

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

KATEDRA APLIKOVANÉ ELEKTRONIKY A TELEKOMUNIKACÍ

BAKALÁŘSKÁ PRÁCE

Návrh a realizace poloautomatické leptací stanice

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta elektrotechnická
Akademický rok: 2016/2017

ZADÁNÍ BAKALÁŘSKÉ PRÁCE (PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jan MORÁVKA**
Osobní číslo: **E14B0155P**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Elektronika a telekomunikace**
Název tématu: **Návrh a realizace poloautomatické leptací stanice**
Zadávací katedra: **Katedra aplikované elektroniky a telekomunikací**

Z á s a d y p r o v y p r a c o v á n í :

Cílem BP je navrhnout a zhotovit tříkomorovou leptací kyvetu (z plexiskla), kde jedna komora (leptací) bude vybavena teplotním senzorem, topným tělesem a cirkulačním čerpadlem. Součástí leptací stanice bude dále řídicí modul, který bude umožňovat: regulaci teploty, vypnutí/zapnutí vzduchového čerpadla a zvukovým signálem oznámí uplynutí nastavené doby (funkce časovače).

1. Nastudujte a popište metody používané pro výrobu plošných spojů.
2. Navrhněte a zhotovte tříkomorovou leptací kyvetu, kde jedna z komor bude vyhrazena pro leptání motivu vodivé vrstvy a bude vybavena teplotním senzorem, topným tělesem a cirkulačním čerpadlem.
3. Navrhněte a realizujte řídicí modul leptání, který bude umožňovat: nastavení a udržování požadované teploty, zapínání/vypínání cirkulačního čerpadla a po uplynutí nastavené doby oznámí konec cyklu leptání.
4. Na základě teoretických poznatků a série vytvořených zkušebních vzorků stanovte optimální čas a teplotu pro výrobu 1dm² jednovrstvého plošného spoje s různou minimální šířkou cesty

Rozsah grafických prací: podle doporučení vedoucího

Rozsah kvalifikační práce: 30 - 40 stran

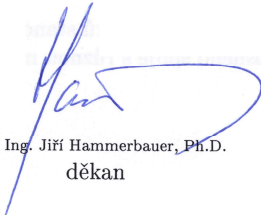
Forma zpracování bakalářské práce: tištěná/elektronická

Seznam odborné literatury:


1. Záhlava, Vít : Návrh a konstrukce desek plošných spojů : principy a pravidla praktického návrhu /1. vyd.. Praha : 2010. ISBN 978-80-7300-266-4
2. Margolis, Michael : Arduino cookbook /2nd ed.. Sebastopol : c2012. ISBN 978-1-449-31387-6

Vedoucí bakalářské práce: Ing. Martin Šindelář
Katedra aplikované elektroniky a telekomunikací

Datum zadání bakalářské práce: 14. října 2016
Termín odevzdání bakalářské práce: 8. června 2017



Doc. Ing. Jiří Hammerbauer, Ph.D.
děkan



Doc. Dr. Ing. Vjačeslav Georgiev
vedoucí katedry

V Plzni dne 14. října 2016

Abstrakt

Předložená bakalářská práce pojednává o různých metodách výroby desek plošných spojů a jejich výhodách či nevýhodách. Dále se práce zabývá návrhem a posléze zhotovením poloautomatické leptací stanice pro výrobu desek plošných spojů. Základem je tříkomorová kyveta, kde jedna z komor je vyhrazena pro leptání motivu desek plošných spojů, další pro vyvíjení fotorezistu, a třetí pro vodu pro opláchnutí desky od leptacího a vyvíjecího roztoku. Leptací komora je dále vybavena teplotním senzorem, cirkulačním čerpadlem a topným tělesem. O řízení těchto komponent se stará řídicí systém založený na mikrokontrolérové vývojové desce Arduino Uno. Řídicí systém umožňuje uživateli stanice si zvolit udržovanou teplotu v leptací komoře a také čas, po jehož uplynutí stanice oznámí konec cyklu leptání.

Klíčová slova

Leptací stanice, metody a výroba desek plošných spojů, Arduino, Atmel, ATmega328P, řídicí modul, topné těleso, teplotní senzor.

Abstract

The presented bachelor thesis deals with various methods of production of printed circuit boards and their advantages or disadvantages. Thesis also deals with the design and then with the production of a semiautomatic etching station for production of printed circuit boards. The base is a three chamber cell, where one of the chambers is designated for etching the printed circuit board motive, another for photoresist evocation, and the third one for water to rinse the board off from etching and evocation solution. The etching chamber is further equipped with a temperature sensor, a circulation pump and a heating element. The management of these components takes over the control system based on the Arduino Uno microcontroller development board. The control system allows user of the station to choose the temperature maintained in the etching chamber and also the time after which the station will announce the end of the etching cycle.

Key words

Etching station, methods and production of printed circuit boards, Arduino, Atmel, ATmega328P, control module, heating element, temperature sensor.

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této bakalářské práce.

Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

.....
podpis

V Plzni dne 12.6.2017

Jan Morávka

Obsah

OBSAH	7
ÚVOD	8
SEZNAM SYMBOLŮ A ZKRATEK	10
1 METODY VÝROBY DESEK PLOŠNÝCH SPOJŮ	11
1.1 DESKA PLOŠNÝCH SPOJŮ.....	11
1.1.1 Materiály pro výrobu DPS.....	12
1.1.2 Typy DPS.....	17
1.2 TECHNOLOGIE VÝROBY DPS.....	22
1.2.1 Subtraktivní technologie.....	22
1.2.2 Aditivní technologie.....	24
1.2.3 Semiaditivní technologie.....	25
2 NÁVRH A REALIZACE KYVETY	33
2.1 KOMPONENTY STANICE.....	35
3 NÁVRH A REALIZACE ŘÍDÍCÍHO SYSTÉMU	36
3.1 ARDUINO UNO REV3.....	39
3.2 ARDUINO IDE.....	40
3.2.1 Používání.....	41
3.2.2 Programovací jazyk.....	42
3.3 ŘÍDÍCÍ PROGRAM STANICE.....	44
4 VÝSLEDKY TESTOVÁNÍ	49
ZÁVĚR	50
SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ	52
PŘÍLOHY	1
<i>Příloha A – Ukázky objednávkových listů výrobců DPS</i>	<i>1</i>
<i>Příloha B – obrázky součástí kyvety</i>	<i>5</i>
<i>Příloha C – Arduino IDE</i>	<i>7</i>
<i>Příloha D – Řídící program stanice</i>	<i>9</i>

Úvod

V dnešní době se mnoho mladých, a nejen mladých, lidí zabývá elektrotechnikou. Ať už čistě zájmově, nebo i profesně, často se setkáváme s problematikou výroby desek plošných spojů (dále jen DPS), kdybychom rádi přenesli svůj návrh do reality a vytvořili si tak naši vlastní DPS. Bohužel pro tyto účely nemáme vždy k dispozici dostatečné prostředky.

Jedinou možností pak zbývá svůj návrh zaslat vybranému výrobcí a nechat si danou DPS zhotovit. Ačkoli je tento postup, z hlediska profesionality zpracování a funkčnosti výsledného výrobku, pravděpodobně tím nejspolehlivějším, naskýtá se také řada omezujících faktorů a nevýhod, které nám mohou naši práci znepríjemnit.

Mezi tyto nevýhody můžeme zařadit například dodací dobu, cenu nebo také u některých výrobců minimální počet kusů. Především je ale problémová možnost opravy chyb. Může se totiž stát, že po týdnu čekání dorazí od výrobce deska a my při osazování součástek nebo až při testování našeho výrobku zjistíme, že jsme někde v návrhu udělali chybu. Taková deska je pak na vyhození, stejně tak jako čas a peníze, které jsme do ní investovali. Musíme návrh přepracovat, znovu odeslat do výroby a znovu čekat a platit náklady. To je také jedním z důvodů, proč toto téma na bakalářskou práci vzniklo.

Předkládaná práce se zabývá návrhem a realizací poloautomatické leptací stanice pro výrobu DPS. Účelem takové stanice je právě výroba prototypů DPS, a tak i případné zabránění opětovnému zasílání opravovaných návrhů do výroby. Další výhodou mohou být nižší náklady a také znatelně kratší výrobní čas. Výroba prototypu desky se tak krátí na jednotky hodin a výrobcí pak můžeme zaslat návrh již kompletní a bez chyb.

Tato leptací stanice pak nenalézá využití pouze v profesní sféře pro výrobu prototypů a následnému zaslání návrhu do výroby, ale samozřejmě také pro aplikace, kde i tato kvalita zpracování je dostačující. To ocení zejména čtenáři a domácí kutilové, kteří se o elektrotechniku zabývají ve svém volném čase.

Text je dělen do tří částí, resp. kapitol. V první části jsou popsány metody výroby DPS, využívané materiály, výrobní postupy a procesy. Druhá část se zabývá návrhem leptací kyvety a její realizací včetně veškerých dalších součástí potřebných k vytápění a cirkulaci roztoku uvnitř leptací lázně. Třetí kapitola se zabývá návrhem řídicího systému pro všechny tyto součásti, výběrem a seznámením s mikrokontrolérem a jeho vývojovým prostředím. Dále je také uvedeno několik málo ukázek z ovládacího programu.

Práce je zpracována v tištěné i elektronické podobě. Obě tyto formy jsou obsahově shodné, avšak k elektronické formě je přiložen zdrojový kód ovládacího programu včetně komentářů.

Seznam symbolů a zkratek

DPS.....	Deska plošných spojů
SMT.....	Technologie povrchové montáže součástek (Surface-Mount Technology)
EMC.....	Elektromagnetická kompatibilita
PTFE.....	Polytetrafluoretylen
CEM.....	Kompozitní materiál (Composite Epoxy Material)
THT.....	Technologie montáže součástek s drátovými vývody skrze DPS (Through-Hole Technology)
HDI.....	označení pro plošné spoje s velkou hustotou spojů (High Density Interconnect)
CNC.....	Číslicové řízení strojů počítačem (Computer Numeric Control)
HAL / HASL.....	technologie žárového nanesení pájky na motiv desky plošných spojů (Hot Air Leveling / Hot Air Solder Leveling)
DIP.....	označení obdélníkového typu pouzdra čipu většinou pro THT montáž s vývody v řadě po dvou stranách (Dual In-line Package)
QFP.....	označení čtvercového typu pouzdra čipu většinou pro SMD montáž s vývody po všech čtyřech stranách (Quad Flat Package)
I ² C (IIC).....	typ počítačové sériové sběrnice (Inter-Integrated Circuit)
ICSP.....	typ počítačové sériové sběrnice (In-Circuit Serial Programming)

1 Metody výroby desek plošných spojů

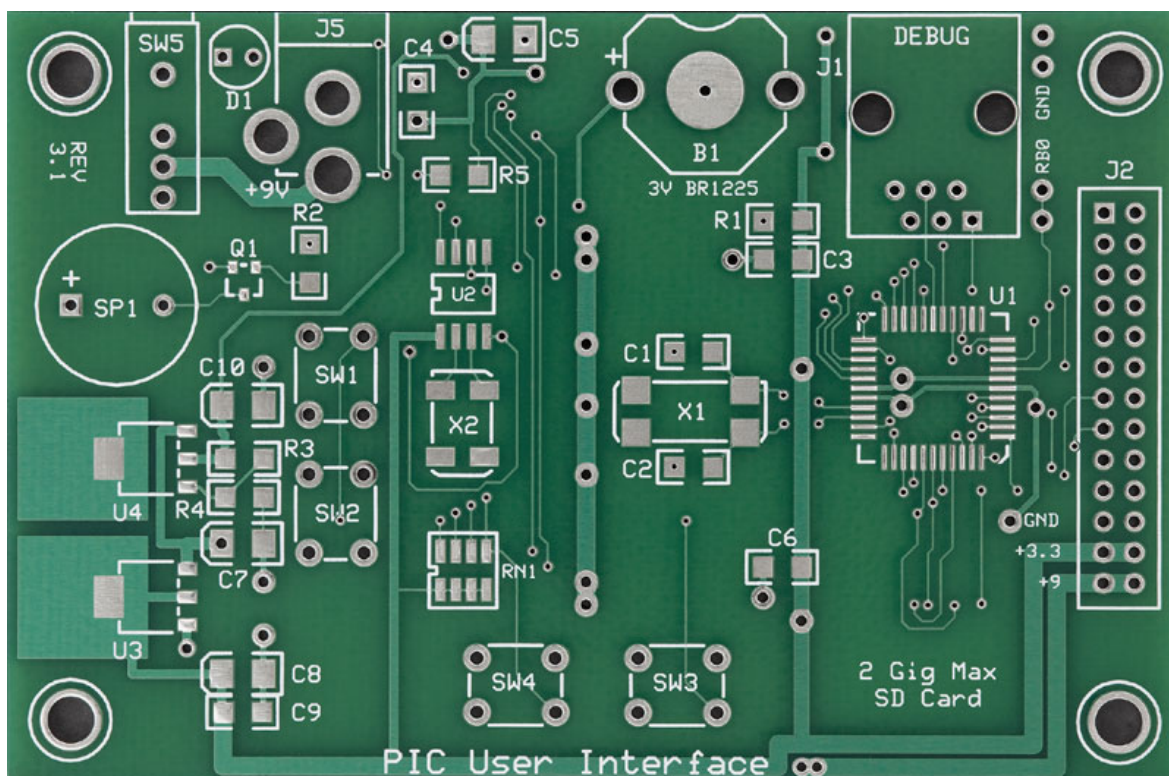
1.1 Deska plošných spojů

Ještě než se pustíme do metod výroby DPS, je třeba si říci, co to vlastně deska plošných spojů je, z jakých materiálů se vyrábí a jaké typy desek existují. Jak již název napovídá, deska plošných spojů (zkráceně DPS, anglicky Printed Circuit Board, PCB) je základní substrát, na jehož povrchu jsou vytvořeny vodivé cesty. Mezi hlavní funkce DPS v elektrotechnice se řadí mechanická podpora pro osazené součástky a zároveň elektrické propojení vývodů těchto součástek. Další významnou funkcí je odvod ztrátového tepla produkovaného součástkami.

Historie desek sahá daleko do minulosti. Již v padesátých letech 19. století se objevila první prkénka, na kterých byly umístěny první elektronické součástky propojené kovovými dráty. Avšak první DPS v dnešním slova smyslu vznikala kolem roku 1925, kdy Charles Ducas tiskl vodivý materiál na izolační podložku pomocí standardní tiskové planžety. Odtud také název tištěné spoje, který se drží dodnes. Tato technologie měla mnoho nedostatků, např. malou vodivost spojů. Zlom nastal po roce 1943, kdy Paul Eisler patentoval technologii leptání měděné fólie. Zpočátku byly jednostranné desky dostačující, ale s postupem času, miniaturizací součástek a počátkem vzniku prvních integrovaných obvodů přestaly být tyto obvody na jedné vrstvě propojitelné. Dalším zlomem v historii desek se tak stává technologie vícevrstvých DPS a také technologie povrchové montáže součástek. [1]

Dnes jsou na DPS kladeny značné požadavky. Desky musí být spolehlivé a zároveň i cenově dostupné. Musí vyhovovat nejen elektricky, mechanicky a tepelně, ale také musí vyhovovat podmínkám prostředí, v jakých je dané zařízení provozováno. Důležitými elektrickými vlastnostmi jsou vnitřní a povrchová rezistivita, vodivost, relativní permitivita, ztrátový činitel, elektrická pevnost. Mechanickými vlastnostmi jsou tepelná roztažnost, pevnost, rovinnost a nehořlavost. Mezi tepelné vlastnosti se řadí především tepelná vodivost, teplota skelného přechodu a odvod ztrátového tepla (vedením, prouděním, sáláním). Vlastnosti ochraňující DPS před podmínkami prostředí jsou například chemická odolnost, odolnost vůči klimatickým změnám a v dnešní době také

velmi významná elektromagnetická kompatibilita. Za zmínku také stojí požadavky na celkový vzhled desky, což zahrnuje především potisk technologických značek a označení rozmístění součástek. Nejvýznamnějšími faktory ovlivňující celkovou funkčnost a spolehlivost DPS jsou materiály použité při výrobě desky a technologie výrobního procesu.



Obr. 1.1 Deska plošných spojů; převzato z [2]

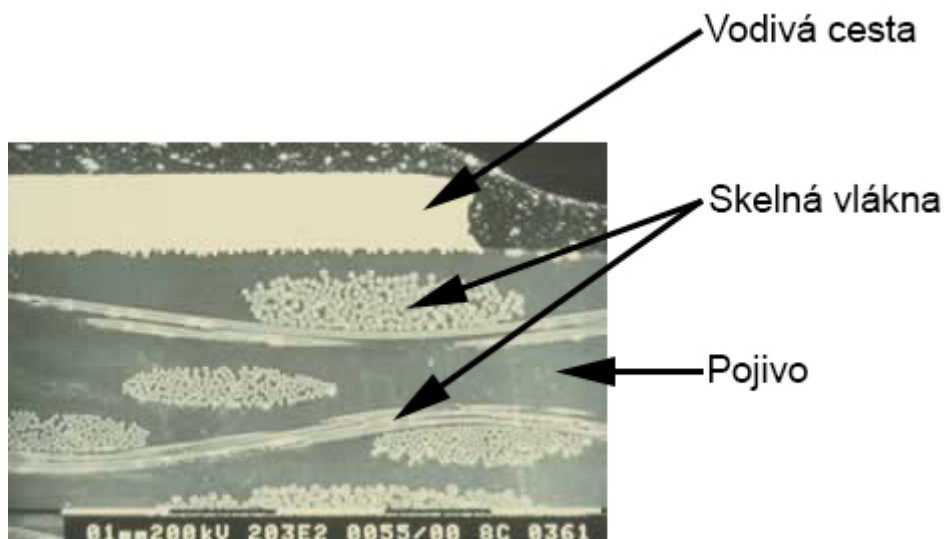
1.1.1 Materiály pro výrobu DPS

Základními skupinami materiálů pro výrobu substrátu (resp. nosné desky) pro DPS jsou pojiva a výztuž. Jako výztuž se nejčastěji používají skelné vlákna a tkaniny, tvrzený papír, aramidová a křemenná vlákna. Tyto výztuže udávají mechanické vlastnosti DPS. Pojiva určují především elektrické vlastnosti a jsou to převážně umělé pryskyřice. Nejčastěji používanými pojivy jsou fenolformaldehydové, epoxidové, polyesterové a polyimidové pryskyřice, polytetrafluoretylen (PTFE) a polyeterimid. Výsledný substrát vzniká spojením několika listů tzv. prepregu, který se vyrábí sycením výztuže pojivy. Výroba substrátu se provádí slisováním za zvýšených teplot a tlaku. Často se také setkáváme s názvem laminát, resp. sklolaminát.

Značení materiálů:

- FR-1, FR-2 – Celulóзовý papír nasycený fenolformaldehydovou pryskyřicí. FR je zkratka odvozená z anglického fire retardant, což v překladu znamená zpomalovač hoření. Rozdíl mezi deskami FR-1 a FR-2 je v hodnotě T_g (teplota skelného přechodu – tzv. zeskelnatění – materiál přestává být tvrdý a pevný). Ta se udává pro FR-1 130 °C a pro FR-2 105 °C. Oba tyto typy materiálu jsou velmi laciné a určené pro nenáročné aplikace. Mají horší mechanické vlastnosti a jsou málo odolné proti vlhkosti.
- FR-3 – Jako nosný materiál je využit celulóзовý papír jako u materiálů FR-1 a FR-2, avšak jako pojivo se zde používá pryskyřice epoxidová. Lze jej využívat v náročnějších aplikacích než materiály FR-1 a FR-2. Také disponuje lepšími mechanickými, elektrickými i tepelnými vlastnostmi. V dnešní době se materiál FR-3 ani materiály FR-1 a FR-2 prakticky nevyrábí nebo je jejich výroba velmi omezená.
- FR-4 – Dnes nejpoužívanější materiál. Výztuž tvoří tkanina ze skelných vláken (skelná rohož) a pojivo tvoří epoxidová pryskyřice. Běžně se při výrobě slisovává několik vrstev prepregu. Například pro výrobu standardních desek FR-4 s tloušťkou 1.5 mm se používá 8 vrstev prepregu. Materiál se vyznačuje dobrými elektrickými a mechanickými vlastnostmi. Je málo nasákavý, poměrně teplotně stálý, chemicky odolný a samozhášecí. FR-4 lze provozovat do teploty 150 °C. Standardně se využívá pro jednostranné, dvoustranné s pokovenými otvory a vícevrstvé DPS. Materiál je určený i pro použití v náročných aplikacích.
- FR-5 – Materiál velmi podobný materiálu FR-4. Má lepší vlastnosti a využívá se především kvůli vyšší tepelné odolnosti (až 180 °C).
- G-10, G-11 – Materiály téměř shodné s FR-4 a FR-5. Rozdíl ve značení říká, že tyto materiály nejsou nehořlavé.

