**UNIVERZITA PARDUBICE**

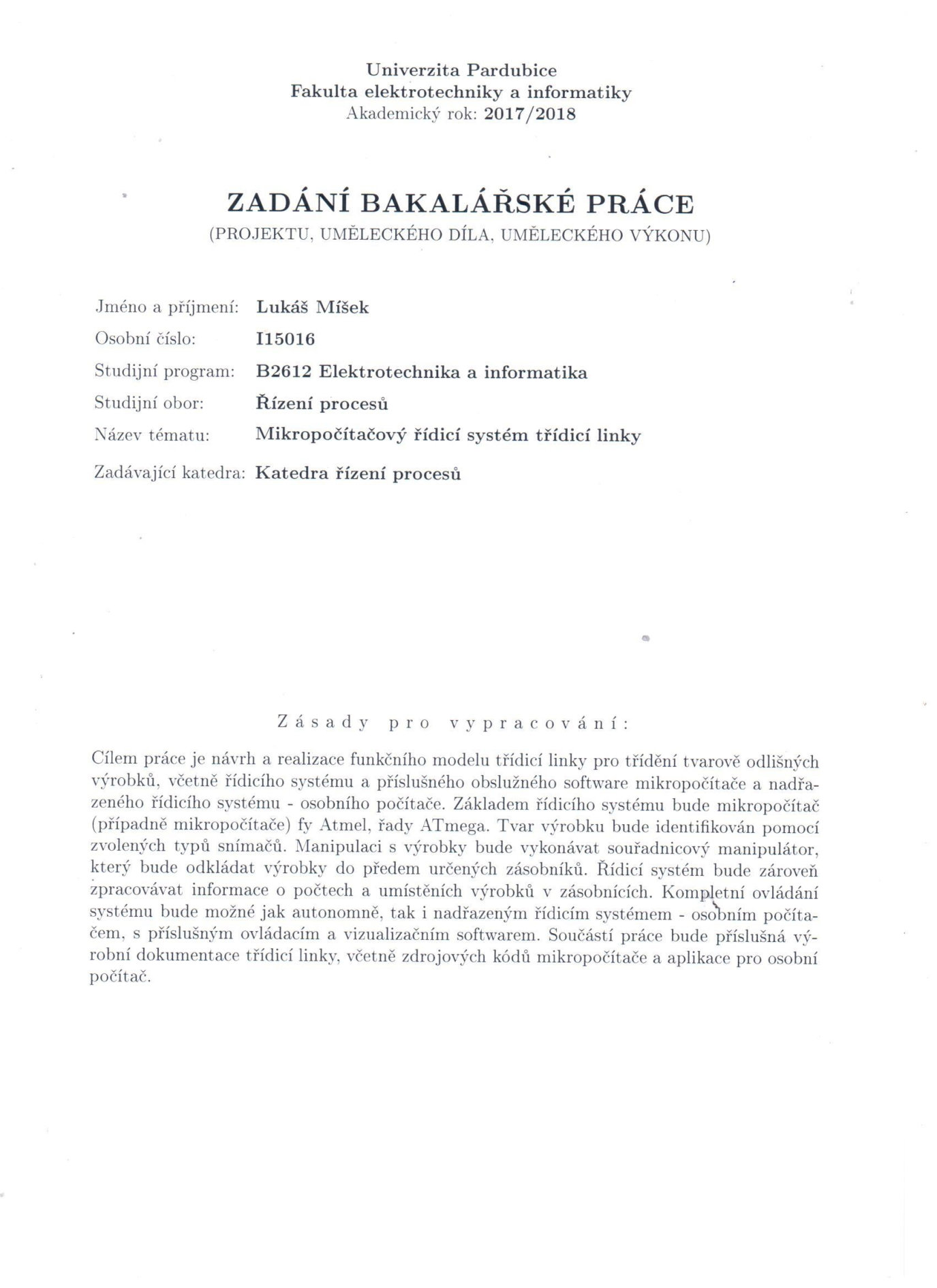
Fakulta elektrotechniky a informatiky

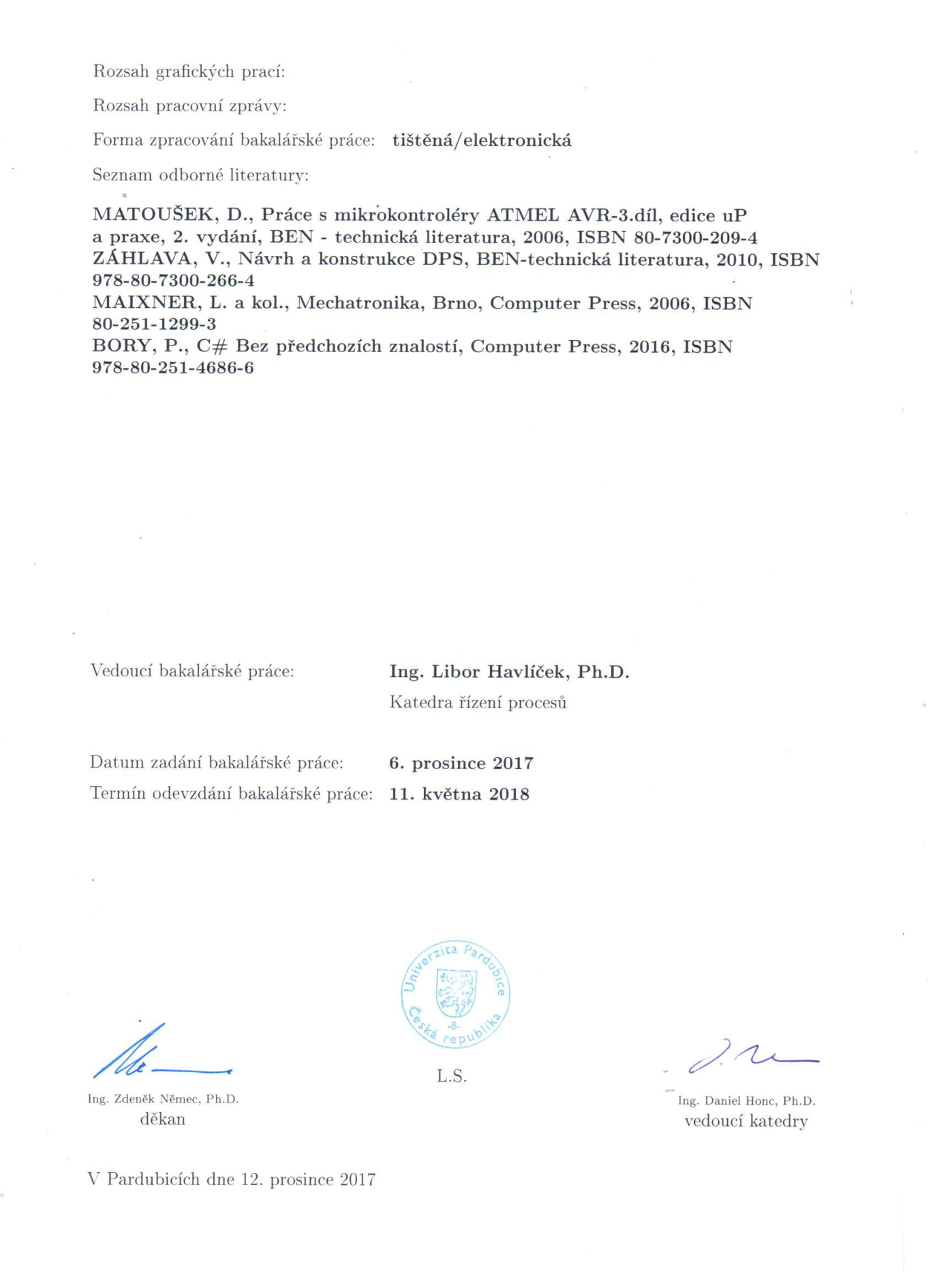
**Mikropočítačový řídicí systém třídící linky**

Lukáš Míšek

Bakalářská práce

2018



**

**Prohlášení**

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 29. 11. 2018

Lukáš Míšek

**ANOTACE**

*Cílem práce je návrh a realizace funkčního modelu třídicí linky pro třídění tvarově odlišných výrobků, včetně řídicího systému a příslušného obslužného software mikropočítače a nadřazeného řídicího systému – osobního počítače. Základem řídicího systému bude mikropočítač (případně mikropočítače) fy Atmel, řady ATmega. Kompletní ovládání systému bude možné jak autonomně, tak i nadřazeným řídicím systémem na osobním počítači, s příslušným ovládacím a vizualizačním softwarem. Součástí práce bude příslušná výrobní dokumentace třídicí linky, včetně zdrojových kódů mikropočítače a aplikace pro osobní počítač.*

**KLÍČOVÁ SLOVA**

*robotický manipulátor, mikropočítač, programování.*

**TITLE**

*Microcomputer control system of the sorting line*

**ANNOTATION**

*Goal of this work is to design and construct functional model of the sorting line to sort objects of the various shapes. Main building block of this system will be microcontroller (or microcontrollers) of company Atmel, series ATMEGA. Shape of the sorted object will be decided by group of chosen sensors. Complete control of the system will be both autonomous, or by superior system, the personal computer with corresponding elements and visual software.*

**KEYWORDS**

*microcontroller, programming, control system.*

[1 Rozbor zadání 11](#_Toc6787610)

[1.1 Rešerše literatury 11](#_Toc6787611)

[1.2 Návrh provedení 11](#_Toc6787612)

[1.3 Blokové schéma 13](#_Toc6787613)

[1.4 Historie automatizace 14](#_Toc6787614)

[1.5 Úvod do projektu 15](#_Toc6787615)

[2 Hardware 17](#_Toc6787616)

[2.1 Snímače 17](#_Toc6787617)

[2.1.1 Binární snímač polohy 17](#_Toc6787618)

[2.1.2 Clonící senzor rozměru 18](#_Toc6787619)

[2.2 Transportní zařízení 19](#_Toc6787620)

[2.2.1 Válečkové tratě 19](#_Toc6787621)

[2.2.2 Pásové dopravníky 19](#_Toc6787622)

[2.2.3 Řetězové dopravníky 19](#_Toc6787623)

[2.2.4 Karuselový dopravník 20](#_Toc6787624)

[2.3 Třídící zařízení 20](#_Toc6787625)

[2.3.1 Přesuvna 20](#_Toc6787626)

[2.3.2 Otočný stůl 21](#_Toc6787627)

[2.3.3 Výklopný shrnovač a tlačka 21](#_Toc6787628)

[2.3.4 Robotické rameno 22](#_Toc6787629)

[2.4 Mikrokontroler ATMEGA 23](#_Toc6787630)

[2.5 Programovací deska Arduino Uno 24](#_Toc6787631)

[2.6 Multiplexor a Demultiplexor 74HC4067 25](#_Toc6787632)

[2.7 Posuvný registr CD4021B 26](#_Toc6787633)

[2.8 Senzor tvaru 27](#_Toc6787634)

[2.9 Senzor tvaru s posuvnými registry 28](#_Toc6787635)

[2.10 LED Matice 8x8 KVM-20881CVB 29](#_Toc6787636)

[2.11 Krokový motor 31](#_Toc6787637)

[2.12 Servo motory SG90 a MSG90 32](#_Toc6787638)

[2.13 Řídící jednotka TB6560 33](#_Toc6787639)

[2.14 Testovací třídicí linka od firmy Merkur 34](#_Toc6787640)

[3 Konstrukce modelu třídící linky 35](#_Toc6787641)

[3.1 Konstrukce dopravníkového pásu 35](#_Toc6787642)

[3.2 Řídící obvod motoru GM37-3530 36](#_Toc6787643)

[3.3 Řídící obvod krokového motoru 28Byj-48 37](#_Toc6787644)

[3.4 Řídící obvod pro servo motory SG90 a MSG90 38](#_Toc6787645)

[3.5 Senzor tvaru 4x4 39](#_Toc6787646)

[3.6 Ovládací deska 40](#_Toc6787647)

[3.7 Tvorba vlastních mechanických součástek 42](#_Toc6787648)

[4 Programování mikrokontroleru 44](#_Toc6787649)

[4.1 Programování Mikrokontroleru pomocí Atmel studia 44](#_Toc6787650)

[4.2 Programování mikrokontroleru pomoci Arduino IDE 46](#_Toc6787651)

[4.3 Použité algoritmy mikrokontroleru 47](#_Toc6787652)

[4.3.1 Detekce stisku tlačítka mikrokontrolerem 47](#_Toc6787653)

[4.3.2 Rozšíření počtu vstupních pinů pomocí posuvného registru 47](#_Toc6787654)

[4.3.3 Rozsvícení LED, nebo LED matice 47](#_Toc6787655)

[4.3.4 Ovládání krokového motoru 28byj-48 48](#_Toc6787656)

[4.3.5 Ovládání servo motoru SG90 48](#_Toc6787657)

[4.3.6 Ovládání GM37-3530 motoru 48](#_Toc6787658)

[4.3.7 Řízení toku programu 49](#_Toc6787659)

[4.4 Program pro mikrokontroler 50](#_Toc6787660)

[4.4.1 Hlavní program 50](#_Toc6787661)

[4.4.2 Inicializační blok Setup a blok SerialEvent 51](#_Toc6787662)

[4.4.3 Blok Loop 51](#_Toc6787663)

[4.4.4 Blok Manual mode 52](#_Toc6787664)

[4.4.5 Obslužné funkce akčních členů 53](#_Toc6787665)

[4.4.6 Detekce tvaru výrobku pomocí funkce SHAP 54](#_Toc6787666)

[4.4.7 Určení kombinace uložiště a tvaru pomocí funkce WHERE 55](#_Toc6787667)

[4.4.8 Automatický režim 56](#_Toc6787668)

[4.4.9 Funkce Default a bloky MoveDest 57](#_Toc6787669)

[4.5 Ukázka běhu programu 58](#_Toc6787670)

[5 Nadřazený systém 59](#_Toc6787671)

[5.1 Aplikace na počítači uživatele: 59](#_Toc6787672)

[5.2 Main window 59](#_Toc6787673)

[5.3 Menu Configuraiton 60](#_Toc6787674)

[5.4 Menu Data storage 61](#_Toc6787675)

[5.5 Menu Manual mode 62](#_Toc6787676)

[5.6 Menu Automat mode 63](#_Toc6787677)

[6 Seznam literatury: 64](#_Toc6787678)

[Obrázek 1: Blokové schéma systému 13](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787679)

[Obrázek 2: Kolo na hřídelí9 14](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787680)

[Obrázek 3: Páka10 14](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787681)

[Obrázek 4: Výroba pečiva11 15](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787682)

[Obrázek 5: Výroba automobilu12 16](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787683)

[Obrázek 6: Výroba integrovaných obvodů13 16](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787684)

[Obrázek 7: Světelná závora, binární snímač polohy14 17](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787685)

[Obrázek 8: Clonící senzor rozměru15 18](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787686)

[Obrázek 9: Princip clonícího senzoru rozěmru16 18](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787687)

[Obrázek 10: Válečková trať17 19](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787688)

[Obrázek 11: Pásový dopravník18 19](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787689)

[Obrázek 12: Řetězový dopravník19 19](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787690)

[Obrázek 13: Karuselový dopravník od firmy LOGSYS20 20](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787691)

[Obrázek 14: Přesuvna od firmy LOGSYS21 20](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787692)

[Obrázek 15: Otočný stůl od firmy FEIFER22 21](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787693)

[Obrázek 16: Výklopný shrnovač23 21](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787694)

[Obrázek 17: Robotický manipulátor od firmy Buddy Toys BCR 1024 22](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787695)

[Obrázek 18: Robotický manipulátor bez pohyblivých kloubů26 22](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787696)

[Obrázek 19: Mikrokontroler ATMEGA27 23](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787697)

[Obrázek 20: Vývojová deska Arduino Uno29 24](#_Toc6787698)

[Obrázek 21: Princip multiplexoru30 25](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787699)

[Obrázek 22: Integrovaný obvod 74HCT406732 25](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787700)

[Obrázek 23: Integrovaný obvod CD4014B34 26](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787701)

[Obrázek 25: Logika senzoru 27](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787702)

[Obrázek 26: Vnitřní zapojení LED Matice KVM-20881 CVB36 29](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787703)

[Obrázek 27: LED Matice KVM-20881 CVB37 29](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787704)

[Obrázek 28: Zapojení LED Matice k mikrokontroleru 30](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787705)

[Obrázek 29: Konstrukce krokového motoru39 31](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787706)

[Obrázek 30: Krokový motor 28byj-4840 31](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787707)

[Obrázek 31: Servo motor42 32](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787708)

[Obrázek 32: Servo motor43 32](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787709)

[Obrázek 33: Vnitřní zapojení krokového motoru45 33](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787710)

[Obrázek 34: Řídící jednotka TB656046 33](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787711)

[Obrázek 35: Motor GM37-353048 34](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787712)

[Obrázek 36: Dopravníkový pás firmy Merkur47 34](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787713)

[Obrázek 37: Tranzistor IRLZ44N49 36](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787714)

[Obrázek 38: GM37-3530 řídící obvod 36](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787715)

[Obrázek 39: Řídící jednotka pro 28Byj-48 37](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787716)

[Obrázek 40: Zapojení servo motorů 38](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787717)

[Obrázek 41: Senzor tvaru 4x4 39](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787718)

[Obrázek 42: Schéma ovládací desky 41](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787719)

[Obrázek 43: Návrh hlavice senzoru tvaru v programu DesignSpark Mechanical 4.0 42](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787720)

[Obrázek 44: Návrh výrobků v programu DesignSpark Mechanical 4.0 42](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787721)

[Obrázek 45: Návrh náhradních merkur součástek v programu DesignSpark Mechanical 43](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787722)

[Obrázek 46: Vývojové prostředí Atmel studio 7.0 44](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787723)

[Obrázek 47: AND-Load aplikace 45](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787724)

[Obrázek 48: Ukázka vývojového prostředí Arduino Uno 46](#_Toc6787725)

[Obrázek 49: Řadič ULN200358 48](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787726)

[Obrázek 50: Hlavní program 50](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787727)

[Obrázek 51: Blok Loop 51](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787728)

[Obrázek 52: Blok manual mode 52](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787729)

[Obrázek 53: Funkce S1UP 53](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787730)

[Obrázek 54: Funkce lineForward 53](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787731)

[Obrázek 55: Funkce S1DO 53](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787732)

[Obrázek 56: Funkce STUP 53](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787733)

[Obrázek 57: Funkce STDO 53](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787734)

[Obrázek 58: Detekce tvaru výrobku 54](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787735)

[Obrázek 59: Funkce WHERE 55](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787736)

[Obrázek 60: Automatický režim 56](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787737)

[Obrázek 61: Bloky MoveDest 57](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787738)

[Obrázek 62: Ukázka běhu programu 58](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787739)

[Obrázek 63: Main window 59](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787740)

[Obrázek 64: Menu Configuration 60](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787741)

[Obrázek 65: Menu Data storage 61](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787742)

[Obrázek 66: Menu manual mode 62](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787743)

[Obrázek 67: Menu Automatic mode 63](file:///C:\Users\Lukáš%20Míšek\Desktop\2019-04-11%20BP\MisekL2.docx#_Toc6787744)

**Seznam zkratek:**

Ucc ... Všechny řídící obvody, servo motory a krokový motor je napájen napětím 5V. Napájení těchto obvodů je označeno Ucc.  
UccLine ... Motor GM37-3530 dopravníkového pásu má jako jedinný napájecí napětí 12V.  
GND ... Označení pro uzemnění. Uzemnění spojuje všechny elektrické vodivé spojení se zemí.  
Latch ... Řídící signál pro použitý posuvný registr CD4021b. Signálem je řízena vnitřní pamět posuvného registru.  
Clock ... Hodinový signál pro použitý posuvný registr CD4021b. Signálem je řízeno čtení výstupu registru.  
Data ... Výstupní datový signál posuvného registru. Signál nese informace o vstupních pinech registru.

* 1. Rozbor zadání

Úkolem práce je navrhnout řídicí systém obsluhující třídící linku. Součástí tohoto systému bude dopravníkový pás, robotický manipulátor manipulující s výrobky a software zajištující činnost zmíněných komponentů a komunikaci s osobním počítačem.

