



Moderní výuka
robotiky s roboty

DOBOT

Magician

Obsah

Úvod	1
Technické vybavení.....	2
Výuka a příprava na výuku	3
Úloha 1: Přímá a inverzní kinematika robotů.....	4
Teoretický rozbor úlohy.....	4
Přímá kinematika.....	5
Výpočet.....	5
Inverzní kinematika	6
Úloha 2: Rozšíření robota Dobot Magician o běžně dostupný senzor barvy TCS3200	9
Teoretický rozbor úlohy.....	9
Digitálně analogový převodník MAX522	10
Ukázka programu	12
Senzor barvy TCS3200	13
Ukázka kódu pro čtení barvy	14
Kód by potom mohl vypadat takto:.....	15
Program pro načtení a zobrazení hodnoty z EIO15 (DAC GP2) může potom vypadat takto:	16
Úloha 3: Dobot Magician a strojové vidění	17
Teoretický rozbor úlohy.....	17
Převod na šedý odstín	17
Thresholding.....	17
Hledání kontur.....	18
Závěr	20
Při práci na projektech získají dovednosti z těchto klíčových oblastí:.....	21
Základy robotiky	21
Programování	21
Strojové vidění.....	21
Interdisciplinární dovednosti.....	21
Kritické myšlení a inovace	21
Praktické aplikace teoretických konceptů.....	21
Seznam obrázků.....	24
Seznam tabulek	24
Zdroje.....	24

Úvod

Robotika prošla v posledních desetiletích mimořádným rozvojem. Spojuje v sobě prvky inženýrství, informatiku a v dnešní době také stále častěji skloňovanou umělou inteligenci k vytváření strojů, které jsou schopny vykonávat úlohy od jednoduchých mechanických operací, až po složitější operace, které pro svoji činnost často vyžadují určitou množinu kognitivních funkcí.

S příchodem nového oboru L+H, který spojuje obory 26-41-L/01 Mechanik elektrotechnik a 26-51-H/01 Elektrikář, byl do výuky začleněn předmět Robotika.

Klíčové pro vzdělávání budoucích mechatroniků je vybavení, které škola pro výuku technických předmětů má. Vybavení škol v oblasti robotiky v současné době značně pokulhává za skutečnými trendy, které dnes hýbou světem průmyslu.

Pokud škola má do výuky zařazeny průmyslové roboty, často se jedná o jednoho zapůjčeného robota a pár výukových hodin pro ukázkou, jak vlastně daný typ robota funguje.

Společnost Dobot Robotics byla založena v roce 2015 a je tvůrcem prvního stolního kolaborativního robota na světě.

Roboty Dobot jsou v současné době v provozu ve více než 15 průmyslových odvětvích včetně spotřební elektroniky, automobilového průmyslu, zpracování kovů, polovodičů, zdravotnictví, chemického průmyslu a maloobchodu.

Mimo výrobu profesionálních průmyslových robotů se firma zaměřila i na školství a tak dnes mají školy možnost používat výukové roboty Dobot Magician – Education, díky kterým mají žáci možnost pochopit základní principy návrhu průmyslových robotů od mechaniky, přes přímou a inverzní kinematiku nebo programování, až po možnost návrhu vlastního hardware pro vylepšení kognitivních funkcí robota.

Tento dokument se zabývá využitím robotů Dobot Magician na Střední odborné škole a Středním odborném učilišti v Lanškrouně v předmětech Robotika a Odborný výcvik.

Technické vybavení

Roboty Dobot Magician a příslušenství jsou aktuálně používány v učebnách odborného výcviku a výpočetní techniky.

V rámci odborného výcviku jsou žáci rozděleni do skupin po 8 žácích na učebnu a výuka probíhá formou rotace mezi specializacemi po 14 dnech.

Máme celkem 4 roboty a to znamená vždy skupinu dvou žáků na robota, což je ideální.

Na počátku bylo zamýšleno používat roboty primárně v rámci výuky odborného výcviku, ale při přípravách na výuku se ukázalo, že roboty se hodí i do teoretických předmětů, byť na nich žáci budou muset pracovat ve větších skupinách.

Roboty jsou tedy určeny pro výuku předmětů:

- **Robotika** – 3. a 4. ročník (33 hodin za školní rok)
- **Odborný výcvik** – 3. a 4. ročník (28 hodin za školní rok)

Začátkem školního roku jsme zahájili práce na učebně zaměřené čistě na robotiku, kde je cílem vytvořit robotické pracoviště (menší výrobní linku), díky kterému bude výuka robotiky komplexnější, k čemuž nás roboty a příslušenství doslova vyzývají.

Již několik let je škola aktivně zapojena do projektu Technohrátky, což je projekt na podporu studijní volby technických a odborných oborů pro žáky základních škol. Letošní ročník odstartuje 27. března právě na naší škole, kde chceme roboty ukázat i těmto žákům a seznámit je s tím, jak snadno a rychle jsou schopny se naučit takového robota naprogramovat.

Pořádáme také projektové dny v rámci podpory základního a středního školství IKAP, kde roboty Dobot byly využity v rámci projektového dne zaměřeného na 3D tisk nebo robotiku.

Robot Dobot Magician	4ks
- Sada laserového gravírování	-
- Sada pro 3D tisk	-
- Sada uchopovačů	-
- Kompresor	-
Pásový dopravník	2 ks
Lineární pohon	1 ks

Tabulka 1 – Zapůjčené technické vybavení



Vědění je poklad, ale praxe je klíč k němu. – *Thomas Fuller*



Výuka a příprava na výuku

V následující části dokumentu se zaměříme na tři úlohy, díky kterým žáci mohou rozvinout své znalosti a prohloubit zájem o robotiku.

- **Přímá a inverzní kinematika robotů**
- **Návrh a připojení senzoru barvy**
- **Strojové vidění – propojení robota Dobot Magician s webkamerou**

Roboty Dobot Magician umožňují celkem snadnu práci v rámci softwarové výbavy dodané firmou Dobot.

K výuce zájmových kroužků pro ZŠ nebo pro základní seznámení s roboty bylo dříve potřeba nainstalovat aplikaci DobotStudio nebo DobotLab. V současné době stačí nainstalovat aplikaci [DobotLink](#) a používat webový prohlížeč.

Vše „běží“ v cloudu, a tak je potřeba se zdarma zaregistrovat na stránkách: <https://dobotlab.dobot.cc/>

Úlohy jsou propojeny s nižší vrstvou, která s dobotem komunikuje prostřednictvím programovacího jazyka [Python](#). Proto je potřeba mít jej v počítači nainstalovaný ideálně spolu s některým z volně dostupných vývojových prostředí, které celkové programování zpříjemní např. [PyCharm](#) nebo [Visual Studio Code](#).

Bez potřebných knihoven by ovšem programování robota nefungovalo, proto je potřeba použít soubory pro [secondary development](#).

*Codito ergo sum
„I code, therefore I am“*

Úloha 1: Přímá a inverzní kinematika robotů

Cílem výuky kinematiky robotů je žáky seznámit s tím, jak určit natočení jednotlivých servomotorů robotického ramene v závislosti na poloze **TCP (Tool Center Point)** v kartézském souřadnicovém systému. Žáci díky tomu pochopí, jakým způsobem probíhá řízení pohybu robotů a k čemu jsou dobré přímá a inverzní kinematika robotů.

Teoretický rozbor úlohy

Kinematika robota popisuje jeho pohyblivost, manipulační možnosti, předurčuje jeho konstrukci i počet stupňů volnosti.

Počet stupňů volnosti se rovná počtu os, podle kterých se hlavice robota může pohybovat a často se shoduje s počtem pohonů.

