

ENGR 3390 Final Project – Path B

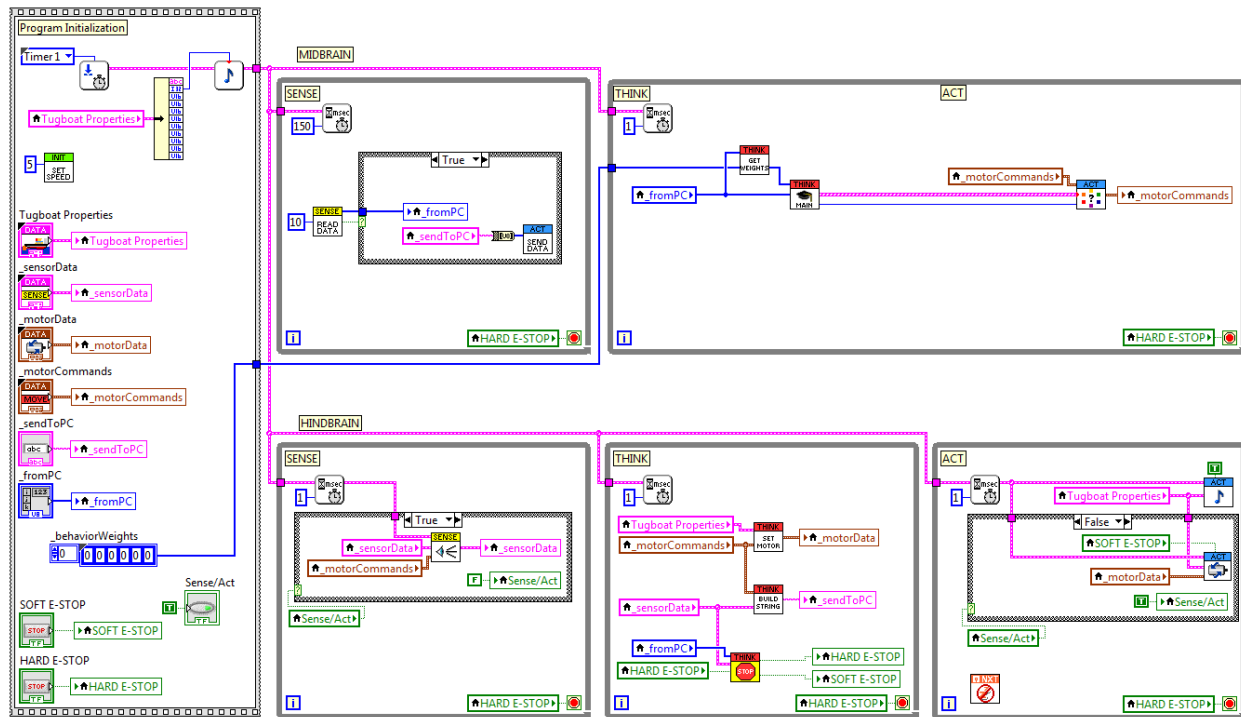


Table of Contents

ENGR 3390 Final Project – Path B	1
List of Figures	2
Executive Summary.....	3
Start-up Guide.....	4
NXT Side	4
PC Side.....	5
The Code	5
Initialization.....	5
Midbrain.....	5
Hindbrain	5
Random Thoughts.....	9
Coding on the NXT	9
This document	9

List of Figures

Figure 1: Program initialization structure. There is heavy use of typedef's and local variables to move information around the brains.	6
Figure 2: The NXT Midbrain.	7
Figure 3: The NXT Hindbrain	8

Executive Summary

This code was developed by the NINJA's based on their work last year on the final project. It contains a complete hindbrain and a complete midbrain structure. You will need to calibrate the 'tugboat properties' to your specific boat to ensure best performance, but the defaults will work reasonably well on any boat.

The hindbrain should not be modified without good reason. The NINJA's will be reluctant to help you debug hindbrain issues, because we have tested it for many hours and it is functioning well. That being said, if you present us with an improvement, we are open to hearing it and helping you implement it.

The midbrain is complete except for behaviors. We have provided one behavior for you to use as a template when you create other behaviors. There are five open slots- if you decide you need more than five behaviors the NINJA's can help you expand the code. The arbiter is functional as-is, and while we recommend you do not change it, other arbiters can be developed if the team wishes.

Some work will be needed to integrate this code with the Forebrain. The Forebrain communicates with Path A's NXT code in a specific way that is not shared with Path B's NXT code. A translator block will need to be developed for the Forebrain to translate Forebrain Path A commands into NXT Path B commands.