* 1. Podobné robotické manipulátory

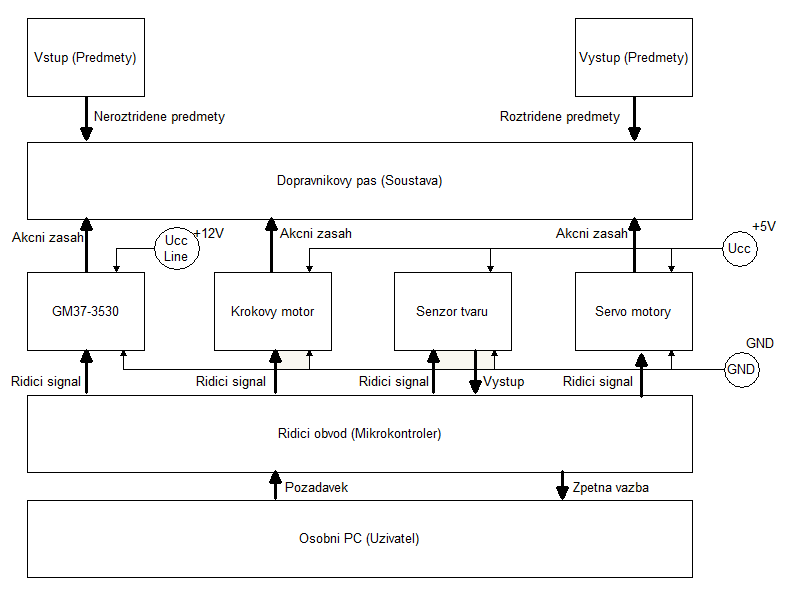
Dopravníkovým systémem, který dopravuje výrobky do dosahu robotického ramene, se zabývá diplomová práce Libora Barcucha1. V této práci jsou rozebrány druhy transportních zařízení, jako jsou válečkové tratě a pásové dopravníky, včetně jejich pohonu.   
 Robotickým ramenem se zabývá bakalářská práce Michaela Konečného2. V této práci je rozebrán princip funkce robotických ramen a dostupné open source druhy robotických ramen na internetu. Tohle robotické rameno může být použito jako akční člen. V systému jsou použity pouze jednoduché mechanické akční členy jako například tlačka. Více v nadcházejících kapitolách.  
 Software pro řízení ramene je zmíněn v pracích Michaela Konečného2, Tomáše Bendy3 a Michala Prokopa4. Nicméně ani jedna z prací se nezabývá pouze softwarem více než jednou kapitolou. Tohle bude jeden z hlavních bodů této práce. Podobná práce, která se zabývá dopravníkovým pásem je Pavla Vicana5.

Komunikací osobního počítače s vývojovým kitem Arduino se zabývá projekt, jehož autor je Nick Koumaris6. Podobný projekt pro komunikace osobního počítače a vývojového kitu Arduino se nazývá C# Serial Port Communication Arduino, kde autor vystupuje pod označením 3756684577.

* 1. Návrh provedení

Dopravníkový pás se bude pohybovat vpřed. Na trase výrobku bude několik výhybek, které budou pohánět servo motory. Samotný senzor tvaru bude poháněn krokovým motorem. Software ovládající tento systém bude složen z mikropočítače firmy Atmel, řady ATmega, ovládajícího snímače, dopravníkový pás, motory a bude komunikovat se softwarem na osobním počítači. Software pro osobní počítač bude v jazyce C# a bude umožňovat uživateli ovládat celý systém. Uživateli bude umožněno zjistit informaci o tom, kolik je výrobků v jednotlivých uložištích, možnost výběru, do kterých uložišti budou výrobky ukládány a možnost výběru tvarů z předdefinovaného seznamu, které bude systém schopen rozpoznat. Zároveň bude možné uživatelem vytvořit nový tvar výrobku, který systém rozezná.

* 1. Blokové schéma



Obrázek 1: Blokové schéma systému

Předmět putuje po dopravníku ze vstupu k senzoru tvaru. Pohyb zprostředkovává motor GM37-3530. Krokový motor následně předmět přitiskne na senzor tvaru. Mikrokontroler následně pomocí senzoru tvaru přečte tvar předmětu. Tvar předmětu je následně identifikován a řídicí logika určí cílové uložiště. Předmět následně putuje po dopravníku před servo motory, které vybočí předmět do předem určených uložišť.

Řídicí obvod ovládá všechny motory a pomocí řídicího signálu je možné si vyžádat tvar pomocí senzoru tvaru. Uživateli je celou dobu umožněno proces zastavit, znovu pustit, nebo přepínat mezi manuálním a automatickým režimem.

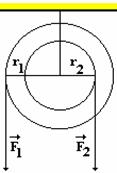
V automatickém režimu mikrokontroler dopravuje výrobky před senzor tvaru, následně přitiskne výrobek na senzor, vyhodnotí skutečný tvar a určí předem definované uložiště.

Během manuálního režimu je možné libovolně ovládat všechny akční členy, testovat jejich funkčnost, nebo uměle vytvářet události. Celý proces je zaznamenáván a je samotný záznam exportovat do textového souboru.

* 1. Historie automatizace

Již od počátku lidstva člověk využíval nástroje kolem sebe pro vlastní užitek, ať už to bylo pro přežití, nebo pro všední užití. Původně byly tyto nástroje, případně potraviny vyráběny přímo na místě, kde byly nalezeny, nebo sklizeny.

Postupem času se začaly vyvíjet specializované dílny, které měly za účel vyrábět nějaký produkt. Tyto dílny byly poměrně velkým přínosem pro společnost, protože umožnili mnohem efektivnější a rychlejší výrobu. Další výhodou byl fakt, že bylo možno lépe koordinovat větší výrobní procesy, ať už z pohledu samotné výroby, logistiky, nebo prodeje produktu. 8



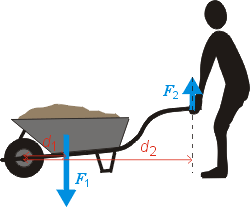
Obrázek 2: Kolo na hřídelí9

První manufaktury na našem území v Čechách začaly vznikat ve čtrnáctém století. Tyto manufaktury byly velké dílny, které soustředily větší množství dělníků, za účelem efektivnější výroby.

V této fázi bylo k dispozici v dostatečné míře množství odborníků a dělníků na jednom místě, společně s materiálem. Nicméně dalším krokem k zefektivnění postupu mnoha primitivních úkonů bylo právě použití robotů.

Využití robotů v dějinách lidstva se datuje už do starověku, kde se vyskytovaly první jednoduché stroje jako kladky, kola na hřídeli, nebo klíny v podobě padacích mostů, studen, nebo poloautomatických výtahů, poháněných buď lidskou silou, nebo zvířaty.

Výhodou robota oproti člověku je schopnost pracovat prakticky bez přestávky. Je nutná pouze přestávka pro případnou údržbu. Robot je schopen vykonávat velmi těžkou práci po velmi dlouhou dobu. A na rozdíl od člověka, s vysokou přesností, rychlostí a v podmínkách, kde člověk pracovat nemůže.



Obrázek 3: Páka10

Problémem do nedávna byl pravděpodobně zdroj energie, kterým by bylo možné takový systém pohánět. Tento problém byl prakticky vyřešen zavedením parního pohonu a později elektrickým proudem do výrobního procesu. Tento skok v technologii umožnil existenci velkých automatických továren, které jsou pouze kontrolovány člověkem, a fungují samostatně.

Dalším problémem je efektivnost těchto robotů. V případě, že je robot navržen a realizován, stále je tu otázka, jestli je robot dostatečně výkonný, aby nahradil lidského operátora.

* 1. Úvod do projektu

V dnešní době je velké množství produkce právě automatizované. Mezi tyto produkty patří nejrůznější potraviny, které denně konzumujeme, nebo výrobky jako je elektronika, dopravní prostředky, či přenos dat, jako je například internet. Tohle všechno pro nás dělají roboty.



Obrázek 4: Výroba pečiva11

Cílem této práce je vytvořit automatický systém, který je schopen na základě požadavku od obsluhy, vlastního programu a s přísunem energie, samostatně pracovat a vykonávat činnost, která kompletně nahradí lidského operátora. Tento systém bude schopen s menšími modifikacemi spolupracovat s libovolnými akčními členy v libovolném prostředí.

Hlavní myšlenka systému je umožnit velmi rychlým a přesným způsobem přemisťovat a třídit předměty, které budou dále použity v procesu, buď na další třídící lince, nebo budou představovat vstup do jiného výrobního procesu, či koncové výrobky.

Systém sám o sobě je nezávislý na vstupních předmětech, se kterými bude manipulovat. Parametry těchto předmětu jsou vždy určitým způsobem identifikovány senzory. Vzhledem k velkému množství dostupných materiálů, ze kterých mohou být sestrojeny komponenty jako akční členy, dopravníky, nebo samotné předměty, můžeme systém přizpůsobit potravinářským produktům, jak pevným, tak kapalným produktům, nebo i substancím, které jsou člověku nebezpečné.

Celá soustava se skládá z dopravníku, akčního členu, nebo členů a řídící jednotky. Dopravník je zařízení, které přemísťuje předměty do dosahu akčních členů. Jeho varianty jsou popsány v dalších kapitolách. Skupina akčních členů představuje jednotlivé třídící prvky a robotické manipulátory, které mohou předměty na dopravníku třídit, či libovolným způsobem s nimi manipulovat.



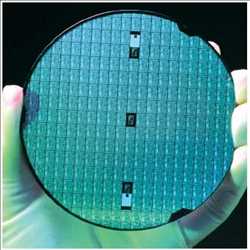
Obrázek 5: Výroba automobilu12

Příkladem systému je výrobní linka v pekárně, nebo i výrobní proces integrovaných obvodů, které jsou příliš malé pro člověka, než aby s nimi efektivně manipuloval.

Tento konkrétní system byl určen ke třídění předmětů podle jejich tvaru. Nicméně je možné třídit výrobky podle jiných fyzikálních vlastností. Některé z těchto vlastností jsou barva, odrazivost, elektrický odpor, teplota a mnoho dalších.

Automatický režim systému je možné použít k sériové výrobě různých mechanických předmětů, jejich transportu, nebo jako součást většího samoobslužného systému.

Ruční režim je možné použít v systémech, kde se vyskytuje příliš mnoho proměnných. Například jako součást výrobního stroje, který bude tisknout popisky na klávesnicích, kde obsluha bude pouze nastavovat konkrétní předlohu, která má být vytištěna.



Obrázek 6: Výroba integrovaných obvodů13

1. Hardware

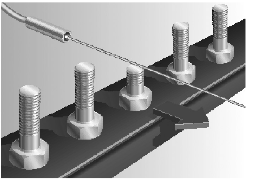
V této kapitole jsou popsány všechny hardwarové součásti celého systému. Systém se skládá z několika motorů, dopravníkového pásu a mikro kontroléru pro ovládání celého procesu a komunikaci s uživatelem.  
 Systém byl sestaven ze stavebnice Merkur a několika dalších součástek, které jsou použity pro uchycení akčních členů. Jmenovitě se jedná o vývojový kit Arduino Uno, servo motory SG90 a MG90S, krokový motor 28BYJ-48, elektromotor GM37-3530 a další elektronické součástky jako jsou rezistory, tranzistory, spínače a svorkovnice. Celý systém se skládá ze samotného dopravního pásu, připojovací desky a senzoru tvaru.

* 1. Snímače

Snímač je zařízení, které detekuje určitou fyzikální veličinu a podle vnitřní logiky zjištěnou informaci vyhodnocuje a následně posílá výstupní signál. Některé snímače vrací kalibrovaný signál, například počet otáček, vzdálenost, úhel. Jiné snímače vysílají nekalibrovaný signál, který bude potřeba změřit a pomocí vhodných převodníků převést na požadovanou hodnotu.

* + 1. Binární snímač polohy

Binární snímač polohy snímá polohu a rozhoduje, jestli mezi detektorem a vysílačem signálu je předmět, nebo není.



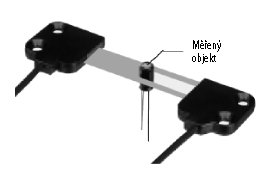
Obrázek 7: Světelná závora, binární snímač polohy14

Snímač funguje vysíláním signálu, který se odráží od takzvané závory, odrazové plochy, nebo samotného objektu, který má být snímán.

V systému bude tento typ snímače použit k detekci pohybu robotického manipulátoru. Robotický manipulátor se bude pohybovat po ose X směrem k uložištím předmětů. Snímač bude dodávat informaci o tom, jestli se robotický manipulátor nachází nad požadovaným místem.

* + 1. Clonící senzor rozměru

Clonící senzor rozměru funguje jako skupina binárních snímačů polohy. Tento snímač má určité rozlišení, které je realizováno skupinou detektorů.  
Například pokud mám 8 detektorů, potom je rozlišení 8. Tím pádem je senzor schopen rozlišit až 28 poloh.



Obrázek 8: Clonící senzor rozměru15

Clonící senzor rozměru snímá, dvoj rozměrnou plochu o konkrétním rozlišení a postupně posílá posloupnost 8 bitů, které jsou dále vyhodnocovány. Pohybem předmětu v dosahu senzoru vznikne informace o přibližném tvaru objektu. Podle tohoto tvaru je řídicí systém schopen rozeznat skutečný tvar.



Obrázek 9: Princip clonícího senzoru rozěmru16

* 1. Transportní zařízení

Transportní zařízení je takové zařízení, které přesouvá objekty na něm ležící. Součástí zařízení je motor, který pohání pásy, povrch linky. Linka může být tvořena válečkovými pásy, pásovými dopravníky, nebo karuselovými dopravníky. Podrobným popisem se zabývá práce Libora Barcucha1. Zde jsou zkráceně popsány nejčastější varianty, které mohou být použity v tomto systému.

* + 1. Válečkové tratě



Obrázek 10: Válečková trať17

Nejjednodušší skupina válečkových tratí není nijak poháněna. Jedná se o řadu k pevné konstrukci připevněných válečků, které se otáčí kolem své osy a umožnují se předmětu pohybovat téměř bez tření.

Pohon je realizován lidskou silou, která na předmět zapůsobí a předmět se následně pohybuje po pásu. Existují zároveň varianty válečkových tratí, které jsou poháněny motorem.



Obrázek 11: Pásový dopravník18

* + 1. Pásové dopravníky

Pásový dopravník je tvořen pružným pásem a motorem. Výhoda této varianty dopravníku je ta, že je možno dopravovat předměty i do vyšších výškových rozdílů.

Doprava probíhá položením předmětu na pás, který je následně přemístěn pomoci motoru, který pohání celou linku.

* + 1. Řetězové dopravníky

Řetězové dopravníky jsou dražší než předchozí varianty a jsou používány zejména pro sypké materiály a přesun palet.

Výhodou je, že je možné přesouvat i materiály za vysokých teplot a agresivní materiály. Součástí řetězového dopravníku je motor.



Obrázek 12: Řetězový dopravník19

* + 1. Karuselový dopravník

Obrázek 13: Karuselový dopravník od firmy LOGSYS20

Karuselový dopravník je používán zejména k přesunu předmětů v zatáčkách. Celý dopravník je tvořen segmenty spojenými řetězem, který je tažen motorem.

Tyto dopravníky se používají na letištích pro přepravu zavazadel, ale je možno je použít i ve výrobních prostředích, pokud není možno použít válečkový dopravník v zatáčce.

* 1. Třídící zařízení

Třídící zařízení může být takzvaná přesuvna, operátor, nebo robotické rameno. Přesuvna může být použita pro přemístění z jednoho dopravníku na jiný.

Robotické rameno je použito pro přemístění po složitější trajektorii. Například uchopit, zvednout vložit dovnitř do pece a za chvíli položit zpět na dopravník.



Obrázek 14: Přesuvna od firmy LOGSYS21

* + 1. Přesuvna

Přesuvna je kombinace dvou dopravníkových ploch, které jsem na sebe kolmé. Tímto způsobem je předmět pohybující se po dopravníku přesunut na druhý dopravník.

Tento třídicí prvek se většinou používá jako nepohyblivá část dopravníkového pásu. Předmět může být na přesuvnu vytlačen a následně silou setrvačnosti pokračuje ve své dráze.

* + 1. Otočný stůl



Obrázek 15: Otočný stůl od firmy FEIFER22

Otočný stůl je konstrukce, na které je připevněna válečková trať. Celá konstrukce může být vybočena motorem dočasně do jiného úhlu, čímž bude předmět poslán jiným směrem.

Otočný stůl je možné použít v kombinaci s některým akčním členem jako je servo motor. Servo motor může pohybovat konstrukcí otočného stolu, čím dochází ke třídění předmětů.

* + 1. Výklopný shrnovač a tlačka

Tyto dva třídiče jsou velmi podobné. Výklopný shrnovač je rám, co se točí motorem kolem své osy a posouvá předměty z dopravníku. Tlačka je součástka, která se posouvá vpřed a vzad a tímto posouvá předměty z dopravníku.

Třídicí prvek shrnovač byl použit například při tvorbě tohoto systému. Samotný shrnovač je připevněn na hřídel servo motoru, která na základě signálů od Mikrokontroleru posouvá hřídel mezi úhlu 0° a 90°. Tímto pohybem je vytvořená jednoduchá závora, která ovládá dráhu předmětu, který se nachází na dopravníkovém pásu.



Obrázek 16: Výklopný shrnovač23

* + 1. Robotické rameno



Obrázek 17: Robotický manipulátor od firmy Buddy Toys BCR 1024

Robotické rameno je konstrukce, která umožňuje uchopit a manipulovat s předmětem po osách X, Y a Z. Rameno obsahuje několika kloubů, ve kterých se může ohnout pro dosažení patřičných souřadnic. Nevýhodou těchto robotických ramen je vysoká cena a složitější ovládání. Na druhou stranu jsou schopny prakticky libovolných rozsahů pohybu.

Většinou se používají dva nebo více kloubů. První nejdůležitější kloub je mezi úchopnou hlavicí a samotným tělem ramena. Druhý kloub je obvykle ve spodní části ramena, aby se mohlo rameno sklonit a uchopit předmět vedle základny manipulátoru.



Obrázek 18: Robotický manipulátor bez pohyblivých kloubů26

Další variantou robotického ramena je rameno bez pohyblivých kloubů. Toto rameno je schopno pouze uchopit předmět. Aby se pohybovalo, je nutné jej připevnit k motoru, který by tento pohyb umožnil. Po konstrukční stránce se robotické ramena skládá ze senzorů pro detekci úhlu natočení kloubů a polohy, tímto způsobem je ovládáno a manipuluje s předmětem. Rameno je ovládáno vlastním řadičem, nebo mikro kontrolérem. Podrobnou konstrukcí robotického manipulátoru se zabývá Tomáš Skalický. 25

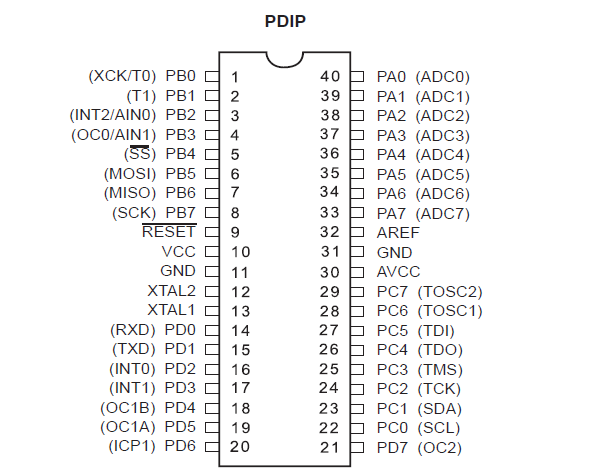
1. 1. Mikrokontrolér ATMEGA

Pro ovládání systému jsem použil mikro kontrolér ATMEGA od firmy Atmel. Jedná se o mikro kontrolér pracující s instrukcemi RISC architektury. K dispozici je zde i 32KBytová Flash paměť, 1024 Bytová EEPROM a 32x8 registrů použitelných pro data programu. Mikro kontrolér má k dispozici až 32 vstupně výstupních pinů, které zároveň mají mnoho nastavitelných módů jako je USART, kanál pro vysílání a přijímání dat.

