

Walking Robot Workshop

step gait kinematics motion

NTK

50°6'14.083"N, 14°23'26.365"E

Národní technická knihovna

National Library of Technology

0x 48 65 6c 6c 6f 20 77 6f 72 6c 64 21 0a

What is written in the heading?

What is written in the heading?

... today, we are about to create a »hello world« alà robotics.

Today's Workshop

Workshop Goals

- Make robots walk with own program.
- Describe the walk mathematically.
- Get advanced insight into robotics.
- See, that STEM is worth learning.
... A to vše za pouhé 4 hodiny.

Rules

- Investigate, invent, try out!
- Cooperate as you want!
- No abrupt movements with the robots.
- Work in Plocha/roboty/<today>.

Martin Zoula (✉) | Filip Kučera (✉)

comrob.fel.cvut.cz



FACULTY
OF ELECTRICAL
ENGINEERING
CTU IN PRAGUE



AI CENTER
FEE CTU

Walking Robot Workshop



https://gitlab.fel.cvut.cz/crl/public/courses/kracejici_roboly_ntk

Our Robots – Today's »Robota«

Hexapod ~ ant



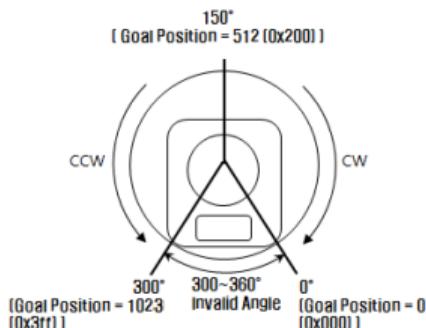
ROBOTIS PREMIUM King Spider

- 6 legs, 3 joints per leg.
- 24 degrees of freedom altogether.
- Statically stable walk is possible.
- Different gaits - 3pod, 4ped, 5pod...

Humanoid ~ biped ~ člověk



- 6 joints per leg, 3 joints per arm.
- 18 *controllable* degrees of freedom.
- Walk not stable!

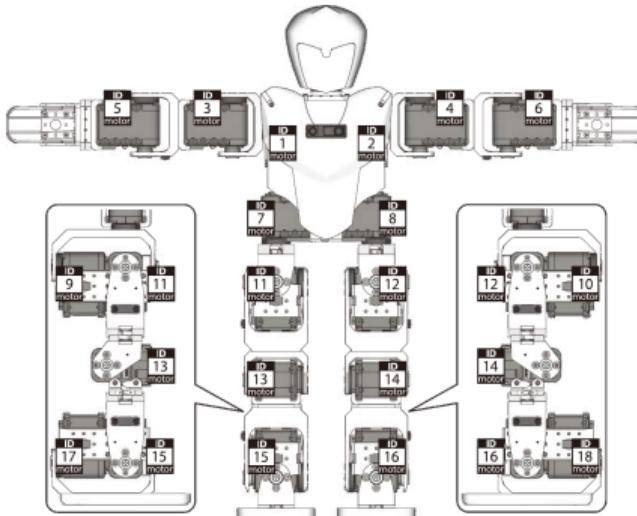
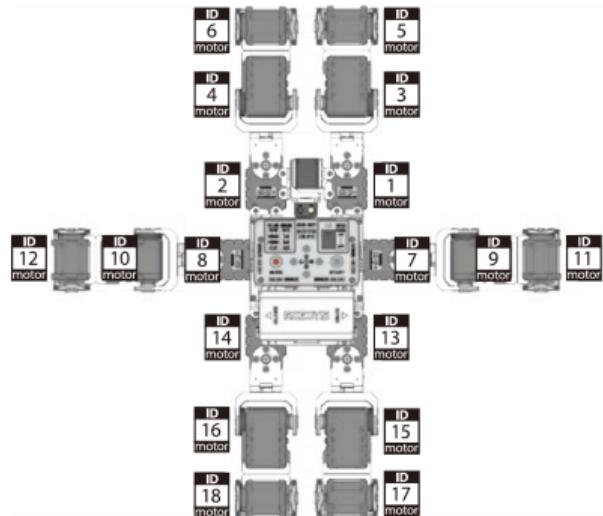


Robotis Dynamixel AX-12W

- DC gear electromotor with position control.
(Beware! Shock may damage the gearbox!)
- Position feedback by potentiometer.
(Dead angle! Rotor can not stop upside down.)
- Commanded from control unit by digital protocol.
(Each servo was assigned unique ID number!)
- We can read or set different servo properties.
(Speed, maximum torque, angle, temperature, stall...)
- 🔗 Reference manual here.