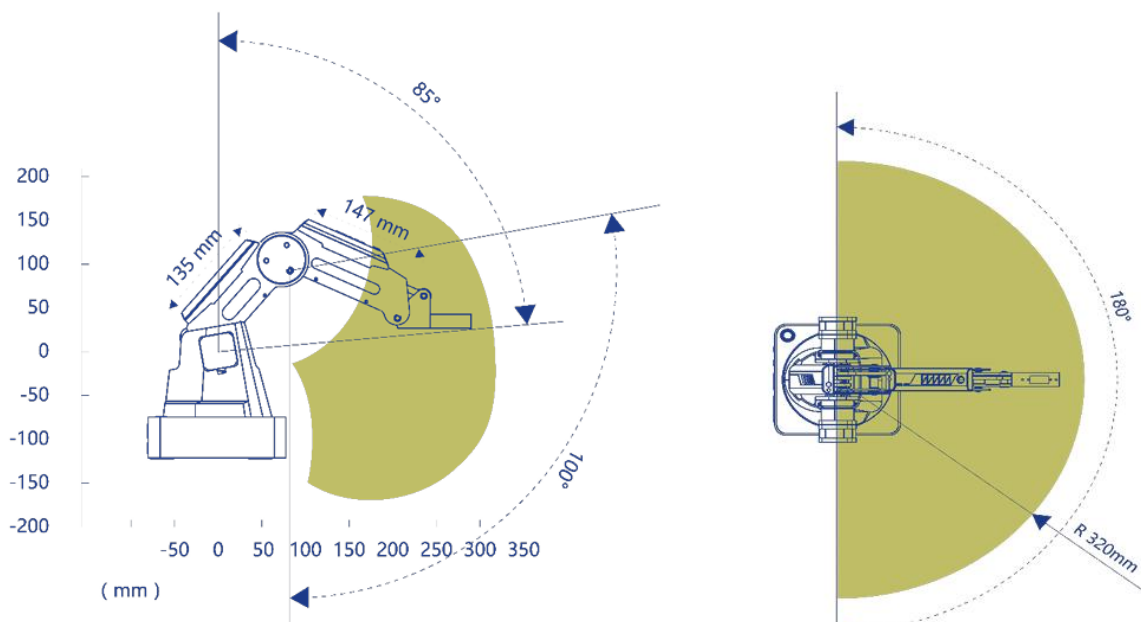
Dobot Magician disponuje čtyřmi stupni volnosti a konstrukčně spadá do kategorie angulárních robotů (RRR).

Rozlišujeme dva druhy kinematiky:

- Přímá kinematika (*forward kinematics*)
- Inverzní kinematika (*inverse kinematics*)

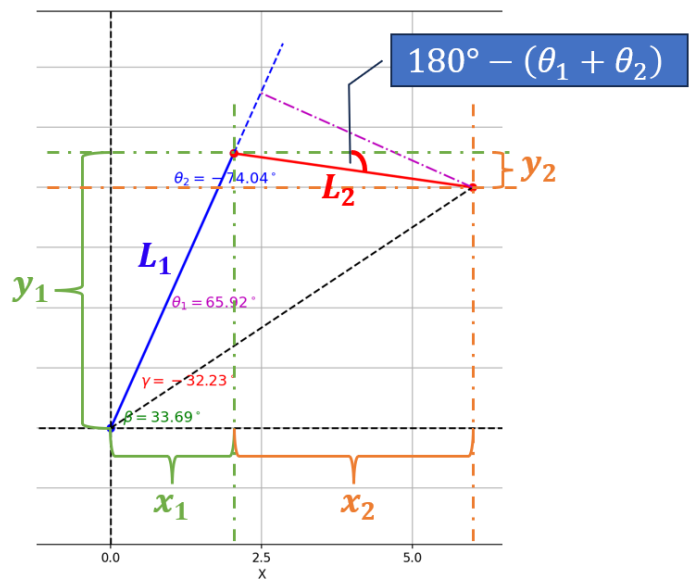
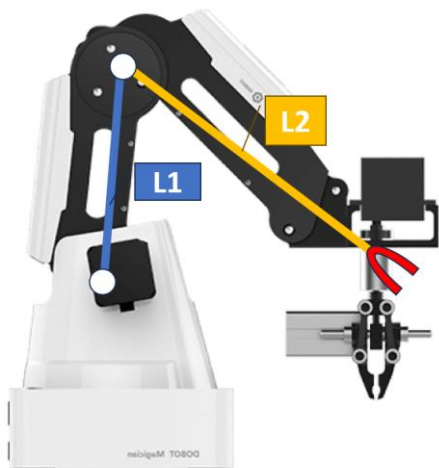
V případě přímé kinematiky řešíme úlohu, kdy známe úhly natočení jednotlivých kloubů robota, délku jeho ramen a chceme vypočítat bod ve kterém se TCP nachází.

Při výpočtech inverzní kinematiky známe souřadnice bodu, ve kterém se TCP nachází, délky jeho ramen a chceme získat hodnoty úhlů natočení jednotlivých kloubů robota.



Obrázek 1 – Pracovní prostor robota Dobot Magician s uvedenými délkami ramen.

Přímá kinematika



Při výpočtech přímé kinematiky vycházíme ze základních poznatků z oblasti trigonometrie.

Mějme 2D robota, podobného bočnímu pohledu na robota Dobot Magician. Tento 2D robot bude mít pouze dva stupně volnosti a dvě ramena – L1 a L2. Známe velikosti úhlů θ_1 a θ_2 a chceme zjistit, na jakých souřadnicích se nachází koncový bod TCP.

Výpočet

$$\sin(x) = \frac{\text{protilehlá}}{\text{přepona}} \rightarrow \sin(\theta_1) = \frac{y_1}{L_1}$$

$$y_1 = L_1 \cdot \sin(\theta_1)$$

$$y_2 = L_2 \cdot \sin(180^\circ - (\theta_1 + \theta_2))$$

$$\cos(x) = \frac{\text{přilehlá}}{\text{přepona}} \rightarrow \cos(\theta_1) = \frac{x_1}{L_1}$$

$$x_1 = L_1 \cdot \cos(\theta_1)$$

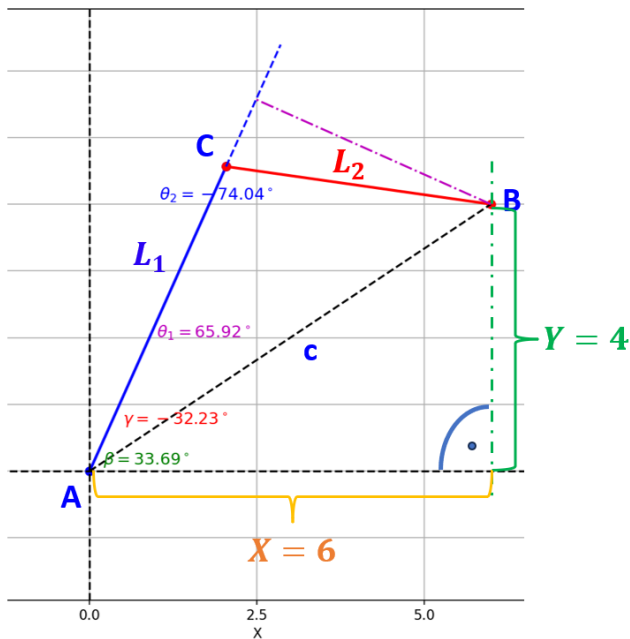
$$x_2 = L_2 \cdot \cos(180^\circ - (\theta_1 + \theta_2))$$

$$Y = y_1 - y_2$$

$$X = x_1 + x_2$$

$$\underline{\underline{\text{TCP} = [X, Y]}}$$

Inverzní kinematika



Známe délku ramen robota a souřadnice bodu, do kterého se chceme s TCP dostat.

Pro jednoduchost opět uvažujme 2D souřadnicový systém, kdy na robota koukáme z boku.

Využíváme základních poznatků z oblasti trigonometrie.