Programovací prostředí, ve kterém je mikro kontrolér naprogramován se nazývá „Ardunio for Atmel Studio 7“ a programovací jazyk je podobný jazyku „Assembler“. Jednou ze stěžejních částí práce je rozpoznání a kalibrace vstupních signálů pro vstupně výstupní zařízení připojené na mikro kontrolér.

Další programovací prostředí se nazývá „Ardunio IDE“. Jedná se o podobný nástroj jako je „Ardunio for Atmel Studio 7“.

Mikro kontrolér v tomto systému funguje jako řídící prvek. Zároveň plní funkci prostředníka při komunikaci mezi systémem a osobním počítačem, ovládaným uživatelem.



Obrázek 19: Mikrokontroler ATMEGA27

Součástí Mikrokontroleru je napájecí napětí, které je možné použít k napájení některých nízkonapěťových akčních členů jako jsou servo motory.

Zároveň existuje mnoho modulů a samotných obvodů, které je možné kombinovat s Mikrokontrolerem. Všechny tyto obvody a moduly mají většinou stejné napájecí napětí jako Mikrokontroler ATMEGA.

Detailní popis mikro kontroléru, jeho funkcí a zapojení může být nalezen v manuálu poskytnutým výrobcem. 28

* 1. Programovací deska Arduino Uno

Jedná se vývojovou desku, obsahující mikro kontrolér ATMEGA 328. Na desce je port USB, který slouží k napojení se na počítač, přes který bude mikro kontrolér programován. Dále deska obsahuje 14 digitálních vstupních portů, 6 analogových vstupních portů. Pro testovací účely mohou být použity porty s vyvedeným napětím 5V, 3.3V a zemnícím potenciálem GND. Tohle byly hlavní porty, které jsem použil v práci.

Všech 14 digitálních vstupů jsem použil pro připojení snímače, který má 10 vstupů. Následně pro ovládací tlačítka jsou použity zbývající 4 piny.

Porty s vyvedeným napětím 5V, 3.3V a zemnícím potenciálem GND jsou použity zejména pro testovací účely, kdy jsem zobrazoval například vstup senzoru na LED matici.

Arduino Uno je vývojový kit, který umožňuje připojit mnoho modulů, které v sobě již kombinují některé navržené řídicí prvky, které byly pro tento systém navrženy. Tyto moduly většinou mají mnoho dalších funkcí, které nejsou pro tento konkrétní systém vhodné.

K programování samotného vývojového kitu byl použit konektor USB, který byl připojen na osobní počítač, kde byly napsány všechny programy, které byly následně nahrány do vývojového kitu.

V tomto konkrétním systému byly použity pouze základní funkce Mikrokontroleru. Pokročilé funkce jsou detailně popsány v technických manuálech pro patřičné Mikrokontrolery.



Obrázek 20: vývojová deska Arduino Uno29

* 1. Multiplexor a Demultiplexor 74HC4067

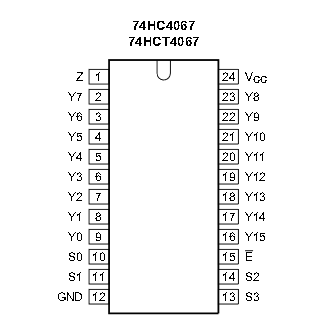
Vzhledem k tomu, že některé měřící členy, nebo akční členy potřebují více vstupních pinů na mikro kontroléru ATMEGA, je možné použít multiplexory.



Obrázek 21: Princip multiplexoru30

Multiplexory jsou integrované obvody, které se zaobírají manipulací se vstupy a výstupy. Obvod na základě řídících bitů přiřazuje výstupu vstupy, výstupu je přiřazen vstup nezávisle jaká data se na konkrétním vstupu nachází. Na vstupu může být libovolné množství datových vstupů, nebo výstupů, zvýší se tím pouze počet adresních vstupů. 31

Pin Z je společný vstup nebo výstup. Y piny jsou nezávislé vstupy, nebo výstupy. S piny jsou adresové vstupy. E pin, aktivní v logické nule slouží k aktivaci obvodu. VCC a GND slouží k napájení, obvod může být napájen mezi 0 až +10 V. Podrobnější popis integrovaného obvodu je k dispozici v technické dokumentaci. 32



Obrázek 22: Integrovaný obvod 74HCT406732

V případě tohoto systéu použiji napájení 5V. Je nutné použít multiplexor, protože senzor tvaru je tvořen 16 tlačítky. To znamená celkem 16 pinů, které je nutné mít k dispozici na vývojovém kitu pro připojení senzoru.

Multiplexor sníží tento počet vstupů na 7 a během přidání pouze jednoho dalšího řídicího signálu je možné rozšířit počet vstupů o dalších 8 vstupů.

* 1. Posuvný registr CD4021B

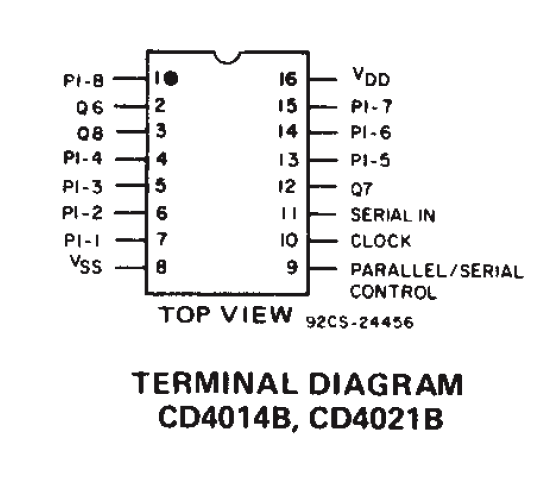
Druhou možností pro rozšíření vstupních, nebo výstupních pinů je použití posuvných registrů. Na rozdíl od multiplexoru a demultiplexoru tento prvek nepřepíná mezi vstupy, či výstupy, ale posílá jednotlivé stavy v sérii.

Posuvný registr má celkem 8 pinů pro vstupy, 2 piny pro napájení, 2 kontrolní piny a 5 pinů pro vstup a výstup. Piny 1, 4, 5, 6, 7, 13, 14 a 15 slouží jako vstupy. Na tyto vstupy je přiváděná informace, která má být zpracována dále. Piny 8 a 16 slouží k připojení napájení. Pin 10 slouží k připojení hodinového signálu, dále použiji označení Clock. Pin 9 slouží k ovládání zápisu vstupů do paměti posuvného registru, dále jako Read. Pin 11 slouží k propojení s jinými registry, dále SerialIn. Piny 2, 3, 12 a 11 slouží jako výstup.

Posuvný registr zapisuje stavy ze vstupů, když je na pinu Read logická 1. Když je na pinu Read logická 0, tak stavy ze vstupů si pouze pamatuje. Následně je možné tyto stavy číst z výstupu z pinu 3. Vždy, když je na pin Clock přiveden puls obdélníkového signálu, tak je na výstupní pin 3 přiveden následující stav. Tímto způsobem je možné číst všech 8 vstupních stavů pomocí jednoho výstupního pinu.

Pin SerialIn slouží k připojení dalšího posuvného registru pro připojení dalších 8 vstupů. V tomto případě výstupní pin 3 druhého registru se připojí na vstupní pin SerialIn prvního registru. Následně je možné obdobným způsobem číst až 16 stavů.

Výstupní piny 2 a 12 obsahují stejný stav jako výstupní pin 3, jen je tento stav posunut o 1, případně 2 doby. 33



Obrázek 23: Integrovaný obvod CD4014B34

Během konstrukce senzoru tvaru byly použity dva posuvné registry. Kombinací registrů bylo možné redukovat 16 výstupů ze senzoru na pouze 4 výstupy. Řídicí signály Clock, Read a Latch jsou pro oba posuvné registry společné. Výstup druhého registru je připojen na vstup prvního registru SerialIn. Případné další posuvné registry je možné zapojit výstupem na vstup předchozího registru SerialIn.

* 1. Senzor tvaru

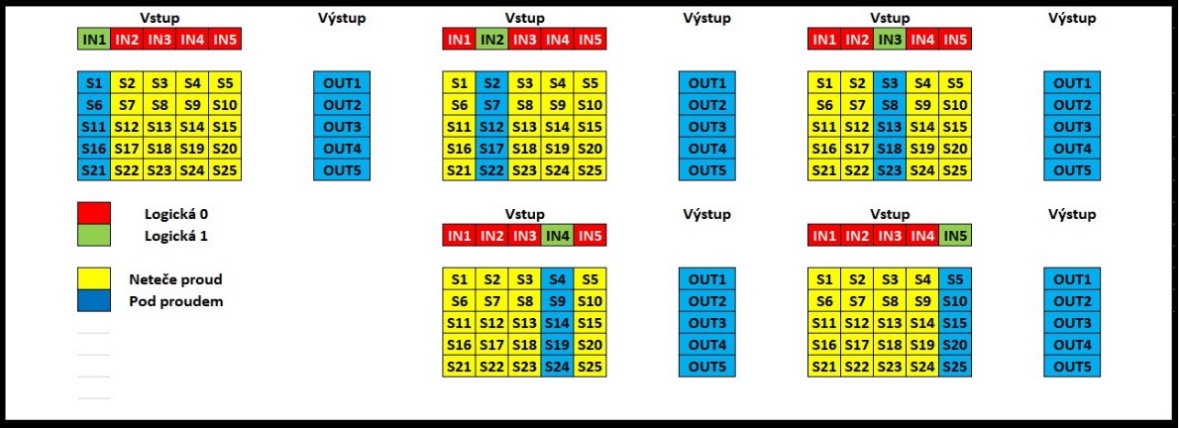
Pro účely detekce výrobků byl zhotoven senzor. Tento senzor je tvořen 25 tlačítky umístěných do matice 5x5. Každé tlačítko v matici představuje jeden bit. Tím pádem rozlišení senzoru je 5x5 bitů a je možné rozlišit až 255 tvarů.

Tlačítka jsou rovnoměrně umístěny od sebe, takovým způsobem, aby každý bit představoval část výrobku. Výrobek je přitisknut na senzor a každé tlačítko, které je stačeno po přitisknutí představuje aktivní bit. Tyto bity jsou vyhodnoceny a následně je sestaven tvar výrobku.

Snímač má celkem 5 vstupů a 5 výstupů. Čtení je navrženo tak, že vstupy jsou aktivovány v cyklech a výstupy jsou čteny ve vnořeném cyklu. Tímto způsobem není nutné mít na senzoru 25 výstupů, ale pouze 5.

Díky tomu, že výstupy jsou čteny v cyklech, je možné rozšířit snímač o další řád pouhým přidáním dvou pinů, to znamená 1 vstup a 1 výstup. Tím pádem pro 36 výstupů není potřeba 36 výstupních pinů, ale pouze 12.

Princip snímače spočívá v tom, že 5 vstupních pinů je připojeno na piny desky, které budou sloužit jako napájecí napětí. Zbylých 5 pinů je připojeno na piny desky, které budou sloužit ke čtení. Vždy bude aktivován pouze jeden vstupní pin a budou čteny výstupní piny. Tento děj se bude opakovat periodicky a mikro kontrolér tím pádem postupně přečte všechny sloupce.



Obrázek 24: Logika senzoru

Pokud je na vstup IN1 přivedena logická 1, potom celým sloupcem protéká proud a na výstupech OUT1-5 je hodnota podle toho, jestli je tvar výrobku na patřičném místě.

Celý senzor se skládá z 2 svorkovnic o 10 pinech a 25 tlačítkách. Tento senzor se prokázal být příliš komplikovaný, a proto byl nahrazen senzorem s posuvným registrem.

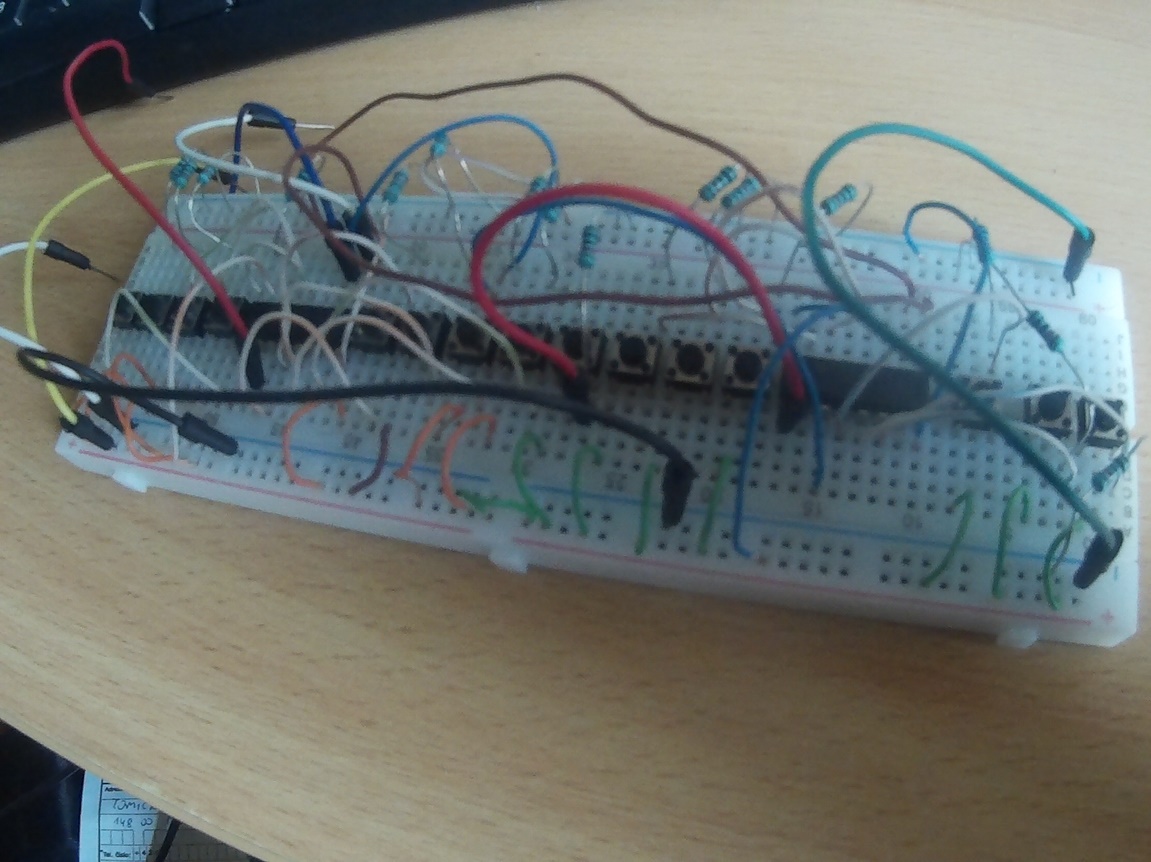
* 1. Senzor tvaru s posuvnými registry

Tato varianta senzoru se projevila jako jednodušší. Senzor je tvořen deskou a 16 tlačítky. Každé tlačítko představuje 1 bit rozlišení senzoru. Deska dále obsahuje 2 posuvné registry CD4021 a svorkovnice s 4 vstupy a 1 výstupem.

2 Vstupy slouží k připojení napájení o velikosti +5V a další dva vstupy slouží jako ovládací signály pro posuvný registr. Jedná se o obdélníkový signál Clock a signál pro čtení z tlačítek Read. Výstupní pin Data následně umožňuje číst stavy tlačítek. 33

Senzor funguje tak, že je celá deska přitisknuta na měřený výrobek. Podle tvaru výrobku budou stisknuty patřičné tlačítka, podle kterých je možné vyhodnotit tvar výrobku. Registrům je poslán puls z mikropočítače na pin desky Read. Posuvné registry si následně drží zaznamenané stavy z tlačítek. Následně je přiveden obdélníkový signál na pin Clock, a zároveň je čten výstupní pin Data. Každým pulsem je na výstupní pin Data poslán následující stav tlačítka. Tohle je provedeno pro všech 16 stavů. Senzor je následně odtáhnut od měřeného výrobku.

Pro účely bakalářské práce byl vytvořen prototyp na univerzálním pájivém spoji. Je možné realizovat senzor také na kontaktním nepájivém poli, nebo vytvořit desku s plošným spojem.



Obrázek 25: Senzor tvaru s posuvnými registry na nepájivém poli

* 1. LED Matice 8x8 KVM-20881CVB

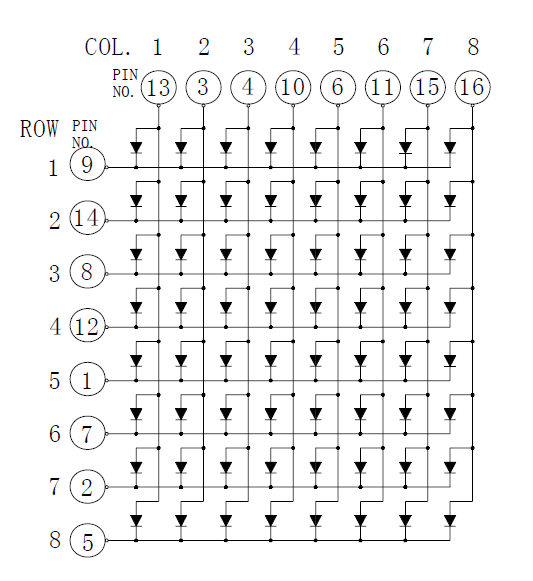
Jedná se o LED matici 8x8. Matice má celkem 16 vstupů. Z toho 8 pinů jsou vstupy pro napájecí napětí do 5v a dalších 8 pinů jsou vstupy pro zemnící potenciál. Je možné rozsvítit jednotlivé LED pomocí propojení patřičných pinů.

Pro testování jsem použil vývojový kit EvB 5.1 s mikro kontrolérem ATMEGA.

PortB je port, který připojím ke katodám LED, kde budu přivádět logickou 1 pro rozsvícení. PortC je port, který připojím na anody. Na anodách budu přivádět logickou 0. LED v matici se nezničí, protože mikro kontrolér ATMEGA má omezený výstupní proud pro každý výstupní pin. Tento proud má velikost 20 mA, což je hodnota, kterou LED snese.

Například, pro rozsvícení LED na pozici 5:5 (řádek: sloupec) je potřeba přivést na pin 6 logickou 1 a na pin 5 logickou 0. Tohle je realizováno dosazením patřičné hodnoty do výstupního registru mikro kontroléru ATMEGA.

Tím pádem je potřeba na výstupní port PortB přivést hodnotu 0b01000000 a na portC hodnotu 0b11011111. V tomhle stavu bude LED na pozici 5:5 zapojena v propustném směru a proto svítit. Ostatní LED nejsou pod napětím.