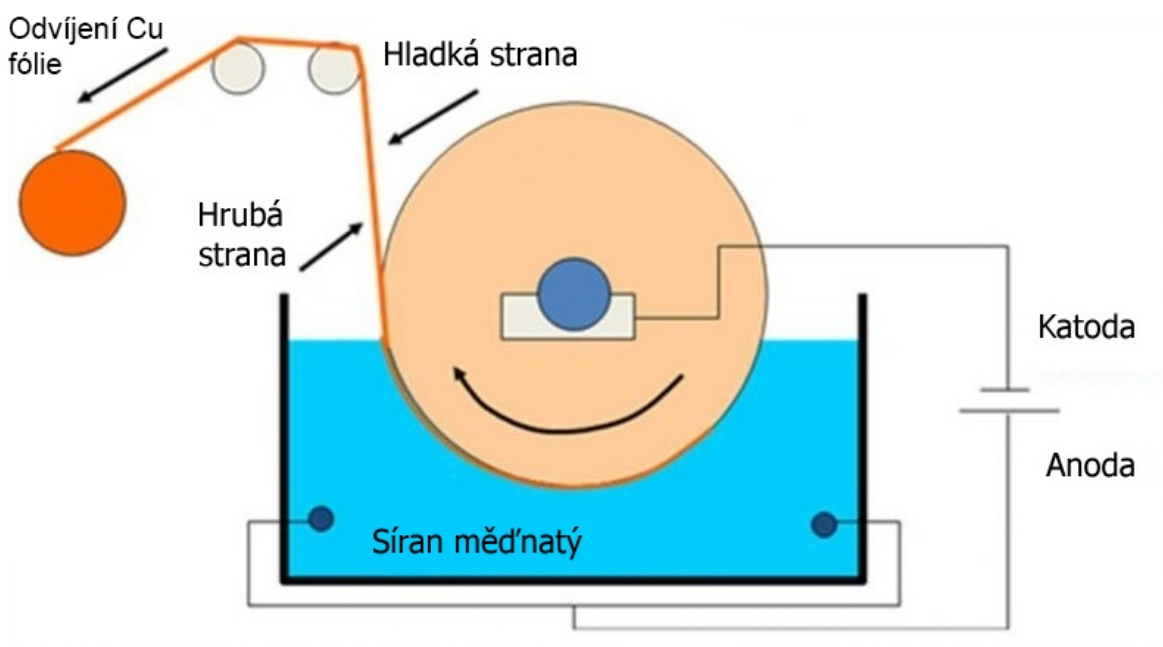
- G-30 – Polyimidový skelný laminát. Má výborné mechanické a elektrické vlastnosti a zároveň se vyznačuje výbornou teplotní stálostí. Díky tomu se také využívá pro tepelně velmi namáhané aplikace. Desky z materiálu G-30 lze provozovat při teplotách až 260 °C. Nevýhodou polyimidových pryskyřic je jejich navlhavost, proto se využívají jen ve speciálních případech.
- CEM-1 – Materiály označované zkratkou CEM jsou materiály kompozitní (obsahují nejméně dva materiály výztuže). Výsledné vlastnosti kompozitních materiálů jsou kombinací vlastností materiálů, ze kterých je vyroben. Jádro výztuže materiálu CEM-1 je tvořeno celulózovým papírem a kryje jej vrstva skelné rohože. Jako pojivo se používá epoxidová pryskyřice. Často se používá jako laciná náhrada za FR4. CEM-1 není vhodný pro dvoustranné desky s pokovenými otvory.
- CEM-3 – Výztuž materiálu tvoří společně tkané a netkané skelné vlákno. Tento materiál lze pokovovat a svými vlastnosti se vyrovná nejběžnějšímu FR4.
- Značení dalších materiálů – FR-6, FR-8, CEM-2, CEM-4, CEM-5, PTFE, RF-35, GPY, GPO-2, GPO-3



Obr. 1.2 Zvětšený příčný řez deskou FR-4; převzato a upraveno z [3]

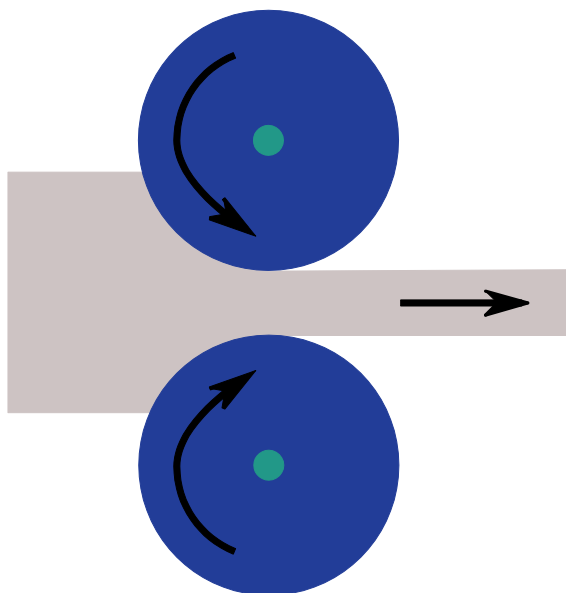
Mezi další substráty se také řadí korundové keramiky, beryliové keramiky a substráty s kovovými či křemíkovými jádry. Výhodou těchto substrátů je velmi dobrá tepelná vodivost, chemická odolnost a malá teplotní roztažnost. Naproti tomu jsou těžší a dražší. Další nevýhodou keramik je jejich křehkost a beryliová keramika je navíc i toxická.

Třetí hlavní skupinou materiálů pro výrobu DPS jsou materiály elektricky vodivé. Vhodnými elektricky vodivými materiály jsou měď (Cu), cín (Sn), stříbro (Ag), zlato (Au), nikl (Ni), olovo (Pb), hliník (Al), železo (Fe), kobalt (Co), platina (Pt) a další. Z hlediska výroby DPS je ale jednoznačně převažující právě měď. Měď se zpracovává do fólií s různou tloušťkou (např. 18, 35, 70 a 105 μm). Z konstrukčního hlediska se fólie 18 μm a nižší používá pro jemné spoje, 35 μm pro běžné aplikace a 70 μm , 105 μm a vyšší pro aplikace, které vyžadují větší proudovou zatížitelnost nebo zvýšenou odolnost vůči otřesům (např. automobilový průmysl). Pro speciální aplikace lze samozřejmě vyrobit i fólie tenčí či tlustší. Měděná fólie se vyrábí nejčastěji elektrolytickým pokovováním nerezového válce.



Obr. 1.3 Ilustrace výroby měděné fólie galvanickým pokovováním nerezového válce

Druhý způsob výroby měděných fólií je válcováním. Válcování je tváření materiálu průchodem mezi dvěma otáčejícími se válci válcovací stolice. Materiál se při průchodu přetváří a posunuje.



Obr. 1.4 Princip válcování materiálů; převzato z [4]

Výroba polotovaru desky plošných spojů se provádí následným plátováním měděných fólií na daný substrát. Plátování se provádí obdobně jako výroba základního substrátu lisováním desky substrátu společně s měděnou fólií. To vše za velmi vysokého tlaku a teploty. Výsledný polotovar takto plátované desky plošných spojů je tedy kombinací třech hlavních složek (výztuž, pojivo, měděná fólie). Doplněním speciálních přísad dokáže výrobce vytvořit takové typy DPS, které budou svými vlastnostmi vyhovovat širokému spektru aplikací.

1.1.2 Typy DPS

Desky plošných spojů se dělí podle dvou základních kritérií. Jednak podle struktury vodivých cest na:

- **Jednostranné DPS**
- **Oboustranné DPS**
- **Vícevrstvé DPS**

a dále podle použitých materiálů na:

- **Neohebné DPS**
- **Ohebné DPS**

Jednostranné desky plošných spojů jsou z jedné strany základního substrátu plátovány měděnou fólií. Tento typ desky se využívá především pro jednoduché obvody a podomácku vyráběná zařízení. Výroba jednostranných desek je poměrně levná, přesto se příliš nevyužívají z důvodu omezených možností návrhu složitějších obvodů. V takovém případě je třeba sáhnout po oboustranné desce.



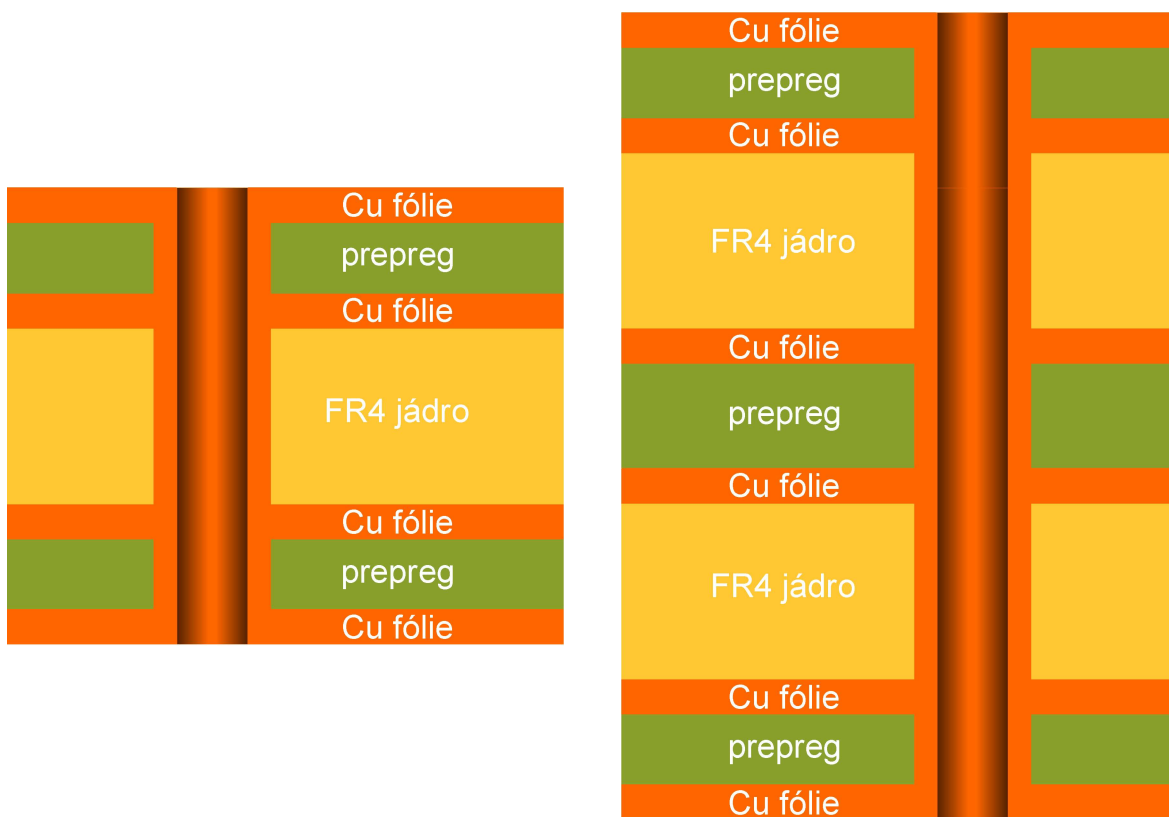
Obr. 1.5 Ilustrace jednostranné desky plošných spojů se dvěma otvory

Oboustranné DPS jsou více rozšířené. Využitím obou stran plošného spoje se docílí zmenšení celkového prostoru potřebného k realizaci obvodů. Například při použití dvoustranné desky společně s SMT technologií osazování součástek lze docílit až 70 % úspory plochy oproti jednostranným DPS s THT součástkami. Je samozřejmostí, že obě dvě technologie osazování součástek se dají kombinovat jak s jednostrannými, tak i s oboustrannými deskami. Spodní a vrchní strana dvoustranné desky se propojuje pomocí pokovených otvorů.



Obr. 1.6 Ilustrace oboustranné desky plošných spojů se dvěma pokovenými otvory

S postupem času, miniaturizací součástek a stále stoupajícími požadavky na nová elektronická zařízení přestaly být i oboustranné DPS s pokovenými otvory dostačující. Obvody s vysokou integrací součástek a čipy s desítkami vývodů (někdy i stovkami) přestaly být propojitelné. Například dnešní počítačové procesory mají až dva tisíce vývodů. Takto vysoká úroveň integrace si již vyžaduje vícevrstvé DPS.

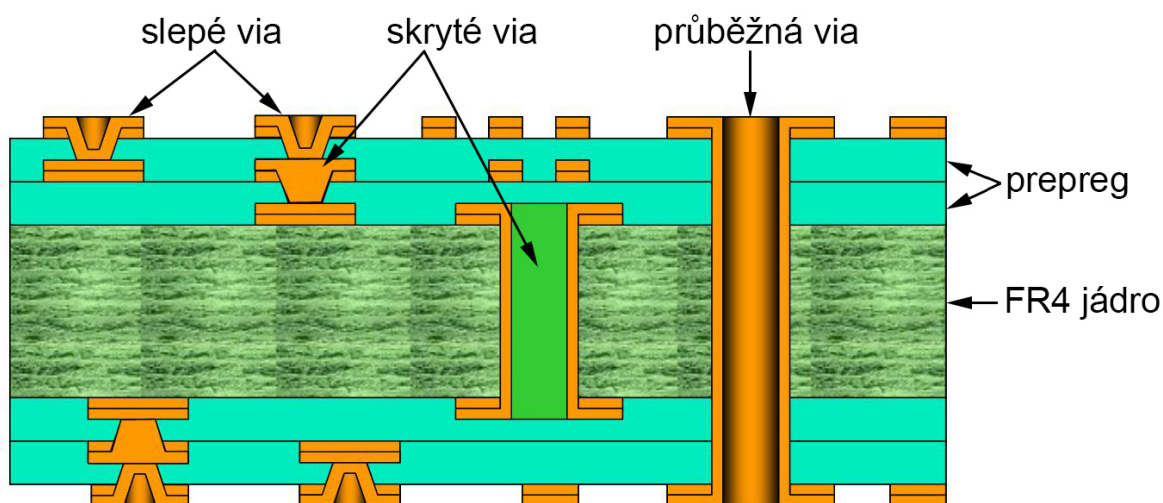


Obr. 1.7 Ilustrace vícevrstevých DPS; 4 vrstvy (vlevo), 6 vrstev (vpravo)

Základ vícevrstvé DPS tvoří tenčí oboustranná deska, avšak po vytvoření motivu plošných spojů se deska vrací zpět do lisu, kde se na ní nalisují další vrstvy prepregu a další měděná fólie. Propojení mezi vrstvami zprostředkovávají opět pokovené otvory. Takto vytvořených vrstev může být i několik desítek. Dnes se v základních deskách počítačů, grafických kartách a podobných komponentech můžeme běžně setkat až s 12

vrstvými DPS. Stejně tak jsou 12 vrstvé desky v nabídkách téměř všech firem zabývajících se výrobou DPS. Pro speciální a extrémně náročné aplikace se vyrábí i desky, které mají 38 nebo dokonce až 42 vrstev. Vícevrstvé DPS dovolují nahustit mnoho spojů na malý prostor a vytvářet velmi komplexní návrhy.

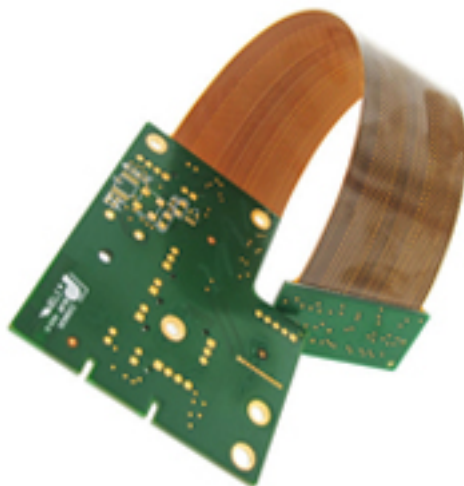
Mnohavrstvé desky plošných spojů s vysokou hustotou spojů (HDI) se také nazývají microvia. Vrstvy jsou propojeny pokovenými otvory nebo se do nepokovených otvorů vtírá vodivá pasta. Tyto otvory však nesmí být dle definice microvia širší než $150\ \mu\text{m}$. Otvory se dělí do tří skupin: skryté (buried), slepé (blind) a průběžné (through-hole). Skryté otvory vytvářejí propojení mezi dvěma a více vnitřními vrstvami. Slepé tvoří spojení mezi jednou vnější vrstvou a alespoň jednou vnitřní. Průběžné otvory procházejí všemi vrstvami (resp. celou deskou), avšak všechny vrstvy nemusí být s otvorem vodivě spojeny. Způsoby vytváření otvorů jsou například vrtání, ražení, laser, fotolitografie, chemické leptání (mokré), plazmové leptání (suché) a abraze (pískování).



Obr. 1.8 Ilustrace microvia desky; převzato a upraveno z [5]

Doposud byla řeč o neohebných DPS, avšak ani ty nemusejí být vždy ploché. Pro zvláštní případy využití lze během výroby desky tvarovat. S takovými deskami se ale téměř nesečkáme. Jsou náročnější na výrobu a kvůli tomu jsou také velmi drahé. Konstrukteři zařízení tak využívají spíše klasické rovné desky propojené různými konektory nebo se také využívají ohebné DPS. Ohebné desky jsou založeny na podobné bázi jako tuhé, s tím rozdílem, že neobsahují výztuže. Jako pojivo se nejčastěji využívají polyimidy nebo polyestery. Výhodou je samozřejmě jejich ohebnost a také nízká hmotnost.

Nevýhodami jsou těžko pokovitelné otvory a rozměrová nestálost. Ohebné DPS lze také vyrábět jako jednostranné, oboustranné nebo vícevrstvé. Pozitivní vlastností je také možnost kombinovat neohebné a ohebné DPS.

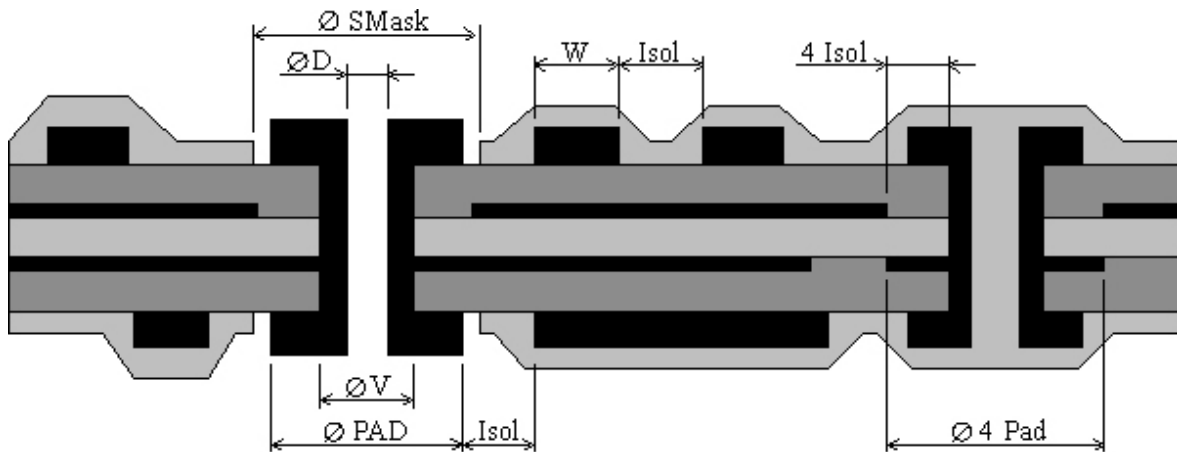


Obr. 1.9 Kombinace ohebné a neohebné DPS; převzato z [6]

Dalším kritériem pro dělení typů DPS jsou konstrukční třídy přesnosti. Mezi základní specifika tříd přesnosti patří: minimální šířka spojů, minimální izolační vzdálenost, minimální velikost pájecích plošek, minimální průměr vrtaných otvorů a minimální odstup nepájivé masky. V České Republice neexistují žádné platné státní normy definující výše zmíněná specifika. Proto je při návrhu plošného spoje třeba řídit se technologickými možnostmi výrobce DPS. Často se také můžeme setkat s údaji v milech namísto milimetrů. Mil je jednotka vycházející z imperiálního systému (1 mil = 0.001 inch = 0.0254 mm). Tříd přesnosti je celkem šest. Následující tabulka uvádí nejdůležitější položky ve třech nejpoužívanějších třídách přesnosti. Uvedené hodnoty jsou v milech. Význam jednotlivých položek je znázorněn na obr. 1.10. [7]

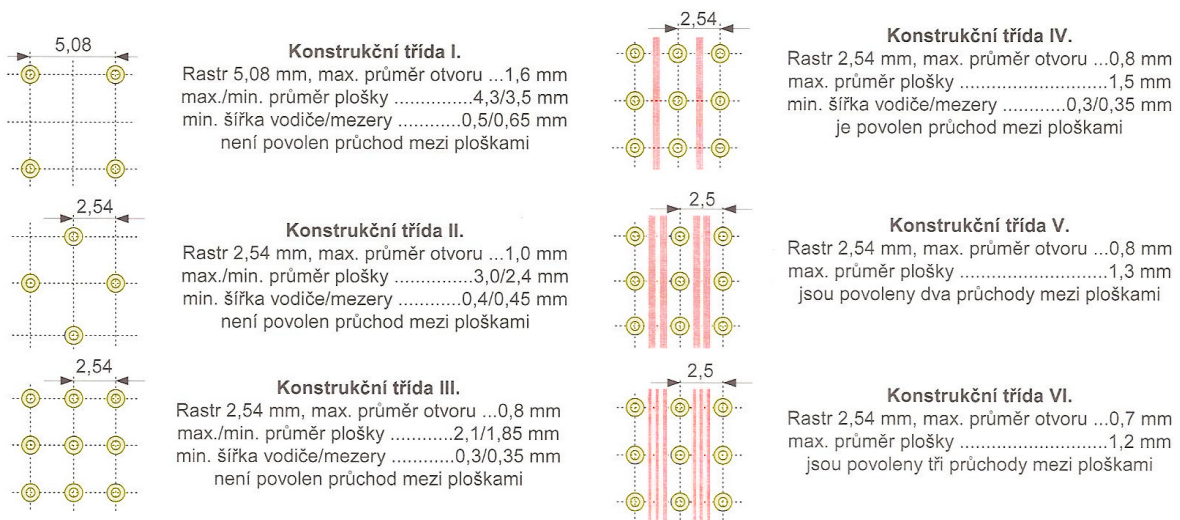
Tab. 1.1 Třídy přesnosti; převzato z [7]

Třída přesnosti	IV.	V.	VI.
Šířka cesty (W)	12	8	6
Izolační vzdálenost (Isol)	12	8	6
Průměr vrtáku (ØV)	28	20	16
Průměr pájecí plošky (ØPAD)	ØV + 24	ØV + 16	ØV + 12
Průměr nepájivé masky (ØSMask)	ØPAD + 10	ØPAD + 8	ØPAD + 6



Obr. 1.10 Vyznačení parametrů tříd přesnosti; převzato a upraveno z [7]

U vícevrstvých DPS se navíc uvádějí izolační vzdálenosti a průměry pájecích plošek ve vnitřních vrstvách (**4 Isol** a **Ø 4 Pad**). V závislosti na přesnosti sesazování jednotlivých vrstev DPS a přesnosti vrtání otvorů se tyto parametry zvětšují o 4 až 8 milů oproti standardním hodnotám. Průměr pájecích plošek a nepájivé masky je vztažen na průměr vrtáku ØV , ale výrobci se zpravidla zadává výsledný průměr otvoru ØD . Tento fakt musíme brát v úvahu již při návrhu a je-li to nutné, průměry ØPAD a ØSMask náležitě upravit. [7]



Obr. 1.11 Konstrukční třídy; převzato a upraveno z [7]

1.2 Technologie výroby DPS

V dnešní době jsou využívány tři metody výrobních postupů:

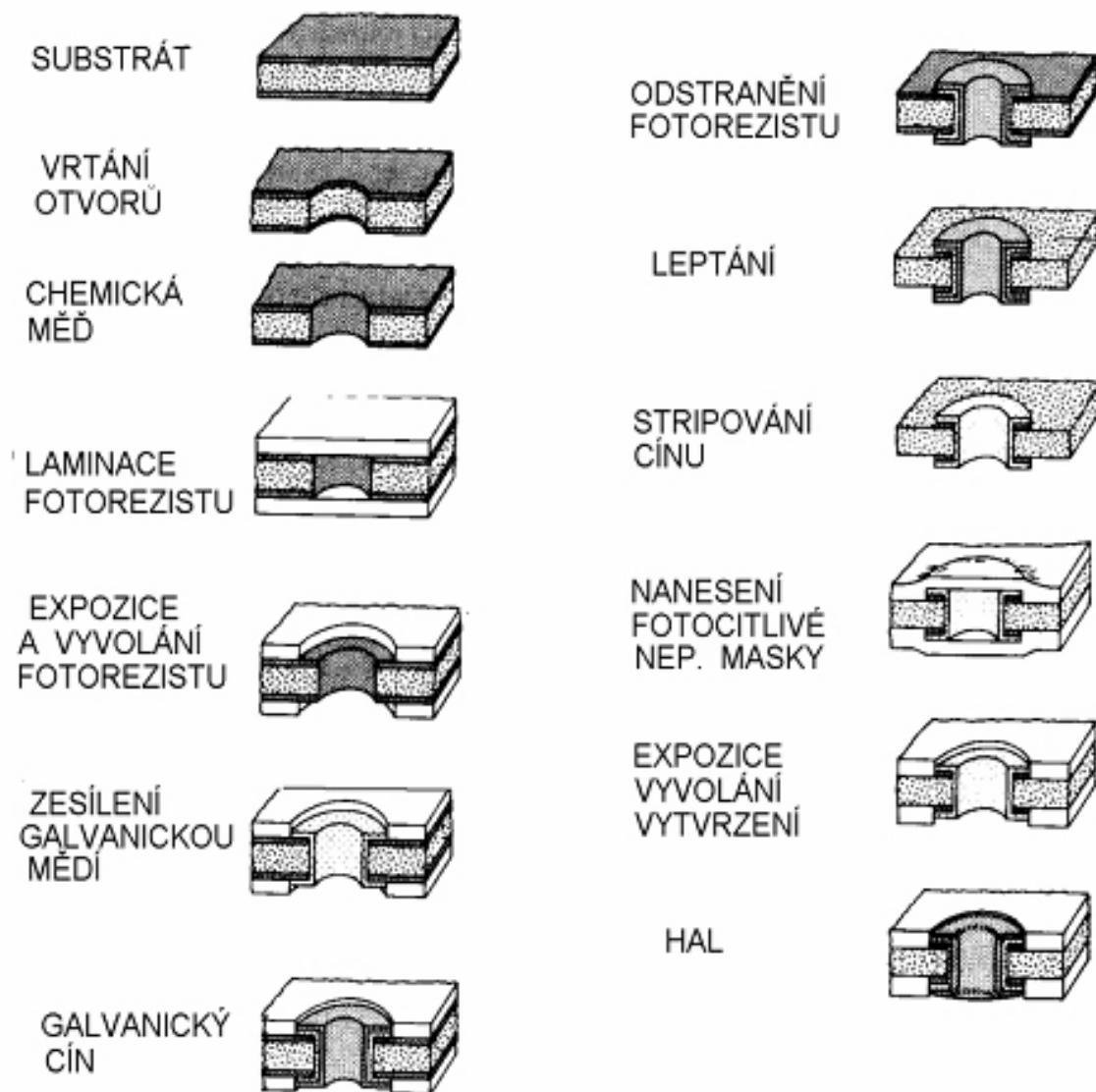
- *Subtraktivní*
- *Aditivní*
- *Semiaditivní*

Metoda subtraktivní spočívá v odstraňování přebytečné mědi z plátovaných substrátů, aditivní metodou se naopak nanáší vodivé cesty na neplátovaný substrát. Semiaditivní postup je kombinací obou předchozích metod.

