastavení všech modulů a zejména zpracování konfiguračního souboru, načež je zobrazena úvodní nabídka a automat je připraven k použití. Automat je pak ovládán pomocí tlačítek a LCD displeje umístěných nad prostorem pro výdej nápoje.



Obrázek 4.2: Umístění čerpadel



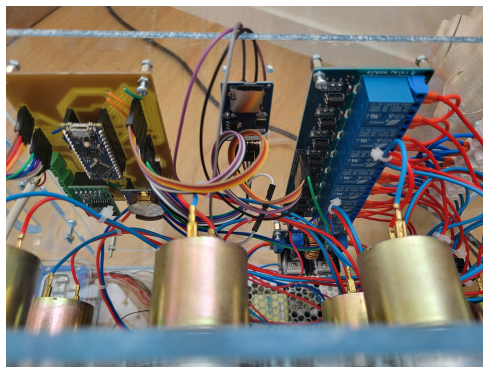
Obrázek 4.1: Přední část automatu

Funkce každého tlačítka je na displeji znázorněna vlastním znakem. Automat má možnost připravovat koktejly, ale i jednotlivé nápoje v definovaném rozsahu. Čerpadla, zobrazena na obrázku 4.2 mají sací část vyvedenou pomocí silikonových hadiček s potravinovou certifikací horní stranou automatu, kam je možno umístit nádoby na jednotlivé nápoje. Automat je otestovaný a plně funkční, bylo vyzkoušeno několik konfigurací, všechny úspěšně, včetně testu různých kombinací i jednotlivých čer-

padel. Varianta hlídání přečerpaného objemu kapaliny pomocí váhy se mi osvědčila jako spolehlivá, váhy se mi podařilo zkalibrovat na přesnost jednotek gramů, což odpovídá i přesnosti jednotek mililitrů, jedinou nevýhodou tohoto přístupu je zpracování informace o váze. Při zpracování počítám průměr z deseti měření, což zpomaluje reakci systému na vypnutí čerpadla. To znamená, že může být tekutiny načerpáno o několik mililitrů více, než bylo zamýšleno.

Během testování jsem narazil na dva nedostatky. Prvním nedostatkem je návrh zapojení tlačítek, jelikož jsem ani u jednoho z nich nenavrhl zapojení na interrupt pin, vykonání příkazu stop může občas trvat déle. Druhým nedostatkem je spíše mechanický problém, čerpadla při zapojení na napájení 24 V při chodu poměrně vibrují. Upravil jsem tedy výstup ze step-down měniče na přibližně 16 V, což je stále v napájecím rozsahu Arduina, a čerpadla již nevibrují tak výrazně, aby to bylo problém.

Na závěr také uvádím tabulku použitých materiálů a součástek.



Obrázek 4.3: Zapojení čerpadel

Tabulka 4.1: Použité materiály a součástky

Součástka	Počet
Plexisklo 4 mm	2 x 1 m ²
Šroub M5x20 a čtyřhranná matka M5	20 ks
Šroub M3x20 a matka M3	48 ks
Distanční sloupek 5 mm	24 ks
Šroub M3x6	24 ks
Peristaltická čerpadla	8 ks
Faston 2,8 mm	16 ks
Vodič červený 0,75 mm ²	5 m
Vodič modrý 0,75 mm ²	5 m
Faston 6,3 mm	7 ks
Kolébkový spínač	1 ks
Euro zásuvka	1 ks
Přívodní vodič	1 ks
Svorkovnice	1 ks
Šroubovací svorkovnice 2,54 mm	8 ks
Osmikanálový relé modul s optočlenem	1 ks
Arduino Nano Every	1 ks
SD modul	1 ks
Senzor váhy	4 ks
AD převodník HX711	1 ks
RTC modul	1 ks
Step-down měnič	1 ks
LCD displej 20x4	1 ks
Mikrospínač TC-1212T	4 ks
Rezistor 10 kΩ	4 ks
Silikonová hadička	4 m
Zdroj Mean-Well RS-100-24	1 ks



Závěr

Cílem této práce byla rešerše existujících řešení pro výdejové automaty, návrh konfigurovatelného nápojového automatu, jeho implementace pomocí vhodné embedded platformy a jeho otestování v reálném provozu. V první části jsem provedl rešerši existujících řešení na základě které jsem v druhé části navrhl konfigurovatelný nápojový automat.

