
FLL Programming For Coaches Documentation

Release 2.0

Winston

September 17, 2012

CONTENTS

1	Sources	3
1.1	References	3
2	Notes	5
2.1	Tony Ayad FLL Programming Beginner and Advanced	5
2.2	Coverage from My Blocks Tutorial	6
2.3	Tasking with AND, OR, etc	6
2.4	Putting it all together	7
2.5	More Tips and Tricks (and things I didn't have time to cover!)	7
3	Indices and tables	9

Contents:

Table of Contents

- FLL Programming For Coaches - Notes
 - Sources
 - * Sample Programs
 - * References
 - Notes
 - * Tony Ayad FLL Programming Beginner and Advanced
 - * Coverage from My Blocks Tutorial
 - * Tasking with AND, OR, etc
 - * Putting it all together
 - * More Tips and Tricks (and things I didn't have time to cover!)
 - Indices and tables

SOURCES

Note:

- I'm using other coaches fine presentation materials for the most part.
 - Therefore, most notes simply refer to the slide/presentation I'm working from.
 - So I'll start by listing their fine resources.
-

1.1 Sample Programs

Here are the programs I used during the training:

[Download Programming Examples](#)

Note: I *don't* provide the My Block examples here. However, the versions I went through in class are *ready* to be easily turned into My Blocks - try it!

1.2 References

FLL Basic and Advanced NXT-G Programming for Coaches - By Tony Ayad

- Posted on the *Los Angeles Region FLL Team Resources Page*
- Hereafter referred to as [Ayad](#)

This is a newer presentation and one of the best I've found. I strongly suggest that you review this presentation. Anything you don't already know? Stop and read/study carefully. This presentation (the basics part) was used on one of the Coaches conference calls just within the last week or two so go find a playback of that call on the FLL site.

2011 Cougar Advanced NXT-G Programming Workshop v2.1 - By Cougar Robotics

- Hereafter referred to as [Cougar Presentation](#)

Another excellent and recent presentation. Look on the web page for a couple of good samples that go with it. Covers some topics not covered well elsewhere. For example, good coverage of a simple "logical AND" alternative using a 1 count loop for synchronization.

NXT Tutorials by Dale Yocum - By Dale Yocum

- Hereafter referred to as [Dale Yocum Tutorial](#)

An oldie, but a goodie - primarily for beginners. Notable primarily because of the very nice video format and patient pace of the presenter. Doesn't cover all the latest techniques.

Creating and Using your own Blocks with My Blocks - By Dave Parker (nxtprograms.com)

- Hereafter referred to as [My Blocks Tutorial](#)

Very thorough overview and tips on creating My Blocks! Some good tips on wiring here as well.

Team Hassenplug NXT-G Code Snippets Index - By "Team Hassenplug"

- Hereafter referred to as [Hassenplug Tips](#)

Excellent well documented tips! Also, some really good information on data logging here. Hassenplug and friends are serious adult Lego enthusiasts so some of what is on their site is not appropriate for FLL. For example, they use specialized blocks created in Labview in some cases that wouldn't be legal in FLL competitions.

FLL Programming 101 with NXT-G - HighTechKids.org

- Hereafter referred to as [NXT-G 101](#)

Another oldie, but goodie. However, it is a bit dated and almost all of it is covered in one of the above. However I may refer to a slide or two and you may want to review...

Specialized Tools and Techniques

- [nxtRICEditV2](#) - Tool to create custom graphics for the NXT Screen!
- [Make a custom My Block icon](#) - Describes how to create/use custom graphics for use in My Block icons.

NOTES

Note: Important Note: At the moment these notes are for an **advanced** class so the below notes suggest skipping slides and resources that I assume an attendee at an advanced class should know. If you are not ready for an advanced class then **don't skip anything!**

2.1 Tony Ayad FLL Programming Beginner and Advanced

Coverage from [Ayad](#)

1. Slides 1 through 9, SKIP
2. Slide 10, Mention Profiles Mention Custom My Block Folders
 - Create a folder called “Peter” at the appropriate spot
3. Slide 12, Mention difference between Motor and Move blocks
4. Slide 16 and 17, Talk about the PID and accumulated error
 - Introduce concept of **reset motor** block and use of **Rotation Sensor** block to “start from 0”.
5. Slide 23, brief pause to talk process.
 - Program one distinct action at a time, test, make reusable (if team is knowledgeable about My Blocks, don't force it)
6. Light Sensors - **GO to Black Line** program
 - (a) Slide 25 through 37. Will have to skim some, but need to cover...
 - Calibration
 - View Sensor Program (download from [Dale Yocum Tutorial](#))
 - Worth pausing to discuss screen output and number to text
 - **Example 1A:** Simple go to black with Loop till sensor
 - Point out problem (how do we tell it the light value?)
 - Cover Light Sensor Block (not covered till slide 47 unfortunately)
 - Note that it takes input (target light value)
 - **Example 1B:** Introduces Light Sensor Block in a Loop
 - **Example 1C:** Adds a constant to feed in the target light value

- At this point we *could* make a my block
But let's think first of other things we might want to control
 - Option to stop at end or keep going?
 - Option to provide a power level?
 - Do we have to get past a *dark* area first before we start looking?
- **Example 1D:** Has these additions

Note: Note how we've already set up constants to provide these values!

2.2 Coverage from My Blocks Tutorial

1. My Block Time!

[My Blocks Tutorial](#) covers this very nicely!

- **Example 1_D-MB:** Demonstrate creating a my block from 1D. Can view finished example.
- Input naming
- Save location
- Try Using 1_D-MB in a program

2. My Block Output!

- **Example 1_E:** Same at 1_D, but it is printing out the distance travelled.
 - Demonstrate making a my block out of 1_E
- **Example 1_E-MB:** Shows the completed My Block. **Example 1_F-MB** shows adding sound effects, just for fun!
- **Example 1_F_USE:** Demonstrates use of the program.
 - Show a program that goes over two lines with just this block.

3. **Example 3:** Follow line till Ultrasonic. Nothing new really,...

2.3 Tasking with AND, OR, etc

In programming a logical AND means we two things need to be true for the “whole” thing to be true.

A logical OR means one or the other thing needs to be true for the “whole thing” to be true.

Here we are talking about a similar concept, but with tasks.

1. AND

Means I want my program to wait until two distinct things are done before continuing with the rest of the program.

- We've seen one already with the register to wall
- Other options?

Can be done with complex conditions, but a neat trick is to use a 1 count loop!

2. OR

A good example can be found in the [NXT-G 101](#) presentation on **slide 81**.

- AND can also be done in a similar way, but the task synchronization trick is usually simpler.

2.4 Putting it all together

A quick look at a complex program that uses multiple my blocks.

2.5 More Tips and Tricks (and things I didn't have time to cover!)

1. The [Hassenplug Tips](#) page is an **excellent** resource, review it!

2. Wiring into a switch block

- Generally wiring is a pain! The [My Blocks Tutorial](#) covers that, review it and practice!

All the more reason to use **My Blocks**!

3. Debugging Tips (there are many)

- Sound
- Stop for 1 second
- Video tape robot and play back in slow motion!
- Data Logging (more in a moment)
- Print stuff to screen

4. Tricky to move things between computers

- Try the “pack and go” feature in 2.0

5. Organize blocks by making custom myblock folders

6. Really crazy stuff!

- Make your own “.ric” files which can be displayed on the NXT screen.

Our team did that “way back when” to display a large number on the NXT screen for what program we were on.

- Make “.png” files which can then be put in a special folder and used for custom My Block pictures.

INDICES AND TABLES

- *genindex*
- *search*