1.2.1 Subtraktivní technologie

Subtraktivní technologie je známá téměř všem, kteří si někdy zkusili vyrobit desku plošných spojů. Jedná se o leptání základního plátovaného materiálu, přičemž požadovaný vodivý motiv je chráněn leptuvzdorným rezistem. Tento postup se vyznačuje nízkými výrobními náklady. Subtraktivní technologii se dále dělí do tří základních typů: Pattern-plating, pattern-plating s diferenčním leptáním a panel-plating.

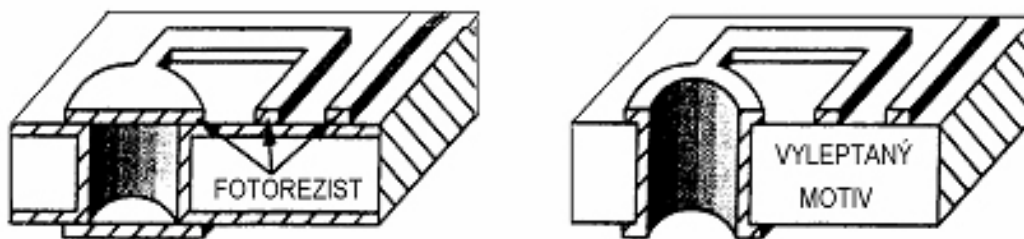
Pattern-plating je nejrozšířenějším typem subtraktivní technologie. Tento typ využívá pokovení požadovaného vodivého motivu leptuvzdorným rezistem. Motiv na desce je vytvořen fotocestou a jako rezist se nejčastěji používá Sn (cín) nebo Sn/Pb (cín s olovem). Po stripování fotorezistu se odhalená měď odleptá. Postup je znázorněn na obrázku 1.12.



Obr. 1.12 Znárodnění postupu pattern-plating; převzato z [8]

Pattern-plating s diferenčním leptáním je určen pro plátované materiály s velmi malou tloušťkou měděné fólie ($5\ \mu\text{m}$, $10\ \mu\text{m}$). Takto tenká vrstva je chráněna hliníkovou fólií s tloušťkou $40\ \mu\text{m}$. Hliníková fólie se před nanášením fotorezistu odstraňuje (mechanicky sloupne nebo odleptá v kyselině chlorovodíkové). [8]

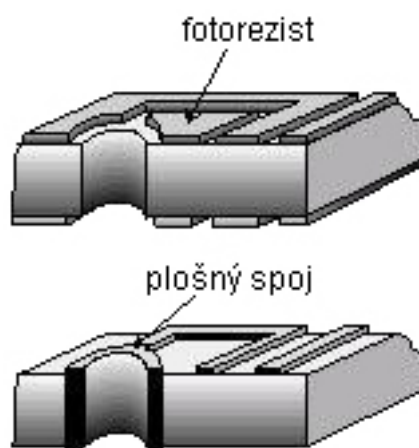
Panel-plating, také označován jako tenting, je postup, při kterém se celý plátovaný materiál včetně již vyvrtaných otvorů chemicky a galvanicky pokovuje mědí. Dále se fotocestou vytváří motiv DPS. Vodivé cesty a otvory zůstávají během leptání motivu kryty fotorezistem (viz. obr. 1.13).



Obr. 1.13 Vytváření motivu technikou panel-plating; převzato z [8]

1.2.2 Aditivní technologie

V případě aditivní technologie se používá neplátovaný základní materiál. Vodivé cesty se vytvářejí nanášením měděné vrstvy podle fotocestou nebo sítotiskem vytvořeného motivu (viz. obr. 1.14). Měď se pokovuje chemicky, bez přítomnosti elektrického proudu. Aditivní technologie přináší mnoho výhod, mezi které patří i nižší výrobní náklady. Velkou výhodou je, že odpadá proces leptání a problémy s leptáním spojené. Nedochozí například k přepleptání nebo nedopleptání motivu a odpadají potíže se zpracováním odpadních produktů (např. oplachových vod a rozpouštědel). Využití základního materiálu bez měděné fólie přináší další výhodou, kterou je nižší mechanické pnutí. Mechanické pnutí může způsobit prohýbání a zkroucení DPS. I přes nesporné výhody je aditivní technologie málo rozšířená. Tomu brání určité technologické problémy a omezení.



Obr. 1.14 Aditivní technologie; převzato a upraveno z [9]

1.2.3 Semiaditivní technologie

Upraveno podle [7][10][11].

Semiaditivní technologie je pravděpodobně tou nejrozšířenější technologií, kterou využívá většina výrobců desek plošných spojů. Touto metodou lze vyrábět jednostranné, oboustranné i vícevrstvé DPS. Základní postup je přiblížen na oboustranné desce s prokovenými otvory, nepájivou maskou a servisním potiskem.

Celý postup se skládá z následujících 19 hlavních kroků (které jsou dále podrobněji popsány):

1. *Zadání výroby*
2. *Technologický rozbor a příprava dat*
3. *Vykreslení filmových matic*
4. *Formátování základního materiálu*
5. *Vrtání*
6. *Prokovení otvorů*
7. *Laminace fotorezistu*
8. *Osvit motivu*
9. *Vyvolání negativního motivu*
10. *Galvanické zesílení mědi*
11. *Leptuvzdorný rezist*
12. *Odstranění negativního fotorezistu*
13. *Leptání*
14. *Odstranění leptuvzdorného rezistu*
15. *Testování*
16. *Nepájivá maska*
17. *HAL / Zlacení*
18. *Potisk*
19. *Formátování na výsledný rozměr*

Dále jsou také nejdůležitější kroky postupu znázorněny na obrázku 1.15. Na řezu deskou lze vidět, jak se postupně vytváří pájecí ploška, prokov a spoj.

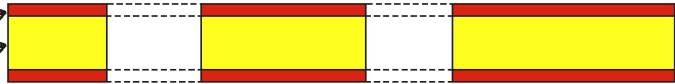
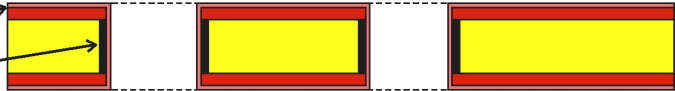
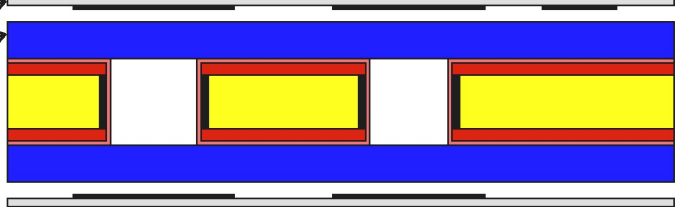
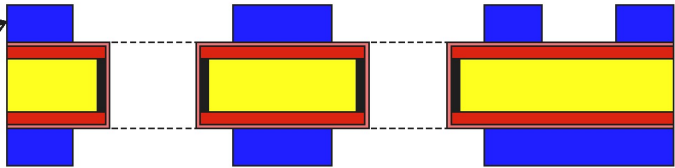
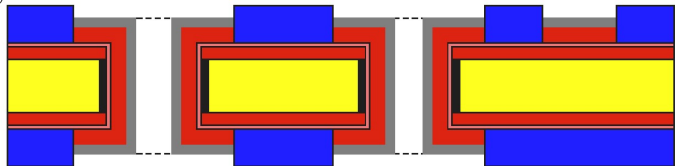
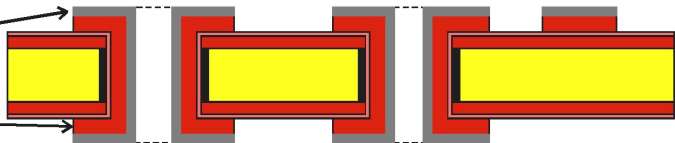
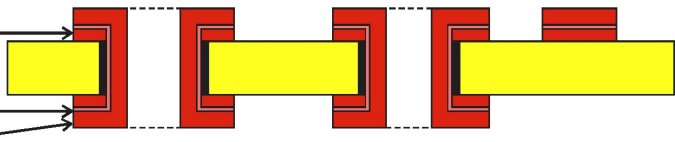
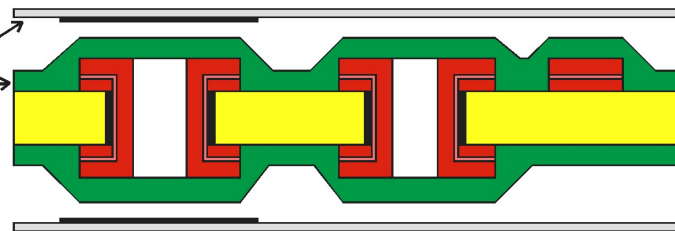
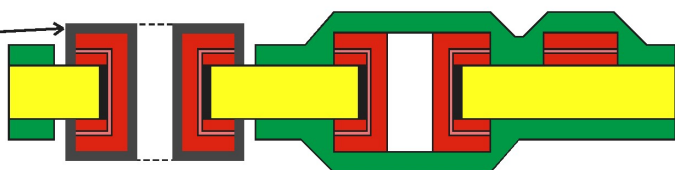
Prvním krokem je tedy zadání výroby. Spočívá v dodání výrobních podkladů, resp. technologických dat, výrobci a vyplnění objednávkového listu. Formát technologických dat je nutné předem dohodnout s vybraným výrobcem. Objednávkový list pak obsahuje přesnou specifikaci všech důležitých parametrů výsledného plošného spoje. Ukázky několika konkrétních objednávkových listů najdete v příloze B.

Před vlastní výrobou provádí výrobce technologický rozbor zakázky a upraví technologická data. Mezi základní operace v této fázi patří například kontrola, umístění sesazovacích značek, úprava ostříhových značek, umístění testovacích obrazců a vytvoření galvanické protiváhy. Tato galvanická protiváha je zhruba 2 cm široký prstenec okolo celé DPS, který napomáhá homogenizaci proudových toků během galvanických procesů.

Dalším krokem je vykreslení filmových matic a výrobních kopií. Filmové matrice se vykreslují na fotoplotru. Fotoplotr je zařízení, které pomocí laseru vykresluje požadovaný motiv na fotocitlivou fólii. Tloušťka této fólie je 7 milů (0.18 mm) a vyznačuje se vysokou rozměrovou stálostí. Motiv je vykreslován s přesností 1-10 μm . Výše zmíněné výrobní kopie se vytvářejí z matic osvitem na fotocitlivý materiál (transparentní film, který osvitem zčerná) a používají se především při sériové výrobě.

Prvním krokem přímých prací na výrobě desky plošného spoje je formátování základního materiálu. Jako základní materiál se u semiaditivní technologie používá oboustranně plátovaný nosný substrát (nejčastěji FR-4). Formátování zahrnuje nastříhání plátované desky na předem daný rozměr, obroušení hran po stříhání a vrtání technologických otvorů pro uchycení desky při některých výrobních operacích.

Po formátování je na řadě vrtání otvorů na počítačem ovládané souřadnicové vrtačce, vyčištění otvorů a odstranění otřepů po vrtání (obr. 1.15a). Pro uchycení vrtáků se používají vysokootáčková vřetena (20.000 – 150.000 ot./min). Polohování vřetena je řízeno s přesností až 10 μm . Používají se vrtáky ve škále průměrů od 0.4 mm až po 6.4 mm. Otvory větších průměrů se frézují. Vrtačka sama je schopna samostatně měnit vrtáky připravené v zásobnících. Po vyvrtání otvorů je nutné desku připravit k prokovení otvorů. Kartáčováním se začistí otřepy děr a dále je třeba odstranit z povrchu desky mastnoty a další nečistoty.

a) VrtáníZákladní plátování - měďNosný materiál**b) Prokovení otvorů**Galvanická měď 1Palladium**c) Laminace fotorezistu a osvit**Film s motivem spojůFotorezist**d) Vyvolání negativního motivu**Fotorezist - negativní motiv**e) Galvanické zesílení mědi a leptuvzdorný rezist****f) Odstranění negativního fotorezistu**Galvanický rezist - cínGalvanické zesílení - měď 2**g) Leptání a odstranění Sn rezistu**Základní plátování - měďGalvanická měď 1Galvanické zesílení - měď 2**h) Fotocitlivá nepájivá maska a osvit**Film s motivem nepájivé maskyNepájivá maska**i) Žárové nanesení SnPb pájky (HAL)**SnPb pájka (HAL)

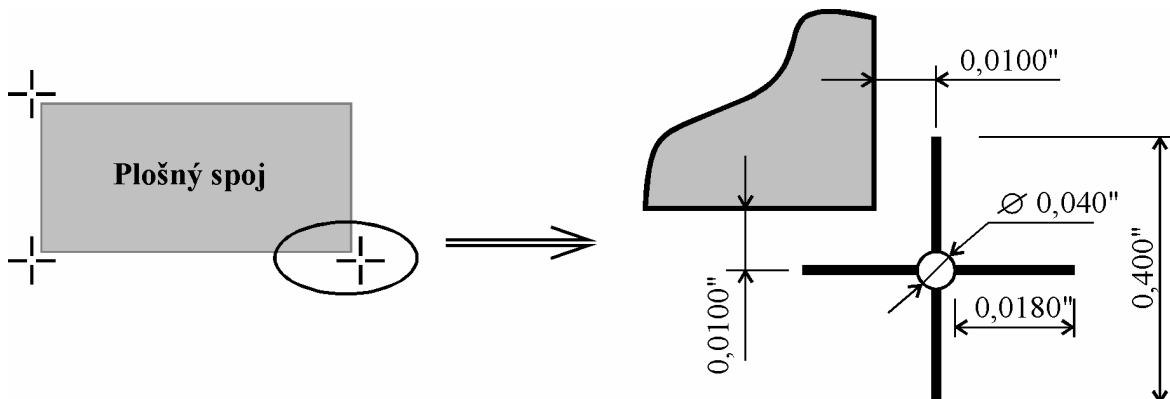
Obr. 1.15 Znárodnění postupu semiaditivní technologie při výrobě dvoustranné DPS;
převzato a upraveno z [7]

Prokovení otvorů se provádí metodou tzv. přímého prokovu. Tato metoda se skládá z několika kroků. Nejdříve je nutné vyvrtané otvory chemicky vyčistit a zároveň narušit strukturu povrchu stěn otvoru. Do takto připraveného otvoru se v katalyzační lázni elektrostaticky nanese $0.1 \mu\text{m}$ tlustá vrstva palladia. Tím se vodivě propojí obě strany DPS ve všech vyvrtaných otvorech. Nakonec se galvanicky nanese 6 až $8 \mu\text{m}$ tlustá vrstva mědi (obr. 1.15b).

Dále se provádí laminace fotorezistu, osvit motivu a vyvolání negativního motivu (obr. 1.15c, 1.15d). Fotorezist je $38 \mu\text{m}$ tlustá fólie z fotocitlivého polymeru, která se laminuje laminátorem na připravenou desku. Místo fotocitlivé fólie se také využívá tekutý fotorezist (především při sériové velkovýrobě). Nanášení tekutých rezistů se provádí navalováním, clonovým litím, sítotiskem, odstředěním, ponorem anebo stříkáním. Existují dva druhy fotocitlivých emulzí, podle kterých se také liší tvorba předlohy. Tyto dva druhy jsou pozitivní a negativní. Obě tyto emulze jsou citlivé na osvětlení. Struktura pozitivní emulze se osvětlením narušuje a ty části desky (resp. emulze), které jsou osvětleny, se následně odplaví ve vývojce. Filmová matrice je tedy shodná s výsledným motivem plošného spoje. Negativní rezist funguje přesně naopak, jeho struktura se osvětlením vytvrzuje a při vyvolání se odplaví neosvětlené části. Předloha se tedy vyrábí tak, že tmavá místa zakrývají vše kromě vodivých cest. Použití negativních emulzí má oproti pozitivním výhody. Nanášení emulze není tak náchylné na čistotu a je tedy menší riziko vzniku trhlin na vodivých cestách. Dále také při osvětlení redukuje jev podsvícení. Podsvícení je negativní jev, kdy světlo nedopadá kolmo na filmovou matrici, což ovlivňuje výsledné rozměry motivu.

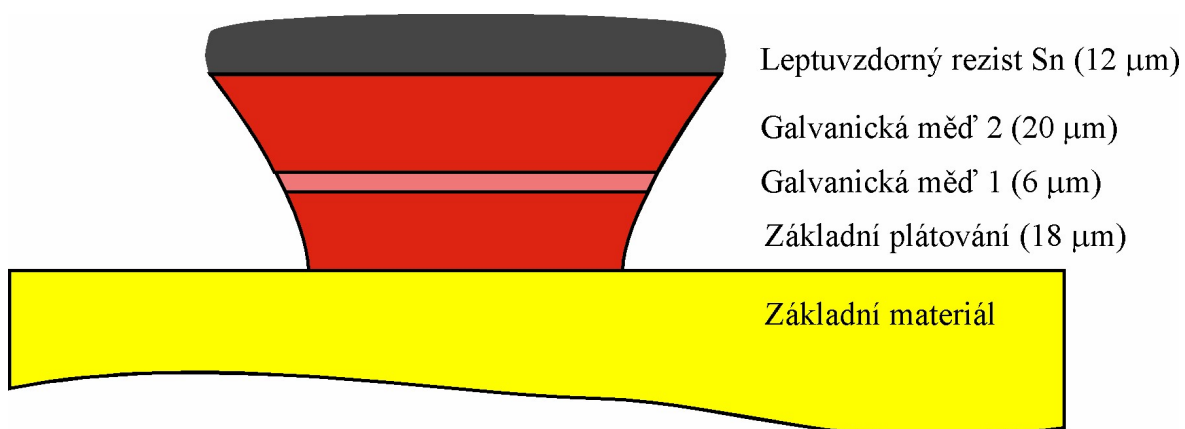
Na takto připravenou desku s naneseným fotocitlivým materiálem se poté přiloží film s motivem a provede se osvit. Jako zdroj záření se používají výkonné UV lampy, případně LED diody, s vlnovou délkou záření $300\text{-}400 \text{ nm}$, na které jsou fotocitlivé materiály nejcitlivější. Film je během osvitů k desce fixován vakuovým rámem, což zajišťuje důkladné přilnutí filmové matrice k povrchu fotorezistu a zabraňuje tak podsvícení. Tato operace je velmi citlivá na čistotu prostředí, částice o velikosti $10 \mu\text{m}$ se již jeví jako nečistoty. Dále je také samozřejmě nutné přiložit film s motivem na již vyvrtanou a prokovenou desku s co nevyšší přesností. K přesnému sesazení slouží právě sesazovací značky, které se umisťují v rozích plošného spoje (obr. 1.16). Osvícený fotocitlivý rezist se

následně vyvolá v 1-2% roztoku hydroxidu sodného (NaOH) (tzv. vývojka). V tomto stádiu jsou na desce odhaleny budoucí cesty plošného spoje a zbylá část desky je zakryta fotorezistem.



Obr. 1.16 Umístění a ukázka sesazovacích značek; převzato z [7]

Na odhalených místech se následně provede galvanické zesílení mědi a nanese leptuvzdorný rezist (obr. 1.15e). Tloušťka galvanického zesílení mědi je typicky $20 \mu\text{m}$. Jako leptuvzdorný rezist se používá $12 \mu\text{m}$ silná vrstva cínu nanášená taktéž galvanicky. Dalším krokem je odstranění fotorezistu (tzv. stripování) pomocí organických rozpouštědel (ethanol, aceton). Deska je tak připravená k leptání (obr. 1.15f). Výhodou semiaditivní metody je, že výsledná tloušťka spojů je větší než tloušťka odleptávaná. Galvanické zesílení mědi se totiž provádí pouze na fotorezistem odkrytých místech, resp. v místech budoucích spojů, pájecích plošek a prokovů. Při leptání dochází také k negativnímu jevu podleptání spojů (obr. 1.17).