Start-up Guide

Before you begin, make sure you have installed the LabVIEW for LEGO Mindstorms module. This is provided in the course folder. Copy the .zip file out of the Path B subfolder and unzip it locally. (Or to the location that you would like to share code with your team.) The NINJA's recommend uploading your project to a Google Code repository and using Tortoise SVN for version control. Ask us if you are unsure of how to do that.

NXT Side

- 1) Open up the hatches to the boat. Connect the two shore power lines (Blue and Red connectors) and plug the USB connector into your laptop.
- 2) Power on the NXT. It should be recognized by your laptop.
- 3) Open the file 'Tugboat.lvrbt.' This is the NXT project file for the code.
- 4) Under 'Choose NXT,' select the NXT that you've connected to your computer.
- 5) Open 'nxt main.vi.' If it asks you to find files, direct it to look through the folder up a level: MindSensors-LVLM-Toolkit-v2p04\MindSensors Palette
- 6) If the NXT were smart about arrays, you could download the code to the NXT immediately and start running your boat. However, this is not the case as the NXT doesn't have the indexing-out-of-bounds protection that stock LabVIEW does. There are a few places where you will need to initialize arrays by hand every time you open the code from scratch. (It's lame the first few times but you get used to it.) Those places are:
 - a. `nxt main.vi`
 - i. `_sensorData: initialize _NXTButtonStates`
 - ii. `_fromPC`
 - b. `NXT_MIDBRAIN Get Behavior Weights.vi`
 - i. `_fromPC`
 - ii. `_behaviorWeights`
 - c. `NXT_MIDBRAIN Master Brain.vi`
 - i. `_fromPC`
 - ii. `_behaviorWeights`
 - iii. Switch to the block diagram and open `NXT_MIDBRAIN_BEHAVIOR Forebrain Control.vi`
 1. `_fromPC`
 2. `_behaviorWeights`
 - d. `NXT_MIDBRAIN Arbiter.vi`
 - i. Behavior Data
 - e. `NXT_HINDBRAIN Get Sensor Data.vi`
 - i. `_sensorData: initialize _NXTButtonStates` (Only have to do the control, not the indicator)
 - f. `NXT_HINDBRAIN Build Message String.vi`
 - i. `_sensorData: initialize _NXTButtonStates`
 - g. `NXT_HINDBRAIN ESTOP.vi`
 - i. `_sensorData: initialize _NXTButtonStates`
 - ii. `_fromPC`

- 7) Download the code to the NXT and run it. The boat should start beeping! If it says “File Error: -1” and quits, you missed one of the initialization locations.

PC Side

- 1) Until you write the translator block, this code doesn’t work with the Forebrain or the Base Station. A VI is provided for manual control of the boat so that the midbrain team can test behaviors while the forebrain gets up and running. Open ‘PC Manual Control.vi.’
- 2) If it complains about dependencies, those VI’s are stored in the same folder as PC Manual Control.vi. It should be able to find them.
- 3) Plug in the USB cable for the xBee radio and determine what COM port it is. Select that one from the drop down menu on the front panel.
- 4) Make sure the NXT code is running on the boat.
- 5) Run the vi. If it can’t see the boat, your laptop will beep at you. If your laptop isn’t beeping, congrats! (Or turn up the speakers...)
- 6) Turn off the ‘soft stop’ button and set Behavior Weight #0 to 1. The tugboat will then respond to the commands you give it on the front panel controls.

The Code

Initialization

The first case structure in the code is essentially our variable declarations- we’re setting up everything we’re going to need throughout the code. As you can see there isn’t a lot of room in here, so we’ve used typedefs to package groups of variables that relate to the same things. (These are the blocks with the little black triangle in the upper left corner.) Unless you’re completely overhauling (NOT RECCOMENDED) a section of the code, you shouldn’t have to go in here much at all.

Midbrain

The top two loops in the code make up the Midbrain. The first loop is the SENSE(act) loop. We call it that because the outgoing communication ended up inside that loop instead of in the act loop. This was so that the NXT didn’t have competing subVI’s in separate loops going after the use of the xBee radio. The sense loop sends the string to the PC whenever it gets a string from the PC- this way the radio is only ever outgoing or incoming and the NXT doesn’t have to decide who gets priority.

The other loop is the THINK/ACT loop. For speed reasons, these have been combined into a single loop. Originally, the link between the behavior engine and the arbiter was through local variables so that they could be run in two different loops. However, the act of writing all of the behavior data to a variable and then turning around and reading it all slowed the execution of the code down tremendously, so the two loops were merged and the read/write time was saved.