1. Pythagorova věta

$$c = \sqrt{x^2 + y^2}$$

2. Výpočet úhlu θ_2

$$c^2 = L_2^2 + L_1^2 - 2 \cdot L_2 L_1 \cdot \cos(\theta_2)$$

$$\cos(\theta_2) = -\frac{c^2 - L_2^2 - L_1^2}{2 \cdot L_1 \cdot L_2}$$

$$\theta_2 = \arccos\left(-\frac{c^2 - L_2^2 - L_1^2}{2 \cdot L_1 \cdot L_2}\right)$$

3. Výpočet úhlu θ_1

$$\sin(\beta) = \frac{y}{c} \rightarrow \beta = \arcsin\left(\frac{y}{c}\right)$$

$$L_2^2 = L_1^2 + c^2 - 2 \cdot L_1 \cdot c \cdot \cos(\gamma)$$

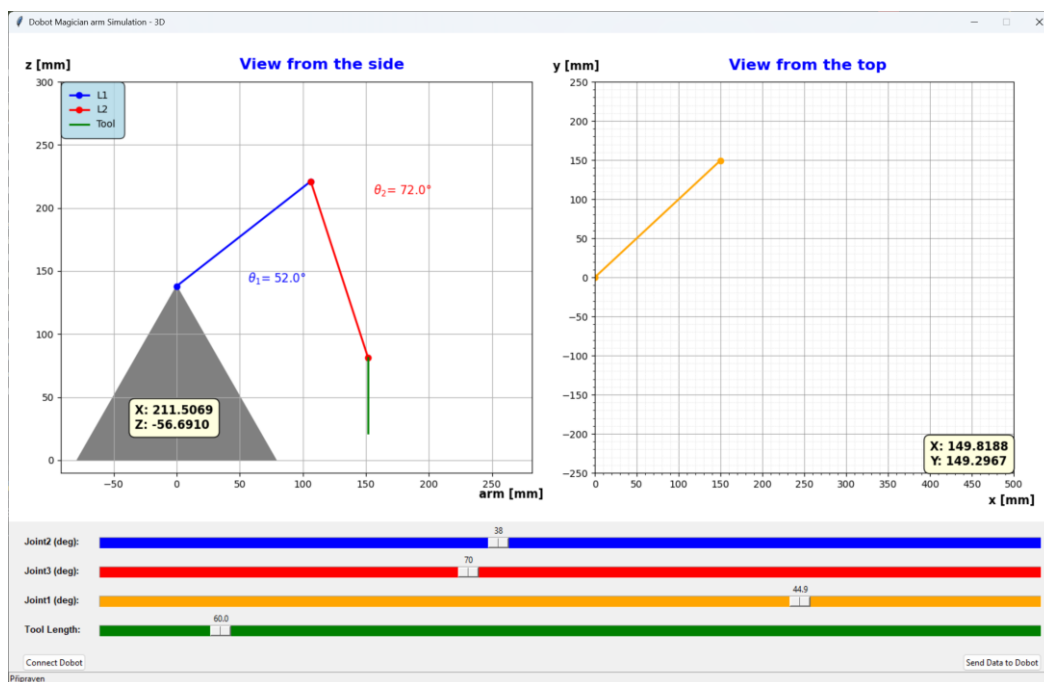
$$\gamma = \arccos\left(-\frac{L_2^2 - L_1^2 - c^2}{2 \cdot L_1 \cdot c}\right)$$

$$\theta_1 = \gamma + \beta$$

Při výuce kinematiky jsou žáci seznámeni s výpočty přímé a inverzní kinematiky, řeší konkrétní zadání, které si následně ověří zadáním správných souřadnic do předem připraveného programu, který byl pro tento účel vytvořen.

Zadání může vypadat takto:

- V pracovním prostoru robota se nachází kostička o rozměrech 25 x 25 x 25 mm.
- Souřadnice kostičky jsou [150, 150, 25]
- Vypočítej jednotlivé úhly naklonění ramen L1 a L2, výpočet ověř zadáním hodnot připojenému robotu.



Obrázek 2 - Aplikace pro výuku přímé a inverzní kinematiky

V další fázi je možno požadovat, aby žáci vytvořili jednoduchý program v Pythonu, který kostičku uchopí např. pomocí přísavky a zvedne ji do určité výšky.

Příklad scriptu napsaného v Pythonu:

```
1 import DobotDllType as dType
2
3 lastIndex = 0
4
5 CON_STR = {
6     dType.DobotConnect.DobotConnect_NoError: "DobotConnect_NoError",
7     dType.DobotConnect.DobotConnect_NotFound: "DobotConnect_NotFound",
8     dType.DobotConnect.DobotConnect_Occupied: "DobotConnect_Occupied"
9 }
10
11 api = dType.load()
12
13 state = dType.ConnectDobot(api, portName: "COM3", baudrate: 115200)[0]
14 print("Connect status:", CON_STR[state])
15
16 if state == dType.DobotConnect.DobotConnect_NoError:
17     lastIndex = dType.SetPTPCmd(api, ptpMode: 4, x: 52, y: 72, z: 45, rHead: 0, isQueued=1)[0]
18     lastIndex = dType.SetEndEffectorSuctionCup(api, enableCtrl: 1, on: 1)
19     lastIndex = dType.SetPTPCmd(api, ptpMode: 4, x: 24, y: 49, z: 44.9, rHead: 0, isQueued=1)[0]
20
21     dType.SetQueuedCmdStartExec(api)
22     while lastIndex > dType.GetQueuedCmdCurrentIndex(api)[0]:
23         dType.dSleep(100)
24
25     dType.SetQueuedCmdStopExec(api)
26     print(dType.GetPose(api))
27     dType.DisconnectDobot(api)
```

Žáci se aktivně zapojili do výuky, dychtili po dalším zadání, soutěžili mezi sebou, kdo rychleji vypočítá jednotlivé úhly. Zkoušeli sami vymýšlet různá zadání, která zahrnovala i více pohybů robota.

Díky jednoduchým úlohám pochopili, že trigonometrie hraje v robotice nezastupitelnou roli i přesto, že dodávané vývojové prostředí vše počítá automaticky.

Úloha 2: Rozšíření robota Dobot Magician o běžně dostupný senzor barvy TCS3200

Cílem úlohy je provést žáky kompletním návrhem a připojením senzoru barvy k robotu Dobot Magician. Seznámit je s I/O porty robota a ukázat jim, jakým způsobem s těmito porty v rámci vývojového prostředí nebo pythonu pracovat.

Teoretický rozbor úlohy

Robot Dobot Magician disponuje několika I/O porty, které je možno softwarově ovládat a umožnit tak připojit k robotu další periferie – pásový dopravník, lineární pohon nebo například různé senzory, díky kterým má možnost reagovat na změny v jeho okolí.

Jednou z takových periférií může být senzor barvy díky kterému je robot schopen poznat, jakou barvu má materiál, se kterým manipuluje.

Nejběžněji dostupným senzorem barvy je senzor s čipem TCS3200. Tento senzor disponuje 8 piny, z nichž dva jsou určeny pro připojení napájecího napětí 2,7 až 5,5 V a GND. Dále pak jedním pinem, který poskytuje informaci o detekované barvě (PWM modulace), dva pro omezení výstupní frekvence a dva pro nastavení snímaného spektra.

V úvahu lze brát pro připojení k robotu několik způsobů z nichž jedním může být připojení senzoru na komunikační interface (UART) anebo můžeme využít rozhraní pro periferní zařízení.

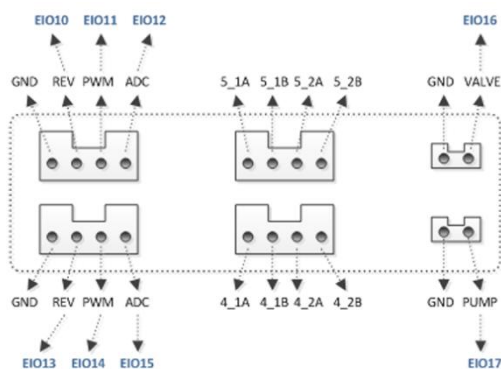
Při hlubším studiu dokumentace zjistíme, že PWM je zde pouze jako výstup a to znamená, že bude potřeba vytvořit hardware pro čtení informací ze senzoru a následné posílání informací prostřednictvím sběrnice robotovi.