Obr. 1.17 Ukázka reálného profilu leptaného spoje; převzato a upraveno z [7]

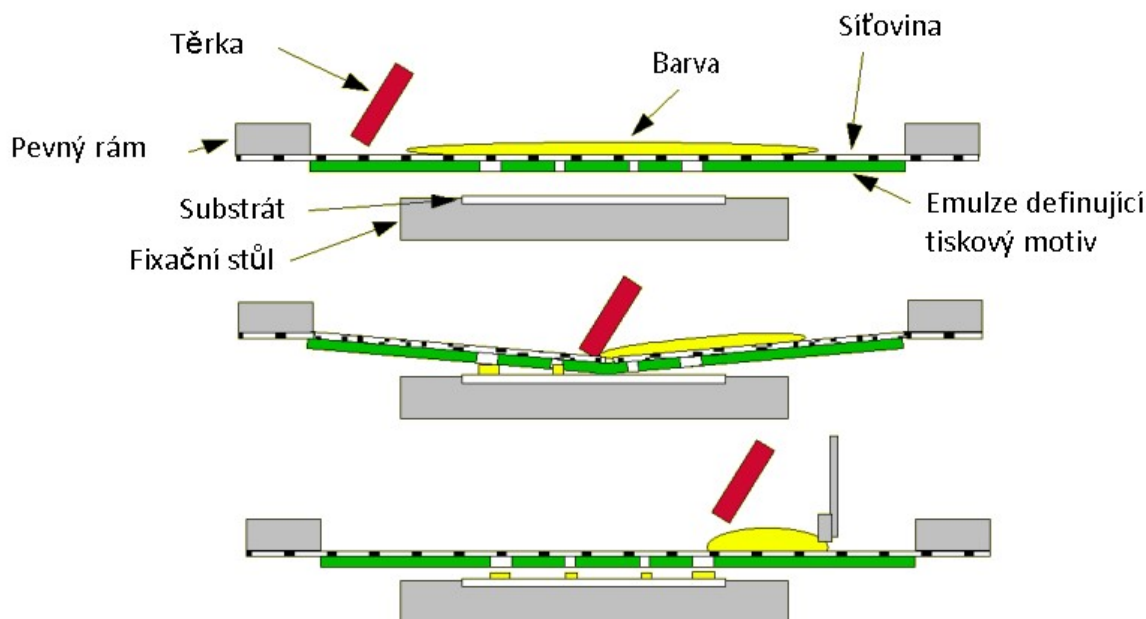
Na obrázku 1.15g lze vidět výsledek leptání a následného odstranění cínového leptuvzdorného rezistu. V této fázi je deska připravena k testování. Testovat lze dvěma způsoby – elektricky nebo opticky. Optický tester scanuje povrch DPS a tyto snímky porovnává a zjišťuje odchylky od referenčními dat. Elektrický tester měří odpor mezi definovanými místy na DPS. Využívají se dva druhy elektrických testerů – statické pole jehel a nebo pohyblivé individuální jehly. Dále se také využívá rentgenového testeru, kterým je možné odhalit závady ve vnitřních vrstvách desky.

Jakmile je testování dokončeno, nanese se na očištěnou desku fotocitlivý nepájivá maska, přiloží se odpovídající film s motivem a provede se osvit (obr. 1.15h). Neexponovaná místa jsou vymyta ve vyvolávacím zařízení a maska je následně tepelně vytvrzena. Nepájivá maska chrání měděné spoje před vnějšími vlivy, chrání motiv před nežádoucím zkratováním a také zakrývá místa, na které nemá být v následujícím kroku nanesena SnPb pájka.

Po vytvrzení nepájivé masky se provádí žárové nanesení SnPb pájky. Tato operace se také nazývá HAL, resp. HASL (Hot Air Leveling, resp. Hot Air Solder Leveling). Postup je následující: celá deska se ponoří do tavidla a poté na několik málo vteřin do nádoby s roztavenou SnPb pájkou. Při vynořování se přebytky pájky odfouknou horkým vzduchem. Takto se na nepájivou maskou odhalených místech vytvoří přibližně 8 μm silná vrstva SnPb pájky. Pájka chrání pájecí plošky před klimatickými podmínkami (cín je za normálních podmínek značně odolný vůči korozi) a také zaručuje snadnou pájitelnost spojů. Namísto SnPb pájky nabízejí výrobci také možnost zlacení. Zlacení lze provádět jak chemicky, tak i galvanicky. Avšak před nanášením zlata se ještě provádí niklování. Jelikož je galvanicky vytvořená vrstva mědi velmi porézní, znamenalo by to velikou spotřebu zlata. Proto se z ekonomických důvodů nejdříve nanáší nikl a až poté zlato. Při chemickém pokovení se nanáší 1 μm niklu a 0.1 μm zlata. Při galvanickém pokovení pak 5 μm niklu a 1 μm zlata. Výhodou zlata je, že nevytváří oxidy a je tedy vhodný jako ochrana při dlouhodobém skladování desek anebo pro zlacení přímých konektorů. Výroba DPS je v této fázi téměř u konce, následovalo by již jen formátování na výsledný rozměr.

Často ale požadujeme z důvodu přehlednosti, aby byly jednotlivé pozice součástek na desce nějak označeny, nebo abychom měli nějak označenou desku samotnou (např.

k čemu bude deska sloužit, verze desky, datum návrhu, apod.). Toto je zajištěno takzvaným servisním potiskem (viz. obr. 1.1). Servisní potisk je zpravidla bílá dvousložková epoxidová barva, která se na DPS nanáší nejčastěji sítotiskem. Minimální možná tloušťka potiskové čáry je 0.12 nm , což je třeba zohlednit již během návrhu. Při návrhu je také nutné dbát na to, aby žádné části servisního potisku nezasahovaly do pájecích plošek. Po nanesení na desku se barva tepelně vytvrzuje.



Obr. 1.18 Znáznornění principu sítotisku

Posledním krokem výroby je formátování desky na výsledný rozměr. Lze postupovat třemi způsoby. Prvním postupem je ostříh na padacích nůžkách. Je to nejjednodušší a nejrychlejší postup, ale zároveň je také nejméně přesný ($\pm 0.25\text{ mm}$). Ostříh se provádí podle ostříhových značek. Na okrajích desky zůstávají po ostříhu otřepy skelného laminátu, které je vhodné zabrousit. Z důvodu přesnosti není ostříh vhodný pro desky, které mají být zasunuty do přesných drážek. Dále je kvůli mechanickému namáhání materiálu při ostříhu vhodné dodržet minimální vzdálenost vodivého motivu od okraje desky alespoň 2.5 mm .

Druhým postupem je frézování. Frézování se zpravidla provádí na stejném stroji jako vrtání, pouze s tím rozdílem, že namísto vrtáku se do vřetena umísťuje frézka o průměru $1.5\text{ až }2.5\text{ mm}$. Přesnost frézování je 0.1 mm a minimální vzdálenost motivu od okraje desky je 1.5 mm . Frézování se také používá při nutnosti vytvoření otvorů o větším průměru

než 6.4 mm (největší používaný průměr vrtáku) a dále také při vytváření nepravidelných otvorů a výřezů.

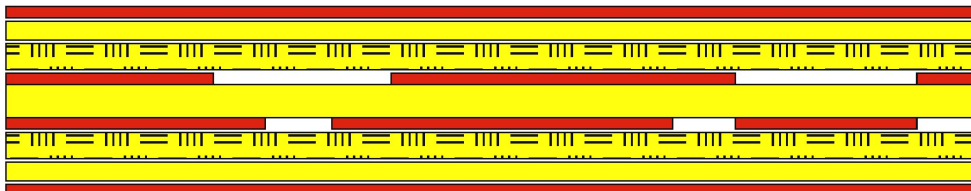
Třetím postupem je drážkování, které je vhodné při výrobě velkého množství malých desek vedle sebe a které navíc mají být osazovány a pájeny strojně. Drážkováním je myšleno oboustranné naříznutí obrysu DPS, přičemž tyto malé desky zůstávají stále držet pohromadě jako jedna velká deska. Naříznutí ztenčí desku na 0.4 mm , šířka drážky je 0.8 mm . Po osazení a zapájení komponent se tyto malé jednotlivé desky snadno oddělí rozříznutím. Minimální vzdálenost motivu od okraje desky je 2.5 mm .

Tímto je proces výroby DPS semiaditivní metodou u konce. Výroba vícevrstvých desek je odvozena od postupu pro oboustranné desky. Obecně je možné vícevrstvé DPS vyrábět buď postupným vrstvením prepregu a měděných fólií na základní oboustrannou desku nebo vrstvením dílčích tenkých oboustranných desek. Princip výroby vícevrstvého (v tomto případě čtyřvrstvého) plošného spoje je znázorněn na následujícím obrázku.

a) Oboustranně vyleptané jádro



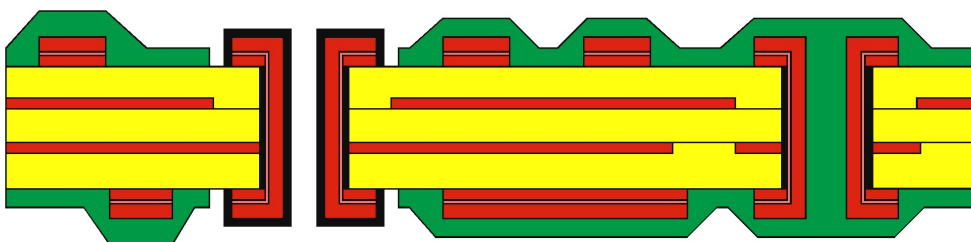
b) Vrstvení fólií nevytvrzených laminátů a mědi



c) Laminace



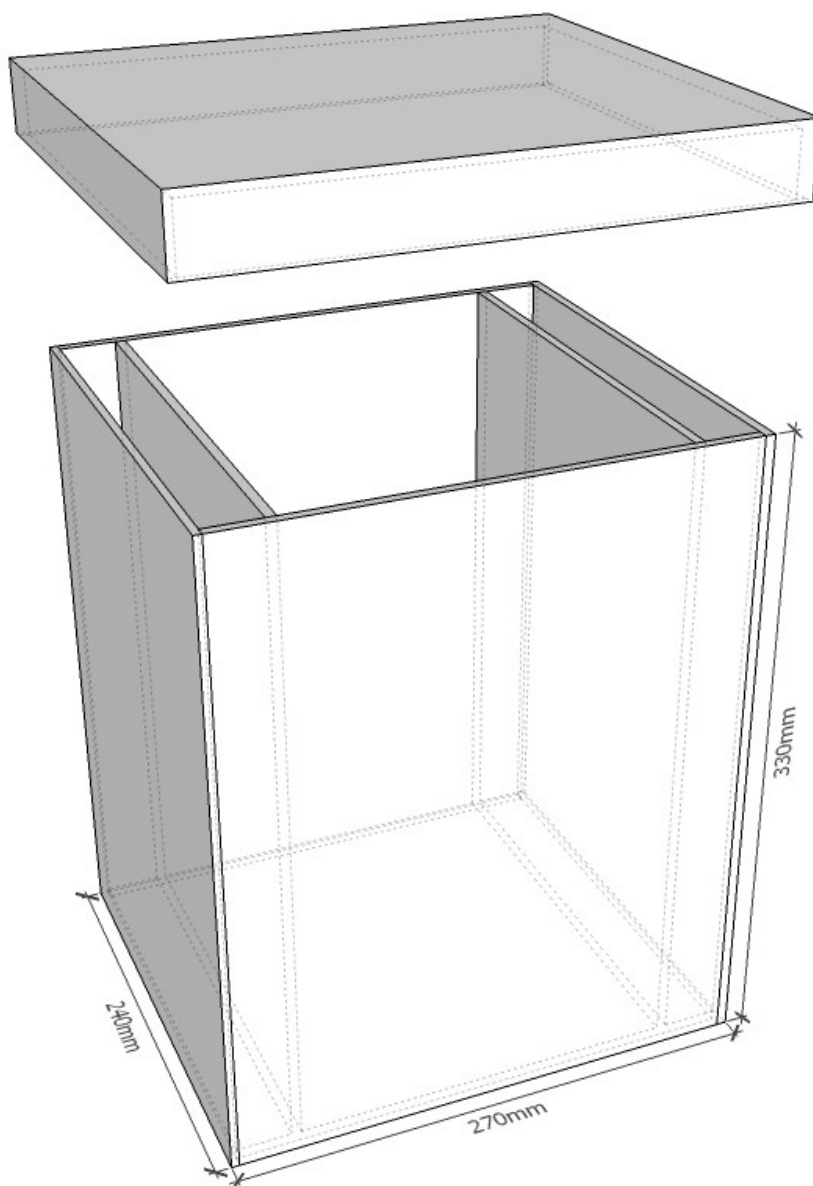
d) Finální výrobek



Obr. 1.19 Princip postupu výroby čtyřvrstvé desky; převzato a upraveno z [7]

2 Návrh a realizace kyvety

Prvním krokem při návrhu kyvety bylo rozmyslet si, jaké všechny součásti má obsahovat, jak velké tyto součásti jsou, jaké rozměry desek se budou vyrábět a jaké jsou materiálové možnosti. Základním požadavkem na tuto leptací stanici je tříkomorová kyveta. Jedna komora slouží k leptání motivu DPS, další k vyvíjení fotocitlivého motivu DPS a poslední komora je určena pro oplachovou vodu. Tuto základní myšlenku návrhu zobrazuje následující obrázek.



Obr. 2.1 Návrh tříkomorové leptací kyvety

Na obrázku lze vidět již zmíněné tři oddělené komory. Dvě užší pro leptací a vyvíjecí roztok a jedna větší uprostřed pro oplachovou vodu. Důvody pro zúžení postranních komor byly ekonomické. Jak leptací, tak vyvíjecí roztok nejsou příliš levnou záležitostí, proto je zúžení komor logickým krokem. Jelikož je nutné při leptání ponořit desku do leptacího roztoku kompletně, je v užší komoře zapotřebí méně roztoku pro vyleptání větších desek, než by bylo v komoře širší. Dalšími důvody jsou také efektivnost využití koncentrovaného leptacího roztoku a rychlost ohřevu. Nejen při výběru materiálu, ze kterého je kyveta vyrobena, ale i při výběru dalších součástí kyvety je nutné brát zřetel na to, že se bude pracovat s roztokem chloridu železitého. Chlorid železitý totiž reaguje téměř se všemi kovy.

Jednou z možností materiálu pro výrobu kyvety je klasické sklo, jehož nákupní cena je sice nižší, avšak zpracování skla je díky jeho křehkosti a amorfnímu charakteru výrazně technologicky náročnější. Další možností jsou plasty. Pro materiál kyvety byl nakonec zvolen plný polykarbonát, což je jeden z druhů plexiskla. Oproti klasickému plexisklu (polymetylmetakrylát) je polykarbonát měkký, houževnatý a odolný vůči nárazům. Díky těmto vlastnostem je polykarbonát netříštivý, ohebný, je lépe zpracovatelný a nepraská. Jeho měkkost je ale zároveň také jeho nevýhodou, povrch polykarbonátu je poměrně snadno poškoditelný. Případné rýhy a škrábance jsou pak velmi obtížně odstranitelné.

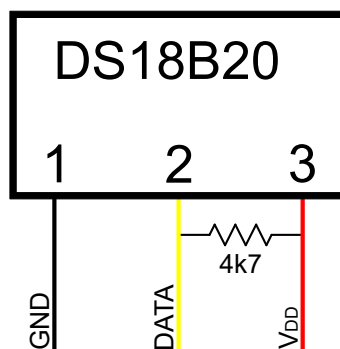
Dle návrhu byly na zakázku nařezány jednotlivé desky polykarbonátu s příslušnými rozměry. Další práce (vytváření otvorů a lepení desek) byly již prováděny samostatně, případně za pomoci vedoucího práce. Desky byly lepeny speciálním rychleschnoucím lepidlem, určeným přímo pro lepení polykarbonátu. Před lepením musí být povrch čistý, zbavený prachu, mastnoty, vlhkosti a dalších nečistot. Lepidlo povrch plexiskla naleptává a tak je spoj po vytvrzení velmi pevný.

2.1 Komponenty stanice

Na předchozím obrázku (obr. 2.1) nejsou zobrazeny žádné další součásti stanice. V přední části každé z komor byly vytvořeny otvory pro vypouštěcí kohouty. Ve stěně jedné z úzkých komor (v té, kterou jsme si určili jako leptací) byly vytvořeny další dva otvory pro dvě průchodky, mezi které je vloženo cirkulační čerpadlo (obrázky kohoutů a průchodek jsou součástí přílohy B). Dále byly v horní části kratších stěn leptací komory vyhloubeny zářezy pro plastový rám, který zajišťuje uchycení leptané DPS v leptací komoře. Posledním zásahem do konstrukce kyvety bylo vyříznutí otvoru v rohu víka z důvodu vyvedení napájecích a signálových vodičů všech komponentů stanice.

K průchodkám a kohoutům byly také zakoupeny příslušné matice a gumové těsnící podložky. Cirkulační čerpadlo (viz. obrázek v příloze B) slouží k cirkulaci leptacího roztoku v komoře. To zajišťuje rovnoměrné leptání celé plochy leptané desky a zároveň zajišťuje rovnoměrnou teplotu v celém objemu leptacího roztoku. Čerpadlo může být napájeno napětím až 12 V, při kterém byl naměřen odběr 355 mA. Avšak při těchto hodnotách je průtok čerpadlem až zbytečně vysoký. Proto bylo zvoleno provozovat jej při napětí 5 V s odběrem 165 mA. Při těchto hodnotách je proud leptacího roztoku slabší a postačující. Zároveň je výhodou, že i všechny ostatní zařízení stanice (kromě topného tělesa) pracují na napěťové hladině 5 V a tudíž není zapotřebí přidávat další napájecí zdroj.

Dalšími komponenty leptací komory jsou teplotní senzor a topné těleso. Jako teplotní senzor je použit digitální teploměr firmy Dallas Semiconductor, typ DS18B20, zapouzdřený v malé kovové patroně (viz. obr. 3 v příloze B). Jelikož je senzor zapouzdřen v kovové patroně, bylo nutné zajistit, aby se patrona nedostala do kontaktu s leptacím roztokem. Tento problém byl vyřešen teplem smršťovací bužírkou s lepidlem. Senzor DS18B20 umožňuje měření teploty v rozsahu od $-55\text{ }^{\circ}\text{C}$ až do $125\text{ }^{\circ}\text{C}$, přičemž v rozsahu $-10\text{ }^{\circ}\text{C}$ až $85\text{ }^{\circ}\text{C}$ s garantovanou přesností $\pm 0.5\text{ }^{\circ}\text{C}$, což je pro toto použití dostačující. Senzor je napájen napětím 5 V, má nastavitelné rozlišení v rozmezí 9 až 12 bitů, má tři vývody z pouzdra a pro komunikaci s mikrokontrolérem potřebuje pouze jeden signálový vodič. Další dva vodiče jsou určeny pro napájení a zem. Při zapojení je pak nutné přidat mezi napájecí a datový vodič $4.7\text{ k}\Omega$ pull-up rezistor (viz. obr. 2.2).



Obr. 2.2 Zapojení teplotního senzoru DS18B20

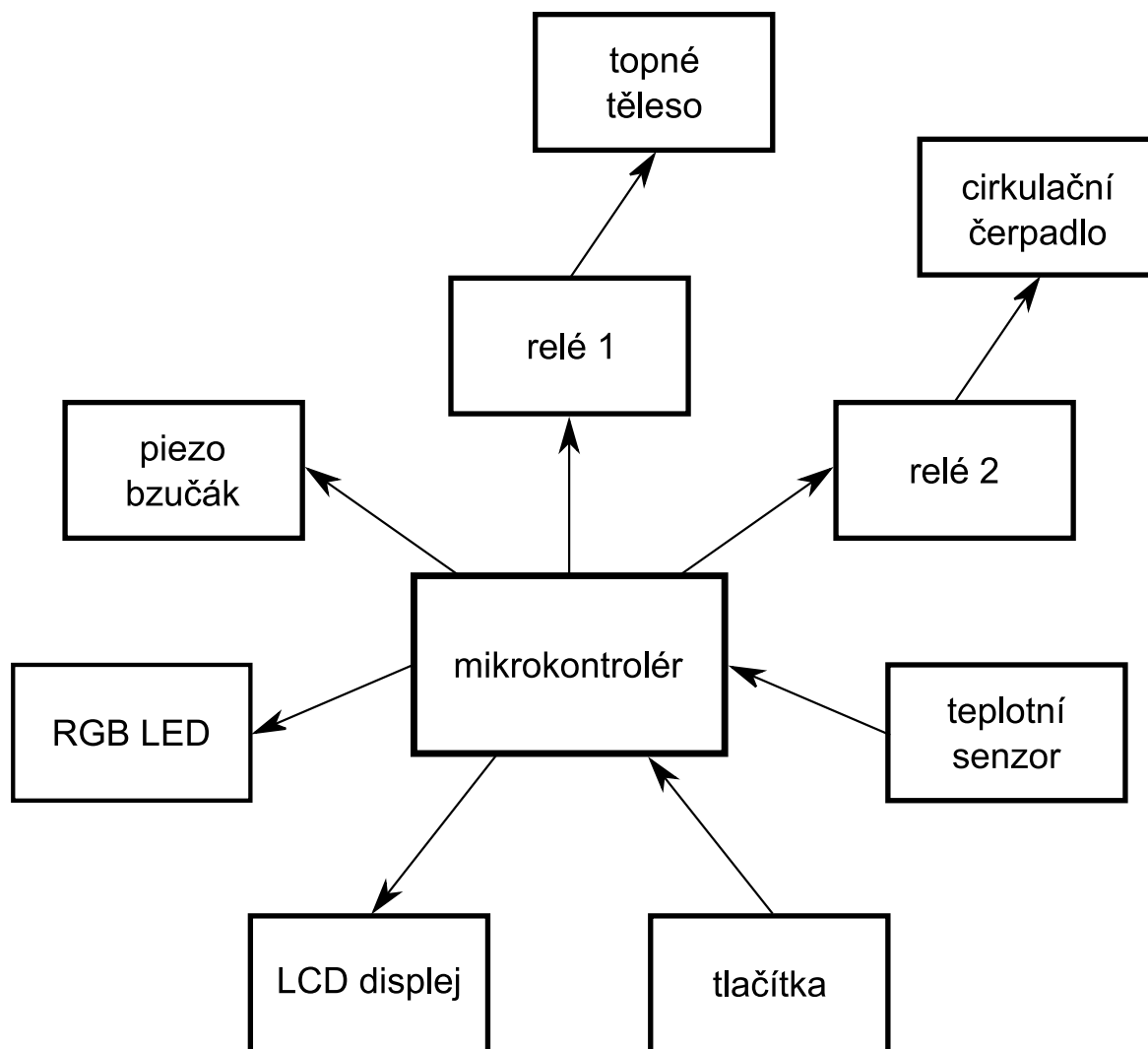
Pro vytápění leptací komory je použito elektrické topné těleso s příkonem 200 W. Těleso má tvar válce, je napájeno střídavým napětím 230 V a je umístěno přímo uvnitř leptací komory. Z toho důvodu bylo také nutné zvolit těleso zapouzdřené v materiálu, který odolá chloridu železitému. V úvahu připadly dvě varianty. Keramická topná patrona nebo keramické topné těleso (nejčastěji keramické jádro s navinutou topnou spirálou) v uzavřené skleněné silnostěnné zkumavce. Důležitým kritériem při výběru byl průměr, jelikož kolem topného tělesa musí proudit ohřívaná kapalina a to uvnitř 2.75 cm široké leptací komory.

