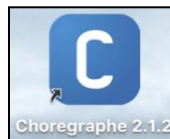




Choregraphe Next Steps: Walking



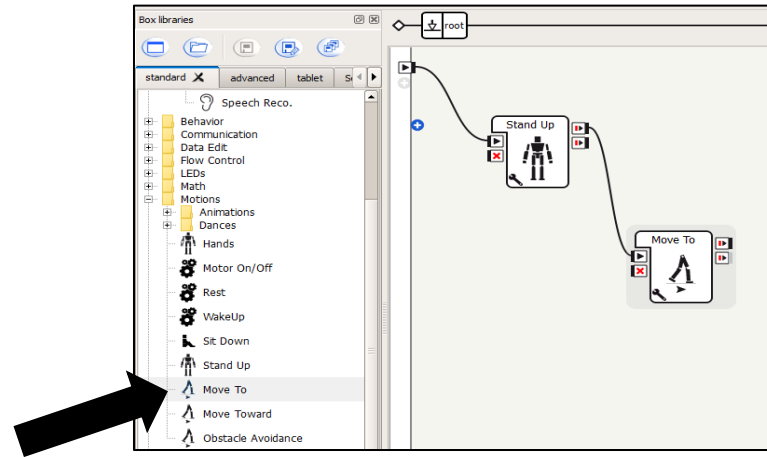
Goals for this session:

- Understand the purpose and use of the "Move To" and "Counter" boxes.
- Program the robot to walk in a square.
- *This lesson assumes you know the basics of Choregraphe (box libraries, root directory, etc.).

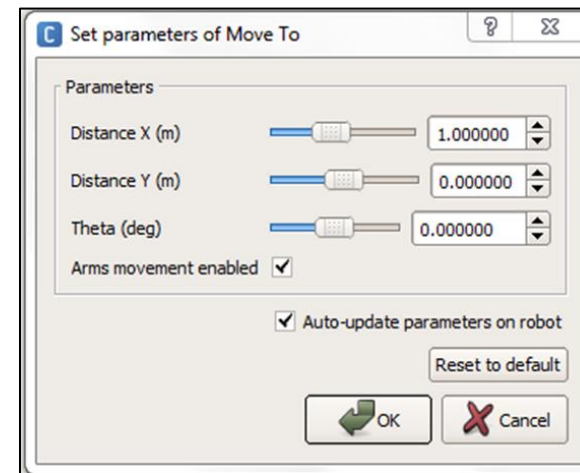
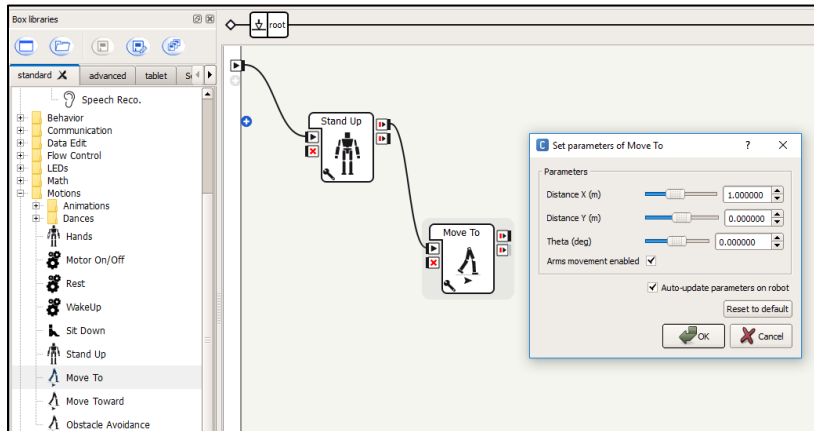
Moving the robot's legs is the trickiest programming because of the high risk of the robot over-balancing and falling down. The robot can never be put in an off-balance position or it will fall over. Because of the balance issues and the complicated programming involved, the robot comes pre-programmed with a couple of walking boxes and some other leg movements.

Part 1:

1. First, have the robot stand up by dragging the "Stand Up" box into the workspace. It's good practice to have a program start by having the robot stand up, otherwise it will start executing the command from whatever position it happens to be in. Next, use the "Move To" box (the "Move Toward" box is similar except the robot will not stop walking until it receives another command). Drag the "Move To" box (found under motions and dances) to the work space and connect with a noodle.