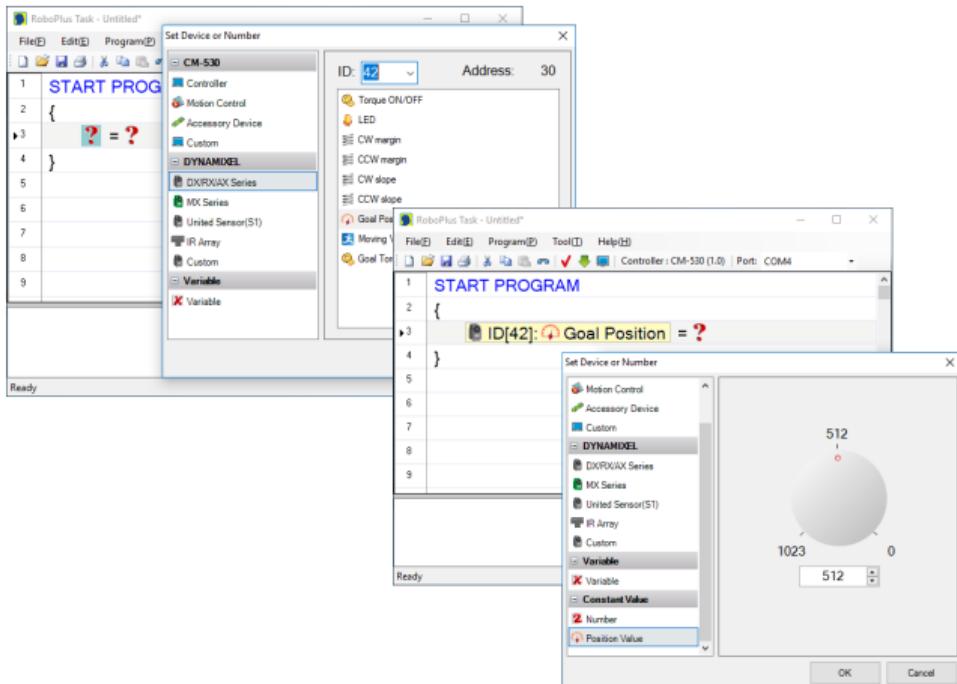
Task: Get to Know Your Robot

- Move the robot by hand.
- See how the robot moves.
- »RoboPlus Manager« displays angles.
- Draw a robot schematics.
- Mark angle reckoning orientation.
- Where is fixed the stator/rotor?



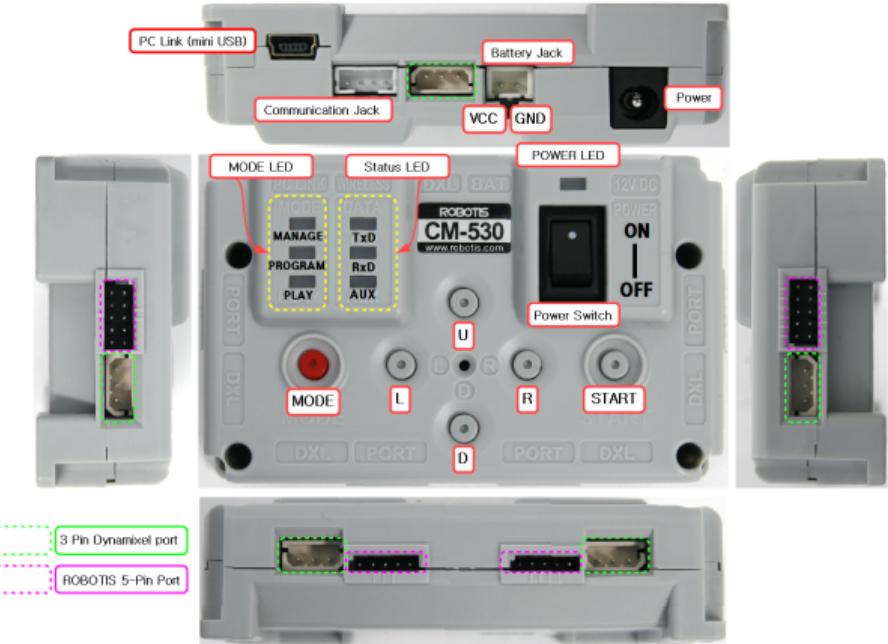
Task: First Steps

1. Launch »RoboPlus« → »Task«.
2. New file, choose controller type.
3. Spacebar to insert empty line.
4. Double-click to insert code.
5. Set some/all servos to 0°.
6. Upload over correct COM port.