3 Návrh a realizace řídicího systému

Řídicí systém slouží k ovládání všech komponentů stanice. Jádrem systému bude mikrokontrolér, který bude zajišťovat jak samotné řízení, tak i zprostředkování uživatelského rozhraní. Uživatelské rozhraní bude sloužit k získání informací a požadavků uživatele. Uživatel bude mít například možnost zvolit si teplotu udržovanou v leptací komoře kyvety. Při výběru mikrokontroléru, resp. vývojové desky, se naskýtá mnoho různých možností. Mezi ně patří například i firmy jako ST Microelectronics nebo Raspberry PI. My jsme zvolili desku Arduino, založenou na mikrokontrolérech ATmega firmy Atmel. Důvod, proč padla volba právě na platformu Arduino, je jeho jednoduché a jednoznačné použití a také jednoduché vývojové prostředí (Arduino IDE).

Díky těmto vlastnostem se platforma Arduino těší velké oblibě, je velmi rozšířená a přitahuje stále více lidí, což se samozřejmě odráží v kvalitě a rozsahu dostupné podpory při řešení nejrůznějších příkladů a problémů. To dále napomáhá ke zdokonalování a vývoji nových nejrůznějších typů vývojových desek. Platforma Arduino je navíc takzvaným open-source projektem. To znamená, že veškeré hardwarové a softwarové nástroje jsou veřejně dostupné, a tak mohou lidé samotní tyto nástroje či vývojové desky rozšiřovat a vylepšovat. To je také důvodem ke vzniku nesčetného množství klonů těchto desek. Již relativně levné originální Arduino se tak prostřednictvím těchto klonů stává ještě více dostupným, rozšířenějším a levnějším.

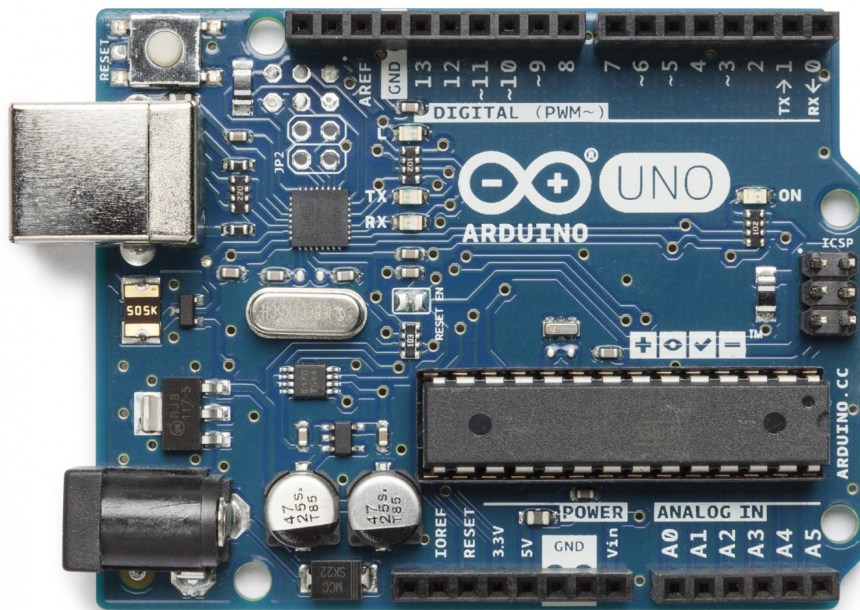
Hlavním kritériem pro výběr konkrétní vývojové desky pro naši stanici je množství součástí, které je třeba k mikrokontroléru připojit. Resp. kolik vstupních/výstupních pinů má konkrétní deska, aby se dalo vše připojit. Je potřeba připojit teplotní senzor, cirkulační čerpadlo, topné těleso, LCD displej, čtyři tlačítka, piezoelektrický bzučák a tři jednotlivé barvy pásku RGB LED diod. Jelikož jakýkoliv typ desky Arduina dokáže na jeden svůj vývod dodávat maximálně 40 mA a cirkulační čerpadlo potřebuje 165 mA, a jelikož je topné těleso napájeno střídavým napětím přímo ze zásuvky, není možné tyto dvě součásti připojit k mikrokontrolérové desce přímo. Namísto toho je spínání těchto prvků řešeno pomocí dvou oddělených elektromagnetických relé, které lze již spínat přímým signálem z desky Arduina. Kompletní návrh blokového schéma zapojení zobrazuje následující obrázek.



Obr. 2.3 Blokové schéma řídicího systému

Obě relé, teplotní senzor a piezo bzučák potřebují každý po jednom pinu. RGB LED tři piny a pro každé ze čtyř tlačítek také po jednom pinu. Komunikace s LCD displejem probíhá po sériovém kanálu I²C, který využívá pouze dvou vodičů. Jeden pro přenos synchronizačních impulzů a druhý pro přenos sériových dat. Celkem je tedy za potřeby třináct vstupních/výstupních digitálních pinů. K tomuto účelu nám plně vyhovuje vývojová deska Arduino Uno v aktuální třetí revizi.

3.1 Arduino Uno Rev3



Obr. 3.1 Vývojová deska Arduino Uno Rev3; převzato z [12]

Arduino Uno je dnes pravděpodobně nejznámější a nejčastěji využívaný typ desky. Je přímým pokračovatelem hlavní vývojové linie, která započala prvním Arduinem v roce 2005. Jádro desky tvoří mikroprocesor Atmel ATmega328P. Dalšími důležitými prvky na desce jsou USB konektor společně s převodníkem USB dat na sériová data pro mikroprocesor, tlačítko Reset, napájecí konektor s napěťovým regulátorem a napájecími obvody. Dále deska Arduino Uno obsahuje 14 digitálních vstupních/výstupních pinů (z čehož 6 může být využito jako PWM výstupy), 6 analogových vstupních pinů, 2 piny vyhrazené pro I²C sběrnici, vyhrazený 6 pinový ICSP header a další zemnicí a napájecí piny. O taktování 8-bit mikrokontroléru se stará 16 MHz krystalový oscilátor. Mikrokontrolér samotný pak dále obsahuje například 10-bit A/D převodníky, 2 KB paměti SRAM, 1 KB EEPROM a 32 KB FLASH. Je schopený pracovat v rozmezí napájecích napětí 1.8 – 5.5 V. [12][13][14]

Existují dvě základní verze čipu ATmega328P, lišící se pouze v odlišném zapouzdření. Klasický čip s označením ATmega328P-PU je dodáván v pouzdře PDIP s 28 vývody (viz. obr. 3.1), druhá verze s označením ATmega328P-AU je verze pro povrchovou montáž v TQFP pouzdře s 32 vývody.

Používání desky Arduino Uno je velmi jednoduché. Pro samostatnou činnost desky se připojí napájení skrze napájecí konektor nebo se připojí přesný zdroj napětí 5 V na napájecí pin Vin. Napájet desku je také možné pomocí USB konektoru. Je nutné zvolit pouze jednu z těchto možností napájení, v opačném případě by mohlo dojít k poškození desky. Pro naprogramování mikrokontroléru se jednoduše připojí deska k počítači pomocí USB kabelu, případně pomocí ICSP konektoru s využitím ICSP programátoru. Programovat pak lze prostřednictvím mnoha vývojových prostředí jako jsou například Visual Studio, Atmel Studio nebo Arduino IDE, což je vývojové prostředí vytvořené přímo pro desky Arduino a také prostředí využitě při tvorbě řídicího programu k této poloautomatické leptací stanici.

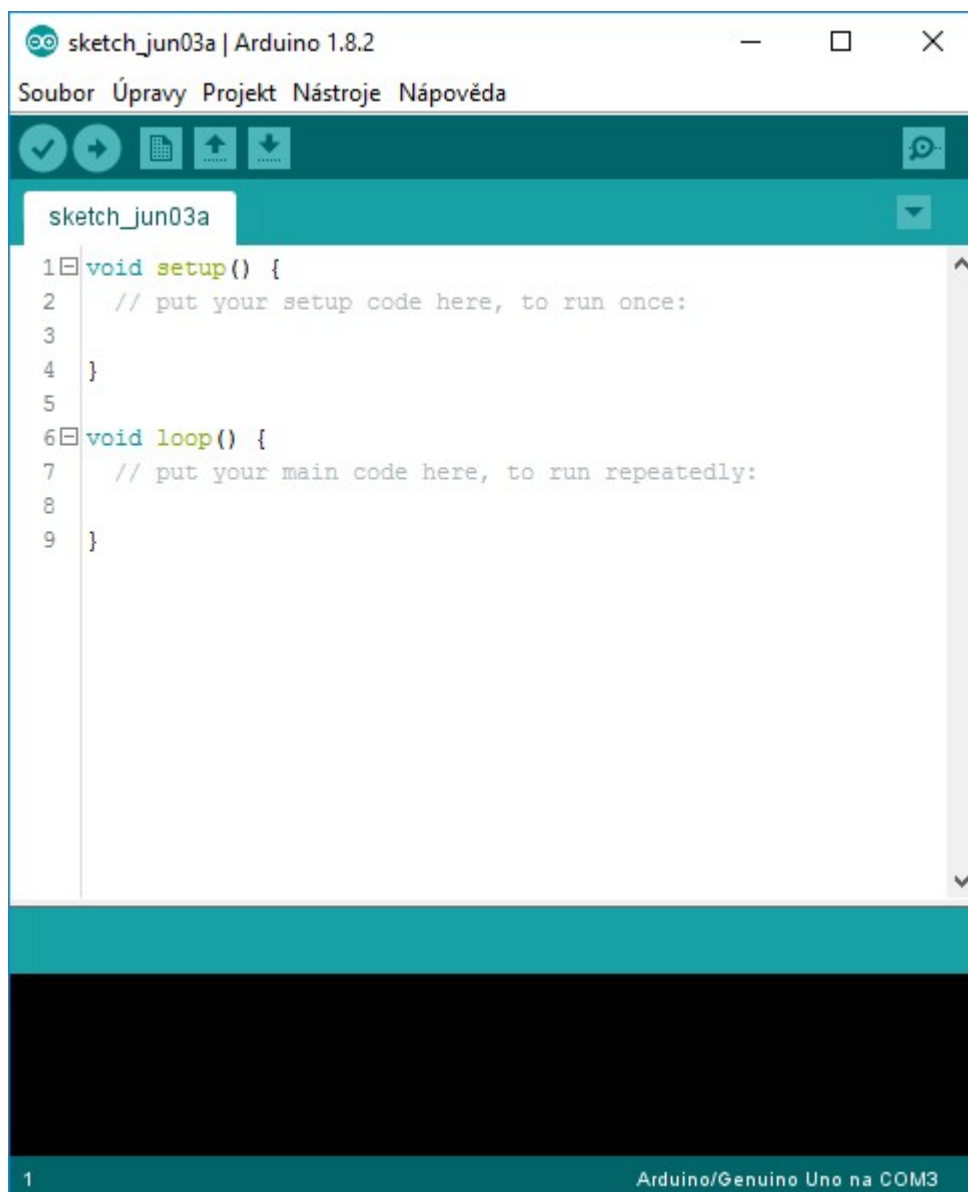
3.2 Arduino IDE

Upraveno podle [12][13][14].

Arduino IDE (Integrated Development Environment = integrované vývojové prostředí) je software napsaný v jazyce Java. Vznikl z výukového prostředí Processing, které bylo upraveno, byly přidány různé funkce a v poslední řadě také podpora jazyka Wiring.

Software, aktuálně ve verzi 1.8.2, je dostupný ke stažení zdarma z oficiálních webových stránek (viz. [12]). Dostupné jsou instalátory jak pro 32-bit tak i 64-bit verze operačních systémů Windows, Linux a MacOS.

3.2.1 Používání



Obr. 3.2 Arduino IDE

Na obrázku lze vidět vývojové prostředí. V první řadě nás bude zajímat nabídka Nástroje, ve které nalezneme nastavení pro připojení a programování desky. V nabídce Nástroje vybereme ze seznamu položku Vývojová deska a dále v seznamu desek vybereme používanou desku. V našem případě tedy Arduino/Genuino Uno (viz. příloha C.1). Dalším krokem je vybrat COM port, na který je Arduino připojeno, většinou je to jediný port v seznamu. Pokud není, zjistíme konkrétní COM port pomocí Správce zařízení v záložce Porty (COM a LTP). Název COM portu většinou odpovídá typu použitého převodníku USB na sériová data. V našem případě je to tedy port COM3 s názvem USB-SERIAL CH340 (COM3) (viz. příloha C.2).

V dalším řádku nalezneme několik ikon. První zleva je ikona s fajfkou – Ověřit. Ta po kliknutí spustí kontrolu našeho kódu a pokud nalezne nějakou chybu, zvýrazní ji. Další je ikona se šipkou doprava – Nahrát. Ta spustí kontrolu programu a pokud nenalezne žádnou chybu, nahraje program do připojeného Arduina. Další je ikona se symbolem přeloženého rohu stránky – Nový, která po kliknutí vytvoří nový soubor. Tlačítka se šipkou nahoru a se šipkou dolů jsou tlačítka pro otevření nebo uložení programu. Z pravé strany řádku najdeme ještě ikonu s lupou – Sériový monitor.

Důležitou součástí vývojového prostředí jsou uživatelské knihovny. Knihovny jsou předpřipravené balíčky kódu, které základní program rozšiřují o nějakou funkcionalitu (např. možnost ovládat či komunikovat se specifickým hardwarem, jako LCD displeje, různé senzory a moduly). Pro použití knihovny vybereme nabídku Projekt > Přidat knihovnu. Tím se na horní konec programu vloží příkaz `#include <název_knihovny.h>`. Příkaz je samozřejmě možné napsat také ručně.

3.2.2 Programovací jazyk

Arduino je možné programovat v jazyce C nebo C++. Nejjednodušší je však používat knihovnu Wiring, která je v současné době velmi rozšířená. Kvůli její komplexnosti se o ní často mluví jako o samostatném programovacím jazyku.

Na ukázkovém kódu (viz. obr. 3.2) si lze všimnout dvou základních věcí. První z nich je přítomnost dvou bloků, resp. funkcí programu. Mezi složené závorky bloku `void setup()` se píše kód, který se provede pouze jednou při startu programu. To znamená buď při připojení napájení, při stisknutí tlačítka Reset nebo při nahrání nového kódu do Arduina. Druhým blokem je `void loop()`, do jehož složených závorek se vkládá kód, který se bude opakovat neustále dokola. Tyto dva bloky musí být součástí programu vždy. Další věcí v ukázkovém kódu je dvojité lomítko, které značí začátek komentáře v programu. Část kódu nebo textu zapsanou za dvojitým lomítkem bude program ignorovat. Komentář za dvojitým lomítkem je jednořádkový komentář. To, co se nachází na další řádce, je již bráno jako pokračování kódu.

Každá proměnná jazyka Wiring má vlastní název, datový typ a hodnotu. Mezi časté datové typy patří `int` (zkratka `integer`), `long`, `float`, `boolean` a `char`. Pokud se chce s danou proměnou pracovat v celém programu, je nutné jí vytvořit vně všech funkcí (včetně funkcí `setup` a `loop`). Pokud stačí používat proměnou jen v rámci jedné funkce a nikde jinde, stačí deklarovat proměnou uvnitř dané funkce.

Kromě funkcí `setup()` a `loop()` jsou důležitou součástí také uživatelsky definované funkce, které si může každý uživatel vytvořit sám. Tyto funkce musí být definovány mimo tělo jiných funkcí a mohou mít vstupní parametry, se kterými se dále pracuje. Každá uživatelská funkce má určitý datový typ. Ten se liší podle typu dat, které funkce vrací. Pokud funkce žádná data nevrací, používá se speciální datový typ `void` (jako mají funkce `setup` a `loop`). Uživatelsky definované funkce se pak volají z jakékoliv části programu, a to i opakovaně.

Dalšími nedílnými součástmi programů jsou struktury podmínek a cyklů. Podmínky slouží k větvení programu. V závislosti na splnění podmínky se určitý příkaz buď provede anebo neprovede. Typicky se používá zápis `if(podmínka){} else{}`. Podmínka může být obecně jakákoliv, ale musí mít jednoznačný výsledek. Buď pravda nebo nepravda. Pro zápis podmínky se často využívají porovnávací operátory (`<`, `>`, `=`), logické operátory (`!`, `&&`, `||`) a i aritmetické operátory (`+`, `-`, `*`, `/`, `%`). Pomocí těchto operátorů lze vytvořit i poměrně složité podmínky, ve kterých se dají dokonce i volat uživatelské funkce. Například: `if(((A>30) && (A<50)) || (uzivatelska_funkce() == true)) {`. Tato podmínka je splněna pokud je proměnná `A` větší než 30 a zároveň menší než 50 nebo pokud je výsledek funkce `uzivatelska_funkce()` roven pravda. Při splnění podmínky se provede následující kód uvnitř složených závorek. V části `else {}` se provede kód, pokud podmínka splněna není. Část `else {}` není nezbytně nutná. Pokud není uvedena a pokud podmínka části `if(podmínka)` není splněna, nestane se nic a program pokračuje normálně dál. Dalším speciálním druhem podmínky je `Switch`. Je speciální v tom, že se zabývá pouze proměnnou (popřípadě výsledkem nějaké volané funkce) a její hodnotou. Program prochází každou větev konstrukce `switch` a testuje hodnotu proměnné. Další rozdíl je v tom, že se může provést i více větví (což u `if` nelze). Pokud se ale chce, aby po provedení větve program pokračoval až za konstrukcí `switch`, je nutné použít na konci větve příkaz `break`. Konstrukci `switch` zobrazuje následující ukázka.

```
1  switch (promenna) {
2      case 1:
3          //pokud je hodnota promenne 1, provede se tato cast kodu
4          break; //po provedeni teto casti konstrukce switch konci
5      case 2:
6          //pokud je hodnota promenne 2, provede se tato cast kodu
7          break;
8      default:
9          /* pokud se hodnota promenne nerovna zadne z nabizenych
10         moznosti, provede se tato cast */
11 }
```

Cyklus přicházejí na řadu, pokud se v programu stále něco opakuje. Základní cykly jsou for a while. Cyklus for se používá nejčastěji v případech, kdy je známý počet opakování cyklu. Zápis je následující: `for(int i = 0; i < 100; i++) {}`. Od ostatních cyklů se liší tím, že potřebuje mít pro funkci svojí vlastní proměnnou (proměnná `i`), která hlídá počet opakování. Dle zápisu výše se v první řadě deklaruje proměnná `i`, následuje testovaná podmínka a nakonec operace, která se provede při každém opakování (zápis `i++` je shodný se zápisem `i = i + 1`). Tento cyklus for se tedy opakuje 100 krát a při každém opakování se provede kód vložený do složených závorek. Dalším cyklem je o něco jednodušší cyklus while se zápisem `while(podmínka) {}`. Všechny příkazy cyklu vložené do složených závorek se provádějí stále dokola, dokud je podmínka pravdivá. Podmínka se testuje vždy na začátku cyklu a je tedy možné, že se cyklus while vůbec neprovede.

3.3 Řídící program stanice

Z důvodu svého rozsahu je kompletní řídicí program stanice součástí přílohy elektronické formy práce. Zde v této kapitole jsou pouze uvedeny jeho základní funkce a několik málo ukázek kódu.

Řídící program stanice musí umožňovat nastavení a udržování požadované teploty uvnitř leptací lázně a nastavení času, po jehož uplynutí oznámí stanice ukončení cyklu. Teplotu je možné nastavit v rozsahu od 20 °C až do 50 °C. Řídící program dále umožňuje nastavení množství použitého leptacího roztoku. Informace o objemu leptacího roztoku může být zdánlivě nedůležitá. Avšak chceme-li využít další přidané funkcionality, je nutné tuto informaci znát. Touto přidanou funkcí je možnost využití přehřevu roztoku leptací lázně. Funkce přehřevu totiž umožňuje ohřát leptací roztok na požadovanou teplotu ještě

před vložením desky do leptací komory a započítím odpočtu nastaveného času cyklu leptání. Pokud si uživatel stanice tuto funkci zvolí, je informace o množství leptacího roztoku využita k výpočtu odhadované doby předehřevu.

Po náběhu napájení (resp. po ukončení předchozího cyklu nebo po resetu) se zobrazí úvodní obrazovka. Na úvodní obrazovce jsou zobrazeny aktuální hodnoty nastavení (požadovaná teplota, čas cyklu leptání a objem leptacího roztoku). Po náběhu napájení nebo resetu jsou tyto hodnoty nastaveny na své výchozí hodnoty (45 °C, 12 min a 1 litr roztoku). Po ukončení předchozího cyklu zůstávají tyto hodnoty takové, jaké byly nastaveny v předchozím cyklu.

Spuštění procesu s hodnotami uvedenými na úvodní obrazovce se provede stiskem tlačítka Start. Chceli uživatel hodnoty změnit, lze přejít do nastavení stiskem tlačítka Stop. Následně lze nastavované hodnoty měnit pomocí tlačítek Plus a Mínus. Pomocí tlačítka Start se právě nastavovaná hodnota potvrdí a nastavení přejde na další hodnotu.

Před samotným započítím procesu leptání (resp. předehřevu pokud byl zvolen) se na LCD displeji zobrazí souhrn aktuálních nastavených hodnot (množství roztoku, požadovaná teplota a čas, volba předehřevu) a po uživateli je vyžadováno jejich potvrzení. Pro ovládání stanice má uživatel k dispozici čtyři tlačítka. Tlačítko Start, Stop, Plus a Mínus. Obsluha těchto tlačítek v programu je řešena pomocí funkce `btnScan()`.

```
int btnScan()
{
    // tlacitko Start
    if (digitalRead(btnStart) == HIGH)
    {
        while (1)
        {
            if (digitalRead(btnStart) == LOW)
            {
                return btnStart;
            }
        }
    }
    // tlacitko Stop
    if (digitalRead(btnStop) == HIGH)
    {
        ...
    }
    // tlacitko Plus
    ...
}
```

Funkce kontroluje stav pinů, na kterých jsou tlačítka připojena. Funkce `digitalRead(btnStart)` vrací hodnotu `HIGH`, je-li tlačítko Start stisknuto, hodnotu `LOW` vrací v opačném případě. Je-li tedy tlačítko Start stisknuto, je splněna podmínka `(digitalRead(btnStart) == HIGH)` a běh programu se zacyklí v nekonečném cyklu `while (1)`. Z tohoto cyklu se lze dostat jediným způsobem, a to následným uvolněním tlačítka Start. Tím je splněna podmínka `(digitalRead(btnStart) == LOW)` a příkaz `return btnStart;` následně ukončí funkci `btnScan()` s návratovou hodnotou `btnStart` (tedy s informací o tom, které tlačítko bylo stisknuto). Tento postup je obdobný pro všechny čtyři tlačítka.

Dosažení požadované teploty v rámci cyklu přehřevu je následně oznámeno zvukovým signálem a zeleně blikajícími LED diodami. Stejný způsob oznámení je využit i při dokončení samotného cyklu leptání (resp. při uběhnutí nastaveného času). Pokud je zvukový signál zrychlený a diody blikají červeně, oznamuje stanice vznik chyby. Chyba může nastat například tehdy, když teplota leptacího roztoku nedosáhne požadované teploty během určité maximální doby ohřevu (to by signalizovalo poruchu topného tělesa).