Obrázek 26: Vnitřní zapojení LED Matice KVM-20881 CVB36

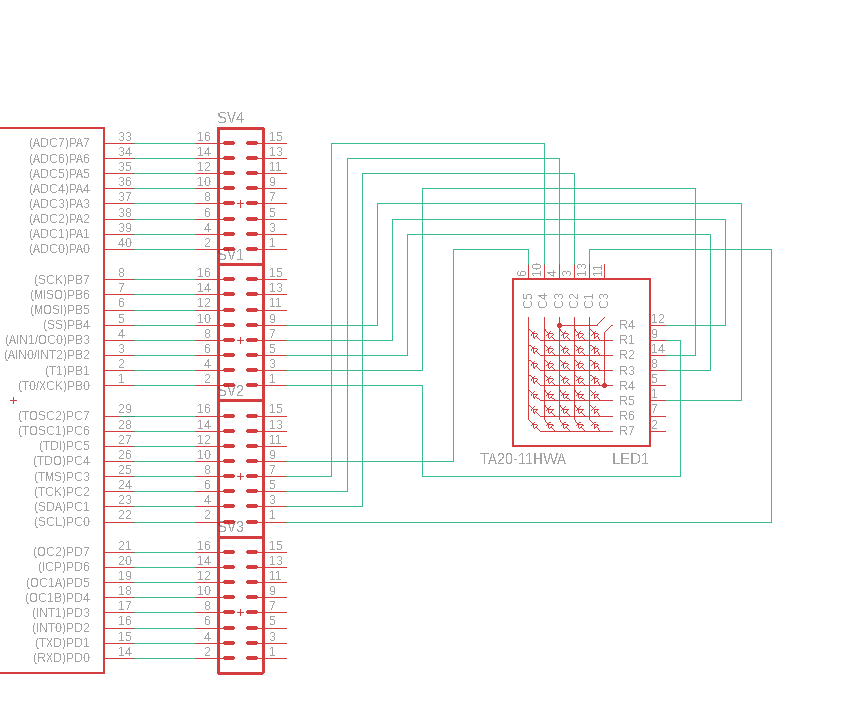
Celá LED matice má 8x8 LED. Nicméně senzor má pouze 5x5 výstupů. Proto použiji pouze krajních 5x5 LED na matici. LED matice bude sloužit jako zobrazovací prvek, zapojený na výstup senzoru. 35 36



Obrázek 27: LED Matice   
KVM-20881 CVB37

Pro účely testování bylo nutné LED matici připojit na vývojovou desku. Byly zapojeny piny anod až katod indexů 1 až 5. Vstupy ze senzoru byly přivedeny na piny vývojové desky. Následně byly data ze senzoru zpracovány a po vyhodnocení byly aktivovány patřičné LED matice.

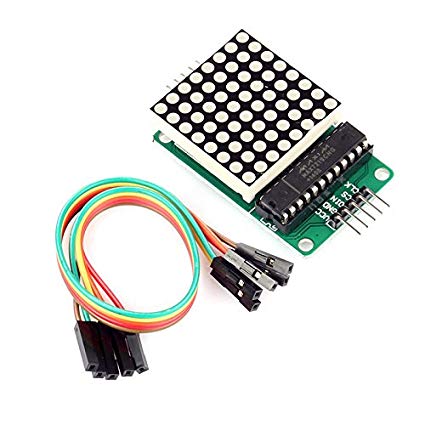
První možností je ovládání LED Matice přímo vývojovým kitem Arduino Uno. Pro rozsvícení konkrétních LED v matici je nutné aktivovat patřičné piny.



Obrázek 28: Zapojení LED Matice k mikrokontroleru

Například pro rozsvícení LED na pozici 1:1 je nutné přivést 5V na pin 13 a pin 9 spojit s GND. Tímto způsobem je možné rozsvítit jednu LED.

Pro rozsvícení většího počtu LED je nutné periodicky rozsvěcovat konkrétní LED. LED musít být rozsvěcovány v pokud možno co nejmenším časovém intervalu. Když bude interval dostatečně krátký, tak LED pouze svítí. Jak se bude interval zvětšovat, tak LED začnou blikat.



Obrázek 29: Modul pro ovládání LED matice 38

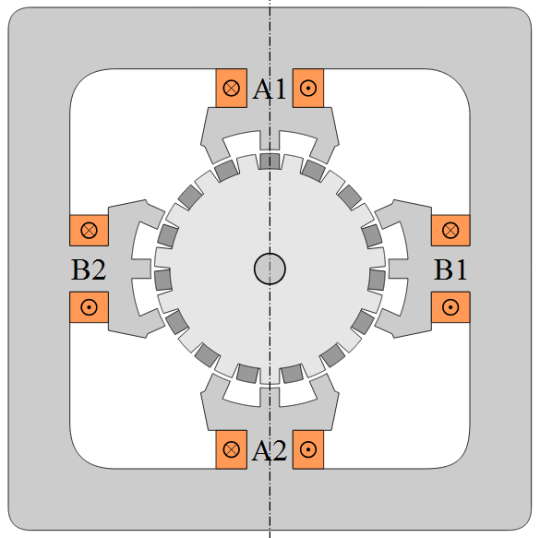
Druhou variantou je připojení LED matice pomocí modulu, který byl přímo navržen k ovládání LED matice. Pro chod tohoto modulu je nutné připojit pouze napájení a řídicí signály. Modul samotný je připojen na LED matici. 37

* 1. Krokový motor 28byj-48

Krokový motor je mechanické zařízení, které se skládá ze dvou částí. Tyto části se nazývají rotor a stator. Stator je statická, nepohyblivá část. Rotor je pohyblivá část. Existuje mnoho druhů motorů fungující různým způsobem. V této práci se zaměřím na krokový motor, poháněný stejnosměrným proudem.

Na statoru jsou navinuty cívky z elektricky vodivého materiálu. Čtyři tyto cívky jsou symetricky rozmístěny po statoru, takovým způsobem, že protikladné póly jsou vedle sebe. Rotor je zase tvořen z permanentního magnetu, který je připevněn na hřídel krokového motoru.

Přivedením impulsu na svorku motoru se zmagnetizují cívky na statoru a rotor tvořený magnetem se posune ve směru otáčení motoru. Přivedením impulsu se motor pohne o jeden krok. Krokový motor má specifickou posloupnost kombinací pořadí, ve kterém musí být cívky aktivovány.



Obrázek 30: Konstrukce krokového motoru39

Při dosažení dostatečné frekvence přivádění impulsu se motor bude pohybovat zdánlivě plynule. Frekvencí impulsů je tedy ovládána rychlost točení motoru. Nicméně jestli bude rychlost moc velká, tak motor otáčky ztratí, protože motor nebude stíhat se otáčet.

Krokový motor je v systému ovládán pomoci vývojového kitu Arduino Uno a několika tranzistorů. Tranzistory propojují motor a napájení. Vzhledem k tomu, že se krokový motor otáčí pomocí kombinace aktivních cívek, tak je nutné pomocí řídícího obvodu aktivovat patřičné tranzistory, který na cívky krokového motoru přivedou napětí. 38

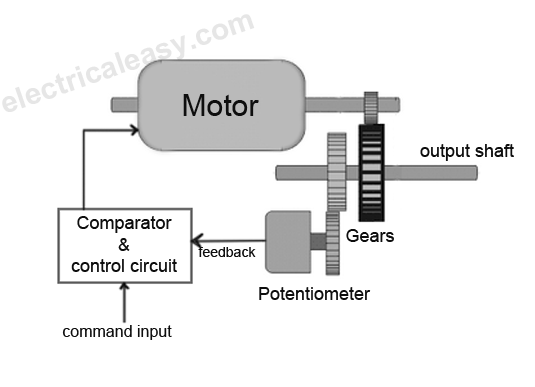


Obrázek 31: Krokový motor 28byj-4840

* 1. Servo motory SG90 a MSG90

Servo motor je zařízení podobné krokovému motoru. Na rozdíl od krokového motoru, kde se rotor otáčí za cívkami, navinutými na statoru, servo motor obsahuje stejnosměrný motor, který se otáčí díky spínání a rozepínání vstupního napájení do motoru.

Servo motor obsahuje další prvky, jako je řídící obvod, který umožňuje přesnější nastavení natočení. Vstupní impuls do servo motoru je ve skutečnosti hodnota napětí, která určuje natočení servo motoru. Servo motor bude tuto pozici držet, dokud nedojde k přerušení napětí, nebo k jeho změně.



Obrázek 32: Servo motor42

Z toho vyplývá, že servo motor se netočí neustále dokola, ale pouze natáčí do konkrétních pozicí. Tohoto je dosaženo pomocí pulzní šířkové modulace. Pulzní šířková modulace je vysílání periodického signálu v pravidelných intervalech. Periodický signál je rozdělen v rámci jedné periody na logickou 1 a logickou 0. Doba logické 1 vzhledem k periodě je udávána v procentech a nazývá se střída.

V systému jsou servo motory ovládány pomocí vývojového kitu Arduino Uno, kde do servo motoru vstupuje napájecí vodiče a jeden řídící vodič, který udává právě úhel natočení. 41 Servo motor je ovládán pomocí pulzní šířkové modulace. Střída odpovídá úhlu natočení servo motoru.

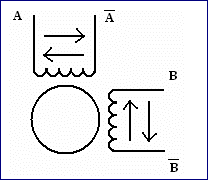


Obrázek 33: Servo motor43

Ve vývojovém prostředí Arduino IDE je k dispozici knihovna Servo.h, která se o ovládání servo motoru postará. Programátor pouze udává úhel a pomocí knihovny je tento úhel přepočítán na střídu, která je následně vyslána na datový vodič servo motoru.

* 1. Řídící jednotka TB6560

Řídící jednotka TB6560 slouží k ovládání krokových motorů. Jednotka je kompatibilní s vývojovými kity řady Arduino. Kity „Arduino Uno“ a „EvB 5.1“, které jsou použity v této práci. Samotná jednotka TB6560 má několik vstupů a výstupů.



Obrázek 34: Vnitřní zapojení krokového motoru45

Vstupní piny CLK+ a CLK- slouží k připojení vstupu, který bude přivádět impulsy. Na základě každého impulsu přivedeného na CLK+ se krokový motor pohne o jeden krok. Na pin CLK- je přiveden zemnící potenciál GND.

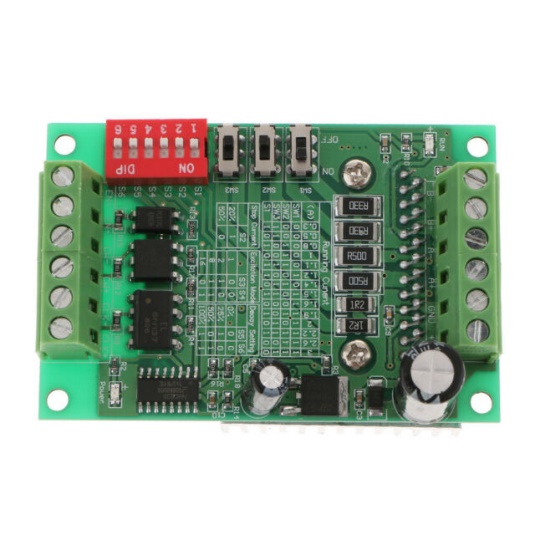
Dalšími vstupními piny jsou CW+ a CW-. Vstup CW+ slouží k určení směru pohybu krokového motoru. Na vstup CW- je přiveden zemnící potenciál GND. Posledními vstupy jsou EN+ a EN-. Po přivedení kladného napětí na EN+ je celá řídící jednotka aktivována. Na EN- je přiveden zemnící potenciál GND.

Posledními vstupy jsou svorky +24V a GND. Tyto svorky slouží k přivedení napájecího napětí.

Následují 4 výstupy A+ A- B+ a B-. Na tyto výstupy je připojen samotný krokový motor. 44

V případě tohoto systému nebyla řídicí jednotka TB6560 použita. Krokový motor 28byj-48 je napájen napětím 5V a spotřebovává proud v řádech desítek mA. Z toho důvodu byla celá řídicí jednotka TB6560 nahrazena čtveřicí tranzistorů BC639.

Řídicí jednotku je možné použít k ovládání větších krokových motorů jako je například krokový motor NEMA23.



Obrázek 35: Řídící jednotka TB656046

* 1. Testovací třídicí linka od firmy Merkur

Pro účely otestování celého systému bylo nutné sestrojit třídicí linku. Tato linka je k dostání od firmy Merkur. Existuje několik variant třídící linky. Základní, standard a úplná. Pro účely této práce jsem vybral základní a elektroniku k dopravníku navrhl a sestrojil.

Použité součástky mohou být připojeny na libovolnou stavebnici. Ovšem bude nutné modifikovat program pro hardware. Podobným systémem ze stavebnice se zabývá práce Pavla Vicana. Tento systém byl sestrojen pomocí podobné stavebnice Merkur.

Na trhu je mnoho stavebnic, které je možné použít k sestrojení systému. Vedle stavebnice merkur existuje i stavebnice s názvem LEGO.

Součástí stavebnice byl pouze dopravníkový pás a motor GM37-3530.

Tyto součásti stavebnice nebyly k realizace systému dostatečné, proto musely být zkonstruovány dodatečné řídicí jednotky a přidány motory, které pohání akční členy. Vedle motorů byl přidán senzor tvaru, který je použit k identifikaci tvaru předmětu, který se nachází na dopranvíku.

Vstupní a výstupní zásobníky dopravníku je možné řešit dalšími samostatnými dopravníky. V tomto případě se jedná o samostatný systém, který není součástí většího procesu, proto výstupní zásobník je realizován pomocí krabičky a vstupní zásobník je obsluhován uživatelem.

Obrázek 36: Motor GM37-353048



Obrázek 37: Dopravníkový pás firmy Merkur47



1. Konstrukce modelu třídící linky
   1. Konstrukce dopravníkového pásu

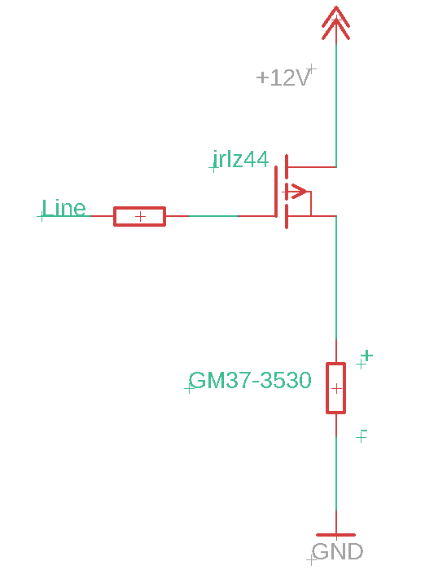
Dopravníkový pás byl pořízen od firmy Merkur. K dopravníku patřil motor GM37-3530 a manuál, jak dopravníkový pás sestavit. Dopravníkový pás nebylo obtížné sestavit. Celý dopravník se sestává z Merkur destiček, kde otvory na šrouby jsou 4 milimetry, mezera mezi otvory je 6 milimetrů. Základna byla rozšířena o merkur destičky, které patřily do jiných stavebnic. Tato základna je použita k upevnění desek řídicích obvodů.



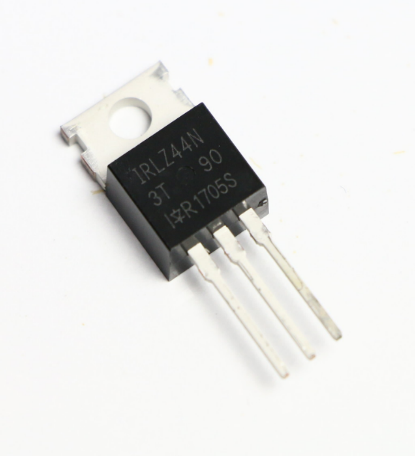
Obrázek 38: Dopravníkový pás

* 1. Řídící obvod motoru GM37-3530

Jako první Bylo nutné k motoru navrhnout řídící obvod, protože bylo pouze možné dopravník vypnout odpojením napájecího napětí. K tomuto účelu jsem použil transistor irlz44 Na kolektor je připojen kladný potenciál napájecího napěti 12V a mezi emitor a zemnící potenciál byl připojen motor GM37-3530. Na bázi tranzistoru je připojen výstup z vývojového kitu Arduino Uno. Přivedením logické 1 na bázi tranzistoru se tranzistor otevře a motor bude pohánět dopravníkový pás. K tomuto byl použit výstupní pin 5 na vývojovém kitu Arduino Uno. 50



Obrázek 39: GM37-3530 řídící obvod



Obrázek 40: Tranzistor IRLZ44N49

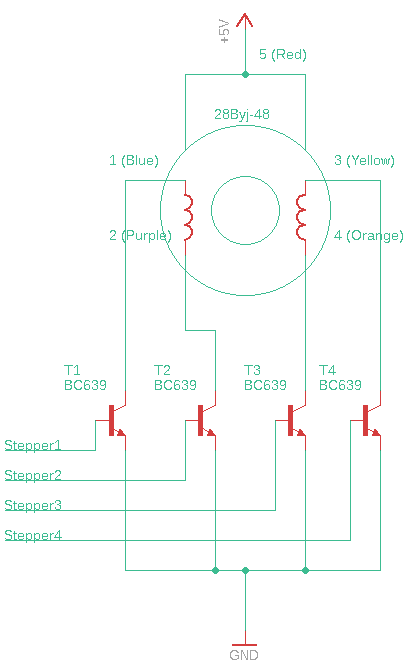
* 1. Řídící obvod krokového motoru 28Byj-48

Nyní je možné ovládat dopravníkový pás pomocí výstupního signálu z vývojového kitu Arduino Uno. Dalším prvek, který je potřeba ovládat je krokový motor 28Byj-48. Tento krokový motor je použit k přitisknutí senzoru tvaru k výrobku. Krokový motor je ovládán pomocí čtyř vstupů. Přivedením kombinace logické 1 na patřičné vstupy docílíme otočení rotoru o 1 fázi. Krok má celkem 8 fází, které je nutno vykonat pro provedení jednoho kroku.

Červený vodič je připojen na pin 5V na vývojovém kitu. Konkrétní vnitřní zapojení krokového motoru se může lišit. 51

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Stepper pin** | | | |  |
| **Stepper1** | **Stepper2** | **Stepper3** | **Stepper4** |  |
| **1** | 1 | 0 | 0 | 0 |  |
| **2** | 1 | 1 | 0 | 0 |  |
| **3** | 0 | 1 | 0 | 0 |  |
| **4** | 0 | 1 | 1 | 0 |  |
| **5** | 0 | 0 | 1 | 0 |  |  |  |
| **6** | 0 | 0 | 1 | 1 |  |  |  |
| **7** | 0 | 0 | 0 | 1 |  |  |  |
| **8** | 1 | 0 | 0 | 1 |  |  |  |

Řídící jednotka pro 28Byj-48 byla vytvořena pomocí 4 tranzistorů BC639. Na bázi těchto tranzistorů byly připojeny výstupy z vývojového kitu. Báze je aktivována logickou 1.



Obrázek 41: Řídící jednotka pro 28Byj-48

Konstrukce řídicí jednotky ja navržena tak, že je možné použít napájecí napětí 5V z vývojového kitu, nebo externí napájecí zdroj. V této aplikaci byl použit k napájení krokového motoru vývojový kit Arduino Uno, ale při větším množství motorů, nebo při použití motorů s vyšším napájecím napětím by bylo nutné použít externího napájecího zdroje.

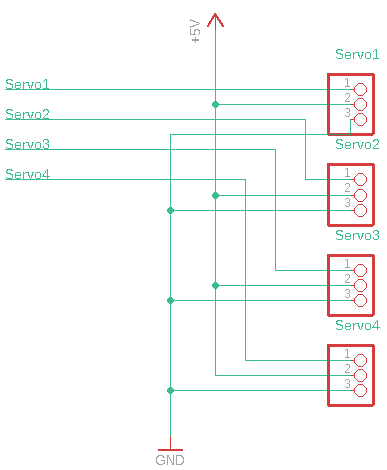
* 1. Řídící obvod pro servo motory SG90 a MSG90

Pro akční členy, byly použity servo motory SG90 a MSG90. Tyto servo motory mají celkem 3 vstupní piny. 2 piny slouží pro připojení vstupního napájení 5V a GND. Poslední pin slouží k připojení datového signálu, který určuje natočení hřídele servo motoru.

Zapojení servo motorů nevyžaduje žádné další elektronické součástky a lze je připojit přímo na vývojový kit. Pro napájení byly použity piny přímo na vývojovém kitu. Jako zdroj napájení pro servo motory je možné použít i 4 1.5V baterie.

V systému servo motory se pohybují v rozsahu 0° až 90° a fungují jako brány na dopravníkovém pásu. Kde 0° je zavřená brána a 90° je otevřená brána. 52

Vývojový kit Arduino Uno má možnost napájet servo motory, proto není nutně potřeba použít tranzistory, nebo externí napájecí zdroj.