Robot disponuje porty označenými jako GP1, GP2, z nichž každý obsahuje čtyři piny (GND, REV, PWM a ADC).

Pin **REV** je schopen při aktivaci na výstup dodat napětí **5V/1A**, což by mohlo stačit i pro napájení senzoru. Jako vstup je zde pin označený zkratkou **ADC** tj. je analogově digitální pin, který obsahuje analogově-digitální převodník.

Potřebujeme tedy získat informace ze senzoru barvy, převést je na analogový signál (úroveň napětí) a pomocí robota tuto úroveň přečíst, a přiřadit ji konkrétní barvu.

Pro načtení hodnoty ze senzoru může být použit jakýkoli mikropočítač. V tomto případě použijeme vývojovou desku Arduino UNO a pro převod na analogový signál DAC převodník MAX522.



Obrázek 3- Rozhraní pro připojení periférií

Digitálně analogový převodník MAX522

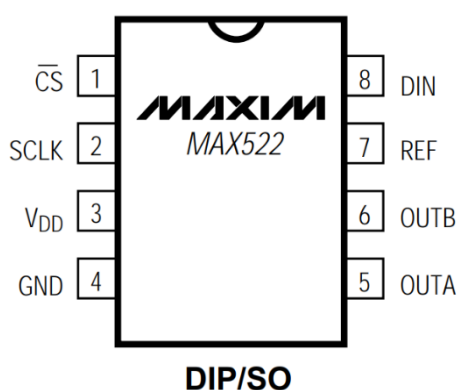
MAX522 obsahuje dva 8-bitové, digitálně-analogové převodníky (DAC A, DAC B) v malém provedení.

MAX522 pracuje s jediným napájecím napětím v rozsahu +2,7 V až +5,5 V.

Obvod využívá třívodičové sériové rozhraní, které pracuje s taktovací frekvencí až 5MHz a je kompatibilní s SPI™, QSPI™ a standardy rozhraní Microwire™.

Sériový vstupní posuvný registr má délku 16 bitů a skládá se z osmi bitů vstupních dat převodníku D/A a osmi bitů pro řízení a vypnutí D/A převodníku. Registry D/A převodníku lze načítat nezávisle nebo paralelně při kladné hraně CS.

MAX522 má velmi nízkou spotřebu energie a malé rozměry.



Obrázek 4- Zapojení pouzdra MAX522

PIN	Název	Funkce
1	CS	Chip Select (aktivní low). Umožňuje posun dat do 16bitového posuvného registru. Programovací příkazy jsou prováděny při náběžné hraně CS
2	SCLK	Hodinový signál clock. Data se přivádějí vzestupnou hranou SCLK.
3	Vcc	Kladné napájení (2,7 V až 5,5 V). Bypass řešen kondenzátorem 0,22μF na GND.
4	GND	Zem.
5	OUTA	Výstup napětí z převodníku D/A. Připojte kondenzátor 0,1 μF nebo větší ke GND.
6	OUTB	Výstup napětí z převodníku D/A. Připojte kondenzátor 0,01 μF nebo větší ke GND.
7	REF	Referenční vstup pro převodníky DAC A a DAC B.
8	DIN	Sériový datový vstup 16bitového posuvného registru. Data jsou do registru posílána se vzestupnou hranou SCLK.

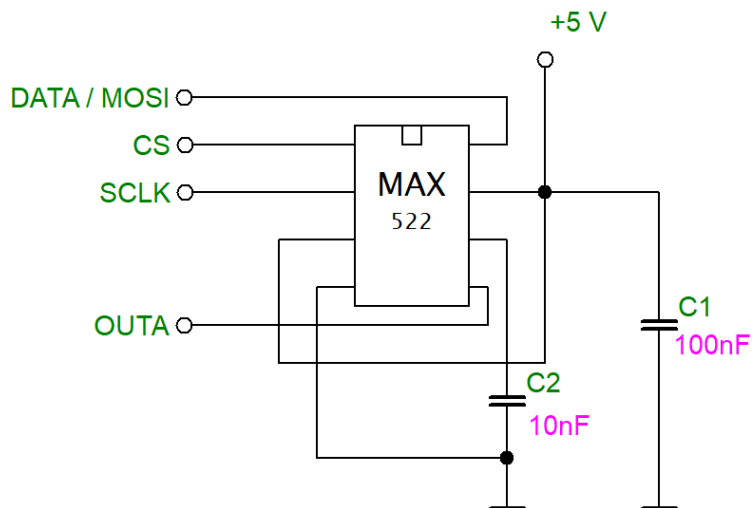
Tabulka 1 - Popis pinů obvodu MAX522

CONTROL									DATA								FUNCTION
UB1	UB2	UB3	SB	SA	UB4	LB	LA	B7 MSB	B6	B5	B4	B3	B2	B1	B0 LSB		
X	X	1	*	*	0	0	0	X	X	X	X	X	X	X	X	No Operation to DAC Registers	
X	X	1	*	*	0	0	0									Unassigned Command	
X	X	1	*	*	0	1	0	8-Bit DAC Data								Load Register to DAC B	
X	X	1	*	*	0	0	1	8-Bit DAC Data								Load Register to DAC A	
X	X	1	*	*	0	1	1	8-Bit DAC Data								Load Both DAC Registers	
X	X	1	0	0	0	*	*	X	X	X	X	X	X	X	X	All DACs Active	
X	X	1	0	0	0	*	*	X	X	X	X	X	X	X	X	Unassigned Command	
X	X	1	1	0	0	*	*	X	X	X	X	X	X	X	X	Shut Down DAC B	
X	X	1	0	1	0	*	*	X	X	X	X	X	X	X	X	Shut Down DAC A	
X	X	1	1	1	0	*	*	X	X	X	X	X	X	X	X	Shut Down All DACs	

X = Don't care.

* = Not shown, for the sake of clarity. The functions of loading and shutting down the DACs and programming the logic can be combined in a single command.

Tabulka 2- Příklad pro vyslání slova CONTROL a DATA do MAX522 pomocí Arduina.



Obrázek 5- Schéma zapojení obvodu k mikropočítači.

CONTROL - Zápis dat do registru DAC A

UB1	UB2	UB3	SB	SA	UB4	LB	LA
0	0	1	0	0	0	0	1

00100001b = 0x21 (hex)

DATA - Zápis dat do registru DAC A

00000000b – 11111111b = 0 – 255 (dec)

Ukázka programu

```
1  #include <SPI.h>
2
3  #define DAC_CS 0
4
5  void setDAC(int digV) {
6      digitalWrite(DacCS, LOW);
7      SPI.transfer(0x21); // send a command byte DAC_A only
8      SPI.transfer(digV); // send digV
9      digitalWrite(DacCS, HIGH); // UnSelect DAC_A
10 }
11
12
13 void setup() {
14     // put your setup code here, to run once:
15     pinMode(DAC_CS, OUTPUT);
16     SPI.begin();
17     digitalWrite(DacCS, LOW); // Chip Select the DAC
18     SPI.transfer(0x30); // Shut down DAC-B
19     digitalWrite(DacCS, HIGH); // UnSelect the DAC
20 }
21
22 void loop() {
23     for(int i=0; i<256; i++) {
24         setDAC(i);
25         delay(100); // wait 100 ms
26     }
27 }
28
```

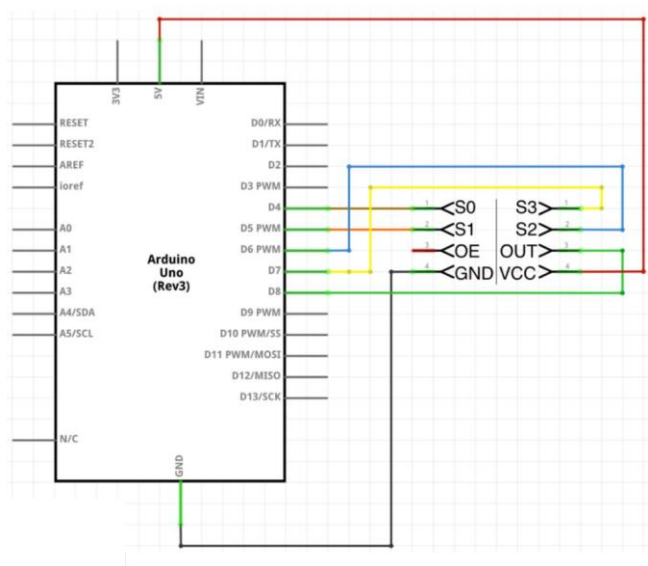
Obrázek 6 - Program zapisující do registru DAC(A) hodnoty pro výstupní napětí od 0 V do 5 V.

