

# Lekce 02

## Zdroje

- [Online příručka, Scratch a Python \(https://www.dji.com/cz/robomaster-s1/programming-guide\)](https://www.dji.com/cz/robomaster-s1/programming-guide)
- [Oficiální příklady v Pythonu \(https://github.com/ROBOMASTER-S1/ROBOMASTER-S1-Python-Examples\)](https://github.com/ROBOMASTER-S1/ROBOMASTER-S1-Python-Examples)
- [Seznam příkazů v Pythonu \(https://github.com/ROBOMASTER-S1/ROBOMASTER-S1-Python-Examples/blob/master/Robomaster%20S1%20Python%20Commands.py\)](https://github.com/ROBOMASTER-S1/ROBOMASTER-S1-Python-Examples/blob/master/Robomaster%20S1%20Python%20Commands.py)

## Úkoly

Cílem dnešní lekce bude samostatnější práce. Tedy samostatné hledání řešení a případně hledání v oficiální dokumentaci. Zase platí, že pokud bude s čímkoliv problém, neváhejte se zeptat.

### #1 Pohyb robota

Vyzkoušejte si pohyb robota skrze nabízené funkce:

```
chassis_ctrl.move_with_distance(0, 1) # argumenty úhel a vzdálenost v metrech
```

Tato funkce přijímá dva argumenty. Prvním argumentem je úhel, pod kterým robot pojede. 0 tedy znamená, že pojede pod úhlem 0°, tedy dopředu. Pokud budeme chtít jet dozadu, použijeme úhel 180° (jako parametr funkce použijeme bez znaku °). Druhým argumentem je vzdálenost, o kterou se má robot pohybovat. Tato vzdálenost je udána v metrech a může být menší než 1, tedy např. 0.1 (10 cm) a podobně (v programování říkáme, že argumentem funkce může být tzv. *float* - což je desetinné číslo).

Můžeme také nastavit, jak rychle se má robot pohybovat:

```
chassis_ctrl.set_trans_speed(0.5) # rychlost v m/s
```

Touto funkcí nastavíme rychlost pohybu robota a to v metrech za sekundu. Toto číslo může být v rozmezí 0 až 3.5. (Pro zajímavost, 3.5 m/s je 12.6 km/h).

#### Rozpohybujte robota tak, aby jezdil "do čtverce"

Vášim úkolem je teď napsat sadu příkazů tak, aby robot jezdil do čtverce o hraně například 1 m. Na konci by měl zůstat na místě, kde začínal.

#### Bonus

Jezděte s robotem místo do čtverce, dokola - robot tedy bude opisovat kružnici.

### #2 Měření vzdálenosti od markeru

Pro apoximaci (odhad) vzdálenosti použijeme informace, která nám robot vrací při rozpoznávání markeru. Pro připomenutí, z minulé lekce - příklad výstupu a popis:

```
[1, 11, 0.5, 0.5666..., 0.123..., 0.277...]
```

Vypsal se nám objekt `detection_info`, viz <https://www.dji.com/cz/robomaster-s1/programming-guide> -> Smart -> 14. Identified marker info. Jednotlivé prvky v poli značí [počet detekovaných markerů, ID detekovaného markeru, souřadnice X, Y, šířka markeru, výška]. Pro zjištění o jakou číslici se jedná budeme potřebovat to 2. číslo. V příkladu výše je tam hodnota 11. Ale pozor, jedná se o tzv. ID markeru, nejedná se o jeho hodnotu (co je na obrázku).

Nás bude pro odhad vzdálenosti zajímat poslední hodnota z vráceného pole - výška. (Bonus: víte proč výška a ne šířka?)

Pro odhad vzdálenosti od markeru tedy použijeme převrácenou hodnotu jeho výšky - čím menší rozpoznaná výška, tím delší vzdálenost.

#### Krok 1

Nastavte detektor markerů, aby rozpoznával markery do vzdálenosti 3 metrů:

```
vision_ctrl.set_marker_detection_distance(3)
```

#### Krok 2

Vypište si, jaké hodnoty se vám vrátí jako výška rozpoznaného markeru v závislosti na jeho vzdálenosti. Tyto hodnoty si poznamenejte.

#### Tip

Pokud rozpoznávání markeru nefunguje spolehlivě na delší vzdálenosti, může být na vině špatné osvětlení - zkuste třeba na marker posvítit mobilem.

#### Bonus

Vypiště odhad vzdálenosti markeru v cm. Pro odhad můžete použít přiřazení do intervalu vzdálenosti (několik podmínek, kterými naměřená hodnota propadne a vrátí se odhad vzdálenosti), dle naměřených hodnot a nebo použít vhodnou aproximaci. Aproximace je funkce (matematický předpis), kterou "naučíte" ze zadaných hodnot. Můžete využít např. online nástroj [Wolfram Alpha \(https://www.wolframalpha.com/input/?i=linear+approximation+%7B%28100%2C+0.125%29%2C+%2880%2C+0.150%29%2C+%2870%2C+0.172%29%2C+%2850%2C+0.233%29%2C](https://www.wolframalpha.com/input/?i=linear+approximation+%7B%28100%2C+0.125%29%2C+%2880%2C+0.150%29%2C+%2870%2C+0.172%29%2C+%2850%2C+0.233%29%2C)

### **#3 Držení vzdálenosti mezi markerem a robotem**

Nyní býde vaším úkolem spojit funkcionalitu z předhozích úkolů. Vyberte si konstantní vzdálenost (např. 40 cm) a úkolem bude, aby se robot držel na tuto zvolenou vzdálenost od rozpoznaného markeru. Pokud tedy vzdálenost mezi markerem a robotem bude větší, bude se muset robot markeru přiblížit a pokud bude větší, bude se muset vzdálit.

### **#4 Co nejrychlejší přiblížení se rozpoznanému markeru**

Můžete vyjít z předhozího úkolu, ale nyní bude cíl trochu jiný. A to, přiblížit se co nejrychleji rozpoznanému markeru. Ale pozor, do rozpoznaného markeru nesmíte narazit! Přiblížte se na vzdálenost cca 30 cm. (tip na řešení - budou se vám hodit hodnoty úkolu 2, kroku 2)