Obrázek 42: Zapojení servo motorů

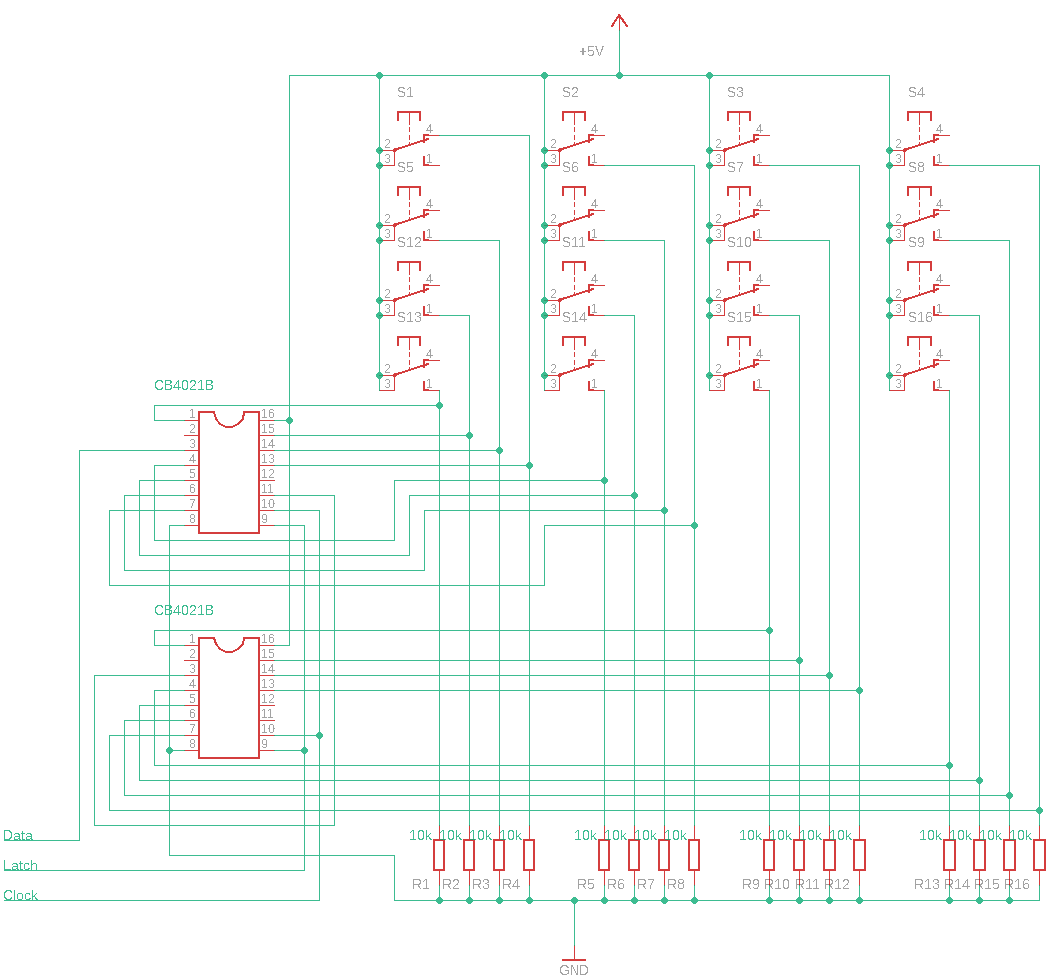
Pro použití většího množství motorů je vhodné použít externí napájecí zdroj, aby nebyl Mikrokontroler přetěžován. Výhodou takového postupu je zároveň to, že je možné použít napájecí zdroje různých napětí pro různé typu motorů.

Právě teď je deska konstruována tak, že všechno motory jsou spojeny s pinem napájecího napětí 5V, který je připojen na 5V pin na vývojovém kitu Arduino.

* 1. Senzor tvaru 4x4

Sensor tvaru 4x4 se skládá z 16 tlačítek a 2 posuvných registrů CB4021B. Vstupy tlačítek jsou připojeny na 5V, kde výstupy jsou připojeny na vstupy posuvných registrů CB4021B a přes rezistory 10kΩ na zemnící potenciál GND.

Během přitisknutí senzoru na výrobek jsou tlačítka stisknuta a na patřičných vstupech posuvného registru stav logická 1 zaznamenán. Následně jsou tyto stavy čteny vývojovým kitem Arduino Uno pomocí vstupů Latch, Data a Clock. Senzor je napájen přímo z desky vývojového kitu napětím 5V.33



Obrázek 43: Senzor tvaru 4x4

* 1. Ovládací deska

Pro účely této práce byla vytvořena řídící deska, která kombinuje všechny předchozí ovládací prvky. Deska obsahuje všechny ochranné a řídící prvky. Dále obsahuje svorkovnice, na které budou připojeny akční členy ze systému.

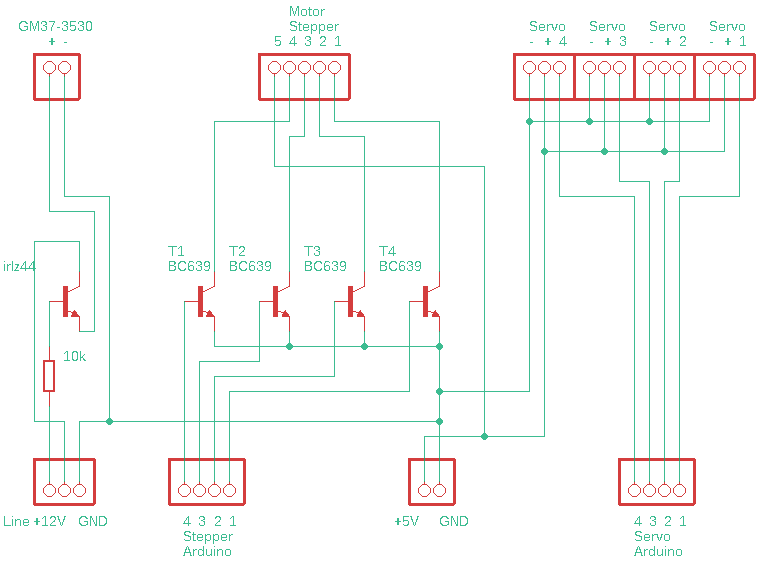
|  |  |  |  |
| --- | --- | --- | --- |
| **Desk pin** | | **Device pin** | |
| **Line +12V** | | External voltage source | +12V |
| **Line GND** | | GND |
| **Line** | | Arduino board pin | 5 |
| **Arduino** | **Stepper1** | A0 |
| **Stepper2** | A1 |
| **Stepper3** | A2 |
| **Stepper4** | A4 |
| **+5V** | | +5V |
| **GND** | | GND |
| **Arduino** | **Servo1** | 2 |
| **Servo2** | 3 |
| **Servo3** | 4 |
| **Servo4** | 6 |
| **Servo1** | **+** | SG90 | + |
| **-** | - |
| **1** | data |
| **Servo2** | **+** | SG90 | + |
| **-** | - |
| **2** | data |
| **Servo3** | **+** | SG90 | + |
| **-** | - |
| **3** | data |
| **Servo4** | **+** | SG90 | + |
| **-** | - |
| **4** | data |
| **Motor Stepper** | **1** | 28Byj-48 | 1 (Blue) |
| **2** | 2 (Purple) |
| **3** | 3 (Yellow) |
| **4** | 4 (Orange) |
| **5** | 5 (Red) |
| **GM37-3530** | **+** | GM37-3530 | + |
| **-** | - |

Ke konstrukci ovládací desky byl použit univerzální plošný spoj, 4 tranzistory BC639, tranzistor irlz44, rezistor 10kΩ, 10 svorkovnic a propojovací dráty.

Celou desku je následně možné připevnit ke konstrukci dopravníku, kde bude sloužit jako propojovací člen s Mikrokontrolerem. Výhodou řešení ovládání akčních členů tímto způsobem je to, že není nutné vždy použít jeden a ten samý Mikrokontroler.

Deska není určena k připojení vyššího napájecího napětí než 5V, protože byla navržena pro akční členy, které pracují s napájecím napětím 5V. Při připojení většího napájecího napětí by mohlo dojít k poškození připojených akční členů.

Na desce nejsou svorkovnice pro připojení senzoru tvaru. Tento senzor tvaru je připojen přímo na vývojový kit Arduino Uno. Důvod je ten, že původně byl navržen jiný senzor tvaru a v budoucnu je možné použít naprosto odlišný způsob vyhodnocení tvaru předmětu, který bude systémem tříděn.

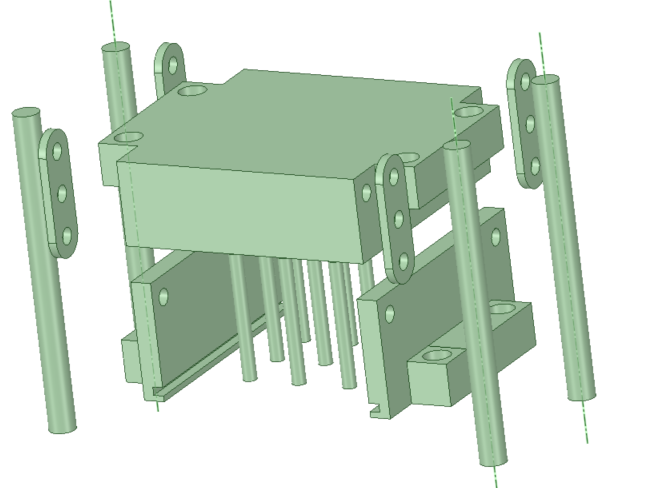


Obrázek 44: Schéma ovládací desky

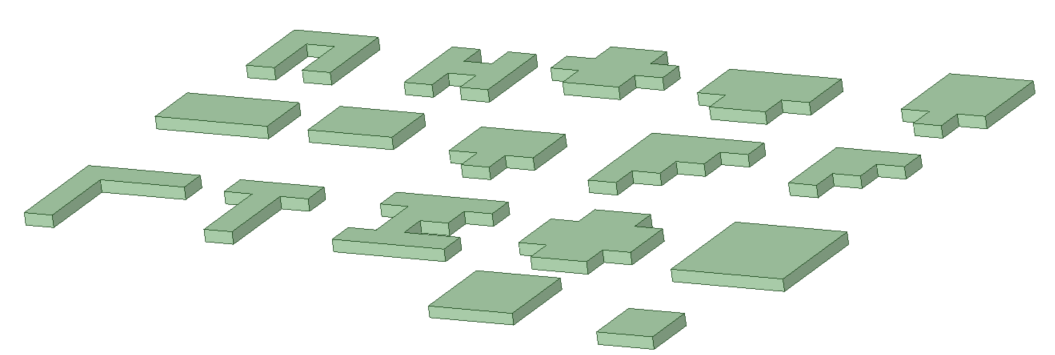
* 1. Tvorba vlastních mechanických součástek

Ke spojení elektronických součástek a dopravníkového pásu bylo nutné navrhnout a realizovat doplňující mechanické součástky. Tyto součástky byly navrženy v programu DesignSpark Mechanical 4.0 a následně vytisknuty pomocí 3D tiskárny.

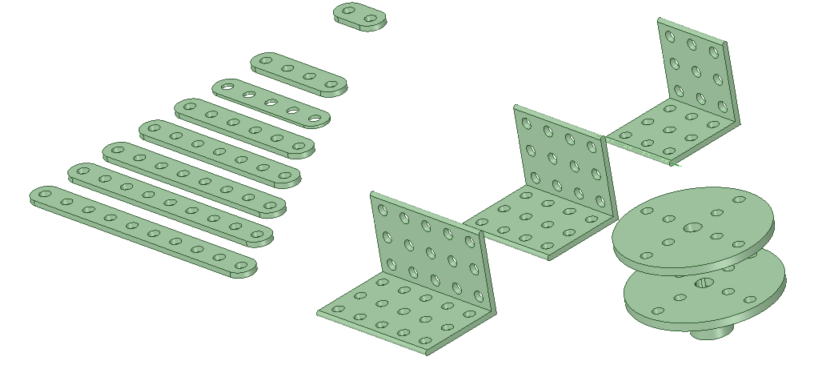
Konkrétně se jedná o hlavici senzoru tvaru, hřídel krokového motoru a několik dalších menších součástek, které slouží k přichycení senzoru tvaru k dopravníkovému pásu. Dále byly vytisknuty výrobky různých tvarů, které bude celý systém třídit.



Obrázek 45: Návrh hlavice senzoru tvaru v programu DesignSpark Mechanical 4.0



Obrázek 46: Návrh výrobků v programu DesignSpark Mechanical 4.0



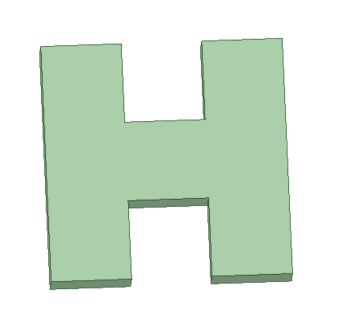
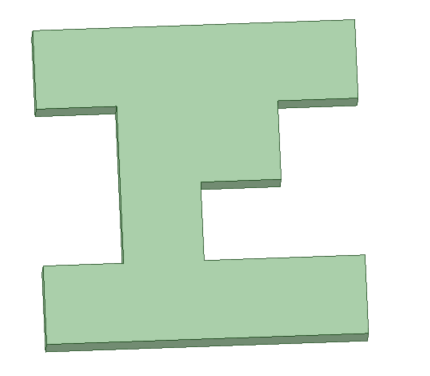
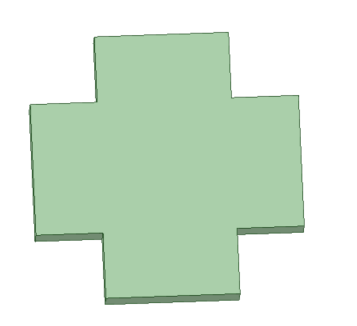
Obrázek 47: Návrh náhradních merkur součástek v programu DesignSpark Mechanical

Všechny mechanické součástky, které byly vytisknuty na 3D tiskárně jsou na rozdíl od klasických kovových merkur součástek vyrobeny z plastu. Proto bylo nutné zvolit větší šířku pro tisk. Dále bylo nutné navrhnout mechanické součástky jako je hlavice senzoru tvaru z několika částí, které je možné spojit dohromady pomocí šroubů.

Tato varianta senzoru byla navržena pro výrobky o maximálním rozměru kvádru s podstavou čtverce o straně 50 milimetrů a výšce do 10 milimetrů. Výrobky musí být pevného skupenství, protože senzor tvaru je přitisknut na samotný výrobek.

Senzor tvaru má celkem 16 tlačítek, které představují jednotlivé bity rozlišení. Tvar předmět je vyhodnocen podle stisknutých tlačítek po přitisknutí předmětu na senzor.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **1** | **0** |  | **0** | **1** | **1** | **0** |  | **1** | **0** | **1** | **0** |
| **1** | **1** | **1** | **1** |  | **1** | **1** | **1** | **1** |  | **1** | **1** | **1** | **0** |
| **1** | **1** | **1** | **1** |  | **1** | **1** | **1** | **1** |  | **1** | **0** | **1** | **0** |
| **0** | **1** | **1** | **0** |  | **0** | **1** | **1** | **0** |  | **0** | **0** | **0** | **0** |

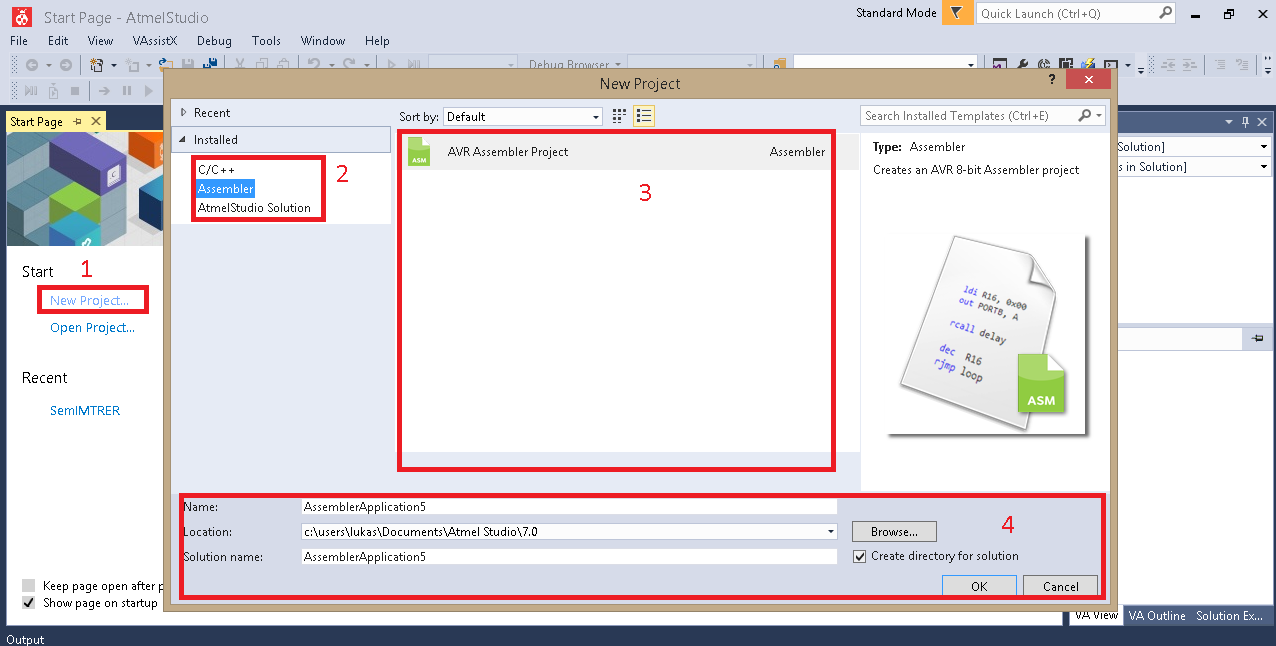


Obrázek 48: Vyhodnocení tvaru předmětu

1. Programování Mikrokontroléru
   1. Programování Mikrokontroleru pomocí Atmel studia

Mikro kontrolér ATMEGA je možné naprogramovat pomocí USB portu. K naprogramování tohoto mikro kontroléru je použito programovací prostředí „Ardunio for Atmel Studio 7“53 a do mikro kontroléru nahrán pomocí aplikace „AND-Load“54.

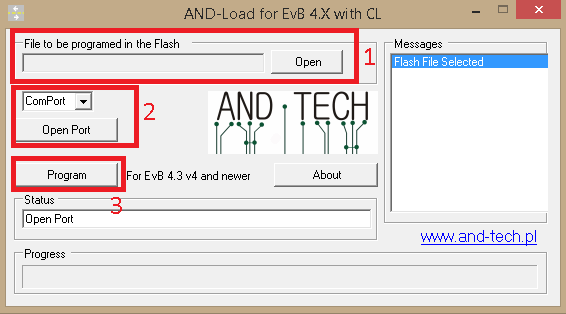
1. Kliknutím na „new project“ založíme nový projekt.
2. V nabídce si vybereme typ programovacího jazyka, je potřeba nainstalovat potřebné balíčky
3. V třetím kroku vybereme správný mikro kontrolér.
4. Následně nastavíme cestu uložení souboru a potvrdíme.



Obrázek 49: Vývojové prostředí Atmel studio 7.0

Po úspěšném programování program je nahrán do mikro kontroléru následovně:

1. Kliknutím na open vybereme napsaný program k nahrání.
2. Připojením vývojového kitu na USB rozhraní počítače se v rolovací nabídce objeví nová položka.
3. Stiskem tlačítka „program“ se program nahraje do mikro kontroléru.
4. Okna status, progress a messages informují o stavu procesu nahrávání.

