Assignment 3

ME 2984

October 21, 2015

Assignments will be turned in as a short (∼5 minute) video showing you have successfully completed the required steps. The video must be submitted via dropbox (please keep files size down - no need for 1080P). You must also submit on Scholar. **Assignments not submitted on Scholar will not be accepted**.

1 Setup Your BeagleBone

1. Install Ubuntu onto the BeagleBone Black (BB), using the instructions in BBB_initialize.pdf.
2. Configure a working internet connection for the BBB, using the instructions in BBB_internet.pdf.
3. Install ROS on your BeagleBone
   (a) Follow the instructions here[1]. The step referencing enabled the "multiverse", "universe", and "restricted" repositories is done as part of the Ubuntu installation, and can be safely skipped. The instructions are complete once python-rosinstall has been installed.
   (b) Initialize a catkin workspace using the same instructions as were used in Assignment 1.
4. Install Git, complete with SSH encryption keys[2].
   (a) Clone the khan_robot repo into the catkin workspace
   (b) Clone the private workspace into the catkin workspace
5. Install the Adafruit BeagleBone GPIO library for Python, using the instructions in BBB_bbio.pdf.

2 Assemble the interface cape

Follow the BBB_cape_wiring.pdf instructions to solder the motor controllers and logic converter together into a single package on your cape. Connect the cape to your BeagleBone. Assemble the wiring harness and add in the voltage splitting circuitry to the IR sensor.

3 Send Python commands to interact with the motors and sensors

These instructions include how to write Python commands which interface with the onboard hardware. At this point, simply demonstrating that the commands can be executed without error is sufficient. The instructions can be found on FleetRobotics/adafruit-beaglebone-io-python repo[3] in the readme. Demonstrate the following commands:

1. Setting a GPIO pin to both a high and low state.
2. Configuring a PWM output and setting the duty cycle.
3. Reading an actual voltage from an ADC. Note this is the raw, integral value, but the actual measurement in volts.

[1] http://wiki.ros.org/indigo/Installation/UbuntuARM

[2] https://help.github.com/articles/generating-ssh-keys/

[3] https://github.com/FleetRobotics/adafruit-beaglebone-io-python

4 Hardware Integration

1. Drive a motor using a motor controller

   (a) Use jumper wires to connect a PWM pin to the PWMA pin on your board and connect two GPIO pins to A_{in1} and A_{in2}.

   (b) Connect the GND pin on the motor controller to the GND on the cape.

   (c) Use a jumper wire to connect the 3.3V Bus to the V_{cc} pin on the motor controller.

   (d) Connect the battery pack to the terminal block connector (the green thing). Ensure the polarity is correct! The red wire goes to the positive (plus) terminal.

   (e) Use two jumper wires to connect to a motor on KHAN chassis to the MotorA pins on the motor controller.

   (f) Use the GPIO and PWM interfaces to command the motor to 2 or 3 different speeds. See page 4 of the motor controller datasheet[4] sheet to see the pin configuration for motor directions. It is recommended to remove the belt from the wheel so that it is not backdriving the second wheel.

   (g) Send a command to stop the motor from moving.

2. Wire the IR Rangefinder into to the ADC (Check the spec sheet!) **WARNING:** The 5V output of the sensor will damage the BeagleBone Black input. The voltage divider described in the instructions must be correctly wired in order to avoid hardware damage.

   (a) Connect the red wire into the 5V bus and the black wire into GND

   (b) Connect the data pin (yellow wire) to P9_39 or A_{in0}. Show the voltage readings from the sensor and how it changes as the distance from the sensor to some object moves.

5 Extra Credit

1. 10% - Convert raw voltages from the distance sensor to actual distances

2. 20% - Output the distance sensor information as a message type Range[5] and use RQT to visualize it

[4] http://www.adafruit.com/datasheets/TB6612FNG_datasheet_en_20121101.pdf

[5] http://docs.ros.org/api/sensor_msgs/html/msg/Range.html