Hindbrain

The Hindbrain sense loops grabs sensor data from the available sensors on the NXT. It packages these up into the `_sensorData` typedef and writes it to the variable so other subVI’s can access the data. The Think loop does three things: it manages the E-STOP conditions, it builds the string that gets returned to the PC, and it translates the `_motorCommands` control (values 0-14) to `_motorData`. (values set by the front panel/tuning your boat. The ACT loop beeps the ‘boat armed’ tone and sets the motors to the commanded values. It also has a keep alive block so the NXT doesn’t turn off.

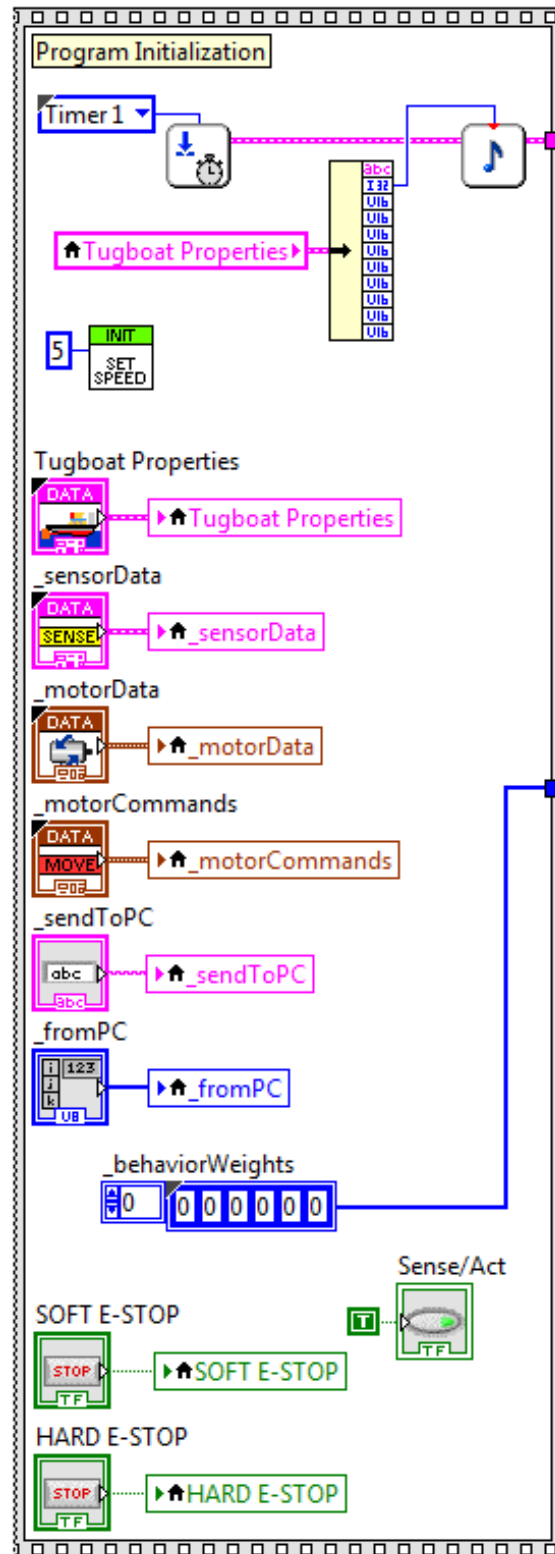


Figure 1: Program initialization structure. There is heavy use of typedef's and local variables to move information around the brains.

