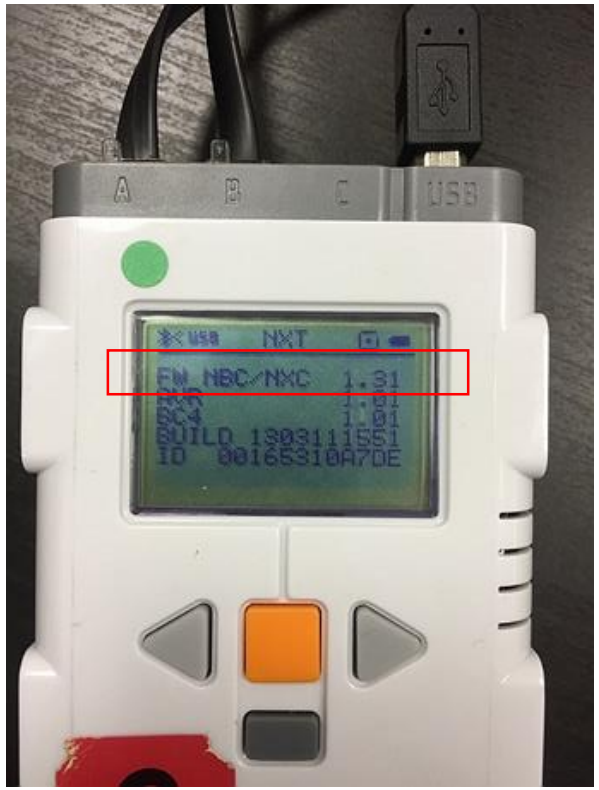
# ME729 Advanced Robotics - Lab #2

2/5/2018

Sangsin Park, Ph.D.

- Before starting, confirm the NXT firmware, FW NBC/NXC 1.31.
  - Settings → NXT Version




If not →

- You can download the NXT firmware file from class web page.

**Week 01 - 01/22/18**

Topic: Introduction and Pre-Requisites

**Handouts**

- Course Outline me729RoboticsCourseOutline011818.pdf
- Weekly Schedule **Last updated 01/20/18** me729RoboticsSchedule011818.pdf
- Lecture:
  - 
  - 
- Lab: (NXC Programming Refresher)
  - labBricxxIntroToProgramming.pdf
- **Homework:** (due next week)
  - 

**Things to do**

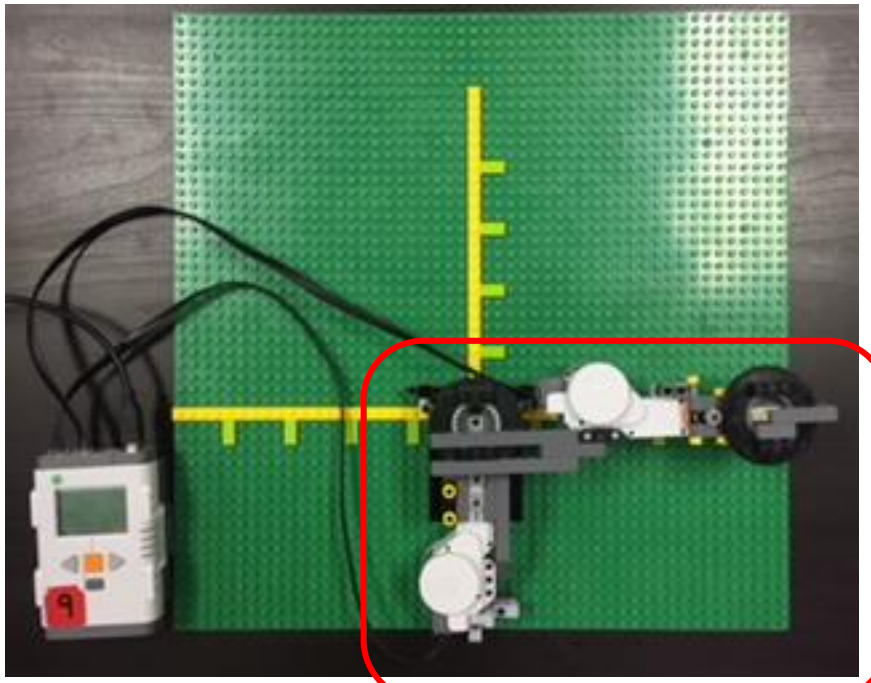
- Bring $50 (check preferred) made out "Paul Oh UNLV", post-dated for "05/14/18". Check canceled if all parts returned.
- Meet each Monday 18:00-21:00 at 1325 E. Flamingo Rd
- Bring NXT Kit (at least Brick and both USB and NXT cables) to class every week
- Check NXT Kit for part completion: Lego Education Kit 9797 Parts List PDF Complete on-line XLS sheet for NXT kit inventory
- Configure NXT Brick and Windows PC (preferable Win7) Bricxcc. Test "Hello World" NXC program