Senzor barvy TCS3200

Senzor detekce barevného spektra slouží k detekci barvy odráženého světla. Na modulu se nachází senzor TCS3200, který je složen z hustého pole fotodiody s barevnými filtry. Okolo čipu jsou umístěny čtyři bílé LED diody sloužící pro nasvícení zkoumaného předmětu.

Základní charakteristika:

- 16 fotodiody s červeným filtrem
- 16 fotodiody s zeleným filtrem
- 16 fotodiody s modrým filtrem
- 16 fotodiody bez filtru



Obrázek 7- Schéma propojení senzoru s Arduinem

Snímání barvy	S2	S3
Červená	L	L
Modrá	L	H
Bez filtru	H	L
Zelená	H	H

Tabulka 3- Druhy filtrů a jejich použití

Výstupní frekvence	S2	S3
Deaktivace senzoru	L	L
12 kHz	L	H
120 kHz	H	L
600 kHz	H	H

Tabulka 4 - Nastavení frekvenčního limitu

Ukázka kódu pro čtení barvy

```
7 int getColor(bool S2, bool S3) {
8     digitalWrite(pinS2,S2);
9     digitalWrite(pinS3,S3);
10    delay(50);
11    return(pulseIn(pinOut, LOW));
12 }
13
14 void setup() {
15     pinMode(pinS0, OUTPUT);
16     pinMode(pinS1, OUTPUT);
17     pinMode(pinS2, OUTPUT);
18     pinMode(pinS3, OUTPUT);
19     pinMode(pinOut, INPUT);
20     // inicializace detektoru barev 12 kHz
21     digitalWrite(pinS0,LOW);
22     digitalWrite(pinS1,HIGH);
23     Serial.begin(9600);
24 }
25
26 void loop() {
27     int frekvenceCervena, frekvenceZelena, frekvenceModra;
28     frekvenceCervena = getColor(LOW, LOW);
29     frekvenceZelena = getColor(HIGH, HIGH);
30     frekvenceModra = getColor(LOW, HIGH);
31
32     Serial.print("R: ");
33     Serial.print(frekvenceCervena);
34     Serial.print(" | G: ");
35     Serial.print(frekvenceZelena);
36     Serial.print(" | B: ");
37     Serial.print(frekvenceModra);
38
39     if (frekvenceCervena < 1300 & frekvenceZelena > 1300) {
40         Serial.print(" | Red. ");
41     }
42     if (frekvenceZelena < 1300 & frekvenceCervena > 1300) {
43         Serial.print(" | Green. ");
44     }
45     if (frekvenceModra < 1300 & frekvenceZelena > 1300) {
46         Serial.print(" | Blue. ");
47     }
48     if (frekvenceCervena < 1300 & frekvenceZelena < 1300 ) {
49         Serial.print(" | Yellow. ");
```

Počet pinů použitých pro připojení k mikrokontroléru můžeme eliminovat, pokud se zaměříme na nastavení jednotlivých pinů, zjistíme například, že S0 a S1 nemusíme ovládat softwarově, ale můžeme je zapojit prostřednictvím jiných součástek k příslušným úrovním napětí: LOW (GND), HIGH (+5 V).

Celé zapojení se tak může být zjednodušeno (Což může být motivací pro další řešení úlohy). Po propojení celého zapojení s předchozím kódem můžeme místo zobrazování hodnot na sériové lince použít jako výstup funkci **setDAC()**, která na základě konkrétních podmínek převede barvu na hodnotu napětí, které se po detekci objeví na výstupu DAC převodníku **MAX522**.

Kód by potom mohl vypadat takto:

```
69     if (frekvenceCervena < 1300 & frekvenceZelena > 1300) {
70         Serial.print(" | Detekce cervene. ");
71         setDAC(63);
72     }
73     if (frekvenceZelena < 1300 & frekvenceCervena > 1300) {
74         Serial.print(" | Detekce zelene. ");
75         setDAC(126);
76     }
77     if (frekvenceModra < 1300 & frekvenceZelena > 1300) {
78         Serial.print(" | Detekce modre. ");
79         setDAC(189);
80     }
81     // detekce vlastní barvy složené z různých složek
82     if (frekvenceCervena < 1300 & frekvenceZelena < 1300 ) {
83         Serial.print(" | Detekce zlute. ");
84         setDAC(252);
85     }
```

Předpokládejme, že hodnota napětí na výstupu MAX522 je při **setDAC(255)** rovna 5 V. Potom můžeme říci, že hodnoty napětí jednotlivých barev jsou:

Barva	Nastavení DAC	Výstup DAC
Červená	63	1.23 V
Modrá	126	2.46 V
Zelená	189	3.69 V
Žlutá	252	4.92 V

Tabulka 5 - Výstup DAC podle nastavených hodnot barev.

Tyto hodnoty lze po připojení k ADC pinu na portu GP1 nebo GP2 robota Dobot Magician snadno číst.

Předpokládejme, že pro senzor použijeme port GP2 jehož zapojení je v tabulce níže.

Označení	GND	REV	PWM	ADC
Pin	-	EIO13	EIO14	EIO15

Tabulka 6 - Označení pinů portu GP2

Pro zjišťování hodnot jednotlivých EIO je potřeba aktivovat vestavěný multiplexing, který v pravidelných intervalech kontroluje stavy EIO patřících pod skupinu datových nebo analogových pinů, které jsou dále rozděleny na skupinu vstupní a skupinu výstupní.

Využijeme metody **dType.SetIOMultiplexing()** a **dType.GetIOADC()**

Program pro načtení a zobrazení hodnoty z EIO15 (DAC GP2) může potom vypadat takto:

```
1 import DobotDllType as dType
2
3 CON_STR = {
4     dType.DobotConnect.DobotConnect_NoError: "DobotConnect_NoError",
5     dType.DobotConnect.DobotConnect_NotFound: "DobotConnect_NotFound",
6     dType.DobotConnect.DobotConnect_Occupied: "DobotConnect_Occupied"
7 }
8
9 api = dType.Load()
10
11 state = dType.ConnectDobot(api, portName: "COM3", baudrate: 115200)[0]
12 print("Connect status:", CON_STR[state])
13
14 dType.SetIOMultiplexing(api, address: 15, multiplex: 4, isQueued: 1)
15 print(dType.GetIOADC(api, addr: 15)[0])
```

Na základě načtených hodnot napětí může robot rozpoznávat barvy – například při třídění různě barevných kostiček.

Tato úloha vzhledem ke své povaze potřebuje k vyřešení větší časovou dotaci, proto je zahrnuta také do výuky odborného výcviku, kde žáci nejdříve sestaví hardware z dostupných modulů (modul senzoru barvy, MAX522, Arduino UNO). Následně navrhnu vlastní desku plošných spojů, což zahrnuje také vlastní návrh s využitím mikroprocesoru firmy Atmel ATMEGA 328P.

Dále pak v hodinách výpočetní techniky zpracují firmware senzoru a následně v hodinách Robotiky propojí senzor s robotem a využívají jej ke kompletnímu vyřešení úlohy.