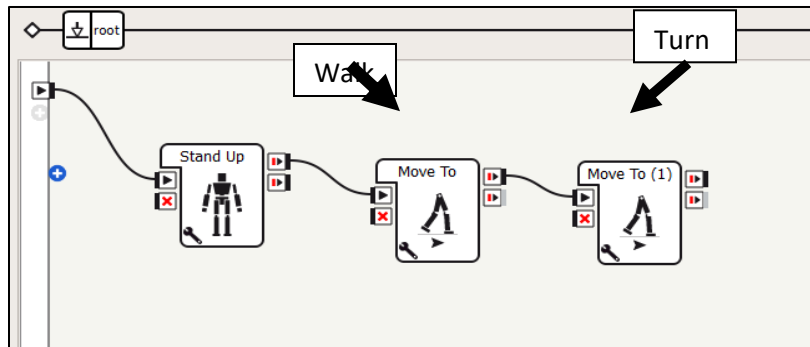
2. From the Root menu, click on the wrench in the corner of the “Move To” box. Set the parameters of the movement.



- The **X** value moves **forward**.
- The **Y** value moves **sideways**.
- If you want it to move **diagonally**, set both values.
 - **Tip:** A value of 1.000000 = one meter. So you are setting distances in meters or fractions of a meter. One meter equals approximately 26 robot steps. 0.3 meters is approximately 9 steps.

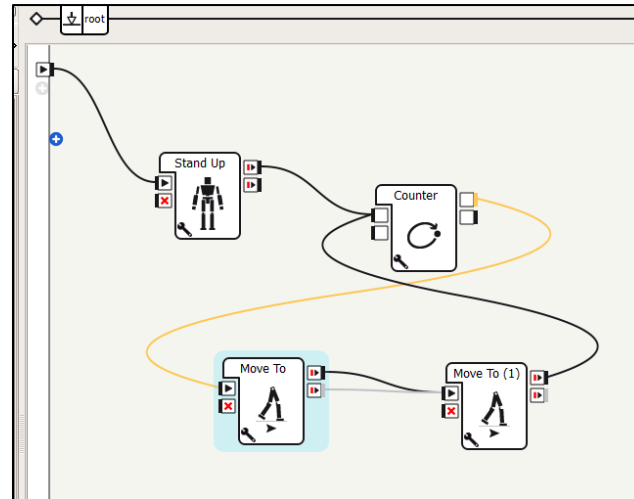
- **The robot only moves while facing forward.** With the Y value set, the robot will shuffle sideways while facing forward. With the X and the Y value set, the robot will slide along on an angle, while always facing forward.
- Set the **Theta** value to make the robot **turn**. Here the Theta numbers stand for degrees out of 360°. For example, setting Theta at 180° would make the robot turn around and face backwards.
 - **Tip:** a **negative** value makes the robot turn to the **right**; a **positive** value makes the robot turn to the **left**. For example: setting Theta at -90° will make the robot turn 90° to the right, but setting Theta at 90° will make the robot turn 90° to the left.
- If the X and/or Y value and the Theta value are set, the robot will execute these commands simultaneously by moving in a line and rotating the set number of degrees. It produces a kind of twirling in slow motion, not the most natural of movements.
- Using two boxes, one to walk and one to turn, the robot will do one after the other, like a soldier.
 - **Tip:** when using two boxes to make the robot perform these motions separately, **remember to change the X value to zero** in the box you're using to set the Theta value. X always defaults to 1.

Walk in a square:

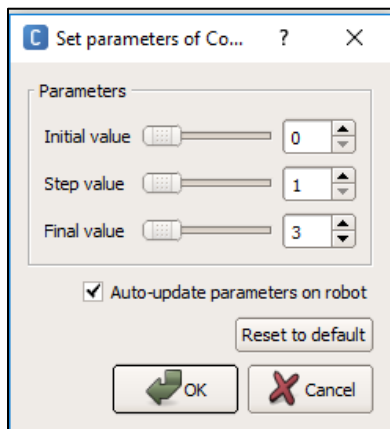


1. The best way to program the robot to walk in a square is to separate the walking movement and turning movement, by setting one "Move To" box to walk and one "Move To" box to make the turn.
2. The "Move To" box is set to walk .25 on the X axis and 0 on the Y axis. The "Move To (1)" box is set to turn 90° on the Theta and 0 on both X and Y axes.

3. Rather than repeating the walking and turning movements by programming eight more boxes, use the “Counter” box. It’s faster and easier. Under “Flow Control” drag the “Counter” box onto the workspace. Insert the “Counter” box between “Stand Up” and “Move To”.



4. Click on the wrench on the “Counter” box to set the counter.



Initial value = How many times the motion has already occurred. Set to 0.

Step value = How many times the robot should perform the action before repeating the action. Set to 1.

Final value = How many times the robot should repeat the action before stopping. Set to 3.

These parameters tell the robot that it has not performed the action yet, but after performing the action once, repeat it three more times for a total of four to make a square.

5. Complete the program with another “Stand Up” box to return the robot to a neutral standing position.