Kontrola teploty je prováděna v pravidelných intervalech po jedné vteřině. Podle získané aktuální teploty se následně rozhoduje, má-li být topné těleso spuštěno či ne (viz. následující úryvky kódu). Společně s kontrolou a aktualizací teploty je prováděna i aktualizace času na LCD displeji.

```
bool isTimeForNextTimeAndTempActualize(unsigned long
nextTimeAndTempActualizeTime)
{
    if (millis() > nextTimeAndTempActualizeTime)
    {
        return true;
    }
    return false;
}
```

Tato funkce kontroluje, je-li již čas pro aktualizaci teploty a času. Jedná se o jednoduché porovnání aktuálního času, který vrací funkce `millis()`, a času, kdy se má provést kontrola.

```
else if (isTimeForNextTimeAndTempActualize(nextTimeAndTempActualizeTime))
{
    nextTimeAndTempActualizeTime = millis() +
ACTUALIZE_TIME_AND_TEMP_PERIOD;
    currentTemp = getCurrentTemp();
    actualizeTimeAndTempInPreheatOrProcessPage(startTime, currentTemp,
requiredTemp);
    actualizeLedColor(currentTemp, requiredTemp, 0);
    regulateTemp(requiredTemp, currentTemp);
}
```

Je-li podmínka pro kontrolu a aktualizaci teploty a času splněna, nastaví se nový čas, kdy se má provést další kontrola. Dále je volána funkce `getCurrentTemp()`.

```
float getCurrentTemp()
{
    senzor.requestTemperatures();
    return senzor.getTempCByIndex(0);
}
```

Funkce `getCurrentTemp()` vrací aktuální hodnotu naměřené teploty ve stupních Celsia. Ke správné komunikaci je nutné připojit k programu knihovnu `OneWire.h`. Tato knihovna ale zajišťuje pouze komunikaci s teplotním senzorem, který vysílá data v binárním kódu. K dekódování těchto dat a následnému přepočtu na stupně Celsia je nutné připojit ještě knihovnu `DallasTemperature.h`.

```
void regulateTemp(int requiredTemp, float currentTemp)
{
    if ((currentTemp >= (requiredTemp + tempHysteresis)))
    {
        stopHeat();
    }
    if ((currentTemp <= (requiredTemp - tempHysteresis)))
    {
        startHeat();
    }
}
```

Tato funkce slouží ke spínání nebo vypínání topného tělesa. Vstupními hodnotami funkce je aktuální a požadovaná teplota. V podmínkách se následně tyto dvě hodnoty porovnávají a podle toho je zapínáno či vypínáno topné těleso. Hodnota `tempHysteresis` slouží k nastavení hystereze regulace kolem požadované teploty a její výchozí hodnota je jeden stupeň Celsia. Topné těleso je tedy zapnuto v případě, že je aktuální teplota nižší než požadovaná teplota snížená o jeden stupeň. A vypnuto v případě, že je aktuální teplota vyšší než požadovaná teplota zvýšená o jeden stupeň.

```
void startHeat ()
{
  if (digitalRead(Heat) == HIGH)
  {
    digitalWrite(Heat, LOW);
  }
}
```

Funkce `startHeat()` spíná topné těleso, pokud bylo v předchozím stavu vypnuto. Pokud bylo zapnuto, neděje se nic a zapnuto zůstává. Obdobně je řešena funkce pro vypínání topného tělesa a pro zapínání a vypínání cirkulačního čerpadla.

Funkce `actualizeTimeAndTempInPreheatOrProcessPage(...)`; se stará o aktualizaci teploty a času na LCD displeji a funkce `actualizeLedColor(...)`; zajišťuje aktualizaci barvy pásku RGB LED diod. Barvu určuje aktuální teplota leptacího roztoku. Mezi teplotou 10 °C a požadovanou teplotou je určeno 6 stupňů, ve kterých svítí vždy jedna určitá barva. Od nejnižší teploty po požadovanou teplotu je to barva fialová, modrá, azurová, zelená žlutá a červená.

Proces přehřevu nebo cyklus leptání nemusí být ukončen pouze automaticky splněním daných podmínek anebo vznikem chyby. Samozřejmostí je možnost předčasného ukončení procesu tlačítkem Stop. Po ukončení procesu se řídicí program stanice vrací zpět na začátek a zobrazí se úvodní obrazovka. V tomto počátečním stavu je přístupná ještě jedna další funkce. Stiskem tlačítka Mínus je možné celý řídicí systém uspat a snížit tak odběr proudu, není-li stanice využívána. Opětovné probuzení systému je umožněno tlačítkem Start.

4 Výsledky testování

První testy vytápění leptací komory byly provedeny s jedním litrem vody a až poté s leptacím roztokem. Výsledky měření se v obou případech příliš nelišily. Aby byly testy co nejvíce přiblíženy reálnému používání stanice, byla ve všech případech ve vedlejší velké komoře přítomna voda. Množství vody bylo voleno tak, aby hladina vody odpovídala hladině roztoku v leptací komoře. Přítomnost vody v oplachové komoře byla také zdrojem nepřesností v měření. Nebylo totiž možné udržet konstantní teplotu vody po dlouhou dobu. Jednak je voda v oplachové komoře ohřívána skrze stěnu komory a jednak nelze při výměně vody v komoře napustit vodu vždy o stejné teplotě.

Testy probíhaly pro ohřev jednoho litru roztoku, a to z počátečních hodnot 10, 15, 20 °C a z pokojové teploty 24 °C, která byla v daném případě naměřena. Koncové (resp. požadované) teploty byly 40, 45 a 50 °C. Zprůměrované výsledky zobrazuje následující tabulka.

Tab. 2 Výsledky měření času vytápění komory na určitou teplotu s jedním litrem roztoku

Počáteční teplota [°C]	10	15	20	24
Požadovaná teplota [°C]				
40	14:15	12:50	9:30	8:00
45	17:30	15:40	12:45	10:30
50	21:15	19:30	16:35	13:05

pozn.: Všechny časy jsou uvedeny ve formátu [minuty]:[vteřiny].

Během udržování teploty v komoře na požadované hodnotě bylo topné těleso v průměru spínáno na 35 až 40 vteřin a s odstupem spínání 75 až 80 vteřin.

Závěr

Cílem teoretické části této práce bylo nastudovat různé metody používané pro výrobu desek plošných spojů a následně využít získaných poznatků v praktické části k vytvoření návrhu a sestrojení poloautomatické leptací stanice. Nastudované metody výroby DPS včetně podkapitol věnovaných různým typům DPS a materiálům ze kterých se vyrábějí, jsou součástí teoretického úvodu. V této části práce byl kladen důraz na materiály a postupy výroby používané v dnešní době nejčastěji. Zvlášť detailně byla popsána metoda výroby desek plošných spojů semiaditivní technologií.

V dalších kapitolách byl následně popsán návrh a sestrojení tříkomorové vytápěné leptací kyvety a řídicího systému, který slouží k obsluze součástí této kyvety a také k obsluze celé poloautomatické stanice uživatelem. Návrh i sestrojení obou částí proběhlo úspěšně. Při prvním testu těsnosti všech tří komor kyvety se objevilo několik nedokonalostí, díky kterým kapalina unikala skrze lepené spoje. Všechny spoje byly po vysušení znovu obnoveny lepidlem a všechny přístupné rohy a hrany byly dále utěsněny silikonem. Další menší závadou při sestrojování kyvety byly vypouštěcí kohouty. Jejich závit se totiž směrem od uzávěru k ústí trochu zužuje a s kohouty dodávané matice se přes zúžené závity převlékaly. Tento problém byl vyřešen omotáním závitu teflonovou těsnicí páskou.

Při sestavení řídicího systému žádné problémy nenastaly. I tak ale nebyl systém použitelný hned od začátku. Na vině byl řídicí program mikrokontroléru. Ačkoli základní myšlenka zůstává již od návrhu téměř nezměněna, první verze programu se od verze prezentované v této práci značně liší. Dále bylo také nutné provést doladění programu na základě testování.

Zprůměrované výsledky všech testování jsou uvedeny v předchozí kapitole. Zde jsou uvedeny jen ty nejčastější případy. Jedním z nich je, že byl leptací roztok skladován v lednici, ve které byla teplota 10 °C. V tom případě trval ohřev na 40 °C 14 minut a 15 vteřin, na 45 °C 17 minut a 30 vteřin a na 50 °C 21 minut a 15 vteřin. Druhá nejčastější situace je, že byl roztok skladován při pokojové teplotě (24 °C). V tom případě trval ohřev na 40 °C 8 minut, na 45 °C 10 minut a 30 vteřin a na 50 °C 13 minut a 5 vteřin.

Díky ztrátám tepla do okolí byla reálná teplota roztoku uvnitř lázně ve všech případech měření nižší než teplota požadovaná. Namísto požadovaných 50 °C se teplota v komoře udržovala na 47.5 °C, namísto 45 °C 43 °C a namísto 40 °C 39 °C. I když jsou reálné hodnoty nižší, byly tyto hodnoty naměřeny v celém objemu roztoku. To poukazuje na správnou funkčnost cirkulace roztoku v komoře. Významným důvodem pro tyto teplotní rozdíly je topné těleso. Těleso se totiž velmi rychle zahřálo na velmi vysokou teplotu. V případě vytápění komory na 50 °C se povrch topného tělesa zahříval až na 70 °C. Takto vysoká teplota je již nepřipustná, protože leptací roztok se při teplotách nad 55 °C začíná rozkládat. Aby se tomuto přehřívání zabránilo, bylo teplotní čidlo umístěno v těsné blízkosti topného tělesa. Problém přehřívání nejspíše není v proudění roztoku kolem topného tělesa. Zvýšení průtoku cirkulačním čerpadlem nijak výrazně nepomohlo. Pravděpodobně proudící roztok nestíhá dosti rychle z takto výkonného tělesa teplo odvádět. Řešením by mohlo být méně výkonné topné těleso.

Únik tepla z leptací do oplachové komory je jedním z nedostatků návrhu kyvety. Vylepšení týkající se těchto ztrát tepla by bylo do oplachové komory nalít teplou nebo horkou vodu nebo vytvořit mezi komorami dvojitou stěnu se vzduchovou mezerou (nebo vyplněnou izolačním materiálem).

V závislosti na koncentraci leptacího roztoku se za zvýšených teplot lázně vyleptá DPS za 10 až 15 minut. Dobu leptání také velmi významně ovlivňuje množství již odleptané mědi (z předchozích cyklů leptání) v roztoku. Při zapnutém ohřevu byl naměřen proud ze sítě 830 mA. Při spuštění stanice, ale ne během ohřevu, bylo naměřeno 3.4 mA a při uspání stanice do sleep módu dokonce 0.8 mA.

Seznam literatury a informačních zdrojů

- [1] PLÍVA Zdeněk. *Z historie plošných spojů*. [online]. 2010 [cit. 1.5.2017]. Dostupné z: <http://www.dps-az.cz/zajimavosti/id:4644/z-historie-plosnych-spoju>
- [2] MiniBoard Service. [online]. Dostupné z: <https://www.expresspcb.com/miniboard-service/>
- [3] *Basic Board Materials*. [online]. Dostupné z: http://www.mtarr.co.uk/courses/topics/0232_bbm/index.html
- [4] *Válcování kovů*. [online]. 17.1.2016. Dostupné z: https://cs.wikipedia.org/wiki/V%C3%A1lcov%C3%A1n%C3%AD_kov%C5%AF
- [5] *Common PCB Builds*. [online]. Dostupné z: <http://www.shipcircuits.com/board-build-2/>
- [6] *Flex & Rigid-Flex PCB Gerber Layout*. [online]. Dostupné z: <http://www.epectec.com/webinar/05-30-2014/rigid-flex-pcb-gerber-layout-requirements.html>
- [7] ZÁHLAVA, Vít. *Návrh a konstrukce desek plošných spojů – principy a pravidla praktického návrhu*. 1. vyd. Praha: 2010. 128 s. ISBN 978-80-7300-266-4
- [8] LIBICH, Jiří. *Návrh koncepce výrobní linky pro výrobu vícevrstevných desek plošných spojů s mikrospoji*. Brno: 2009. bakalářská práce (Bc.) VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, Fakulta elektrotechniky a komunikačních technologií
- [9] *Výroba obrazce plošných spojů*. [online]. 2.6.2010. Dostupné z: <http://www.smtcentrum.cz/vyroba-desek-plosnych-spoju/vyroba-obrazce-plosnych-vodicu/>
- [10] VÍTEK, Václav. *Tvorba pracoviště pro výrobu desek plošných spojů*. C. Bud.: 2013. bakalářská práce (Bc.). JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH. Pedagogická fakulta
- [11] CIRKOVSKÝ, Jaroslav. *Metody návrhu desek plošných spojů*. Brno: 2008. bakalářská práce (Bc.) VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, Fakulta elektrotechniky a komunikačních technologií
- [12] *Arduino UNO & Genuino UNO*. [online]. Dostupné z: <https://www.arduino.cc/>
- [13] VODA, Zbyšek. *Průvodce světem Arduina*. 1. vyd. Bučovice: 2015. 280 s. ISBN 978-80-87106-90-7
- [14] MARGOLIS, Michael. *Arduino cookbook*. 2nd ed. Sebastopol: c2012. ISBN 978-1-449-31387-6

Přílohy

Příloha A – Ukázky objednávkových listů výrobců DPS

a)

8. 5. 2017

Poptávkový formulář

Published on PragoBoard s.r.o. - vícevrstvé, dvou a jednovrstvé plošné spoje (<http://www.pragoboard.cz>)

Časá - Přílohy

Poptávkový formulář



pole označená - jsou povinná !!

Firma, fyzická osoba: *

E-mail adresa zákazníka: *

Název plošného spoje: *

Typ výroby: * nová výroba opakovaná zakázka POOL servis šablona pro SMT

Počet jednotlivých kusů / panelů / šablon: *

Rozměr jednotlivého DPS / panelu / šablony: *

 mm

Základní materiál: *

Výsledná tloušťka DPS: *

Tloušťka Cu folie:

Počet vodivých vrstev: *

 1 2 4 6 8 10 12 pouze laminát šablona
Nepálivá maska: * 1x 2x bez maskyBarva nepálivé masky: Servisní potisk: * 1x 2x bez potiskuBarva servisního potisku: Povrchová úprava: * H.A.S.L. Pb free H.A.S.L. Pb chemické zlato chemický cín bez úpravyGalvanické zlacení konektoru: * ano ne

Počet zlacovaných pinů:

Plocha 1 pinu: mm²Elektrické testování: * ano neObrysová oprava: * stříhání frézování frézování na můstky drážkování bez úpravySpeciální technologie: slepé otvory pohřbené otvory snímatelná maska karbon pasta vyplňované otvory

prokovené plněné otvory

Fakturace v : * měnaTermín výroby: * standard 10 pracovních dní expresní výrobaPočet dní: Připojit data v elektronické podobě: Soubor nevybrán

Velikost zaslaných dat max. 5Mb.

Poznámka:

CAPTCHA

Tato otázka je pro ověření, zda jste opravdový návštěvník a ne automatizovaný spamový robot.



Jaký kód je na obrázku? *

Vložte znaky zobrazené na obrázku.

b)

8. 5. 2017

Poptávkový / objednávkový formulář | MEV

INFORMACE O
POLEČNOSTI

TECHNICKÉ
MOŽNOSTI

INSTALOVANÁ
TECHNOLOGIE

PROTOTYPE
POOL

VÝROBA
ŠABLON

DOOACÍ
PODMÍNKY

KONTAKT

POPTÁVKOVÝ / OBJEDNÁVKOVÝ FORMULÁŘ

Povinné údaje:

Typ: Objedávka Poptávka

Základní*:

Telefon*:

Adresa*:

Fax:

Kontaktní osoba*:

E-mail*:

Označení desky*:

Nová zakázka

Opakovaná výroba

Verze (označení předchozí)

Množství*: ks

panely

Datum dodání (dd.mm.rrrr)*:

Typ: Jednostranná Dvoustranná Vícevrstvá

Počet vrstev:

Přenos dat: Floppy disc / CD-ROM E-mail

Název souboru:

Rozměry: Rozměry panelu (mm)*: x

Matériaž: FR4 jiný

Počet desek na panelu*:

Tloušťka desky (mm)*:

Rozměry desky (mm)*: x

Tloušťka Cu (um)*:

Měroňnost: Nejmenší vrtaný otvor (mm)*:

Povrchová úprava: HAL

organický povrch

chemické zlacení

chemický čit

Nejmenší šířka spoje (mm)*:

Nejmenší značnická mezera (mm)*:

Počet vrtaných otvorů*:

Průhledy: Neokrované tvory BLIND VIAs BURIED VIAs

Nepájivá maska: na straně pájení na straně součástek

Obryš: střídání frézování frézování s mřížky drážkování bez opracování

Smělečná maska: na straně pájení na straně součástek

Počítak: na straně pájení na straně součástek

Elektrický tvar: bez BMD 1x BMD 2x BMD vybrat

Přímý konektor: Zácená plocha (cm2):

vybrat: Počet párů konektorů:

c)

OBJEDNÁVKA výroby plošných spojů č.

PragoBoard s.r.o.
Technologický park Běchovice
190 11 Praha 9 – Běchovice

IČO : 2561 5149
DIČ : CZ2561 5149

Odběratel :

způsob dopravy :
forma úhrady :
cenová nabídka č. :

Objednáváme u Vás :

Položka č.	Název DPS	Počet kusů	Počet vrstev	Materiál	Technologie	Vykreslení filmů	Termín dodání
1.							
2.							
3.							
4.							
5.							
6.							
7.							

Poznámka :

Za objednavajícího :

datum :

název DPS : používejte vždy jednoznačný a jedinečný název plošného spoje (nelze zadat 2 různé desky pod stejným názvem). Tento název bude uveden na dodacím listě a na faktuře. Pro případnou opakovanou zakázku uvádějte v objednávce DPS pod stejným názvem a označením verze

počet kusů : rozumí se počet jednotlivých motivů. V případě potřeby dodání v panelech, zadejte také požadovanou panelizaci

počet vrstev : **1** jednovrstvý, **2** dvouvrstvý, **4** čtyřvrstvý, **6** šestivrstvý atd .

materiál : zadání tloušťky základního materiálu + síly použité Cu folie v mikronech (1,5 35/35) (u dvou a vícevrstvých DPS mějte na paměti, že ve výrobním procesu bude na základní folii nakoveno dále nejméně 25 mikronů. Zadává se tedy výsledná tloušťka Cu folie. V případě požadavku použití jiného materiálu - např. s vyšším Tg bodem - upřesněte. Standardní výrobky jsou vyráběny z materiálu kvality FR4 - Tg bod 130 °C

technologie : zadání kompletní technologie jednotlivých DPS :

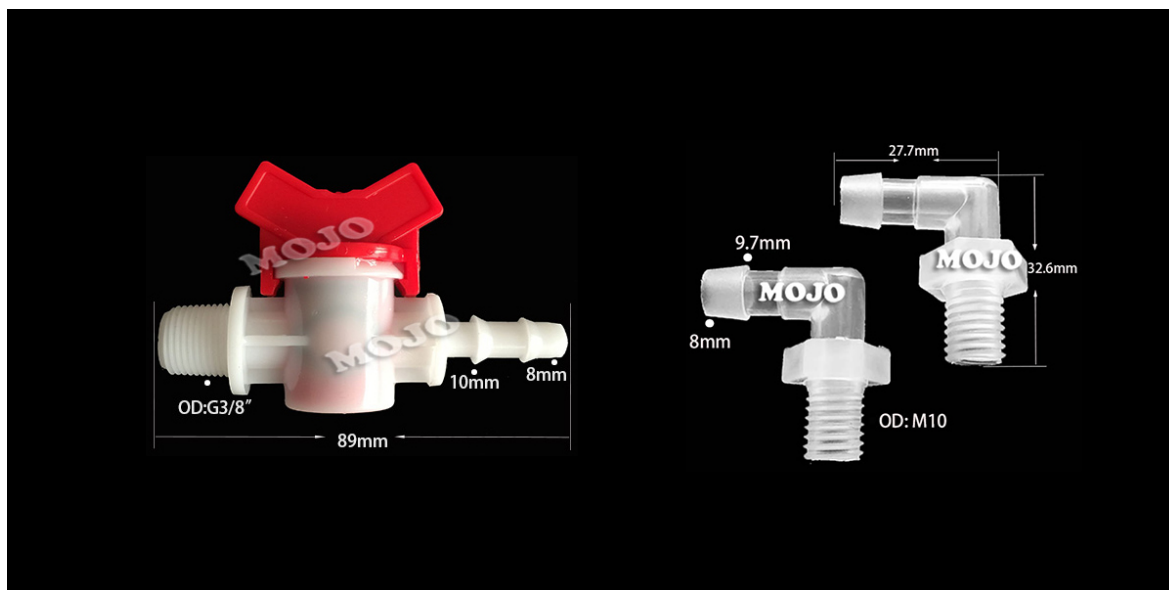
- **NM** nepájivá maska, **H** žárový cín H.A.S.L., **Po** potisk, **ZI** zlacení konektorů, **Fr** frézování výřezů nebo obrysu, **Dr** drážkování, **Te To** elektrické nebo optické testování, **Ni** niklování, **Au** chemické zlacení, **Sn** chemické cínování

vykreslení filmů : zadejte ANO - NE

- v případě nové zakázky **ANO**, v případě opakované výroby nebo výroby z donesených filmových předloh **NE**

termín dodání : termín převzetí hotových DPS nebo jejich předání prvnímu dopravci

Příloha B – obrázky součástí kyvety



Obr. B.1 Kohout a průchodky;
převzato a upraveno z webových stránek výrobce: www.mojo-hk.com