Žákům se práce na takovýchto projektech líbí. Musí propojit znalosti z několika odborných předmětů a následně je použít ke splnění cíle, kterým je právě například program pro třídění barevných kostiček.

Úloha 3: Dobot Magician a strojové vidění

Cílem úlohy je seznámit žáky se základními principy strojového vidění a jeho využití v oblasti robotiky.

Žáci poznají principy strojového vidění, včetně zpracování a analýzy obrazu. Zjistí, jak používat algoritmy pro detekci a rozpoznávání objektů na základě vizuálních dat. Naučí se, jak robot vnímá své prostředí a jak je definována pozice a orientace objektů v prostoru.

Teoretický rozbor úlohy

Strojové vidění je technologie, která umožňuje počítačům a systémům interpretovat, rozumět a reagovat na vizuální data z reálného světa podobně, jak to dělají lidé.

Tato technologie řeší problematiku analýzy a zpracování obrazů a videí za účelem identifikace objektů, sledování pohybu, detekce a rozpoznávání vzorů, měření a kategorizace objektů nebo dokonce interpretaci scén a událostí.

Strojové vidění kombinuje obory, jako jsou digitální zpracování obrazu, umělá inteligence (AI), strojové učení a počítačové vidění, aby bylo možné automatizovat úkoly, které vyžadují vizuální vnímání.

V našem případě se zaměříme na zpracování obrazu pomocí oblíbené knihovny **OpenCV**.

Základní princip detekce pomocí funkce **detect_objects**, která v sobě zahrnuje několik kroků. Funkce jako vstup obdrží oblast, ve které detekuje objekty, které po úspěšné detekci zvýrazní.

Převod na šedý odstín

V barevném obrázku (typicky ve formátu RGB) má každý pixel tři složky odpovídající barvám červená, zelená a modrá. Při převodu na šedý odstín se tyto tři složky kombinují tak, aby se vytvořila jedna hodnota intenzity. Tato hodnota obvykle odpovídá vnímané světlosti barvy.

V OpenCV je pro tento účel vytvořena metoda **cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)**, která vezme obrázek (roi) a převede jej z RGB do šedotónového formátu.

Thresholding

Thresholding je technika binarizace obrazu. V základní formě se jedná o proces převodu obrazu na černobílý formát, kde se vyberou pixely na základě prahové hodnoty jejich intenzity.

Nejprve je zvolena prahová hodnota (např. 60) poté se projde každý pixel obrázku(roi)

- Pokud je hodnota pixelu menší než prahová hodnota, pixel je nastaven na černou (hodnota 0)
- Pokud je hodnota pixelu větší než prahová hodnota, pixel je nastaven na bílou (hodnota 255)

V programu je použit **cv2.THRESH_BINARY_INV**, což je inverzní binární thresholding. To znamená, že pixely, které jsou menší než prahová hodnota, se stanou bílými, zatímco ty, které jsou rovny nebo větší budou černé.

```
gray = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)
_, thresh = cv2.threshold(gray, thresh: 60, maxval: 255, cv2.THRESH_BINARY_INV)
```

V kontextu se oba tyto kroky používají pro zjednodušení obrazu před jeho dalším zpracováním, které zahrnuje např. detekci kontur. Převod na šedý odstín pomáhá odstranit informace o barvě, které nejsou pro detekci tvarů potřeba. Thresholding potom vytvoří jasný kontrast mezi objekty a pozadím, což usnadňuje identifikaci a sledování objektů v obraze.

Hledání kontur

Pomocí metody `cv2.findContours` jsou nalezeny kontury v binárním obrázku – s `cv2.RETR_EXTERNAL` získává program pouze vnější kontury.

```
contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
```

Kód funkce `detect_objects`

```
24 def detect_objects(roi, roi_position, pixel_per_cm):
25     gray = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)
26     _, thresh = cv2.threshold(gray, thresh: 60, maxval: 255, cv2.THRESH_BINARY_INV)
27     contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
28
29     for contour in contours:
30         x, y, w, h = cv2.boundingRect(contour)
31         cv2.rectangle(roi, (x, y), (x + w, y + h), (0, 255, 0), 2)
32
33         # Výpočet středu obdélníku
34         center_x = x + w // 2
35         center_y = y + h // 2
36         cv2.circle(roi, center: (center_x, center_y), radius: 5, color: (0, 0, 255), -1)
37
38         # Výpočet souřadnic X a Y vzhledem k osám
39         coord_x_cm = center_x / pixel_per_cm
40         coord_y_cm = center_y / pixel_per_cm
41
42         # Zobrazování souřadnic
43         cv2.putText(roi, text: f"X: {coord_x_cm:.2f}cm, Y: {coord_y_cm:.2f}cm", org: (x + 10, y - 10),
44                     cv2.FONT_HERSHEY_SIMPLEX, fontScale: 0.5, color: (255, 0, 0), thickness: 2)
45
46     return roi
```

Důležité pro správnou detekci obrazu jsou také – nastavení kamery, její kalibrace a správné nasvícení pracovní plochy.

Díky správné kalibraci je program schopen rozpoznat i velikost objektů a definovat správně rozměry pracovní plochy, která se nachází v pracovním prostoru robota.

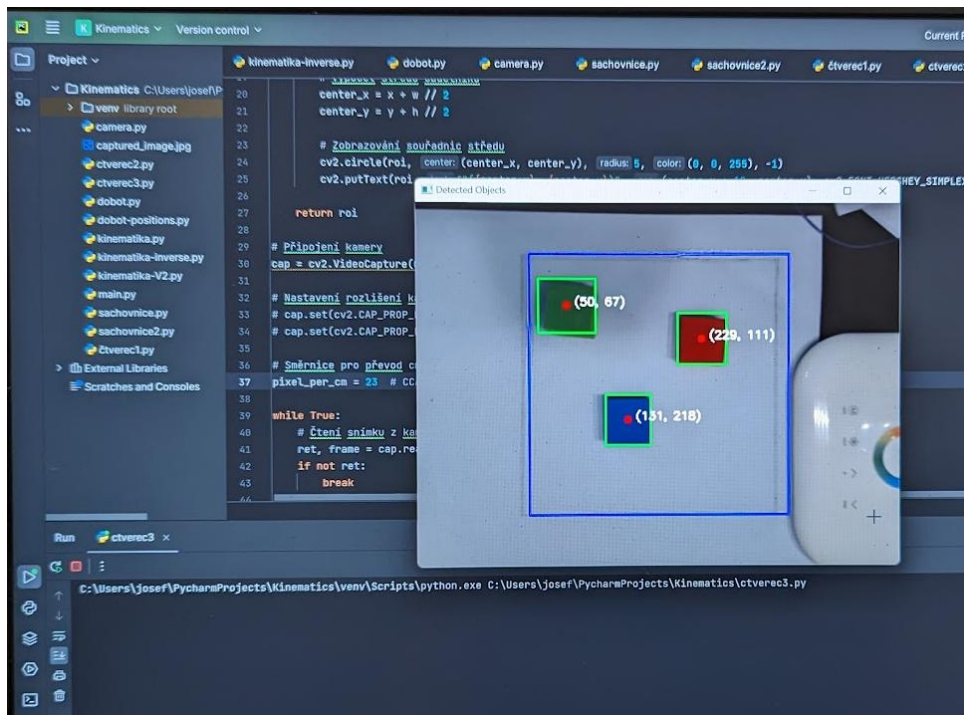
Následně už potom určíme výchozí (referenční) bod, kterým označíme jeden roh pracovní plochy a od tohoto bodu dochází k přepočtu souřadnic robota vůči pracovní ploše.

Tato úloha je jednou z nejnáročnějších, ale lze ji zadat formou projektu skupině žáků.