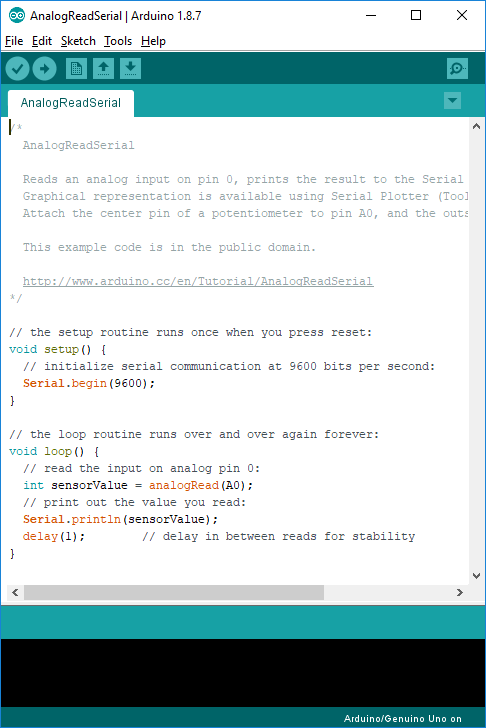


Obrázek 50: AND-Load aplikace

1. 1. Programování mikro kontroléru pomoci Arduino IDE

Další možnost k naprogramování mikro kontroléru je pomocí programovacího prostředí Arduino IDE. Tohle programovací prostředí slouží zároveň jako kompilátor a také jako nástroj pro nahrání programu přímo do mikro kontroléru.

Vývojový kit je opět připojen přes USB port k počítači. Některé části programu, nebo jeho principy jsou velmi podobné, nebo přebrané z návodů, které jsou volně dostupné na internetu. Tyto části jsou popsány podrobně dále. 55



Obrázek 51: Ukázka vývojového prostředí Arduino Uno

* 1. Software Mikrokontroleru
     1. Detekce stisku tlačítka mikro kontrolérem

Velmi důležitým prvkem pro chod programu je detekce stisku tlačítka mikro kontrolérem. V programovacím prostředí Arduino IDE je řešen pomocí funkce digitalRead(pin). Vstup této funkce je číslo pinu a výstup této funkce je logická hodnota HIGH, nebo LOW. 56

* + 1. Rozšíření počtu vstupních pinů pomocí posuvného registru

Senzor tvaru se skládá z celkem 16 tlačítek. Výstup z těchto tlačítek je nutné přivést na vstupní piny vývojového kitu Arduino Uno a vyhodnotit. Vzhledem k tomu, že vývojový kit použitý v tomto systému obsahuje pouze omezené množství pinů, je nutné, aby byl použit posuvný registr k rozšíření vstupních pinů.

Existuje mnoho variant posuvných registrů. V tomto systému byl použit posuvný registr CD4021B. Posuvný registr potřebuje ke svému chodu 2 řídící signály a poskytuje 1 datový signál se stavy tlačítek. 33

* + 1. Rozsvícení LED, nebo LED matice

Dalším důležitým prvkem je rozsvícení LED. LED bylo nutné rozsvítit pro účely otestování správnosti vstupních dat ze senzoru tvaru. LED matice se od samostatné LED liší tím, že LED matice má katodu a anodu společné s ostatními LED. V každém případě na anodu musí být přivedena logická jedna, neboli kladné napětí a na katodu musí být přivedena logická nula, nebo zemnící potenciál GND. Máme tedy několik možností pro rozsvícení.

Můžeme rovnou připojit LED na zemnící potenciál na desce (GND pin) a přes ochranný rezistor na kladné napětí (5V pin).

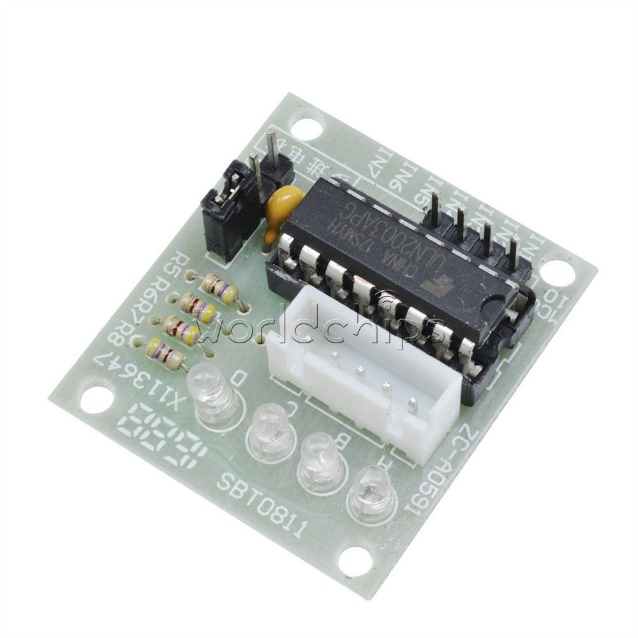
Další možností je připojení katody na zemnící potenciál desky (GND pin) a připojení anody na jeden z digitálních výstupů. (0-13 digitální piny). Tímto způsobem je možné LED, nebo LED matici rozsvěcovat přivedením logické jedničky na pin výstupu a zhasínat přivedením logické nuly.

Následující možností je propojení jak anody, tak katody s digitálním výstupem. V tomto případě je nutné pro rozsvícení přivést na katodu logickou nulu a na anodu logickou jedničku. Tuto variantu jsem uplatnil pro testovací potřeby, kdy všechny společné anody a katody byly propojeny s vývojovou deskou a byly rozsvíceny pouze konkrétní bity LED matice.

Pro přivedení logické jedničky, nebo nuly je zapotřebí funkce digitalWrite(pin, hodnota). Tato funkce nemá žádný výstup, ale má dva vstupy. Prvním vstupem je číslo pinu, na který má být aktivován. Druhým vstupem je hodnota, která má být na digitální výstup poslána. 57

* + 1. Ovládání krokového motoru 28byj-48

Krokový motor v tomto systému slouží k přitisknutí hlavice senzoru tvaru k výrobku. Je možné tento krokový motor ovládat pomocí řadiče ULN2003. Řadič ULN2003 je deska s integrovaným obvodem ULN2003, napájecím napětím, signálními LED a vstupními a výstupními piny.



Obrázek 52: Řadič ULN200358

Tento integrovaný obvod obsahuje 7 tranzistorů. Tyto tranzistory jsou použity k přivedení napájecího napětí na svorky motoru ve správné sekvenci.

Pro účely tohoto systému byl tento řadič nahrazen 4 tranzistory BC639. Tyto tranzistory plní totožnou funkci, jako řadič ULN2003.

Příklady použití tohoto motoru jsou k dispozici na internetu. Ve své práci jsem čerpal zejména z projektu Arduino Simple „Camera Slider“ Electronics. Tento projekt obsahuje stejné elektronické součástky. 59

* + 1. Ovládání servo motoru SG90

Další důležitou částí programu je ovládání servo motorů. Servo motory mají pouze napájecí vstupy a 1 datový vstup. Zároveň jsou napájeny napětím pouze 5V. Díky tomu není potřeba použít k jejich obsluze žádné řídící prvky jako u krokového motoru.

Vývojové prostředí Arduino IDE obsahuje zvláštní knihovnu Servo.h, která umožní programátorovi ovládat tento servo motor. 52

* + 1. Ovládání GM37-3530 motoru

Motor GM37-3530 je napájen stejnosměrným napětím 12V. Proto nebylo možné jej napojit přímo na vývojový kit Arduino Uno. K ovládání byl použit tranzistor irfz44. Z vývojového kitu byl přiveden na bázi tranzistoru signál, který tranzistor otevíral, nebo zavíral. Tímto způsobem byl ovládán motor GM37-3530. 50

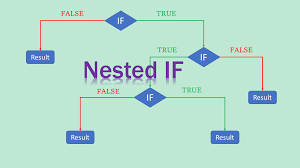
* + 1. Řízení toku programu

Každý program vykonává konkrétní instrukce v určité posloupnosti. Pro řízení programu je nutné použít rozhodovací cykly a proměnné. Proměnné slouží k uchování dat, které mohou označovat libovolný stav, počet, nebo situaci.

V programu byly použity proměnné pro označení indexu iterace, uchování hodnoty zpoždění, nebo čísla pinu. Výhoda proměnné je v tom, že lze ji definovat na začátku programu a následně použít kdekoliv dále. Díky tomu je možné změnit pouze hodnotu proměnné na začátku a hodnota bude změněna v celém programu.

Dále byl použit cyklus while, který je vykonáván, dokud podmínka v argumentu cyklu platí. Tento cyklus byl použit pro rozsvícení LED, které jsou zobrazovány periodicky o velmi vysoké frekvenci. Dále byl použit pro ovládání krokového motoru. Kdybych poslal pouze jeden impuls, tak by se LED jedenkrát rozsvítila a motor se otočil o pouze jeden krok. 60

Posledním prvkem bylo rozhodování pomocí funkce IF. Argumentem této funkce je podmínka, na základě které je rozhodnuto, která větev IF funkce bude vykonána. V případě, že podmínka platí, tak je vykonána větev za podmínkou. V případě, že neplatí podmínka, je vykonána větev za klíčovým slovem ELSE. Klíčovým slovem IF je v programu rozhodováno například, který program má být vykonán. 64



Obrázek 53: Příklad rozhodovacího algoritmu 61

* 1. Program pro Mikrokontrolér

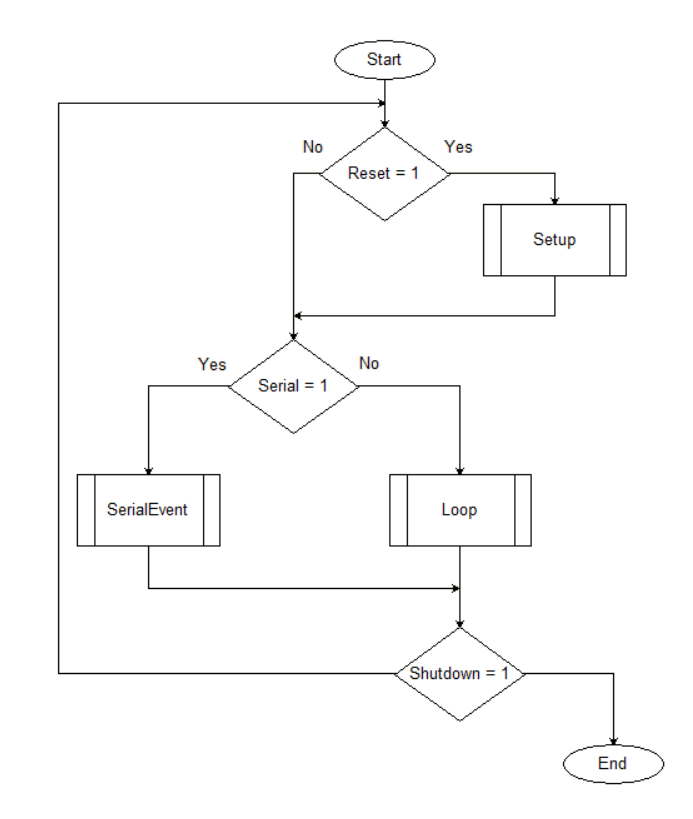
Tato kapitola se zabývá popisem programu, podle kterého je řízena část systému, která ovládá hardwarovou část. Program byl napsán ve vývojovém prostředí Arduino IDE, které je zmíněno výše.

* + 1. Hlavní program

Hlavní program je tvořen bloky Setup, Loop a SerialEvent. Setup je vždy spuštěn po nahrání programu do mikro kontroléru a po resetování systému. Loop je smyčka s hlavním programem. Blok SerialEvent se vykoná vždy, když mikro kontrolér obdrží tisknutelné znaku z nadřazeného systému. Tímto způsobem nadřazený systém a mikro kontrolér komunikují. Program mikrokontroleru běží ve smyčce dokud není systém vypnut. Vypnutí systému je možné odpojením napájení.

Uvnitř hlavního programu je rozhodovací logika, která přijímá instrukce z nadřazeného systému na osobním počítači. Hlavní program obsahuje bloky s manuálním a automatickým režimem. Bloky pro čtení a vyhodnocení sériového vstupu. Zároveň jsou zde bloky pro obsluhu akčním členů.

V manuálním režimu není žádná rozhodovací logika. Jde zde pouze vyhodnocení vstupních insturkcí z osobního počítače. V automatickém režimu je blok s rozhodovací logikou, která využívá další bloky jako je vyhodnocení tvaru předmětu, výpočet cílového uložiště a samotné bloky pro oblushu akčních členů.



Obrázek 54: Hlavní program

* + 1. Inicializační blok Setup a blok SerialEvent

Blok setup slouží k nastavení pinů jako vstupů, případně jako výstupů. Dále je uvnitř bloku deklarace proměnných a konstant, které budou použity uvnitř programu a otevření sériového kanálu. Tento blok se skládá pouze z několika instrukcí.

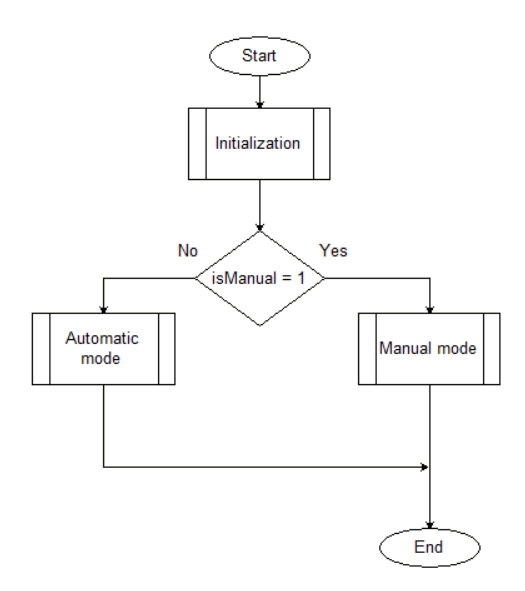
Blok SerialEvent je vždy vykonán, když mikro kontrolér na sériovém portu zaznamená libovolné tisknutelné znaky, které pošle nadřazený systém. Blok SerialEvent je obsloužen, i když se program aktuálně nachází v jiné části programu. 62

* + 1. Blok Loop

Blok Loop se skládá pouze z rozhodovacího návěští If. Toto návěští vyhodnocuje, jestli byl vybrán manuální, nebo automatický režim. 63

Tento blok je hned pod úrovní hlavního programu. V tomto bloku je rozhodováno, jestli uživatel vybral manuální, nebo automatický režim. Vždy je vykonán jeden cyklus a následně je opětovně vyhodnoceno, jestli nebyl vybrán jiný režim.

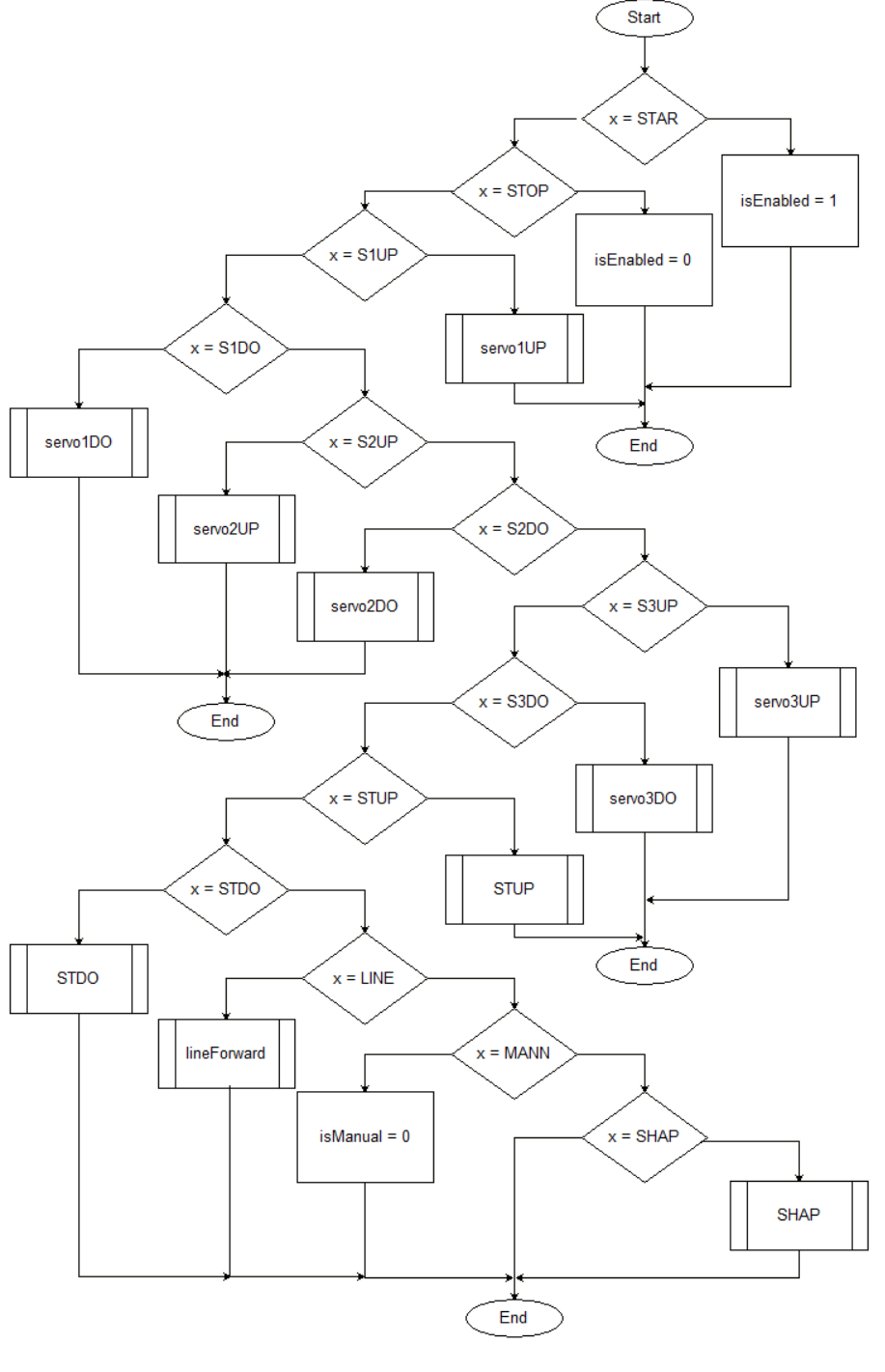
Volba mezi automatickým a ručním režimem je možná na základě instrukce z nadřazeného systému na osobním počítači. Defaultně je systém nastaven na manuální režim. Blok initialization je určen pro nastavení akčních členů do výchozí polohy. Tohle je důležité, protože motory zasahují do soustavy pomocí mechanických součástek.



Obrázek 55: Blok Loop

* + 1. Blok Manual mode

Manuální režim je aktivní, když je proměnná isManual rovna logické 1. Tohoto je možné docílit volbou v aplikaci osobního počítače. V manuálním režimu je možné ovládat všechny akční členy systému nezávisle na sobě. Tento režim je určen pro testování jednotlivých prvků.



Obrázek 56: Blok manual mode

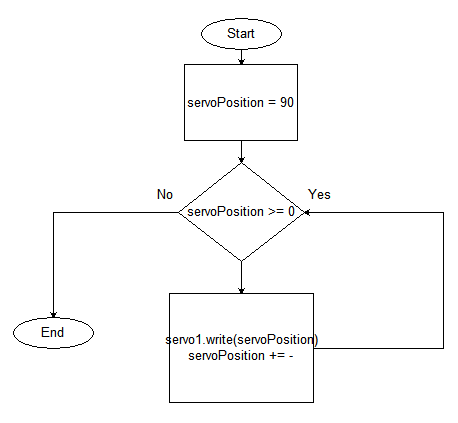
* + 1. Obslužné funkce akčních členů

Všechny akční členy mají obslužné funkce, které jsou řídící logikou programu volány. Řídící logika vždy rozhodne o cílové lokaci a na základě toho je zavolána obslužná funkce s patřičným parametrem. Parametr zde slouží k určení počtu otáček, případně vzdálenosti, kterou motor urazí.

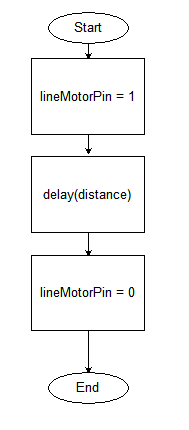
Funkce S1UP, S2UP a S3UP ovládají servo motory. Přesouvají servo motory z natočení 0° do natočení 90°. Funkce S1DO, S2DO, S3DO ovládají servo motory během opačného požadavku. Motory jsou natočeny z 90° dolů na 0°.