Only one app can be connected to the robot at a time.

Control Unit



STM32F103RE ARM MCU, 72 MHz, 512 kB

- Stores and executes programs controlling the servos.
- Programmed in PC graphical interface, flashed over USB.
- Online servo inspection.
- Red button to choose »play« mode, then press »start«.
- Kill program with red button.
- Power switch in emergency.

[Hexapod] Task: Waving the Leg

!!! Preset movement speed may be high!

1. Set all servos movement speed to 50.
2. Set all servos angle to zero.
(Hence define robot's initial pose.)
3. Make a leg to move there & back again.
4. Use WAIT WHILE with Is Moving.

!! Servo is a »worker«, just follows orders.

If the »boss« (CM-530) makes two set-angle commands without waiting, only the latter effectively happens.

[CM-530]-ukol-kyvani-hexa-2

```
1: START PROGRAM
2: {
3: // Zapnulí výkonu serv
4: [ID[All]: ☀ Torque ON/OFF] = TRUE
5: // Nastavení rychlosti serv.
6: [ID[All]: ✎ Moving Velocity] = 50
7: // Nastavení základní pozice robotu.
8: [ID[All]: ⏪ Goal Position] = 512
9: Timer = 2.048sec
10: WAIT WHILE ([Timer] > 0.000sec )
11: // Kývání pravým středním femurem
12: LOOP FOR (i = 0 ~ 6 )
13: {
14: [ID[9]: ⏪ Goal Position] = 860
15: WAIT WHILE ([ID[9]: 🔴 Is Moving] == TRUE )
16: [ID[9]: ⏪ Goal Position] = 512
17: WAIT WHILE ([ID[9]: 🔴 Is Moving] == TRUE )
18: }
19: // Povolení serv - zabráníme přehřívání
20: [ID[All]: ☀ Torque ON/OFF] = FALSE
21: }
```

[Humanoid] Úkol: Kývání nohou

- Humanoidova noha je složitá. Prosté vynulování serv by vzpríčilo díly robotu.

1. Vyvolezte »pohyb č. 25«, nechť toto je naše počáteční pozice.

- Pohyb má definované i rychlosti.
Poslední rychlosť zůstane nastavena.

2. Nastavte všem servům rychlosť 50.

3. Hýbejte servomotorem sem a tam.

4. Kontrola dokončení pohybu viz sl. 11.

- Serva se mohou při zátěži přehrádat.

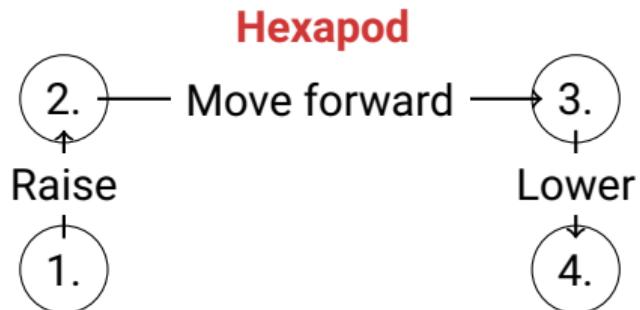
5. Vypínejte mezi programy »torque«.

12 / ?? Úkázka zdrojového kódu vytvořena v RoboPlus Task. (<https://robotis.co.uk/software/roboplus-1-0.html>)

[CM-530]-ukol-kyvani-humanoid

```
1: START PROGRAM
2: {
3: // Inicializace - počáteční poloha humanoida
4: [Motion Index Number] = 25
5: WAIT WHILE ([Motion Status] == TRUE )
6: // Nastavení rychlosti pohybu
7: [ID[All]; Moving Velocity] = 50
8: // Kývání kyčlí
9: LOOP FOR (i = 0 ~ 5 )
10: {
11: [ID[8]; Goal Position] = 700
12: WAIT WHILE ([ID[8]; Is Moving] == TRUE )
13: [ID[8]; Goal Position] = 512
14: WAIT WHILE ([ID[8]; Is Moving] == TRUE )
15: }
16: // Povolení motorů
17: [ID[1]; Torque ON/OFF] = FALSE
18: }
```