**Resources**

- BricxCC Compiler Version 3.3.8.9 bricxcc_setup_3389.exe | SourceForge Site
- (If needed): NXC Firmware Version 1.31 lms_arm_nbcnxc_131.rfw | SourceForge Site
- (May need): USB Driver for NXT Brick NXT_USB_Driver_120.zip
- Primer Notes: labBricxxIntroToProgramming.pdf
- Sample Code: nxcPrimerExampleCode.zip
- (Reference) NxC Programming Guide: PDF | BricxCC Web Site URL
- (Reference) NxC Tutorial: PDF | BricxCC Web Site URL
- (Reference) NxC API: PDF | BricxCC Web Site URL
- Download and install Scilab (open-source Matlab/Simulink) scilab-5.5.2_x64.exe (Vers 5.5.2 64-bit Windows) | Other versions including Vers 6.0.0

- You can update the NXT firmware through Bricx Command Center.
  - Tools → Download Firmware

- We set up the manipulator as figure.
  - Align links with a horizontal axis.
  - That is, $\theta_1 = 0, and\ \theta_2 = 0$.



Home position of the manipulator

- Distance between each stud : 8 mm
- So, the initial position is located at (200 mm, 0 mm).
  - Link 1 : 21 studs * 8 mm = 168 mm
  - Link 2 : 4 studs * 8 mm = 32 mm

- Download ‘Fwd_Kine_blank.nxc’ from class web page.
- A breakdown of the code.

```
1  // ------------------------------------------------------ //
2  // Forward Kinematics for 2R planar manipulator
3  // Sangsin Park, Ph. D.
4  // Feb. 5. 2018
5  // ------------------------------------------------------ //
6
7  // Motor's constants
8  #define JNT1 OUT_A
9  #define JNT2 OUT_B
10 #define FULL_SPEED (100)
11
12 // Gear's constants
13 #define turnTableTeeth (56)
14 #define spurTeeth (8)   // NB: +Angle yields CCW rotation (with eye on top of turntable)
15 #define gearRatio turnTableTeeth/spurTeeth  // 7.0
```

Define constants.

- A breakdown of the code.

```
17 task main()
18 {
19      // button variables
20      bool orangeBtnPushed = FALSE;
21      bool l_ArrowBtnPushed = FALSE;
22      bool r_ArrowBtnPushed = FALSE;
23      bool greyBtnPushed = FALSE;
24
25      int cnt_OrangeBtn = 0;
26      int cnt_l_ArrowBtn = 0;
27      int cnt_r_ArrowBtn = 0;
28
29      // reference joint angles
30      long theta1 = 0;
31      long theta2 = 0;
32
33      PlayTone(TONE_B3, 50);
34      TextOut(0, LCD_LINE1, "Grey BTN Quits");
35      TextOut(0, LCD_LINE2, "Orange BTN Home");
36
37      PosRegEnable(JNT1); // Set Port A current angle as zero [deg]
38      PosRegSetMax(JNT1, 0.4*FULL_SPEED, 0); // Set Port A speed limit (40) and default acceleration (0)
39
40      PosRegEnable(JNT2); // Set Port B current angle as zero [deg]
41      PosRegSetMax(JNT2, 0.4*FULL_SPEED, 0); // Set Port B speed limit (40) and default acceleration (0)
```

Define flags.

Define local variables.

Display info.

Initialize two motors.

- A breakdown of the code.
  - Set infinite loop with a condition.
  - This code is not terminated until a grey button is pushed.

```
43  while(greyBtnPushed == FALSE)
44  {
45      greyBtnPushed = ButtonPressed(BTNEXIT, FALSE);
46      orangeBtnPushed = ButtonPressed(BTNCENTER, FALSE);
47      cnt_l_ArrowBtn = ButtonCount(BTNLEFT, FALSE);
48      cnt_r_ArrowBtn = ButtonCount(BTNRIGHT, FALSE);
49
50      // Code here for updating angles //
51      // 1. Whenever left arrow button is pushed, let the link 1 rotate every 5 degrees.
52      // 2. Whenever right arrow button is pushed, let the link 2 rotate every 5 degrees.
53
54
55      // ---------------------------- //
56
57      PosRegSetAngle(JNT1, theta1);
58      PosRegSetAngle(JNT2, theta2);
59
60      TextOut(0, LCD_LINE4, FormatNum("JNT 1: %d", theta1/gearRatio));
61      TextOut(0, LCD_LINE5, FormatNum("JNT 2: %d", theta2/gearRatio));
62
63      // Code here for going to home postion after oragne button is pushed //
64
65
66      // ---------------------------- //
67  }
```

Check if grey and orange buttons are pushed.

Count how many time are left and right arrow buttons pushed.

**Here is your work.**

Set updated angles. This functions can move motors.

Display link rotation angles

**Here is your work.**