Funkce STUP ovládá krokový motor. Krokový motor reaguje na sekvenci vstupů, proto je nutné jeden celý krok realizovat 8 mikro kroky. Na 4 datové vstupy krokového motoru jsou přiváděny patřičné sekvence, které zajistí pohyb motoru vpřed. Pro zpětný pohyb je na vstupy přivedena obrácená sekvence uvnitř funkce STDO.

Funkce LineForward slouží k obsluze motoru GM37-3530. Tento motor je ovládán pouze jedním tranzistorem, takže není potřeba dělat kroky jako u krokového motoru, ale na bázi tranzistoru přivést logickou 1 a po určitou dobu nechat tranzistor otevřený. Tato doba je závislá na cílové vzdálenosti do uložiště výrobku.



Obrázek 57: Funkce S1DO



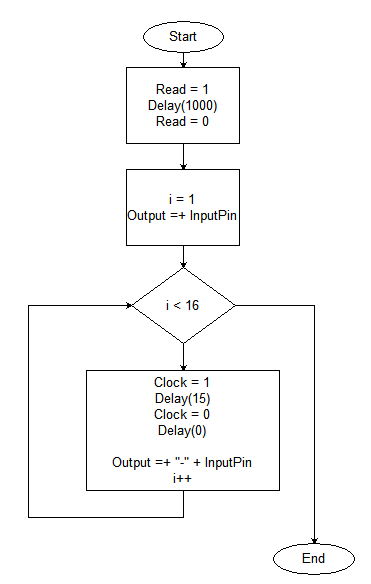
Obrázek 58: Funkce lineForward

* + 1. Detekce tvaru výrobku pomocí funkce SHAP

Tvar výrobku je detekován pomocí 16 tlačítek a 2 posuvných registrů. Tyto registry si načtou data ze vstupů po přivedení signálu a následně vstupy všechny vysílají po 1 sériovém pinu v hodinových cyklech. Podle toho je program navržen.

Na začátku podprogramu je na pin Read přivedena logická 1 a následně po prodlevě 1 vteřiny logická 0. Tímto je do registru zaznamenán stav tlačítek senzoru. Následně je v cyklu čten výstup registru a tento výstup je uložen do proměnné. Tato proměnná nese informaci o tvaru výrobku.

Tento blok je používán zejména automatickým režimem programu. Vždy, když je předmět přistaven na linku, tak je potřeba detekovat jeho tvar. Je zároveň velmi lehce možné zvětšit rozlišení přidáním dalších tlačítek bez přidání dalších pinů. Jedinné, co se změní je délka cyklu, ve kterém se výstup z posuvného registru čte.



Obrázek 59: Detekce tvaru výrobku

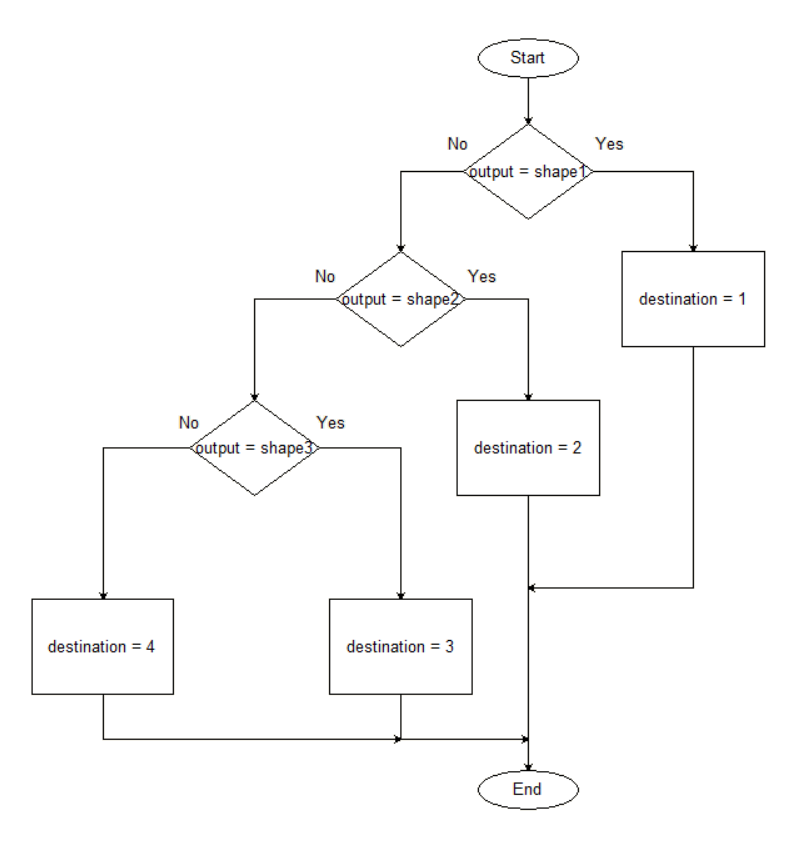
V případě tohoto bloku je to proměnná i, která při každé iteraci představuje jeden ze vstupů.

Během každého pulzu CLOCK je zvýšena proměnná i o 1 a na výstupu posuvného registru je další stav vstupu.

* + 1. Určení kombinace uložiště a tvaru pomocí funkce WHERE

Tato funkce je používána pouze v automatickém režimu a vždy následuje po funkci vyhodnocení tvaru SHAP. Funkce porovná proměnné shape1, shape2 a shape3 s proměnnou output. Proměnná output obsahuje informaci o tvaru výrobku, kterou detekovala funkce SHAP. Proměnné shape1, shape2 a shape3 jsou staženy z nadřazeného systému pomocí funkce UDPATE.

Funkce WHERE po porovnání proměnných uloží do proměnné destination hodnotu, která odpovídá uložišti, do kterého výrobek patří. Zároveň je tímto způsobem identifikován akční člen před uložištěm.



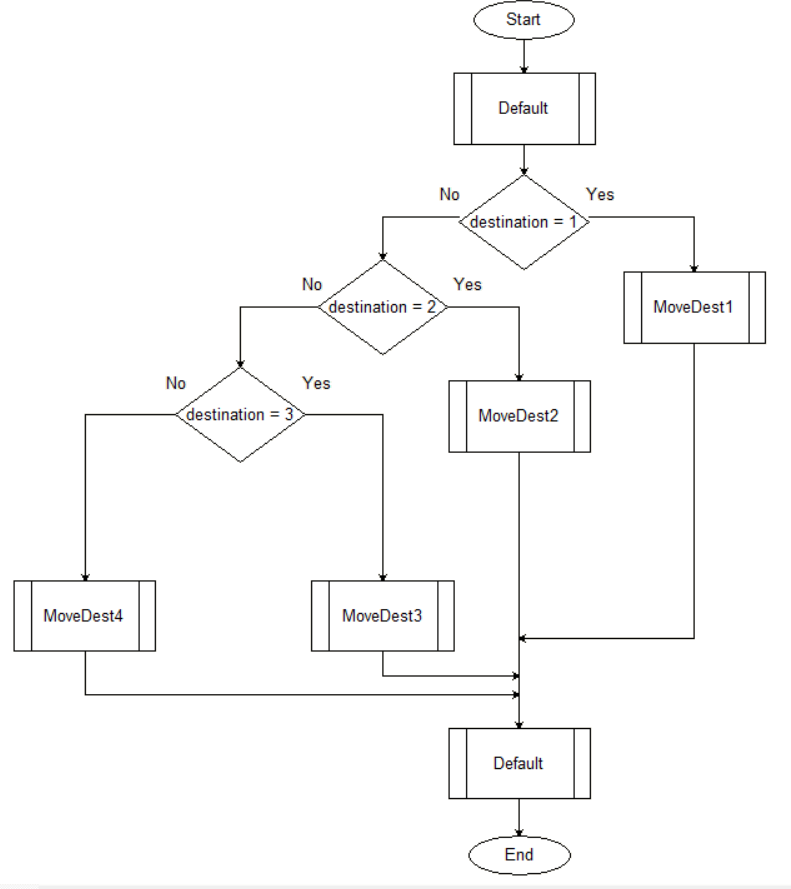
Obrázek 60: Funkce WHERE

* + 1. Automatický režim

Druhá varianta chodu systému je automatický režim. Během automatického režimu jsou používány stejné funkce. Jediný rozdíl je v tom, že není možné ručně ovládat akční členy, ale o procesu rozhoduje řídící logika.

Řídící logika pracuje v cyklu. Výrobek je vždy přiveden před senzor tvaru. Senzor tvaru je přitisknut na výrobek. Tvar je vyhodnocen a je rozhodnuto, kam výrobek patří. Následně je vybrán akční člen u uložiště a výrobek je dopraven do uložiště. Akční členy jsou následně přesunuty do jejich základních pozicí. Celý proces se opakuje ve smyčce.

Na začátku a na konci je vždy zavolána funkce Default. Tato funkce nastaví všechny akční členy do výchozí pozice.



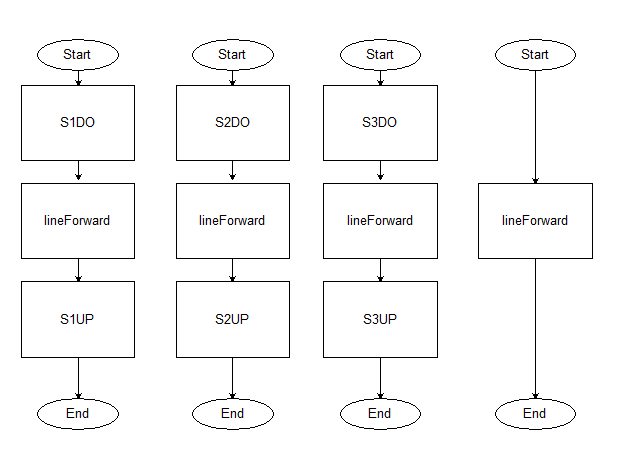
Obrázek 61: Automatický režim

* + 1. Funkce Default a bloky MoveDest

Poslední funkcí je funkce Default. Tato funkce je použita k nastavení všech akčních členů do výchozí pozice. Funkce Default je použita také během inicializace mikro kontroléru. Tedy funkce Setup.

Servo motory je nutné natočit do úhlu 90°. Tohle natočení je stav, kdy je brána před uložištěm otevřená. Výrobek tedy projede dál k dalšímu uložišti. Dále je nutné nastavit krokový motor do vrchní pozice. Tato pozice je stav, kdy je senzor tvaru nepřitisknut. Funkce rozlišuje, jestli už akční členy nejsou v počáteční pozici.

Bloky MoveDest slouží k přesunu výrobku k uložišti. Prvně je servo motor cílového uložiště nastaven na 0°. Následně je použita funkce lineMotorForward. Nakonec je opět zavolán blok Default.

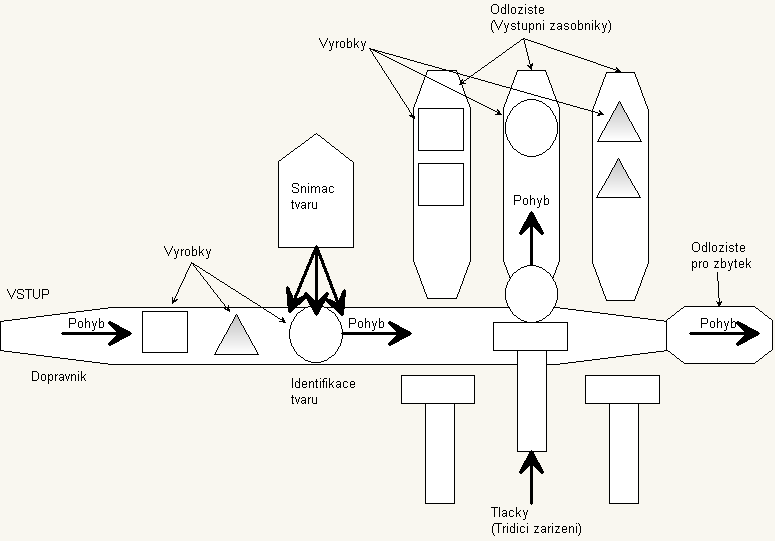


Obrázek 62: Bloky MoveDest

* 1. Ukázka běhu programu

Zde je ukázka běhu programu znázorněného blokovým schématem. Výrobek je položen na dopravníkový pás, kde pokračuje pod senzor tvaru. Senzor tvaru je přitisknut a informace o tvaru výrobku vyhodnocena mikro kontrolérem. Následně výrobek pokračuje po dopravníkovém pásu před cílové uložiště. Akční člen před uložištěm zatlačí na výrobek a vybočí ho do uložiště.

Celý systém může být realizován podobnými akčními členy. Zde v blokovém schématu je ukázka realizace pomocí tlaček. Tlačka na výrobek zatlačí a ten pokračuje na vedlejším dopravníkovém pásu.



Obrázek 66: Ukázka běhu programu

1. Nadřazený systém
   1. Aplikace na počítači uživatele:

Software mikro kontroléru je navržen zejména pro obsluhu akčním členů systému a řídící logiku pro automatický režim. Nadřazený systém je určen pro osobní počítač s operačním systémem Windows 10. Aplikace je napsána v programovacím jazyce C#, ve vývojovém prostředí Microsoft visual studio. 64

Aplikace pro osobní počítač je určena pro komunikaci uživatele s mikro kontrolérem. Uživatel má možnost ovládat akční členy v manuálním režimu, kdy aplikace posílá instrukce po sériovém portu, na kterém je připojen vývojový kit Arduino Uno.

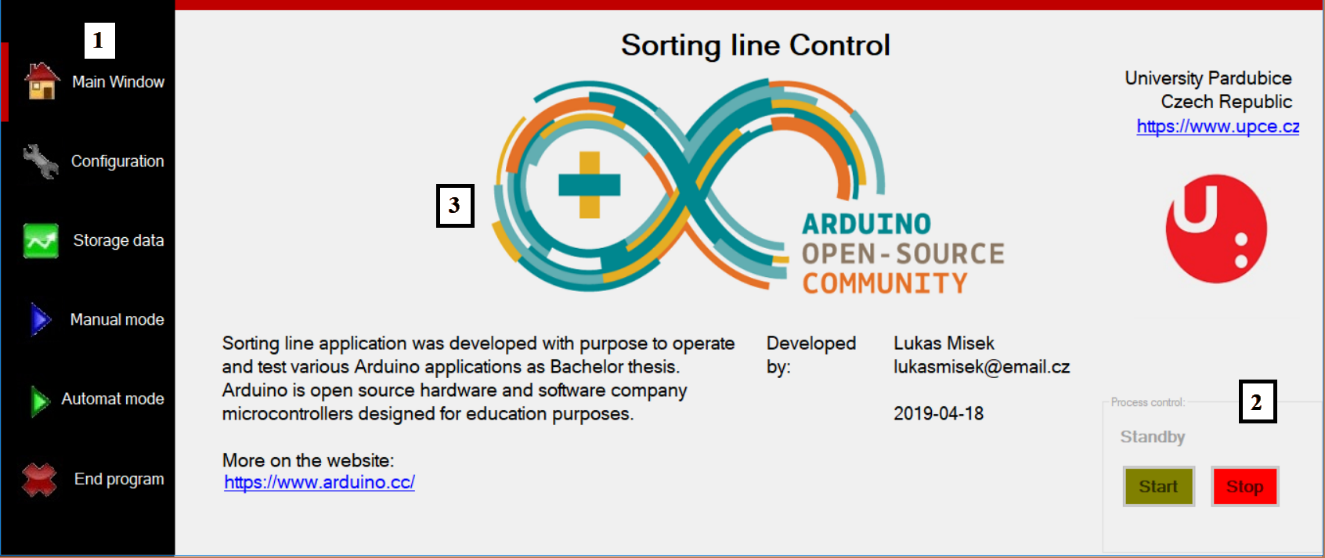
Aplikace umožňuje uživateli ovládat třídící linku dvěma způsoby. Prvním je manuální režim, kdy uživatel může ovládat jednotlivé akční členy a měřící členy. Druhá možnost je nahrát do mikro kontroléru cílové tvary a jejich lokace a nechat třídící linku běžet autonomně. Kdykoliv je možné třídící linku zastavit.

Uživateli je umožněno zaznamenávat data o procesu do databáze a vytvářet konfigurace, podle kterých se bude třídící linka řídit. Tyto data je možné exportovat do txt formátu, případně importovat ze stejného formátu.

* 1. Main window

Úvodní obrazovka je nazvaná Main Window. Tato obrazovka umožňuje přepnutí jednotlivých režimů a obsahuje informace o programu.

1. Postranní lišta k přepínání mezi jednotlivými menu
2. Panel pro ovládání procesu
3. Obrazovka s informacemi o programu



Obrázek 67: Main window

* 1. Menu Configuration

Menu Configuration slouží k vytváření, editování a mazání uživatelských konfigurací. Podle těchto konfigurací se řídí ruční režim programu. Konfigurace je zároveň možné exportovat, nebo importovat ze souboru. Vždy může být aktivní pouze 1 konfigurace. Uživatel tuto konfiguraci vybere v sekci číslo 3.

Uživateli je umožněno dosadit libovolný tvar do libovolného uložiště. Všechny ostatní tvary budou považovány za „undefined“ a bude jim přiřazeno uložiště „Locaiton\_Others“.

1. Postranní lišta k přepínání mezi jednotlivými menu
2. Sekce k vytvoření konfigurací. Jednotlivé konfigurace jsou ukládány jako záznamy do tabulky v sekci 6
3. Sekce k volbě aktivní konfigurace. Aktivní může být pouze jedna konfigurace. Je zároveň možné smazat všechny záznamy konkrétní konfigurace
4. Mazání jednotlivých řádků. Jsou smazány pouze řádky vybrané kurzorem
5. Sekce k exportování a importování konfigurací do souboru ve formátu txt
6. Tabulka se záznamy konfigurací
7. Panel pro ovládání procesu



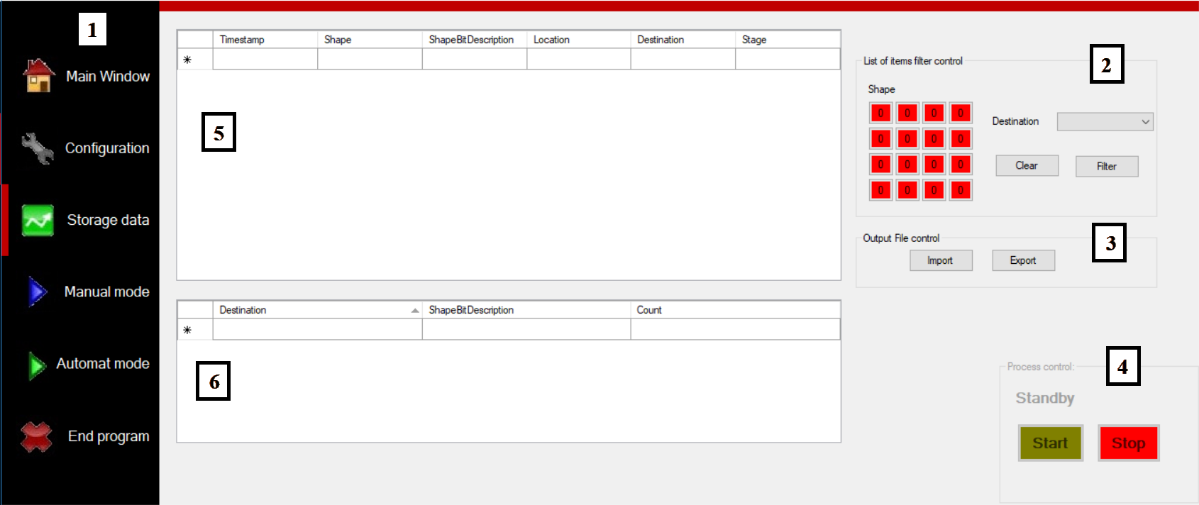
Obrázek 68: Menu Configuration

* 1. Menu Data storage

Manu Data storage slouží k zobrazení dat, které byly v procesu zaznamenány. Jsou rozlišovány stavy, ve kterých se výrobky nacházely, čas vzniku události, místo vzniku a cílové lokace po dokončení kroku. Zároveň všechny výrobky mají přiřazeny tvary formou bitové reprezentace. Tyto bity jsou detekovány senzorem tvaru a představují tvar výrobku.