Task: Program One Leg Step

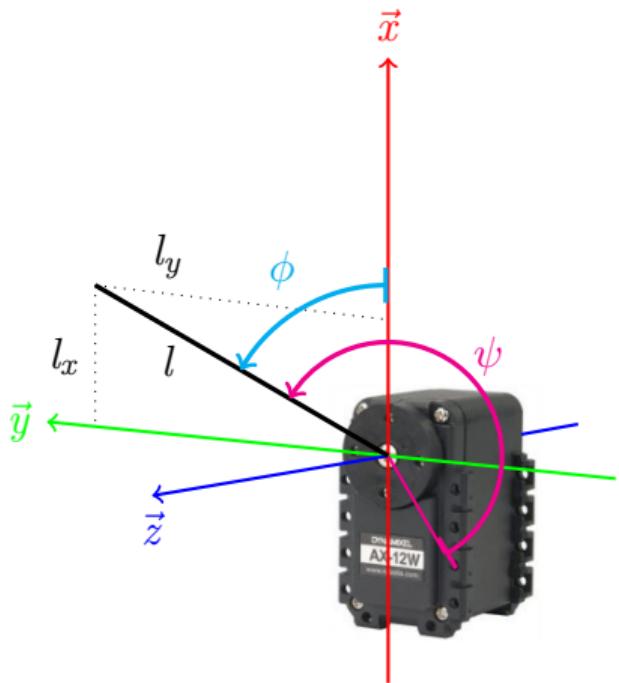


Humanoid

- Mind the mass! Always catch the robot!
- »Motion Index 25« lowers the torso.
- Move only the ankle and hip.

- You need to identify which particular angles to set!
- Divide your work. (Each identify one leg...) Mind the servo orientation!
- There are different step trajectories. (Triangle, square, ellipse, adaptive, ...)
- Do not forget to wait before each intermediate move finishes.

Servo Described by Math



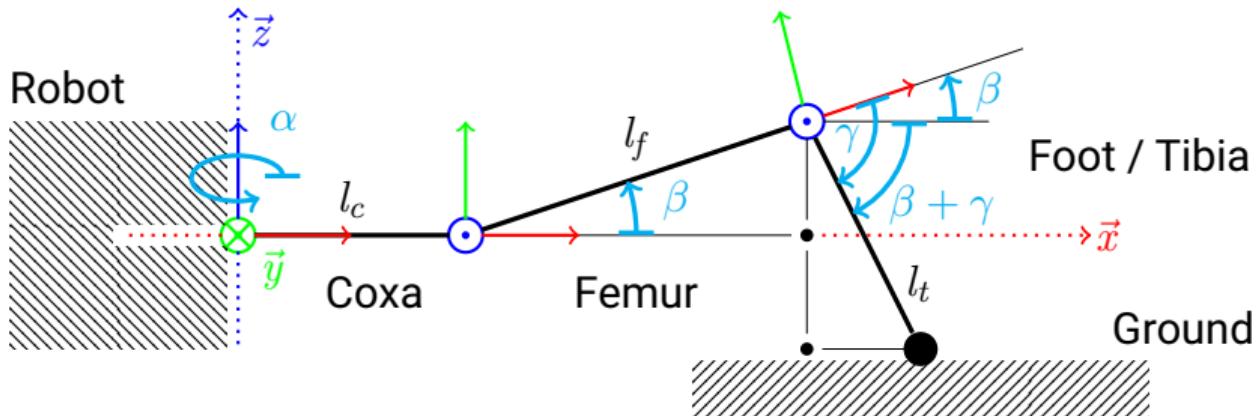
- Angle ψ : Defined by manufacturer.
 - (We set/read this when programming.)
- Angle ϕ : Kinematically meaningful.
 - Positive angle along z -axis.
 - Zero-angle aligned with x -axis.
 - (Right-hand rule.)

$$l_x = l \cos(\phi)$$

$$l_y = l \sin(\phi)$$

$$\phi = \psi - 150^\circ$$

Leg, a.k.a. kinematic chain



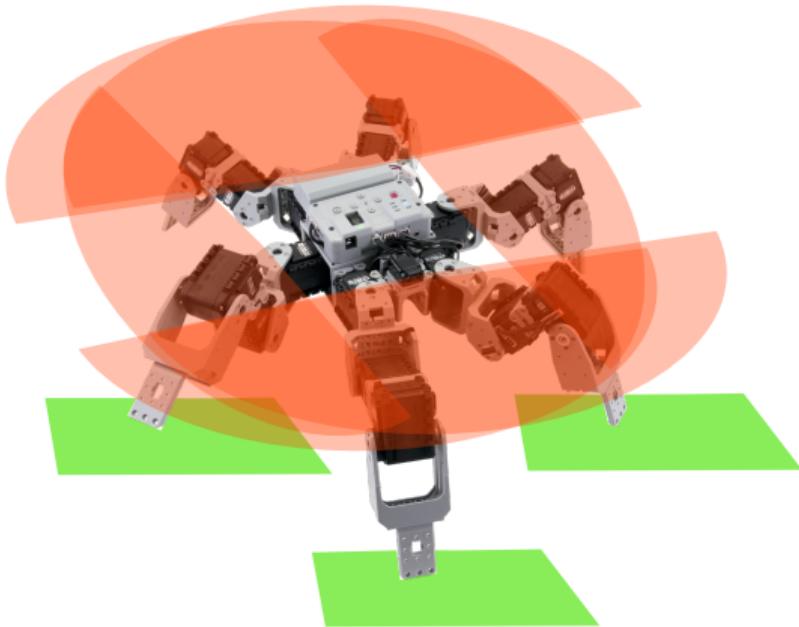
Direct kinematic task: Servomotor angles → foot coordinates.