Automat byl, jak bylo zamýšleno, navrhnout k sestavení pomocí běžně dostupných materiálů, pomocí kterých jsem ho také v poslední části realizoval. Pro projekt jsem použil vývojovou desku Arduino Nano Every a program jsem napsal pomocí vývojového prostředí VS Code a rozšíření Platforimo. Pro jeho konfiguraci je použita paměťová karta, pro jeho přenastavení tedy není potřeba zasahovat do zdrojového kódu, ale stačí pouze upravit konfigurační soubor. Použitá čerpadla dokáží připravit nápoj v řádu desítek vteřin, což přispívá k pohodlnosti jeho používání. Fyzickou schránku jsem navrhl pro lehkou údržbu, která se dá z části provést i bez jeho rozebrání. Projekt také nabízí prostor pro zlepšení či další vývoj, jako je například zlepšení univerzálnosti a robustnosti kódu nebo bezdrátová komunikace s řídicí jednotkou nebo program zajišťující správný formát konfiguračního souboru.



Literatura

- [1] BELLIS, Mary. The History of Vending Machines. Online. *ThoughtCo*. 2019, s. 1. Dostupné z: <https://www.thoughtco.com/the-history-of-vending-machines-1992599>. [cit. 2023-12-04].
- [2] *Post mix*. Online. Drinkotech. Dostupné z: <https://drinkotec.ch/drinks/postmix-for-an-economic-ecological-gain/>. [cit. 2023-12-07].
- [3] DOUGAL, Sabrina. *How does a coffee vending machine work*. Online. Expert Market. 2022. Dostupné z: <https://www.expertmarket.com/uk/coffee-machines/how-does-a-coffee-vending-machine-work>. [cit. 2023-12-07].
- [4] *How do vending machines work*. Online. Nu Vending. C2023. Dostupné z: <https://www.nuvending.co.uk/latest-news/how-do-vending-machines-work/>. [cit. 2023-12-07].
- [5] *POST-MIX ICE DRINK DISPENSER*. Online. In: Lancer Mid West. Dostupné z: <https://www.lancermidwest.com/en/ice-beverage-dispensers/18194-ibd22cb-4500-cold-carb-post-mix-ice-drink-dispenser-6-lev-self-serve-lever-valves-coke-graphics.html>. [cit. 2024-05-04].
- [6] GLIDDEN, Brad a ANGELICA, Mike. *Understanding post-mix systems*. Online. In: Cornelius. 2015. Dostupné z: https://www.cornelius.com/wp-content/uploads/2019/05/Understanding-Post-Mix_December-2018.pdf. [cit. 2024-05-04].
- [7] *Nápojové automaty*. Online. Kafepartner. C2008-2024. Dostupné z: <https://kafepartner.cz/napojove-automaty?cookie=450.3999938964844>. [cit. 2024-05-04].
- [8] ROBERTS, Sam. *How coffee vending machines work*. Online. Vending-times. 2023. Dostupné z: <https://www.vendingtimes.com/articles/how-coffee-vending-machines-work/>. [cit. 2024-05-04].
- [9] *Prodejní automaty*. Online. In: Coste. Dostupné z: <http://coste.cz/automaty/prodejni-automaty/>. [cit. 2024-05-04].