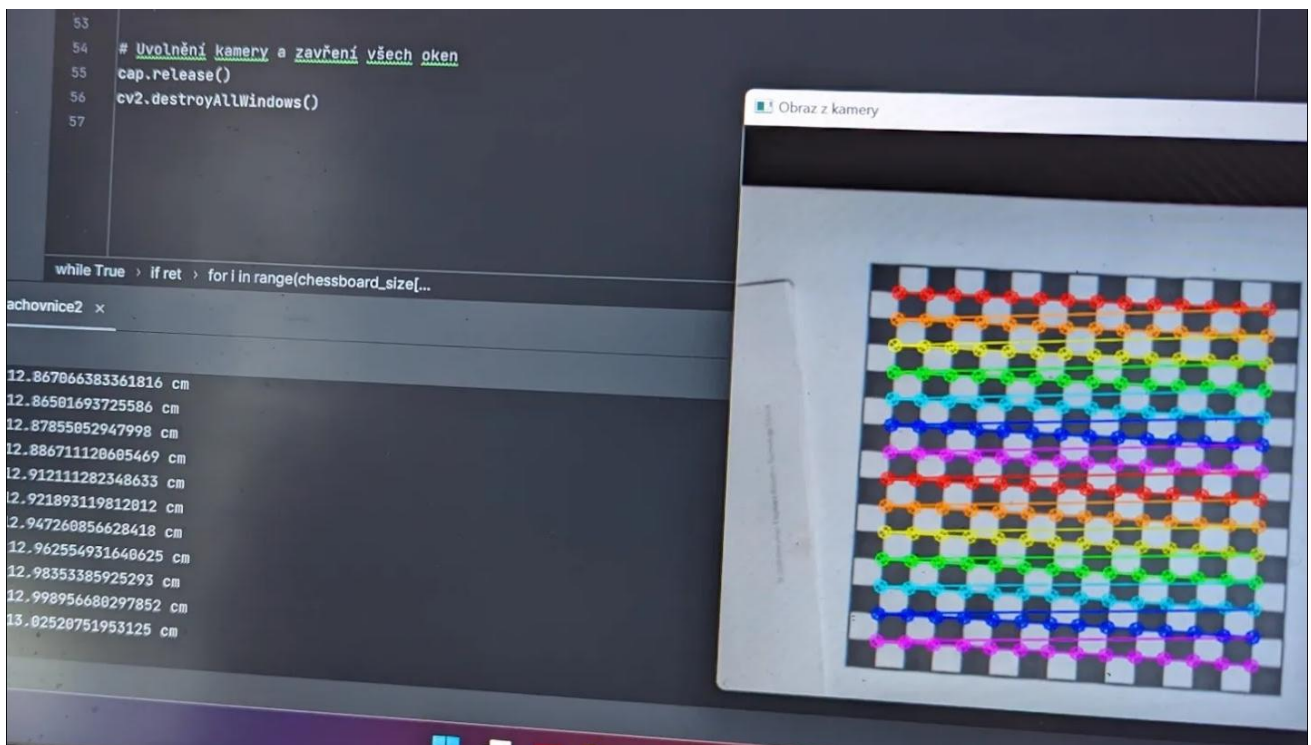
S knihovnou OpenCV je možné použít různé typy web kamer. V našem případě používáme kameru Logitech C920.

Žáci získají přehled o tom, jakým způsobem probíhá zpracování obrazu a následná detekce objektů nebo například textu.

Práce vyžaduje hlubší znalosti a základní dovednosti programování v jazyku Python, který je použit jak pro detekci obrazu, tak i pro naprogramování robota Dobot Magician a je tak nutná provázanost s dalšími odbornými předměty do jejichž osnov je strojové vidění implementováno.



Obrázek 8 - Detekce objektů a zvýraznění jejich středů pomocí knihovny OpenCV.



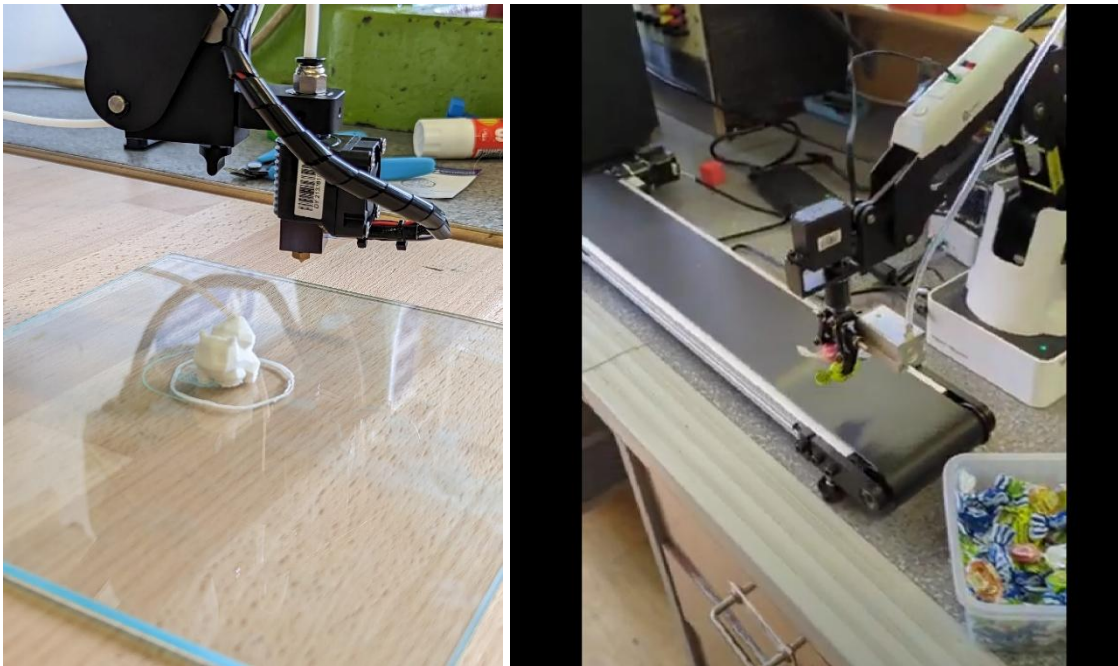
Obrázek 9 - Kalibrace kamery s využitím knihovny OpenCV.

Závěr

S roboty Dobot Magician jsem jako učitel získal neuvěřitelně silný nástroj pro výuku odborných předmětů.

Roboty přinesli do naší školy více možností, jak žákům přiblížit svět robotů. Zahrnují v sobě prvky mechaniky, elektroniky, výpočetní techniky, strojového vidění, AI a jsou tak ideální ukázkou synergické integrace, která je jednou ze základních definic mechatroniky.

Seznámení se se základními principy bylo celkem rychlé a intuitivní, protože software dodávaný k robotům je navržen tak, aby uživatel již z počátku dokázal robota používat k různým činnostem jakými jsou např. Laserové gravírování, 3D tisk nebo manipulace s objekty.



Obrázek 10 - 3D Tisk s robotem Dobot Magician a manipulace se sladkými objekty.

Dodávané vývojové prostředí umožňuje robota programovat, jak pomocí bloků, tak i prostřednictvím programovacího jazyka Python. V případě webového prostředí **dobotlab.cc** je zde možnost používat také knihovny třetích stran, jako je např. zmiňovaná knihovna **OpenCV**, nebo třeba **matplotlib**, která byla využita při tvorbě aplikace pro výuku kinematiky.

Problémy, na které jsem z počátku narazil byly způsobeny nedostatečným množstvím materiálů, které se programováním robotů Dobot Magician zabývají.

Dnes nemám, co bych robotům vytknul. Za celou dobu různých pokusů se nestalo, že by se robot porouchal. Je velmi dobře navržen vůči poškození v důsledku špatného naprogramování.

Vyzkoušeli jsme všechny možnosti, které jsou součástí zapůjčeného balíčku, což nás v důsledku vedlo k vytvoření učebny robotiky, kde bychom rádi z těchto robotů vytvořili robotizovanou linku, která by v sobě zahrnovala všechny prvky, které byly zmíněny ve výše uvedených úlohách (projektech).

