

# Lekce 3

## Cíl lekce

Cílem dnešní lekce bude jeden komplexnější program. Tento program se vám bude hodit při soutěži, proto jeho vypracování dejte maximum a nezapomeňte si ho uložit.

## Úkol

Úkolem robota bude aby hledal číslice od 1 do 5 a to následujícím způsobem. Pokud robot žádnou číslici nevidí, otáčí se dokud nějakou číslici nenajde. Pokud najde 1. číslici, pokusí se k ní přiblížit na vzdálenost cca 20 cm. Poté hledání opakuje s následující číslicí. Stavte program tak, aby vyhledával číslice popořadě tedy 1, 2, 3, 4, 5. Trať bude postavena tak, aby po přiblížení se na požadovanou vzdálenost, byla následující číslice (po vhodném otočení) vidět. Nebude tedy potřeba náhodně jezdit. Dalším zjednodušením nechť je, že jízdní dráha bude pouze ortogonální (bude stačit, když robot bude jezdit po kolmicích, tedy čísla hledat s pootočením o 90°).

Řešení se samozřejmě v první řadě pokuste najít sami, ale následuje malá nápověda, kterou můžete využít. Nicméně nebojte se v průběhu programování ptát.

Pokud budete mít hotovo, tak program optimalizujte tak, aby robot připravenou dráhu projel co nejrychleji. Ale pozor robot se nesmí markerů dotknout ani je srazit.

## Co si z dnešní lekce odnést

- jak fungují v pythonu pole
- kdy se používá klíčové slovo `global`, [viz vysvětlení \(https://www.programiz.com/python-programming/global-keyword\)](https://www.programiz.com/python-programming/global-keyword)
- a samozřejmě jak autonomně hledat markery a jezdit od jednoho k druhému

## Co se může při programování hodit

### Neřešte zvlášť pohyb těla a hlavy

Nejjednodušší je zafixovat polohu těla vůči poloze hlavy. Kam se otočí hlava, tam se otočí také tělo:

```
robot_ctrl.set_mode(rm_define.robot_mode_chassis_follow)
```

### Obejdeme se bez callbacku pro rozpoznanou číslici

V minulých lekcích jsme používali callback:

```
def vision_recognized_marker_number_all(msg):  
    # co má robot udělat když vidí číslici
```

Ale nyní pro nás bude pohodlnější se pouze dotazovat, jestli robot vidí číslici (například po otočení). Dotaz můžeme udělat následujícím způsobem:

```
if vision_ctrl.check_condition(rm_define.cond_recognized_marker_number_one):  
    # tady robot číslici 1 vidí  
else:  
    # tady robot číslici 1 nevidí
```

### Cyklus nad číslicemi

Vzhledem k tomu, že se bude rozpoznávat 5 číslic, nebudeme chtít program napsat pro 1. a následně ho rozkopírovat pro dalších 5 číslic. Proto použijeme cyklus

```
for i in range(5):  
    # tady bude i postupně nabývat hodnot 0, 1, 2, 3, 4
```

### Detekce rozpoznané číslice

Vzhledem k omezení programového rozhraní robota nelze napsat něco jako:

```
rm_define.cond_recognized_marker_number(1)  
rm_define.cond_recognized_marker_number(2)  
...
```

Ale rozhraní nám nabízí pouze:

```
rm_define.cond_recognized_marker_number_one  
rm_define.cond_recognized_marker_number_two  
...
```

Proto použijeme malý trik. Nejprve si tyto konstanty dáme do pole a na příslušný prvek budeme v cyklu přistupovat:

```
markers_recognized = [  
    rm_define.cond_recognized_marker_number_one,  
    rm_define.cond_recognized_marker_number_two,  
    rm_define.cond_recognized_marker_number_three,  
    rm_define.cond_recognized_marker_number_four,  
    rm_define.cond_recognized_marker_number_five,  
]  
  
markers_numbers = [  
    rm_define.marker_number_one,  
    rm_define.marker_number_two,  
    rm_define.marker_number_three,  
    rm_define.marker_number_four,  
    rm_define.marker_number_five,  
]
```

V cyklu pak na příslušnou hodnotu přistupujeme pomocí indexu (číslováno od 0):

```
for i in range(5):  
    markers_recognized[i]
```

Tedy například

```
for i in range(5):  
    if vision_ctrl.check_condition(markers_recognized[i]):  
        # tady víme, že robot rozpoznal naši i-tou číslici
```

### Míření na marker

Může se vám stát, že robot bude trochu pootočený a pokud by jel pouze rovně na číslici se netrefí. V tomto případě se vám může hodit funkce `vision_ctrl.detect_marker_and_aim(rm_define.marker_number_one)`. Robot zajistí to, že na marker zamíří. Ovšem pozor, pouze pokud se mu to povede. Pokud se mu to nepovede, tak tato funkce způsobí to, že se robot bude pouze 5 sekund smutně otáčet. Takže doporučením je, tuto

funci volat až pokud jste markeru blíž (jak zjistit odhad vzdálenosti od markeru, víme z předchozí lekce). Druhým doporučením tedy je, až funkce míření skončí ještě jednou zkontrolovat, že robot opravdu vidí číslici a že se neotočil někam mimo.