- [10] BENSON, Michael. *How Do Vending Machines Work*. Online. Vending locator. 2023. Dostupné z: <https://www.vendinglocator.com/blog/how-do-vending-machines-work>. [cit. 2024-05-04].
- [11] *Mikroelektronika*. Online. 2024. Dostupné z: <https://www.mikroelektronika.com/cs/produkty/kategorie/automaty-na-vydej-jizdenek>. [cit. 2024-05-04].
- [12] *Cecilia AI*. Online. 2024. Dostupné z: <https://cecilia.ai/>. [cit. 2024-05-04].
- [13] *Most Popular Types of Industrial Pumps*. Online. JSG Industrial. C2024. Dostupné z: <https://jsgindustrial.com/types-of-industrial-pumps/>. [cit. 2024-05-03].
- [14] *Membránové čerpadlo*. Online. PROKOP, Luboš. Homesolar. 2020. Dostupné z: <https://www.homesolar.cz/p/membranove-cerpadlo-f10-121/>. [cit. 2024-05-05].
- [15] *Peristaltic Pump Intllab*. Online. Aliexpress. 2024. Dostupné z: <https://www.aliexpress.com/item/1005001353898759.html>. [cit. 2024-05-04].
- [16] KRÁLOVÁ, Magda. *Elektromotry*. Online. Eduportál techmania. 2007. Dostupné z: <https://edu.techmania.cz/cs/encyklopedie/fyzika/elektromagneticka-indukce/elektromotory>. [cit. 2024-05-03].
- [17] MR-Z, Mateusz. *Krokový motor*. Online. Botland blog. 2023. Dostupné z: <https://botland.cz/blog/krokovy-motor-co-to-je-princip-fungovani-a-pouziti/>. [cit. 2024-05-03].
- [18] *What is a servo motor*. Online. Electrical4U. 2024. Dostupné z: <https://www.electrical4u.com/what-is-servo-motor/>. [cit. 2024-05-03].
- [19] *Jaký je rozdíl mezi kartáčovým a bezkartáčovým motorem?* Online. Firstman. C2014-2019. Dostupné z: <https://www.firstman.cz/jaky-je-rozdil-mezi-kartacovym-a-bezkartacovym-motorem/>. [cit. 2024-05-03].
- [20] SHAH, Jesal. *How does an H-Bridge work?* Online. CircuitBread. 2023. Dostupné z: <https://www.circuitbread.com/ee-faq/how-does-an-h-bridge-work>. [cit. 2024-05-03].
- [21] PRASAD, Leela. *What is Relay? How it Works?* Online. ElectronicsHub. 2024. Dostupné z: <https://www.electronicshub.org/what-is-relay-and-how-it-works/>. [cit. 2024-05-03].
- [22] ORLÍKOVÁ, ING., Soňa. *Měření průtoku tekutin - principy průtokoměrů*. Online. Tzbinfo. 2008. Dostupné z: <https://voda.tzbinfo.cz/teorie-voda-kanalizace/4624-mereni-prutoku-tekutin-principy-prutokomeru>. [cit. 2024-05-05].

- [23] *HX711 with load cell calibration*. Online. Forum Arduino. 2020. Dostupné z: <https://forum.Arduino.cc/t/hx711-with-load-cell-calibration/650485>. [cit. 2024-05-05].
- [24] *Top 10 Microcontroller Development Boards*. Online. The Mechatronic Blog. 2023. Dostupné z: <https://www.themechatronicsblog.com/2023/05/top-10-microcontroller-development-boards-for-beginners-to-professionals.html>. [cit. 2024-05-05].
- [25] *Arduino Nano Every*. Online. Arduino. 2024. Dostupné z: <https://store.Arduino.cc/products/Arduino-nano-every>. [cit. 2024-05-05].
- [26] *Arduino Nano Every Headers*. Online. Arduino. 2024. Dostupné z: <https://store.Arduino.cc/products/Arduino-nano-every-with-headers>. [cit. 2024-05-05].
- [27] *Arduino IDE*. Online. Arduino. 2024. Dostupné z: <https://www.Arduino.cc/en/software>. [cit. 2024-05-05].
- [28] *Platformio*. Online. 2024. Dostupné z: <https://platformio.org/>. [cit. 2024-05-05].
- [29] *Zdroj Mean Well RS-100-24*. Online. Eshop Mean Well. C2024. Dostupné z: <https://www.eshop-meanwell.cz/rs-100-24-mean-well-spinany-zdroj-uzavreny-100w-24/>. [cit. 2024-05-18].
- [30] *Autodesk Education*. Online. 2024. Dostupné z: <https://www.autodesk.com/education/home>. [cit. 2024-05-18].