Při návštěvě mezinárodního strojírenského veletrhu v Brně - MSV2023 jsem osobně navštívil stánek firmy Dobot a zajímalo mě, jak moc se odlišuje programování robotů Dobot Magician od průmyslových robotů, které firma Dobot nabízí. S překvapením jsem zjistil, že vývoj probíhá v blocích a programování je velmi podobné tomu, co nabízí software dodávaný právě k těmto výukovým robotům.

Všeobecně však platí, že je potřeba učit žáky základní principy, protože po jejich pochopení se budou schopni adaptovat na jakéhokoliv jiného robota.

Výuka se díky robotům stala zábavou. Žáci mají možnost pracovat na různých projektech, které prohlubují jejich znalosti.

Při práci na projektech získají dovednosti z těchto klíčových oblastí:

Základy robotiky

Porozumění mechanice a kinematice robotických ramen, včetně způsobů, jakými mohou pohybovat a manipulovat s objekty.

Naučí se, jak robot vnímá své prostředí a jak je definována pozice a orientace objektů v prostoru.

Programování

Získají zkušenosti s programovacími jazyky a platformami používanými pro řízení robotů, například Python s ROS (Robot Operating System) nebo vývojové prostředí Dobot.

Naučí se, jak psát algoritmy pro automatizaci úkolů, jako je detekce, uchopení a přemístění objektů.

Strojové vidění

Získají pochopení základních principů strojového vidění, včetně zpracování a analýzy obrazu.

Naučí se, jak používat algoritmy pro detekci a rozpoznávání objektů na základě vizuálních dat.

Interdisciplinární dovednosti

Naučí se řešit problémy. Rozvinou schopnosti analyzovat problém, navrhnout řešení a testovat různé přístupy k dosažení cíle.

Osvojí si týmovou práci a komunikaci. Práce v týmu na společném projektu podporuje rozvoj komunikačních dovedností a schopnosti spolupracovat.

Kritické myšlení a inovace

Proces návrhu a testování umožňuje studentům experimentovat s různými řešeními a iterativně zlepšovat vlastní návrhy.

Naučí se adaptovat a integrovat nové technologie do svých projektů, což je klíčová dovednost v rychle se vyvíjejícím technologickém světě.

Praktické aplikace teoretických konceptů

Použijí matematické a fyzikální principy v praxi, například při výpočtu trajektorií nebo sil potřebných k manipulaci s objekty.

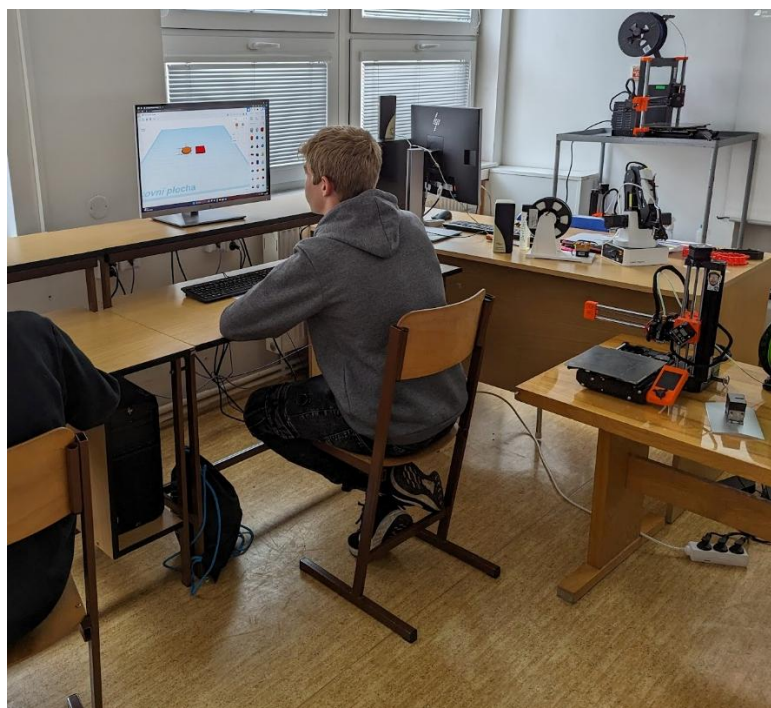
Celkově projekty využívající roboty přinesly žákům ucelený zážitek, který spojil teorii s praxí a ukázal jim, jak se různé disciplíny prolínají v reálných aplikacích.

Zahrnutí robotů do výuky se stalo vynikající příležitostí k rozvoji dovedností, které jsou důležité jak v akademickém prostředí, tak i v průmyslu.

Mimo výuku byly roboty použity také na prezentacích školy, jako jsou dny otevřených dveří, soutěž [Mikrotela](#), projektové dny a burzy škol.



Obrázek 11- Burza škol Česká Třebová



Obrázek 12 - Projektový den zaměřený na 3D Tisk



Obrázek 13 - Burza škol Pardubice



Obrázek 14 – Techné 2023 – Bühler Žamberk

V robotech Dobot Magician vidím obrovský potenciál pro školy, které chtějí žákům co nejvíce přiblížit svět robotiky.

Seznam obrázků

Obrázek 1 – Pracovní prostor robota Dobot Magician s uvedenými délkami ramen.	4
Obrázek 2 - Aplikace pro výuku přímé a inverzní kinematiky	7
Obrázek 3 - Rozhraní pro připojení periférií.....	9
Obrázek 4 - Zapojení pouzdra MAX522.....	10
Obrázek 5 - Schéma zapojení obvodu k mikropočítači.....	11
Obrázek 6 - Program zapisující do registru DAC(A) hodnoty pro výstupní napětí od 0 V do 5 V.	12
Obrázek 7- Schéma propojení senzoru s Arduinem	13
Obrázek 8 - Detekce objektů a zvýraznění jejich středů pomocí knihovny OpenCV.....	19
Obrázek 9 - Kalibrace kamery s využitím knihovny OpenCV.	19
Obrázek 10 - 3D Tisk s robotem Dobot Magician a manipulace se sladkými objekty.....	20
Obrázek 11 - Burza škol Česká Třebová.....	22
Obrázek 12 - Projektový den zaměřený na 3D Tisk	22
Obrázek 13 - Burza škol Pardubice	23
Obrázek 14 – Techné 2023 – Bühler Žamberk.....	23

Seznam tabulek

Tabulka 1 - Popis pinů obvodu MAX522.....	10
Tabulka 2 - Příklad pro vyslání slova CONTROL a DATA do MAX522 pomocí Arduina.....	11
Tabulka 3 - Druhy filtrů a jejich použití.....	13
Tabulka 4 - Nastavení frekvenčního limitu	13
Tabulka 5 - Výstup DAC podle nastavených hodnot barev.	15
Tabulka 6 - Označení pinů portu GP2.....	15

Zdroje

- [1] Shenzhen Yuejiang Technology Co., Dobot Magician V2 User Guide, Dokumentace 2023, [Search for the Right Manual | Dobot Download Center \(dobot-robots.com\)](#). Přístup dne 9. března 2023.
- [2] Maxim Integrated Products, MAX522, Datasheet 1995, [MAX522 Datasheet\(PDF\) - Maxim Integrated Products \(alldatasheet.com\)](#). Přístup dne 17. října 2023.
- [3] Eclipsera s.r.o., Senzor detekce barevného spektra, Datasheet 2021, [1420662593.pdf \(dratek.cz\)](#). Přístup dne 17. října 2023.
- [4] Luboš M., "Arduino detektor barvy TCS230", Dratek.cz, [Arduino detektor barvy TCS230 | Návod Drátek \(dratek.cz\)](#). Přístup 17. října 2023.
- [5] Zdeněk Kolíbal a kolektiv, *Roboty a robotizované výrobní technologie 1. vydání*, VITIUM Brno, 2016, ISBN:978-80-214-4828-5
- [6] Jiří Skařupa, *Průmyslové roboty a manipulátory*, VŠB Ostrava, 2007