V tomto menu je možné zobrazit data o procesu. Uživateli jsou k dispozici filtry, kterými může zúžit seznam záznamů. Je možné filtrovat podle tvaru a podle cílového umístění výrobku.

1. Postranní lišta k přepínání mezi jednotlivými menu
2. Sekce s filtrem. Jestli nejsou žádné parametry vybrány, tak je filtr neaktivní
3. Sekce k exportování a importování konfigurací do souboru ve formátu txt
4. Panel pro ovládání procesu
5. Tabulka se záznamy událostí, které během procesu nastaly
6. Tabulka s počty jednotlivých výrobků v uložištích



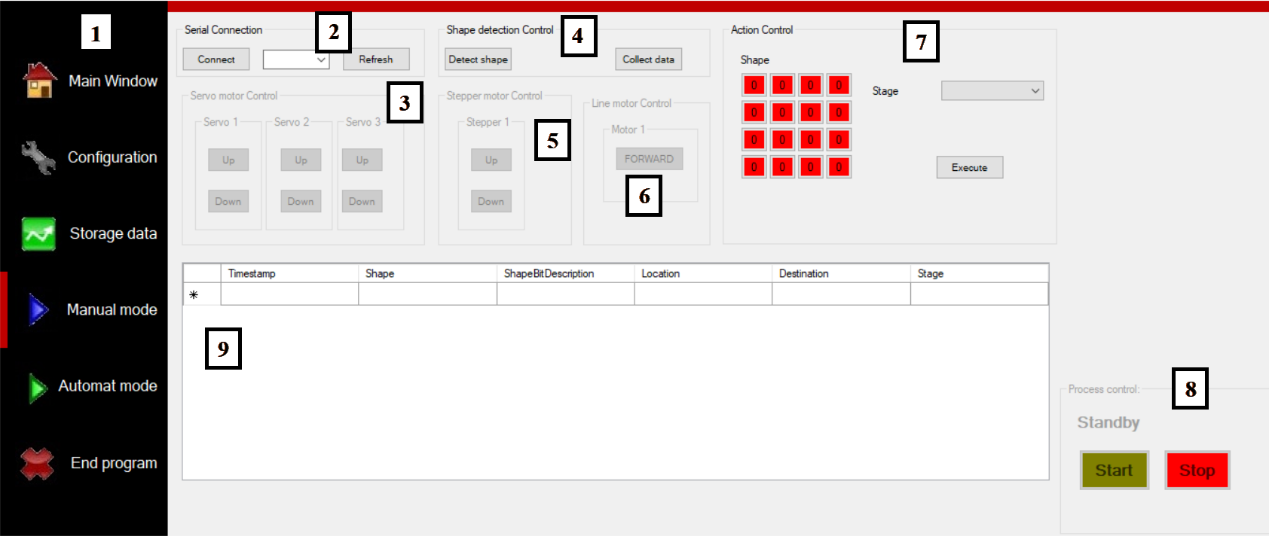
Obrázek 69: Menu Data storage

* 1. Menu Manual mode

Menu Manual mode slouží k ručnímu ovládání jednotlivých akčních členů. V tomto menu je možné připojit mikro kontrolér, ovládat jednotlivé motory, spustit algoritmus pro detekci tvaru výrobku, ručně generovat události systému a zobrazit seznam událostí.

Jednotlivé akční členy jsou neaktivní do doby, než je připojen mikro kontrolér. Mikro kontrolér je připojen přes sériový port. Jestli není žádné zařízení připojené na sériovém portu, není možné spustit akční členy.

1. Postranní lišta k přepínání mezi jednotlivými menu
2. Sekce pro připojení mikro kontroléru. Seznam zařízení je generován po spuštění programu. Pro obnovu seznamu stiskněte Refresh.
3. Sekce pro ovládání servo motorů. Tlačítko Up natočí servo motory do úhlu natočení 90°. Tlačítko Down natočí servo motory do úhlu natočení 0°.
4. Sekce pro ovládání senzoru tvaru. Tlačítko Detect shape spustí algoritmus detekce tvaru. Tlačítko Collect data stáhne data z mikrokontroleru.
5. Sekce pro ovládání krokového motoru. Tlačítko Up spustí zpětný chod motoru. Senzor se tak ocitne v horním stavu, kdy se samotný senzor nedotýká výrobku. Tlačítko Down přitlačí senzor tvaru na výrobek.
6. Sekce pro ovládání motoru dopravníku. Tlačítko Forward spustí chod motoru, který posune výrobek dále po lince.
7. Sekce pro vytváření událostí. Je vytvořen záznam, který obsahuje informace o tvaru výrobku, místu vzniku události, cíli události a času vzniku události.
8. Panel pro ovládání procesu
9. Tabulka se záznamy vzniklých událostí



Obrázek 70: Menu manual mode

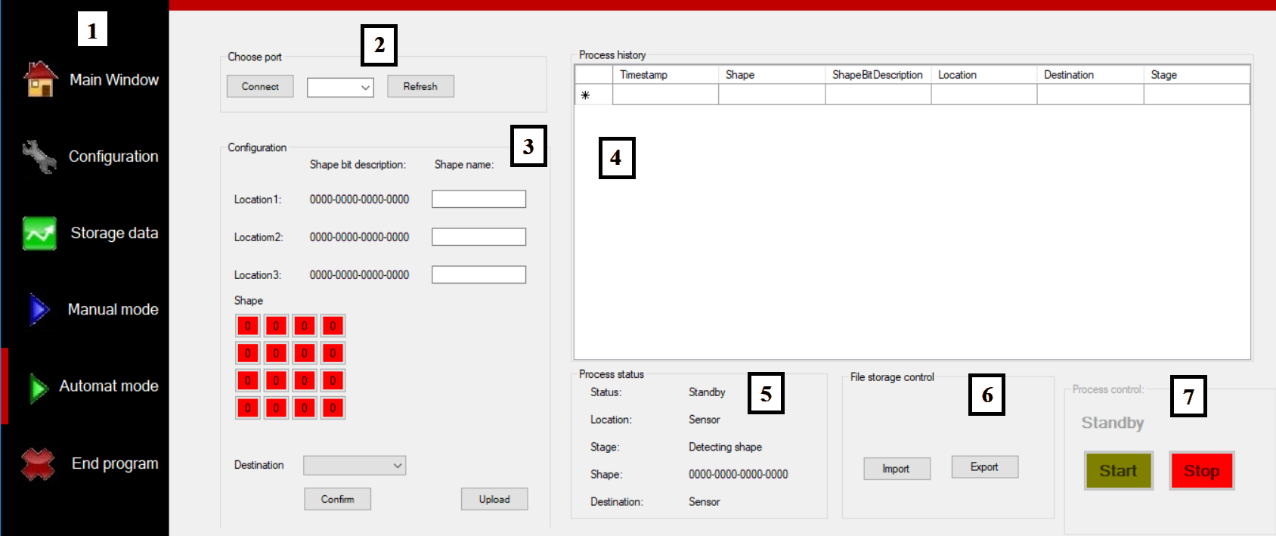
* 1. Menu Automat mode

Menu Automat mode slouží k ovládání autonomního režimu třídící linky. Menu stejně jako ruční režim obsahuje sekci s připojením mikro kontroléru. Na rozdíl od ručního režimu není možné použít přednastavené konfigurace, ale musí být vybrány 3 výrobky, které budou třízeny do jednotlivých uložišť.

Po spuštění procesu mikro kontrolér vykonává jednotlivé kroky a pouze zaznamenává jednotlivé události do tabulky v menu. Tabulku se seznamy událostí je možné importovat, nebo exportovat do textového souboru.

Tabulku se seznamem událostí je možné zobrazit a filtrovat v menu Storage data. Aktuální stav procesu je zobrazován v sekci 5.

1. Postranní lišta k přepínání mezi jednotlivými menu
2. Sekce pro připojení mikro kontroléru. Seznam zařízení je generován po spuštění programu. Pro obnovu seznamu stiskněte Refresh.
3. Sekce pro nastavení konfigurace mikro kontroléru. Ve spodní části sekce je možné definovat tvar výrobku, v nabídce Destination je možné vybrat cílovou lokaci a pomocí tlačítka Confirm potvrdit kombinaci tvaru a uložiště. Třídící linka tuto kombinaci použije pro přiřazení tvaru do uložiště.
4. Tabulka se záznamy vzniklých událostí
5. Sekce s detaily o aktuálním stavu procesu.
6. Sekce k exportování a importování konfigurací do souboru ve formátu txt
7. Panel pro ovládání procesu



Obrázek 71: Menu Automatic mode

1. Seznam literatury:
2. BARCUCH, Libor. *TRANSPORTNÍ ZAŘÍZENÍ PRO DOPRAVU KUSOVÝCH PŘEDMĚTŮ* [online]. Brno, 2017 [cit. 2019-04-22]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/100909> Vysoké učení technické Brno. Vedoucí práce Ing. Jiří Malášek
3. MICHAEL, Konečný. *Návrh robotického ramene* [online]. Brno, 2016/207 [cit. 2019-04-22]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/100537> . Vedoucí práce Tomáš Marada
4. BENDA, Tomáš. *Konstrukce a řízení manipulátoru* [online]. Pardubice, 2015 [cit. 2019-04-22]. Dostupné z: <https://portal.upce.cz/StagPortletsJSR168/PagesDispatcherServlet?pp_destElement=%23ssSouboryStudentuDivId_717&pp_locale=cs&pp_reqType=render&pp_portlet=souboryStudentuPagesPortlet&pp_page=souboryStudentuDownloadPage&pp_nameSpace=G3593&soubidno=16098> . Diplomová. Vedoucí práce Havlíček Libor
5. PROKOP, Michal. *Model třídicí linky v systému Tecnomatix Process Simulate* [online]. Brno, 2017 [cit. 2019-04-22]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/102661>. Vedoucí práce Václav Kaczmarczyk
6. VICAN, Pavel. *Třídící linka* [online]. Česká lípa, 2017 [cit. 2019-04-22]. Dostupné z: http://www1.fs.cvut.cz/stretech/2017/sbornik\_2017/pdf/16.pdf. Maturitní práce. Střední průmyslová škola, Česká Lípa. Vedoucí práce Josef Bašta.¨
7. KOUMARIS, Nick. C# Application Arduino Communication. *Educ8s.tv* [online]. [cit. 2019-04-22]. Dostupné z: <https://educ8s.tv/c-application-arduino-communication/>
8. 375668457. C# Serial Port Communication Arduino. *Instructables* [online]. [cit. 2019-04-22]. Dostupné z: <https://www.instructables.com/id/C-Serial-Communication-With-Arduino/>
9. *Historický vývoj automatizace? Poznejte 12 zásadních dat* [online]. [cit. 2019-04-21]. Dostupné z: <https://factoryautomation.cz/historicky-vyvoj-automatizace-poznejte-12-zasadnich-dat/>
10. [online]. [cit. 2019-04-21]. Dostupné z: <http://nd04.jxs.cz/419/332/3ce8c88974_74039634_o2.jpg>
11. [online]. [cit. 2019-04-21]. Dostupné z: <http://fyzweb.cz/materialy/sily/paka/kolec1.gif>
12. [online]. [cit. 2019-04-21]. Dostupné z: <http://www.j4.cz/uploads/tx_odphotogallery/thumbs/e4c6364f0a0aa08538edd5177649f77b.jpg>
13. [online]. [cit. 2019-04-21]. Dostupné z: https://www.zenavaute.cz/wp-content/uploads/vyroba.jpg
14. [online]. [cit. 2019-04-21]. Dostupné z: <http://www.newsreck.com/wp-content/uploads/2019/03/Global-Wafer-And-Integrated-Circuits-Market.jpg>
15. Maixner, L.; et al. Optoelektrické senzory binární. *Mechatronika*, 1st ed.; Computer Press: Brno, 2006; Chapter 3.2.1.6., p 49.
16. Maixner, L.; et al. Clonicí senzor rozměrů. *Mechatronika*, 1st ed.; Computer Press: Brno, 2006; Chapter 3.2.1.7, p 52.
17. Maixner, L.; et al. Clonicí senzor rozměrů. *Mechatronika*, 1st ed.; Computer Press: Brno, 2006; Chapter 3.2.1.7, p 53.
18. [online]. [cit. 2019-04-21]. Dostupné z: <http://www.riko.cz/image.php?nid=12547&oid=3490858&width=640&height=480>
19. [online]. [cit. 2019-04-21]. Dostupné z: <http://www.zakhodonin.cz/src/img/produkty/dopravniky/dopravniky/dopravnik_1.jpg>
20. [online]. [cit. 2019-04-21]. Dostupné z: <http://www.tmt.cz/cz/CMS/foto/1902.jpg>
21. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.logsys.cz/files/th_8b1a7f5cccad127c10a8fd1f45573c4e.jpg>
22. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.logsys.cz/files/c042868367d48ab0b6a02dfccc0a580a.jpg>
23. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.feifer.cz/obrazky/1678acz.jpg>
24. [online]. [cit. 2019-04-21]. Dostupné z: <https://automatizace.hw.cz//files/styles/full/public/story_automat/11806/vyhybka_5.jpg?itok=2RpPhCv2>
25. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.exasoft.cz/buddy-toys-bcr-10-robotic-arm-kit_ie561785.jpg>
26. SKALICKÝ, Tomáš. *Návrh a realizace robotického manipulátoru* [online]. Pardubice, 2015 [cit. 2017-10-15]. Dostupné z: <https://portal.upce.cz/StagPortletsJSR168/PagesDispatcherServlet?pp_destElement=%23ssSouboryStudentuDivId_745&pp_locale=cs&pp_reqType=render&pp_portlet=souboryStudentuPagesPortlet&pp_page=souboryStudentuDownloadPage&pp_nameSpace=G3593&soubidno=16238> . Vedoucí práce Havlíček Libor
27. [online]. [cit. 2019-04-21]. Dostupné z: <https://t3.ftcdn.net/jpg/01/10/03/22/240_F_110032210_m8lKqMJawvwCrlj2sD1kjB6s8P22klXN.jpg>
28. [online]. [cit. 2019-04-21]. Dostupné z: https://www.souepl.cz/wp-content/ucitele/valecka/at\_mega\_32\_popis%20(Opraveno)\_soubory/image004.jpg
29. [online]. [cit. 2019-04-21]. Dostupné z: <http://ww1.microchip.com/downloads/en/devicedoc/doc2503.pdf>
30. [online]. [cit. 2019-04-21]. Dostupné z: https://i.ebayimg.com/00/s/NDIzWDU4Mw==/z/q10AAOSwBRVae8zd/$\_35.JPG?set\_id=8800005007
31. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.researchgate.net/profile/Shyam_Akashe/publication/257799438/figure/fig1/AS:341731565424653@1458486562832/MUX-graphical-symbol-a-truth-table-b.png>
32. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.researchgate.net/figure/MUX-graphical-symbol-a-truth-table-b_fig1_257799438>
33. [online]. [cit. 2019-04-21]. Dostupné z: <https://assets.nexperia.com/documents/data-sheet/74HC_HCT4067.pdf>
34. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.arduino.cc/en/tutorial/ShiftIn>
35. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.arduino.cc/en/uploads/Tutorial/shftin_cd4021_pins.png>
36. [online]. [cit. 2019-04-21]. Dostupné z: <https://create.arduino.cc/projecthub/SAnwandter1/programming-8x8-led-matrix-23475a>
37. [online]. [cit. 2019-04-21]. Dostupné z: <http://www.lucky-light.com/Dot%20Matrix%20LED%20Displays/5x7_Dot_Matrix/KWM-50571.pdf>
38. [online]. [cit. 2019-04-30]. Dostupné z: <https://create.arduino.cc/projecthub/SAnwandter1/programming-8x8-led-matrix-23475a>
39. [online]. [cit. 2019-04-30]. Dostupné z: <https://images-na.ssl-images-amazon.com/images/I/51653PgxqgL._SX425_.jpg>
40. [online]. [cit. 2019-04-21]. Dostupné z: <https://static3.tme.eu/products_pics/5/4/d/54dcf8dd74a04d30c55078ca113eb5ac/363694.jpg>
41. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.mylms.cz/text-krokovy-motor-princip/>
42. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.mylms.cz/obrazky/elektronika/krokovy-motor-14.png>
43. [online]. [cit. 2019-04-21]. Dostupné z: <https://i.ebayimg.com/images/g/BXoAAOSweBFckLAn/s-l300.jpg>
44. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.electricaleasy.com/2015/01/how-does-servo-motor-work.html>
45. [online]. [cit. 2019-04-21]. Dostupné z: <https://1.bp.blogspot.com/-WWEjibj3iX0/VK1LTqWI9NI/AAAAAAAABCs/FmiULNH04kg/s1600/servo%2Bmotor%2Bworking.png>
46. [online]. [cit. 2019-04-21]. Dostupné z: <https://i.ebayimg.com/images/g/DTQAAOSwFIRbxxlJ/s-l300.jpg>
47. [online]. [cit. 2019-04-21]. Dostupné z: <https://eluc.kr-olomoucky.cz/verejne/lekce/809>
48. [online]. [cit. 2019-04-21]. Dostupné z: <https://eluc.kr-olomoucky.cz/uploads/images/20238/obr.2.gif>
49. [online]. [cit. 2019-04-21]. Dostupné z: <https://i.ebayimg.com/images/g/AhEAAOSwUP5cLCBK/s-l640.jpg>
50. [online]. [cit. 2019-04-21]. Dostupné z: <https://eshop.merkurtoys.cz//data/product/178_352.jpg>
51. [online]. [cit. 2019-04-21]. Dostupné z: <https://images-na.ssl-images-amazon.com/images/I/61pnAVpZjuL._SY741_.jpg>
52. [online]. [cit. 2019-04-21]. Dostupné z: <https://pino-tech.eu/wp-content/uploads/2018/06/irlz44n.jpg>
53. [online]. [cit. 2019-04-21]. Dostupné z: <http://bildr.org/2012/03/rfp30n06le-arduino/>
54. [online]. [cit. 2019-04-21]. Dostupné z: <https://42bots.com/tutorials/bipolar-stepper-motor-control-with-arduino-and-an-h-bridge/>
55. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.instructables.com/id/Arduino-How-to-Control-Servo-Motor-With-Arduino/>
56. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.microchip.com/mplab/avr-support/atmel-studio-7>
57. [online]. [cit. 2019-04-21]. Dostupné z: <http://and-tech.pl/evb-4-3/>
58. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.arduino.cc/en/main/software>
59. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.arduino.cc/reference/en/language/functions/digital-io/digitalread/>
60. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.arduino.cc/reference/en/language/functions/digital-io/digitalwrite/>
61. [online]. [cit. 2019-04-21]. Dostupné z: <https://i.ebayimg.com/images/g/FAcAAOSwLfRbPg0F/s-l640.jpg>
62. [online]. [cit. 2019-04-30]. Dostupné z: <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTYE4F_YMtfyQwVEHx_Ei8wSmfKx58ZBC7T0jfNmKw3UwCVjEZjFg>
63. [online]. [cit. 2019-04-21]. Dostupné z: <https://www.hackster.io/ianabcumming/arduino-simple-camera-slider-electronics-9e9d8b>
64. [online]. [cit. 2019-04-22]. Dostupné z: <https://www.arduino.cc/reference/en/language/structure/control-structure/while/>
65. [online]. [cit. 2019-04-22]. Dostupné z: <https://www.arduino.cc/reference/en/language/structure/control-structure/if/>
66. [online]. [cit. 2019-04-22]. Dostupné z: <https://www.arduino.cc/en/Tutorial/SerialEvent>
67. [online]. [cit. 2019-04-22]. Dostupné z: <https://www.arduino.cc/en/Reference/Loop?setlang=it>
68. [online]. [cit. 2019-04-22]. Dostupné z: <https://visualstudio.microsoft.com/downloads/?rr=https%3A%2F%2Fwww.google.cz%2F>