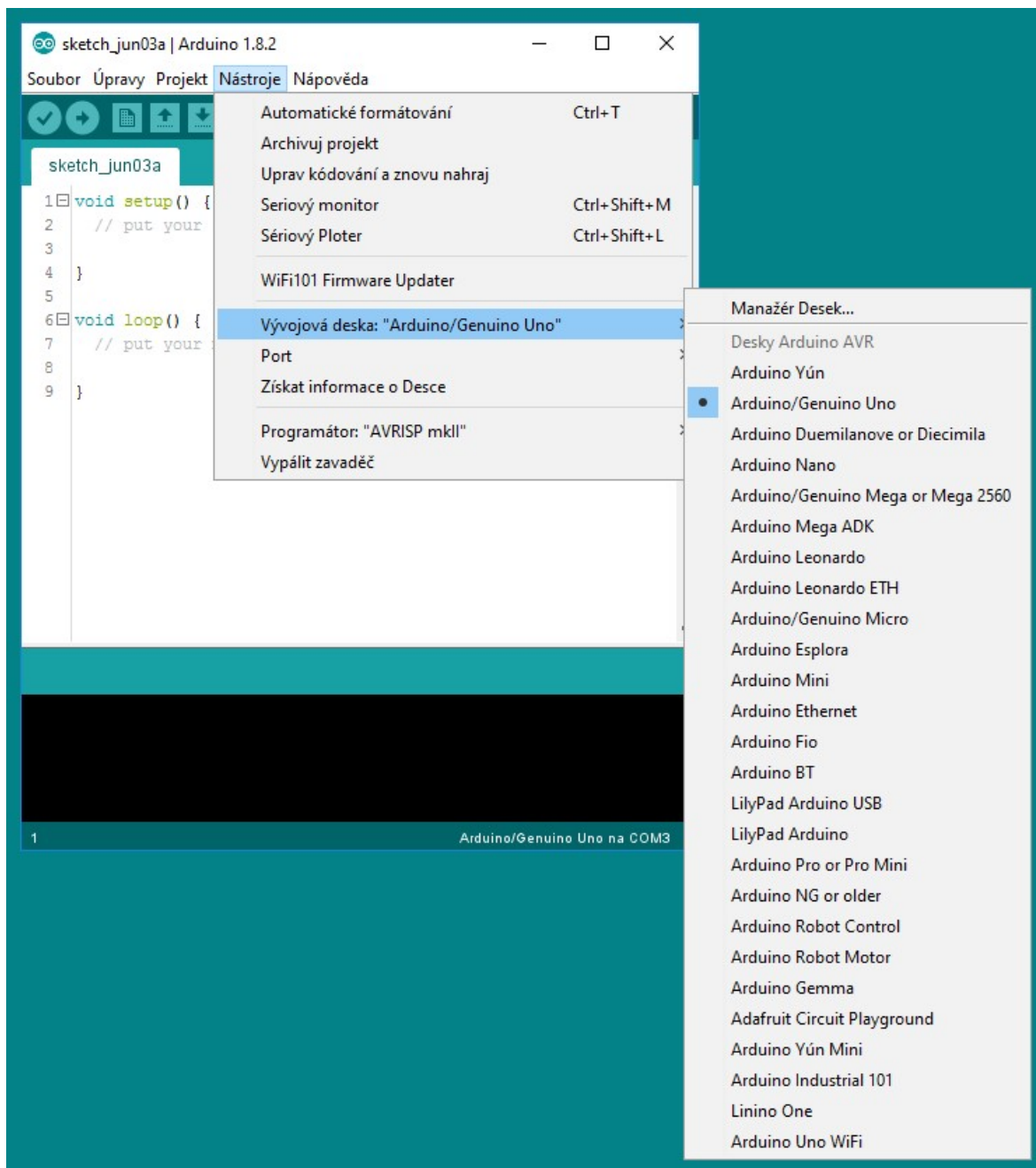


Obr. B.2 Cirkulační čerpadlo

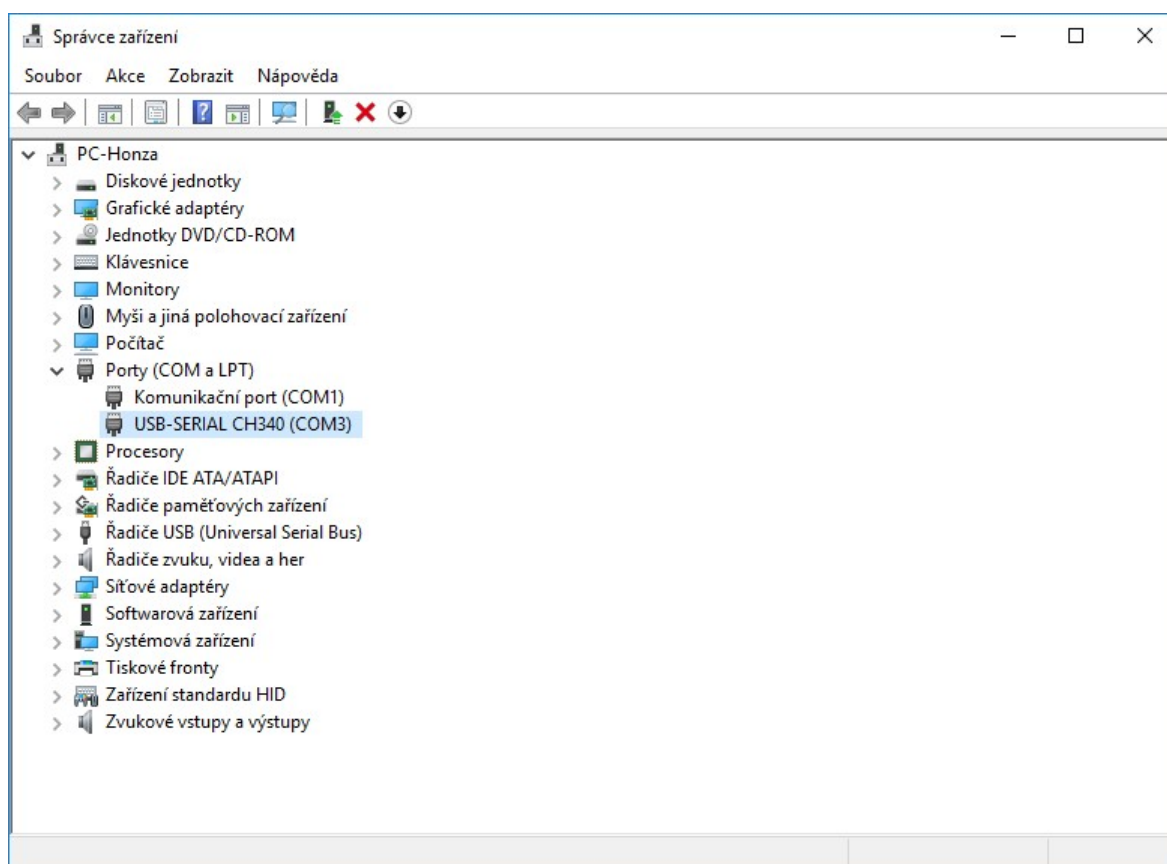


Obr. B.4 Teplotní senzor DS18b20 zapouzdřený s vývody

Příloha C – Arduino IDE



Obr. C.1 Výběr konkrétní vývojové desky



Obr. C.2 Zjištění COM portu

Příloha D – Řídící program stanice

```
//----- DEKLARACE A DEFINICE -----

#include <OneWire.h> // zprostředkovává komunikaci s teplotním čidlem
#include <DallasTemperature.h> // dělá konverzi dat z čidla na teplotu
#include <LiquidCrystal_I2C.h> // zprostředkovává komunikaci s lcd
#include <avr/sleep.h> // umožňuje arduinu sleep mode

// definice názvu pinu
#define btnStart 2 // tlačítko start
#define btnStop 3 // tlačítko stop
#define btnPlus 4 // tlačítko +
#define btnMinus 5 // tlačítko -
#define Sensor 6 // teplotní senzor
#define Pump 7 // rele IN1
#define Heat 8 // rele IN2
#define redLed 9 // červená led
#define greenLed 10 // zelená led
#define blueLed 11 // modrá led
#define Buzz 12 // pípač
// definice lcd
#define lcdAddr 0x27 // I2C adresa lcd
#define lcdRows 4 // počet řádků lcd
#define lcdCols 20 // počet sloupců lcd
// definice UI
#define UI 2 // 1 = seriová komunikace, 2 = lcd

LiquidCrystal_I2C lcd(lcdAddr, lcdCols, lcdRows); // vytvoření instance
lcd z knihovny LiquidCrystal_I2C
OneWire oneWire(Sensor); // vytvoření instance oneWire z knihovny OneWire
DallasTemperature senzor(&oneWire); // vytvoření instance senzor z
knihovny DallasTemperature

// definice konstant
const int MANUAL_END = 1;
const int AUTOMATIC_END = 2;
const int FAILRULE_END = 3;

const int NOTHING = 0;
const int MANUAL = 1;
const int AUTOMATIC = 2;
const int EXIT = 3;

const int DEFAULT_TEMP = 45;
const long DEFAULT_PROCESS_TIME = 720000;
const float DEFAULT_SOLUTION_VOLUME = 1;

const int PREHEAT_SUCCES = 1;
const int PREHEAT_FAILED = 0;
const int ACTUALIZE_TIME_AND_TEMP_PERIOD = 1000;
const int TIME_STEP = 5000;
const float TEMP_STEP = 1;
const float SOLUTION_VOLUME_STEP = 0.25;

const int HEATING_ELEMENT_POWER = 200;

// definice proměnných
int processType = 0;
int requiredTemp = 0;
int tempHysteresis = 1;
```

```
int requiredPreheat = 0;
int endType = 0;
int preheatFinished = 0;
float currentTemp = 0;
float solutionVolume = 0;
float colorStep = 0;
unsigned long requiredProcessTime = 0;
unsigned long preheatTime = 0;
unsigned long startTime = 0;
unsigned long maxProcessTime = 0;
unsigned long maxPreheatTime = 0;
unsigned long nextTimeAndTempActualizeTime = 0;
unsigned long defaultValue = 0;
unsigned long lastBeepAndBlink = 0;
bool succesFinished = false;
String question = "otazka";

//----- SETUP -----

void setup()
{
  if (UI == 1)
  {
    Serial.begin(9600); // inicializace komunikace pres seriovou linku
  }
  else
  {
    lcd.init(); // inicializace komunikace s lcd
    lcd.clear(); // vycistení lcd
    lcd.backlight(); // zapnutí podsvícení lcd
  }

  senzor.begin(); // inicializace komunikace cidla s knihovnou
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, LOW); // vypnutí ledky na desce

  // inicializace pinu podle definovanych nazvu
  pinMode(Pump, OUTPUT);
  digitalWrite(Pump, HIGH);
  pinMode(Heat, OUTPUT);
  digitalWrite(Heat, HIGH);

  pinMode(redLed, OUTPUT);
  digitalWrite(redLed, HIGH);
  pinMode(greenLed, OUTPUT);
  digitalWrite(greenLed, HIGH);
  pinMode(blueLed, OUTPUT);
  digitalWrite(blueLed, HIGH);

  pinMode(btnStart, INPUT);
  pinMode(btnStop, INPUT);
  pinMode(btnPlus, INPUT);
  pinMode(btnMinus, INPUT);

  solutionVolume = DEFAULT_SOLUTION_VOLUME;
  requiredTemp = DEFAULT_TEMP;
  requiredProcessTime = DEFAULT_PROCESS_TIME;
  currentTemp = getCurrentTemp();
}
```

```
//----- LOOP -----  
  
void loop()  
{  
  processType = displayMenu(); // nekonecna smycka, ktera kontroluje  
  tlacitka a ceka na vybranou volbu  
  switch (processType)  
  {  
    case NOTHING: // nic se nedeje  
    {  
      break;  
    }  
    case MANUAL: // manualni nastaveni  
    {  
      solutionVolume = askForSolutionVolume(); // ziskani objemu  
      leptaciho roztoku  
      requiredTemp = askForTemp(); // ziskani pozadovane teploty  
      requiredProcessTime = askForProcessTime(); // ziskani  
      pozadovane doby procesu  
      preheatTime = computePreheatTime(requiredTemp, solutionVolume);  
      // vypocet jak dlouho bude predehrev trvat  
      requiredPreheat = askForPreheat(preheatTime); // ziskani volby  
      predehrevu  
      if (askForStart() == true) // otazka jestli chceme opravdu  
      spustit proces s danymi parametry  
      {  
        process(requiredTemp, requiredProcessTime, requiredPreheat,  
preheatTime); // zapne proces  
      }  
      break;  
    }  
    case AUTOMATIC: // defaultni nastaveni  
    {  
      preheatTime = computePreheatTime(requiredTemp, solutionVolume);  
      requiredPreheat = askForPreheat(preheatTime);  
      if (askForStart() == true)  
      {  
        process(requiredTemp, requiredProcessTime, requiredPreheat,  
preheatTime);  
      }  
      break;  
    }  
    case EXIT:  
    {  
      goToSleepMode();  
      break;  
    }  
  }  
}  
  
//----- UZIVATELSKE FUNKCE -----  
  
int btnScan() // kontroluje stav tlacitek a vraci hodnotu podle toho,  
ktere tlacitko bylo zmacknuto  
{  
  if (digitalRead(btnStart) == HIGH)  
  {  
    while (1)  
    {  
      if (digitalRead(btnStart) == LOW)  
      {
```

```
        return btnStart;
    }
}

if (digitalRead(btnStop) == HIGH)
{
    while (1)
    {
        if (digitalRead(btnStop) == LOW)
        {
            return btnStop;
        }
    }
}

if (digitalRead(btnPlus) == HIGH)
{
    while (1)
    {
        if (digitalRead(btnPlus) == LOW)
        {
            return btnPlus;
        }
    }
}

if (digitalRead(btnMinus) == HIGH)
{
    while (1)
    {
        if (digitalRead(btnMinus) == LOW)
        {
            return btnMinus;
        }
    }
}

return 0;
}

float askForSolutionVolume()
{
    question = "Zvolte objem roztoku";
    defaultValue = DEFAULT_SOLUTION_VOLUME;
    displayQuestion(question, defaultValue);
    return obtainSolutionVolume();
}

float obtainSolutionVolume()
{
    while (1) // smyčka ve které se čeká na nastavení objemu roztoku a
    následně potvrzení volby
    {
        switch (btnScan())
        {
            case btnStop:
            {
                break;
            }
            case btnStart:

```



```
    {
        return solutionVolume;
    }
    case btnPlus:
    {
        if (solutionVolume < 1.5)    // horni mez objemu roztoku
        {
            solutionVolume = solutionVolume + SOLUTION_VOLUME_STEP;
        }
        actualizeSolutionVolumeInSettingPage(solutionVolume);
        break;
    }
    case btnMinus:
    {
        if (solutionVolume > 0.5)    // dolni mez objemu roztoku
        {
            solutionVolume = solutionVolume - SOLUTION_VOLUME_STEP;
        }
        actualizeSolutionVolumeInSettingPage(solutionVolume);
        break;
    }
    default:
    {
        break;
    }
}
}
}

int askForTemp()
{
    question = "Zadejte teplotu.    ";
    defaultValue = DEFAULT_TEMP;
    displayQuestion(question, defaultValue);
    return obtainTempValue();
}

int obtainTempValue()
{
    currentTemp = getCurrentTemp();

    while (1)    // smyčka ve které se čeká na nastavení požadované teploty
    a následně potvrzení volby
    {
        switch (btnScan())
        {
            case btnStop:
            {
                break;
            }
            case btnStart:
            {
                return requiredTemp;
            }
            case btnPlus:
            {
                if (requiredTemp < 50)    // horni mez teploty
                {
                    requiredTemp = requiredTemp + TEMP_STEP;
                }
                actualizeTempInSettingPage(requiredTemp);
            }
        }
    }
}
```

```
        break;
    }
    case btnMinus:
    {
        if (requiredTemp > 20)           // dolni mez teploty
        {
            requiredTemp = requiredTemp - TEMP_STEP;
        }
        actualizeTempInSettingPage(requiredTemp);
        break;
    }
    default:
    {
        break;
    }
}
}
}

unsigned long askForProcessTime()
{
    question = "Zadejte dobu procesu.";
    defaultValue = DEFAULT_PROCESS_TIME;
    displayQuestion(question, defaultValue);
    return obtainProcessTimeValue();
}

unsigned long obtainProcessTimeValue()
{
    while (1) // smyčka ve které se čeka na nastavení doby procesu a
    následně potvrzení volby
    {
        switch (btnScan())
        {
            case btnStop:
            {
                break;
            }
            case btnStart:
            {
                return requiredProcessTime;
            }
            case btnPlus:
            {
                requiredProcessTime = requiredProcessTime + TIME_STEP;
                actualizeTimeInSettingPage(requiredProcessTime);
                break;
            }
            case btnMinus:
            {
                if (requiredProcessTime > 5000)
                {
                    requiredProcessTime = requiredProcessTime - TIME_STEP;
                }
                actualizeTimeInSettingPage(requiredProcessTime);
                break;
            }
            default:
            {
                break;
            }
        }
    }
}
```

```
    }
  }
}

unsigned long computePreheatTime(int requiredTemp, float solutionVolume)
// vypočet obhadovane doby predehrevu
{
  currentTemp = getCurrentTemp();
  if (requiredTemp > currentTemp)
  {
    return (((solutionVolume * 4186 * (requiredTemp - currentTemp)) /
(HEATING_ELEMENT_POWER * 0.98))*1.5); //((objem roztoku * merna tepelna
kapacita * rozdil teplot) / (prikon topneho telesa * ucinnost topneho
telesa))
  }
  else
  {
    return 0;
  }
}

int askForPreheat(unsigned long preheatTime)
{
  question = "Zapnout predehrev? ";
  displayQuestion(question, defaultValue);
  return obtainPreheatOption(preheatTime);
}

int obtainPreheatOption(unsigned long preheatTime)
{
  while (1) // smyčka ve které se čeká na nastavení zda chceme
predehrev a následně potvrzení volby
  {
    switch (btnScan())
    {
      case btnStop:
      {
        break;
      }
      case btnStart:
      {
        return requiredPreheat;
      }
      case btnPlus:
      {
        requiredPreheat = 1;
        actualizePreheatOptionInSettingPage(requiredPreheat,
preheatTime);
        break;
      }
      case btnMinus:
      {
        requiredPreheat = 0;
        actualizePreheatOptionInSettingPage(requiredPreheat,
preheatTime);
        break;
      }
      default:
      {
        break;
      }
    }
  }
}
```

```
    }
  }
}

bool askForStart()
{
  question = "Spustit proces?";
  displaySummary(question, requiredProcessTime, requiredTemp,
  solutionVolume, requiredPreheat, preheatTime); // zobrazuje souhrn
  nastavenych parametru
  return obrtainStartAnswer();
}

bool obrtainStartAnswer()
{
  while (1) // smyčka ve které se čeká na konečné potvrzení nastavených
  parametru
  {
    switch (btnScan())
    {
      case btnStop:
      {
        return false;
      }
      case btnStart:
      {
        return true;
      }
      default:
      {
        break;
      }
    }
  }
}

int process(int requiredTemp, unsigned long requiredProcessTime, int
requiredPreheat, unsigned long preheatTime) // funkce obsluhující celý
proces
{
  if ((requiredPreheat > 0) && (requiredTemp > currentTemp)) //
  predehrev se zapíná pouze pokud jsme tak zvolili a pokud je nám
  požadována teplota vyšší než aktuální teplota
  {
    preheatFinished = preheat(requiredTemp, preheatTime); // čeká ve
    smyčce dokud se nedosáhne požadované teploty a pokud se během nějaké doby
    teploty nedosáhne zavolá se finishProces(FAILRULE_END)
  }
  else // pokud se predehrev nespustí, označí se jako úspěšný
  {
    preheatFinished = PREHEAT_SUCCES;
  }

  if (preheatFinished == PREHEAT_SUCCES) // pokud byl predehrev úspěšný,
  nebo jsme jej ani nezapínali, spustí se proces
  {
    startTime = millis();
    succesFinished = false;
    displayPreheatOrPrcessPage(preheatTime, requiredProcessTime);
    startPump();
    maxProcessTime = getMaxProcessTime(requiredProcessTime);
  }
}
```

```

    succesFinished = false;

    while (millis() < maxProcessTime) // smyčka pokud je proces leptání
spuštěn
    {
        if (isProcessFinished(startTime, requiredProcessTime)) //
automatické zastavení po vypršení času
        {
            succesFinished = finishProcess(AUTOMATIC_END);
            break;
        }
        else if
(isTimeForNextTimeAndTempActualize(nextTimeAndTempActualizeTime)) //
kontrola, regulace a aktualizace teploty a času
        {
            nextTimeAndTempActualizeTime = millis() +
ACTUALIZE_TIME_AND_TEMP_PERIOD;
            currentTemp = getCurrentTemp();
            actualizeTimeAndTempInPreheatOrProcessPage(startTime,
currentTemp, requiredTemp);
            actualizeLedColor(currentTemp, requiredTemp, 0);
            regulateTemp(requiredTemp, currentTemp);
        }
        if (btnStopPressed()) // manuální zastavení
        {
            succesFinished = finishProcess(MANUAL_END);
            break;
        }
    }

    if ((millis() >= maxProcessTime) && (succesFinished == false))
    {
        finishProcess(FAILRULE_END);
    }
}

unsigned long getMaxProcessTime(unsigned long requiredProcessTime)
{
    return (millis() + (requiredProcessTime * 2));
}

bool isProcessFinished(unsigned long startTime, unsigned long
requiredProcessTime)
{
    if (millis() > (startTime + requiredProcessTime)) // pokud je aktuální
čas větší než čas spuštění a požadovaný čas procesu dohromady
    {
        return true;
    }
    return false;
}

bool finishProcess(int endType)
{
    stopPump();
    stopHeat();
    displayFinishedPage(endType); // podle typu ukončení procesu vypisuje
různé zprávy

    while (1) // po skončení procesu se čeká na potvrzení tlačítkem start

```

```
{
    startFinishBuzzAndBlink(endType); // podle typu ukončení procesu
    různé bliká a pípa

    switch (endType)
    {
        case AUTOMATIC_END:
        {
            if (btnScan() == btnStart)
            {
                return true;
            }
        }
        case MANUAL_END:
        {
            if (btnScan() == btnStart)
            {
                return true;
            }
        }
        case FAILRULE_END:
        {
            if (btnScan() == btnStart)
            {
                return false;
            }
        }
        default:
        {
            break;
        }
    }
}

int preheat(int requiredTemp, unsigned long preheatTime) // funkce
obsluhující proces předehřevu
{
    startTime = millis();
    preheatFinished = PREHEAT_FAILED;
    displayPreheatOrProcessPage(preheatTime, requiredProcessTime);
    startPump();
    maxPreheatTime = getMaxPreheatTime(preheatTime);

    while (millis() < maxPreheatTime) // smyčka se opakuje dokud se
nedosáhne maximální doby předehřevu, resp. požadované teploty
    {
        if (isPreheatFinished(requiredTemp, currentTemp)) // automatické
zastavení po dosažení požadované teploty
        {
            preheatFinished = finishPreheat(AUTOMATIC_END);
            return preheatFinished;
        }
        else if
(isTimeForNextTimeAndTempActualize(nextTimeAndTempActualizeTime))
        {
            nextTimeAndTempActualizeTime = millis() +
ACTUALIZE_TIME_AND_TEMP_PERIOD;
            currentTemp = getCurrentTemp();
            actualizeTimeAndTempInPreheatOrProcessPage(startTime, currentTemp,
requiredTemp);
        }
    }
}
```

```
        actualizeLedColor(currentTemp, requiredTemp, 0);
        regulateTemp(requiredTemp, currentTemp);
    }
    if (btnStopPressed()) // manualni zastaveni
    {
        preheatFinished = finishPreheat(MANUAL_END);
        return preheatFinished;
    }
}

if ((millis() >= maxPreheatTime) && (preheatFinished ==
PREHEAT_FAILED))
{
    preheatFinished = finishPreheat(FAILRULE_END);
    return preheatFinished;
}
}

unsigned long getMaxPreheatTime(unsigned long preheatTime) // ziskani
maximalni doby predehrevu
{
    return (millis() + (1000 * preheatTime * 2));
}

bool isPreheatFinished(int requiredTemp, float currentTemp)
{
    if (currentTemp >= requiredTemp)
    {
        return true;
    }
    return false;
}

int finishPreheat(int endType)
{
    stopPump();
    stopHeat();
    displayFinishedPreheatPage(endType); // podle typu ukonceni procesu
vypisuje ruzne zpravy

    while (1) // po skonceni procesu se ceka na potvrzeni tlacitkem start
    {
        startFinishBuzzAndBlink(endType); // podle typu ukonceni procesu
ruzne bliká a pípa

        switch (endType)
        {
            case AUTOMATIC_END:
                {
                    switch (btnScan())
                    {
                        case btnStart:
                            {
                                return PREHEAT_SUCCES;
                            }
                        case btnStop:
                            {
                                return PREHEAT_FAILED;
                            }
                        default:
                            {