$$x_{chod.}(\alpha, \beta, \gamma) = \cos(\alpha) [l_c + l_f \cos(\beta) + l_t \cos(\beta + \gamma)]$$

$$y_{chod.}(\alpha, \beta, \gamma) = \sin(\alpha) [l_c + l_f \cos(\beta) + l_t \cos(\beta + \gamma)]$$

$$z_{chod.}(\alpha, \beta, \gamma) = l_f \sin(\beta) + l_t \sin(\beta + \gamma)$$

Motion Limitations



- Robot has many degrees of freedom.
Can it use them fully?
- Joint angles are limited by design.
- Legs operation spaces intersect.
- Walk needs to take this into account!
- Also, environment might get in the way.
(Terrain, obstacles, people, machines...)

1. Wrap a code for a single step into »function«.
2. Make such function for each leg. Double-check the IDs.
3. Servos may be oriented differently at each leg. Check it.
4. [Hexapod] Make a function to move all Coxas backwards, moving the robot forth.
5. Call the functions in meaningful order.

BEWARE!

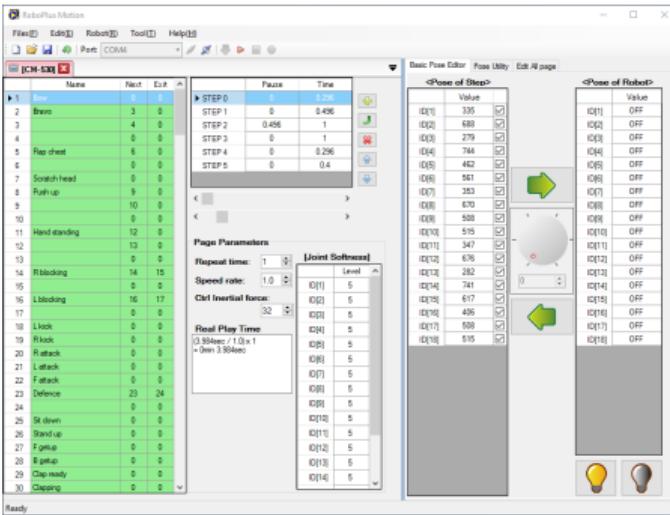
- Pay attention to leg collisions.
- Adjust the walk parameters to avoid collisions.

Preprogrammed Motions

- Robot has several preprogrammed motions always ready.
- Use »RoboPlus Motion« to play them back.
- You can launch the motions also programmatically.

[CM-530]-stub-motion

```
1: START PROGRAM
2: {
3: // Začne vykonávat přednastavený cvik.
4: Motion Index Number = 42
5: // Program je třeba zastavit dokud není cvik dokončen.
6: WAIT WHILE (Motion Status == TRUE )
7: }
```



Robots are furnished with distance sensors and microphone to detect clapping.

[CM-530]-stub-distance

```
1: START PROGRAM
2: {
3:     // Vyčtení hodnoty do proměnné
4:     přečteno = PORT[1]:IR Sensor
5:     // Použití hodnoty přímo v programu
6:     IF ( PORT[1]:IR Sensor <= 100 )
7:     {
8:         // ...
9:     }
10:
11:    // Senzor tlesknutí - čekám na jeden tlesk.
12:    Result of Sound Counter = 0
13:    WAIT WHILE ( Result of Sound Counter < 1 )
14:    // Zahraji melodi.
15:    Buzzer Time = Play Melody
16:    Buzzer index = Melody2
17:    WAIT WHILE ( Buzzer Time > 0 )
18: }
```



Final Task: Make the Robots Dance

1. Robot waits until there is a clap.
2. Then, it executes some motion.
3. Then, waits for another clap.
4. Then, different motion is executed.
5. ... or anything you imagine.
6. Use sensors to avoid obstacles.
7. Accelerate the hexapod by »tripod gait«.