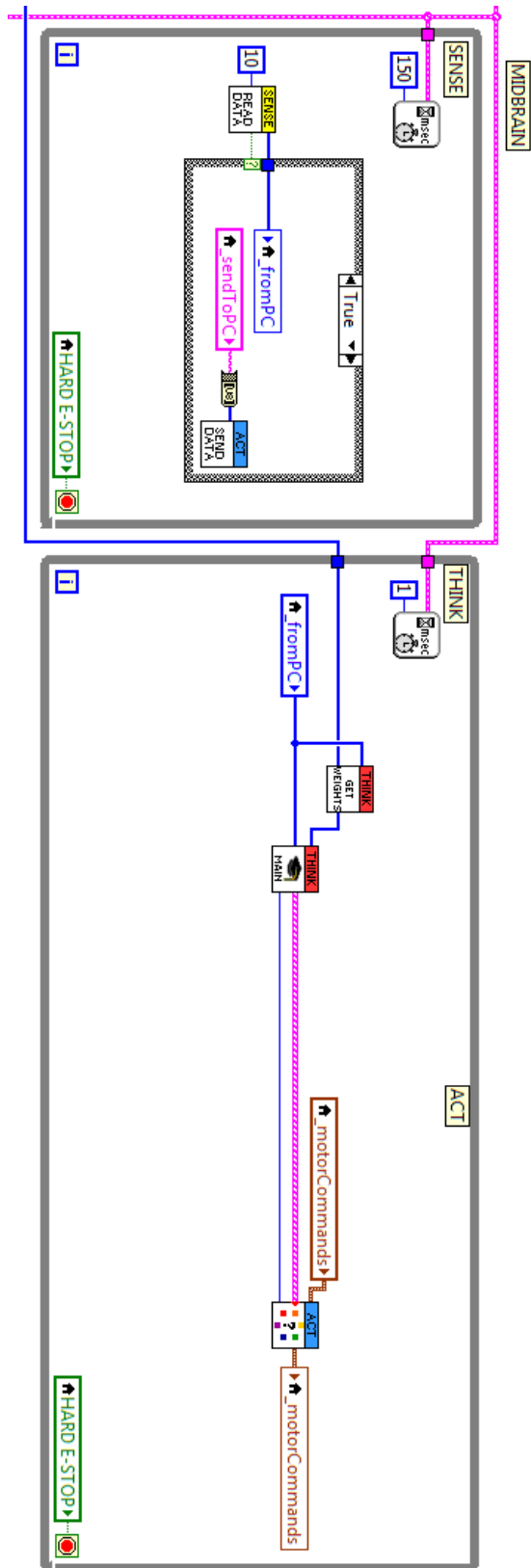


Figure 2: The NXT Midbrain.

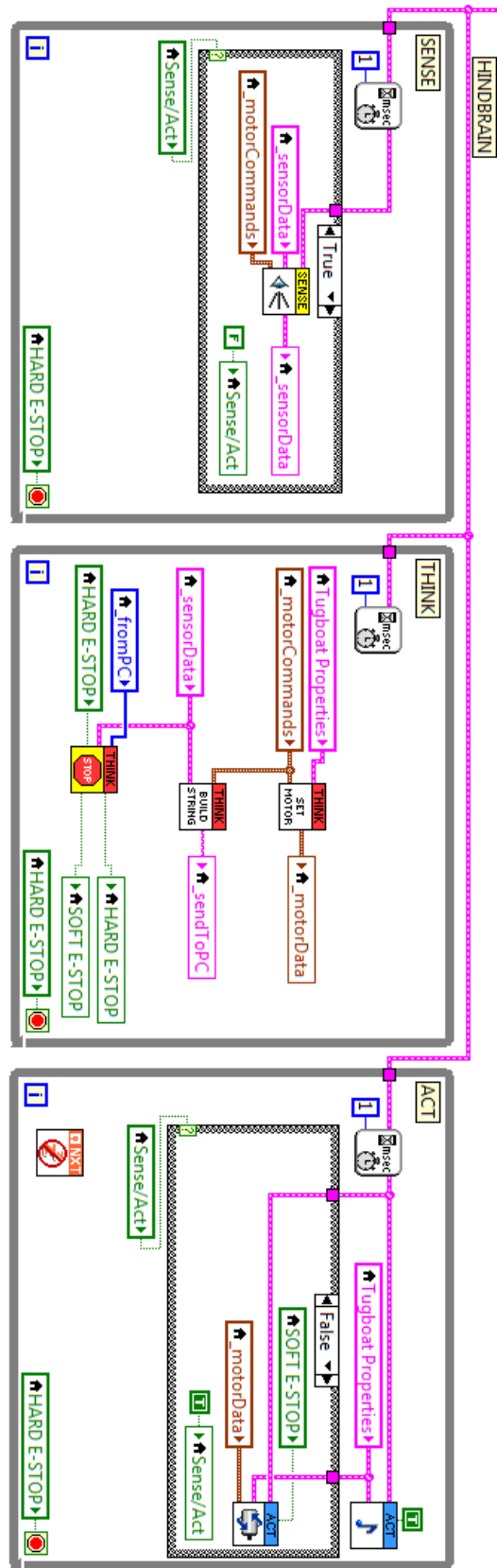


Figure 3: The NXT Hindbrain

Random Thoughts

Coding on the NXT

This applies to either branch of NXT code- the NXT is a monster. It looks innocent- it's not. There are lots of things you can do to make it upset and it won't be able to tell you what you did. Work slowly and carefully, debug in stages, and consult with Kevin (He's been fighting with this for years now and has experience.) if things go horribly wrong.

The NINJA's will hold a special session sometime soon covering best practices for NXT code, so stay tuned for that.

This document

As you can probably tell, this all came together very last-minute. This document will be updated with more complete instructions as time goes on, but honestly, you will probably explore the lab faster than we can keep up. A road map for the code will be posted on the wall and we'll tell you when this gets updated, but other than that you're getting thrown into the deep end with this. Have fun!