```

```
                break;
            }
        }
        break;
    }
    case MANUAL_END:
    {
        if (btnScan() == btnStart)
        {
            return PREHEAT_FAILED;
        }
        break;
    }
    case FAILRULE_END:
    {
        if (btnScan() == btnStart)
        {
            return PREHEAT_FAILED;
        }
        break;
    }
    default:
    {
        break;
    }
}

if (endType != FAILRULE_END)
{
    return PREHEAT_SUCCES;
}
return PREHEAT_FAILED;
}

bool isTimeForNextTimeAndTempActualize(unsigned long
nextTimeAndTempActualizeTime) // kontrola kroku pro aktualizace casu
{
    if (millis() > nextTimeAndTempActualizeTime)
    {
        return true;
    }
    return false;
}

float getCurrentTemp() // funkce ktera vraci teplotu senzoru ve
stupnich Celsia
{
    senzor.requestTemperatures();
    return (senzor.getTempCByIndex(0) + 1);
}

void regulateTemp(int requiredTemp, float currentTemp) // regulace
vytapani
{
    if ((currentTemp >= (requiredTemp + tempHysteresis))) // podminky pro
vypnutí topeni
    {
        stopHeat();
    }
}
```



```
    if ((currentTemp <= (requiredTemp - tempHysteresis))) // podmínky pro
    zapnutí topení
    {
        startHeat();
    }
}

void startHeat() // zapína topení
{
    if (digitalRead(Heat) == HIGH)
    {
        digitalWrite(Heat, LOW);
        if (UI == 1)
        {
            Serial.println("Heating ON");
        }
    }
}

void stopHeat() // vypína topení
{
    if (digitalRead(Heat) == LOW)
    {
        digitalWrite(Heat, HIGH);
        if (UI == 1)
        {
            Serial.println("Heating OFF");
        }
    }
}

void startPump() // zapína pumpu
{
    if (digitalRead(Pump) == HIGH)
    {
        digitalWrite(Pump, LOW);
        if (UI == 1)
        {
            Serial.println("Pump ON");
        }
    }
}

void stopPump() // vypína pumpu
{
    if (digitalRead(Pump) == LOW)
    {
        digitalWrite(Pump, HIGH);
        if (UI == 1)
        {
            Serial.println("Pump OFF");
        }
    }
}

void actualizeLedColor(float currentTemp, int requiredTemp, int endType)
// aktualizuje barvu led diod
{
    colorStep = ((requiredTemp - 10) / 6);
    if (colorStep > 0)
    {
```

```
    if (currentTemp < (10 + colorStep))
    { //fialova
        digitalWrite(redLed, LOW);
        digitalWrite(greenLed, HIGH);
        digitalWrite(blueLed, LOW);
    }
    if ((currentTemp >= (10 + colorStep)) && (currentTemp < (10 + 2 *
colorStep)))
    { //modra
        digitalWrite(redLed, HIGH);
        digitalWrite(greenLed, HIGH);
        digitalWrite(blueLed, LOW);
    }
    if ((currentTemp >= (10 + 2 * colorStep)) && (currentTemp < (10 + 3 *
colorStep)))
    { //azurova
        digitalWrite(redLed, HIGH);
        digitalWrite(greenLed, LOW);
        digitalWrite(blueLed, LOW);
    }
    if ((currentTemp >= (10 + 3 * colorStep)) && (currentTemp < (10 + 4 *
colorStep)))
    { //zelena
        digitalWrite(redLed, HIGH);
        digitalWrite(greenLed, LOW);
        digitalWrite(blueLed, HIGH);
    }
    if ((currentTemp >= (10 + 4 * colorStep)) && (currentTemp < (10 + 5 *
colorStep)))
    { //zluta
        digitalWrite(redLed, LOW);
        digitalWrite(greenLed, LOW);
        digitalWrite(blueLed, HIGH);
    }
    if ((currentTemp >= (10 + 5 * colorStep)))
    { //cervena
        digitalWrite(redLed, LOW);
        digitalWrite(greenLed, HIGH);
        digitalWrite(blueLed, HIGH);
    }
}
else
{
    switch (endType)
    {
        case MANUAL_END:
            { //bila
                digitalWrite(redLed, LOW);
                digitalWrite(greenLed, LOW);
                digitalWrite(blueLed, LOW);
                break;
            }
        case FAILRULE_END:
            { //cervena
                digitalWrite(redLed, LOW);
                digitalWrite(greenLed, HIGH);
                digitalWrite(blueLed, HIGH);
                break;
            }
        case AUTOMATIC_END:
            { //zelena
```

```
        digitalWrite(redLed, HIGH);
        digitalWrite(greenLed, LOW);
        digitalWrite(blueLed, HIGH);
        break;
    }
    default:
    { //zadna barva
        digitalWrite(redLed, HIGH);
        digitalWrite(greenLed, HIGH);
        digitalWrite(blueLed, HIGH);
        break;
    }
}
}
}

bool btnStopPressed() //vraci informaci o tom, jestli bylo stisknuto
tlacitko Stop (pro manualni zastaveni predehrevu / procesu)
{
    if (btnScan() == btnStop)
    {
        tone(Buzz, 1000, 200);
        lastBeepAndBlink = millis();
        return true;
    }
    return false;
}

void startFinishBuzzAndBlink(int endType) // bliká a pípa podle typu
ukončení predehrevu / procesu
{
    switch (endType)
    {
        case MANUAL_END:
        {
            if (millis() < (lastBeepAndBlink + 400))
            {
                actualizeLedColor(0, 0, endType);
            }
            else
            {
                actualizeLedColor(0, 0, 0);
            }
            break;
        }
        case FAILRULE_END:
        {
            if (millis() > (lastBeepAndBlink + 250))
            {
                actualizeLedColor(0, 0, 0);
            }
            if (millis() > (lastBeepAndBlink + 500))
            {
                lastBeepAndBlink = millis();
                tone(Buzz, 1000, 250);
                actualizeLedColor(0, 0, endType);
                break;
            }
        }
        case AUTOMATIC_END:
        {
```

```
        if (millis() > (lastBeepAndBlink + 750))
        {
            actualizeLedColor(0, 0, 0);
        }
        if (millis() > (lastBeepAndBlink + 1500))
        {
            lastBeepAndBlink = millis();
            tone(Buzz, 1000, 750);
            actualizeLedColor(0, 0, endType);
            break;
        }
    }
    default:
    {
        break;
    }
}
}

void goToSleepMode()
{
    set_sleep_mode(SLEEP_MODE_PWR_DOWN); //nastaveni sleep modu
    sleep_enable(); //povoleni sleep funkce
    attachInterrupt(0, afterWakeUp, FALLING); //prirazeni interruptu na pin
2 (tl. Minus)
    lcd.clear(); //vycistení lcd
    lcd.noBacklight(); //vypnutí podsvícení lcd
    actualizeLedColor(0, 0, 0); //vypnutí led
    sleep_mode(); //zapnutí sleep modu - ceka se na interrupt (tl.Start)
    sleep_disable(); //probuzení ze sleep modu - po stisknutí tl. Start
    detachInterrupt(0); //odebrání interruptu z pinu 2 (tl. Minus)
    lcd.backlight(); //zapnutí podsvícení lcd
}

void afterWakeUp() //funkce obsluhující interrupt - nepotřebujeme
{
    ;
}

int displayMenu()
{
    displayWelcomePage();

    while (1) // smyčka ve které se čeká na volbu funkce
    {
        switch (btnScan())
        {
            case btnStop:
            {
                return MANUAL;
            }
            case btnStart:
            {
                return AUTOMATIC;
            }
            case btnPlus:
            {
                return NOTHING;
            }
            case btnMinus:
            {
```

```
        return EXIT;
    }
    default:
    {
        currentTemp = getCurrentTemp();
        actualizeLedColor(currentTemp, requiredTemp, 0);
        break;
    }
}
}
}

void displayWelcomePage()
{
    if (UI == 1)
    {
        Serial.println("Vítejte u poloautomatické leptací stanice. ");
        Serial.println("Stiskem tlačítka 'Stop' přejdete na nastavení objemu
roztoku, teploty a času. Stiskem 'Start' zahájíte proces s defaultními
parametry.");
        Serial.println("Defaultní nastavení je: 40 °C, 10 min a 0,5 l
leptacího roztoku. ");
        Serial.print("Aktuální nastavení je: ");
        Serial.print(requiredTemp);
        Serial.print(" °C, ");
        Serial.print((requiredProcessTime / 1000) / 60);
        Serial.print(" min ");
        Serial.print((requiredProcessTime / 1000) % 60);
        Serial.print(" sec a ");
        Serial.print(solutionVolume);
        Serial.println(" l leptacího roztoku.");
        Serial.print("Aktuální teplota je: ");
        Serial.print(currentTemp);
        Serial.println(" °C.");
        Serial.println(" ");
    }
    else
    {
        lcd.setCursor(0, 0);
        lcd.print("Hi. 'Start' s nast. ");
        lcd.setCursor(0, 1);
        lcd.print("nast.= ");
        lcd.print(requiredTemp);
        lcd.print(" C, ");
        lcd.print((requiredProcessTime / 1000) / 60);
        lcd.print("m ");
        lcd.print((requiredProcessTime / 1000) % 60);
        lcd.print("s ");
        lcd.setCursor(0, 2);
        lcd.print("a ");
        lcd.print(solutionVolume);
        lcd.print(" l lept. rozt.");
        lcd.setCursor(0, 3);
        lcd.print("'Stop' pro nastavení");
    }
}

void displayQuestion(String question, unsigned long defaultValue)
{
    if (UI == 1)
    {
```

```
    Serial.println(" ");
    Serial.println(question);
    Serial.println("Stiskem tlačítka '+' nebo '-' měníte hodnotu.");
    Serial.println("Stiskem tlačítka 'Start' potvrďte nastavení.");
    Serial.println(" ");
}
else
{
    lcd.setCursor(0, 0);
    lcd.print(question);
    lcd.setCursor(0, 1);
    lcd.print("Stiskem tlačítek + -");
    lcd.setCursor(0, 2);
    lcd.print("nastavujete hodnotu.");
    lcd.setCursor(0, 3);
    lcd.print("Start potvrdí hod. ");
}
}

void actualizeSolutionVolumeInSettingPage(float solutionVolume)
{
    if (UI == 1)
    {
        Serial.print("Objem roztoku nastaven na: ");
        Serial.print(solutionVolume);
        Serial.println(" l. ");
    }
    else
    {
        lcd.setCursor(0, 0);
        lcd.print("                ");
        lcd.setCursor(0, 1);
        lcd.print(" Objem nastaven na: ");
        lcd.setCursor(0, 2);
        lcd.print("                ");
        lcd.print(solutionVolume);
        lcd.print(" l                ");
        lcd.setCursor(0, 3);
        lcd.print("                ");
    }
}

void actualizeTempInSettingPage(int requiredTemp)
{
    if (UI == 1)
    {
        Serial.print("Teplota nastavena na: ");
        Serial.print(requiredTemp);
        Serial.println(" °C. ");
    }
    else
    {
        lcd.setCursor(0, 0);
        lcd.print("Teplota nastavena na");
        lcd.setCursor(0, 1);
        lcd.print("                ");
        lcd.print(requiredTemp);
        lcd.print(" C                ");
        lcd.setCursor(0, 2);
        lcd.print("                ");
        lcd.setCursor(0, 3);
    }
}
```

```
        lcd.print("Aktualni temp: ");
        lcd.print(currentTemp);
    }
}

void actualizeTimeInSettingPage(unsigned long requiredProcessTime)
{
    if (UI == 1)
    {
        Serial.print("Cas nastaven na: ");
        Serial.print((requiredProcessTime / 1000) / 60);
        Serial.print(" min ");
        Serial.print((requiredProcessTime / 1000) % 60);
        Serial.println(" sec.");
    }
    else
    {
        lcd.setCursor(0, 0);
        lcd.print("                ");
        lcd.setCursor(0, 1);
        lcd.print(" Cas nastaven na: ");
        lcd.setCursor(0, 2);
        lcd.print("                ");
        lcd.print((requiredProcessTime / 1000) / 60);
        lcd.print(" min ");
        lcd.print((requiredProcessTime / 1000) % 60);
        lcd.print(" sec                ");
        lcd.setCursor(0, 3);
        lcd.print("                ");
    }
}

void actualizePreheatOptionInSettingPage(int requiredPreheat, unsigned
long preheatTime)
{
    if (UI == 1)
    {
        Serial.print("Predehrev nastaven na: ");
        if (requiredPreheat == 1)
        {
            Serial.print("ZAPNUTO");
            Serial.print(" (bude trvat priblizne: ");
            Serial.print(preheatTime / 60);
            Serial.println(" min)");
        }
        else
        {
            Serial.println("VYPNUTO");
        }
    }
    else
    {
        if (requiredPreheat == 1)
        {
            lcd.setCursor(0, 1);
            lcd.print("                ZAPNUTO                ");
            lcd.setCursor(0, 2);
            lcd.print("Bude trvat priblizne");
            lcd.setCursor(0, 3);
            lcd.print("                ");
            lcd.print(preheatTime / 60);
        }
    }
}
```

```
        lcd.print(" min          ");
    }
    else
    {
        lcd.setCursor(0, 1);
        lcd.print("          VYPNUTO          ");
        lcd.setCursor(0, 2);
        lcd.print("          ");
        lcd.setCursor(0, 3);
        lcd.print("          ");
    }
}
}

void displaySummary(String question, unsigned long requiredProcessTime,
int requiredTemp, float solutionVolume, int requiredPreheat, unsigned
long preheatTime)
{
    if (UI == 1)
    {
        Serial.println(" ");
        Serial.print(question);
        Serial.println("Start = OK, Stop = navrat k nastaveni.");
        Serial.print("Nastaveni: Teplota ");
        Serial.print(requiredTemp);
        Serial.print(" C; Cas ");
        Serial.print((requiredProcessTime / 1000) / 60);
        Serial.print(" min ");
        Serial.print((requiredProcessTime / 1000) % 60);
        Serial.print(" sec; Objem roztoku ");
        Serial.print(solutionVolume);
        Serial.print(" l; Predehrev ");
        if (requiredPreheat == 1)
        {
            Serial.print("ZAPNUTO");
            Serial.print(" (bude trvat priblizne: ");
            Serial.print(preheatTime / 60);
            Serial.println(" min)");
        }
        else
        {
            Serial.println("VYPNUTO");
        }
    }
    else
    {
        lcd.setCursor(0, 0);
        lcd.print(question);
        lcd.print("          ");
        lcd.setCursor(0, 1);
        lcd.print("Temp: ");
        lcd.print(requiredTemp);
        lcd.print(" C, Cas:          ");
        lcd.setCursor(0, 2);
        lcd.print((requiredProcessTime / 1000) / 60);
        lcd.print(" min ");
        lcd.print((requiredProcessTime / 1000) % 60);
        lcd.print(" sec, Objem");
        lcd.setCursor(0, 3);
        lcd.print(solutionVolume);
        lcd.print(" l, Predehrev ");
    }
}
```



```
    if (requiredPreheat == 1)
    {
        lcd.print("ON ");
    }
    else
    {
        lcd.print("OFF");
    }
}
}

void displayPreheatOrProcessPage(unsigned long preheatTime, unsigned long
requiredProcessTime)
{
    if (UI == 1)
    {
        if (preheatFinished == PREHEAT_FAILED)
        {
            Serial.print("Predehrev je zapnuty. Bude trvat priblizne: ");
            Serial.print(preheatTime / 60);
            Serial.println(" min.");
        }
        else
        {
            Serial.println("Proces je zapnuty.");
        }
    }
    else
    {
        if (preheatFinished == PREHEAT_FAILED)
        {
            lcd.setCursor(0, 0);
            lcd.print("Predehrev ON. Bude ");
            lcd.setCursor(0, 1);
            lcd.print("trvat asi ");
            lcd.print(preheatTime / 60);
            lcd.print(" min ");
        }
        else
        {
            lcd.setCursor(0, 0);
            lcd.print("Proces je ON. Bude ");
            lcd.setCursor(0, 1);
            lcd.print("trvat ");
            lcd.print((requiredProcessTime / 1000) / 60);
            lcd.print(" min ");
            lcd.print((requiredProcessTime / 1000) % 60);
            lcd.print(" sec ");
        }
    }
}

void actualizeTimeAndTempInPreheatOrProcessPage(unsigned long startTime,
float currentTemp, int requiredTemp)
{
    if (UI == 1)
    {
        Serial.print("Uplynula doba: ");
        Serial.print(((millis() - startTime) / 1000) / 60);
        Serial.print(" min ");
        Serial.print(((millis() - startTime) / 1000) % 60);
    }
}
```

```
    if (requiredPreheat == 0)
    {
        Serial.print(" sec (z celkove doby ");
        Serial.print((requiredProcessTime / 1000) / 60);
        Serial.print(" min ");
        Serial.print((requiredProcessTime / 1000) % 60);
        Serial.println(" sec).");
    }
    else
    {
        Serial.print(" sec (z odhadovane doby ");
        Serial.print(preheatTime / 60);
        Serial.println(" min).");
    }
    Serial.print("Aktualni teplota: ");
    Serial.print(currentTemp);
    Serial.print(" C (pozadovana teplota: ");
    Serial.print(requiredTemp);
    Serial.println(" C).");
}
else
{
    lcd.setCursor(0, 2);
    lcd.print(" Cas: ");
    lcd.print(((millis() - startTime) / 1000) / 60);
    lcd.print(" min ");
    lcd.print(((millis() - startTime) / 1000) % 60);
    lcd.print(" sec ");
    lcd.setCursor(0, 3);
    lcd.print("Temp: ");
    lcd.print(currentTemp);
    lcd.print(" C (");
    lcd.print(requiredTemp);
    lcd.print(" C)");
}
}

void displayFinishedPreheatPage(int endType)
{
    if (endType == AUTOMATIC_END)
    {
        if (UI == 1)
        {
            Serial.print("Predehrev je u konce. Teplota je: ");
            Serial.print(currentTemp);
            Serial.println(" C.");
            Serial.println("Stiskem stiskem tlacitka Start zahajite proces");
            Serial.println("Stiskem stiskem tlacitka Stop se vratite k
nastaveni.");
        }
        else
        {
            lcd.setCursor(0, 0);
            lcd.print("Predehrev je u konce");
            lcd.setCursor(0, 1);
            lcd.print(" Teplota je: ");
            lcd.print(currentTemp);
            lcd.print(" C. ");
            lcd.setCursor(0, 2);
            lcd.print("Start: spusti proces");
            lcd.setCursor(0, 3);

```

```
        lcd.print("Stop: z5 k nastaveni");
    }
}
if (endType == MANUAL_END)
{
    if (UI == 1)
    {
        Serial.print("Predehrev byl prerusen. Teplota je: ");
        Serial.print(currentTemp);
        Serial.println(" C.");
        Serial.println("Stiskem stiskem tlacitka Start se vratite k
nastaveni.");
    }
    else
    {
        lcd.setCursor(0, 0);
        lcd.print("Predehrev prerusen ");
        lcd.setCursor(0, 1);
        lcd.print(" Teplota je: ");
        lcd.print(currentTemp);
        lcd.print(" C. ");
        lcd.setCursor(0, 2);
        lcd.print("Tlacitkem 'Start' se");
        lcd.setCursor(0, 3);
        lcd.print("vratite k nastaveni ");
    }
}
if (endType == FAILRULE_END)
{
    if (UI == 1)
    {
        Serial.println("Predehrev byl neuspesny.");
        Serial.println("Zkontrolujte, zda je vse radne zapojeno.");
        Serial.println("Pokracujte stiskem tlacitka Start");
    }
    else
    {
        lcd.setCursor(0, 0);
        lcd.print("Predehrev neuspesny ");
        lcd.setCursor(0, 1);
        lcd.print("Zkontrolujte, zda je");
        lcd.setCursor(0, 2);
        lcd.print("vse radne zapojeno.");
        lcd.setCursor(0, 3);
        lcd.print("Pokracujte tl. Start");
    }
}
}

void displayFinishedPage(int endType)
{
    if (endType == AUTOMATIC_END)
    {
        if (UI == 1)
        {
            Serial.println("Proces je u konce.");
            Serial.println("Pokracujte stiskem tlacitka Start.");
        }
        else
        {
            lcd.setCursor(0, 0);
        }
    }
}
```

```
        lcd.print("Proces je u konce.  ");
        lcd.setCursor(0, 1);
        lcd.print("Pokracujte stiskem  ");
        lcd.setCursor(0, 2);
        lcd.print("tlacitka 'Start'  ");
        lcd.setCursor(0, 3);
        lcd.print("                ");
    }
}
if (endType == MANUAL_END)
{
    if (UI == 1)
    {
        Serial.print("Proces byl prerusen. Uplynula doba je: ");
        Serial.print(((millis() - startTime) / 1000) / 60);
        Serial.print(" min ");
        Serial.print(((millis() - startTime) / 1000) % 60);
        Serial.println(" sec.");
        Serial.println("Pokracujte stiskem tlacitka Start.");
    }
    else
    {
        lcd.setCursor(0, 0);
        lcd.print("Proces byl prerusen ");
        lcd.setCursor(0, 1);
        lcd.print("Uplynula doba je:  ");
        lcd.setCursor(0, 2);
        lcd.print("                ");
        lcd.print(((millis() - startTime) / 1000) / 60);
        lcd.print(" min ");
        lcd.print(((millis() - startTime) / 1000) % 60);
        lcd.print(" sec ");
        lcd.setCursor(0, 3);
        lcd.print("Pokracujte tl. Start");
    }
}
if (endType == FAILRULE_END)
{
    if (UI == 1)
    {
        Serial.println("Proces byl neuspesny.");
        Serial.println("Zkontrolujte, zda je vse radne zapojeno.");
        Serial.println("Pokracujte tlacitkem Start");
    }
    else
    {
        lcd.setCursor(0, 0);
        lcd.print("Proces byl neuspesny");
        lcd.setCursor(0, 1);
        lcd.print("Zkontrolujte, zda je");
        lcd.setCursor(0, 2);
        lcd.print("vse radne zapojeno.");
        lcd.setCursor(0, 3);
        lcd.print("Pokracujte tl. Start");
    }
}
}
```