

Progress review #2

Gauri Gandhi

Team G Robographers

Teammates:

Rohit Dashrathi

Jimit Gandhi

Tiffany May

Sida Wang

ILR #3

October 29, 2015

a. Individual Progress

For the second Progress Review, I worked on the Turtlebot odometry and gyro calibration and presented my team's work in the review. Besides this, I worked on the ROS tutorials to finish the ROS task. I also prepared the conceptual review for the Power Distribution Board required for our project.

I. **Running the Turtlebots**

For this Progress review, I took up the task of bringing up the turtlebots and doing their calibration. To learn about the hardware of the platform we are using for the project, I first disassembled one of the Turtlebots. This way, I could know about the part number of the chip mounted on the power board of the turtlebot and also understand its assembly (figure 1). The chip was ADXRS652 and its gyro setting number was 250 (figure 2).

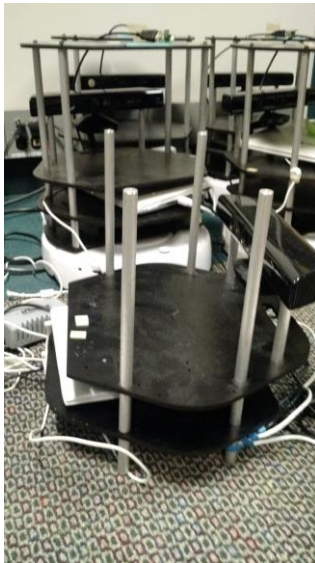


Figure 1 Disassembled Turtlebot

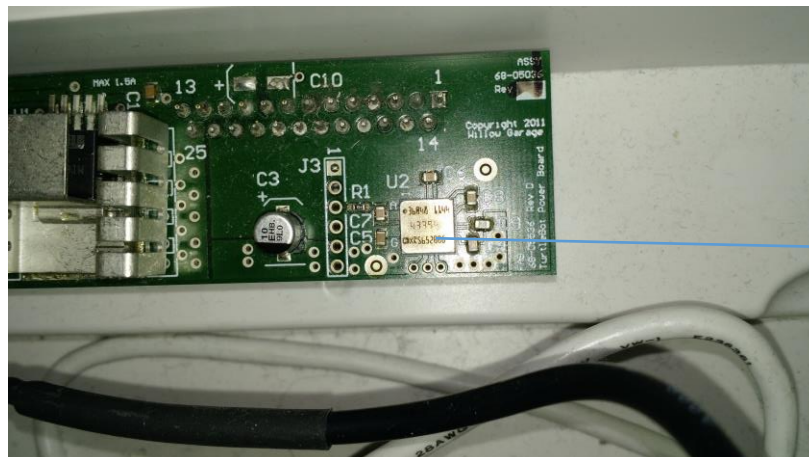


Figure 2 Chip on the power board

After that I assembled it again and started working on the calibration part. To do this, I connected my laptop to the turtlebot network in the Advanced Agents lab on first floor of NSH. Then I connected it to the netbook on the turtlebot through ssh to roslaunch the calibration node files. But it didn't give any result as the turtlebots were out of battery. So I left them to charge for a day and then brought one of them down to the MRSD lab. As the network changed, I had to reconfigure all the network settings again for both the workstation (my laptop) and the netbook on the turtlebot.

First I found out the IP addresses of both the machines on the MRSD lab network using the ifconfig command.

1. Workstation: 192.168.1.101
2. Netbook: 192.168.1.120

On the workstation, I wrote the following commands on Ubuntu terminal for installation:

1. To install chrony
\$ sudo apt-get install chrony
2. To install turtlebot ros library
\$ sudo apt-get install ros-hydro-turtlebot
3. To source setup.sh script from .bashrc
\$ echo "source /opt/ros/hydro/setup.bash" >> ~/.bashrc
4. Debs installation for create base
\$ export TURTLEBOT_BASE=create
\$ export TURTLEBOT_STACKS=circles
\$ export TURTLEBOT_3D_SENSOR=kinect
5. To merge the above commands in .bashrc
\$ nano ~/.bashrc

To configure the network settings on the netbook, I edited the following commands in the .bashrc file:

```
$ echo export ROS_MASTER_URI=http://192.168.1.120:11311  
$ echo export ROS_HOSTNAME=192.168.1.120
```

And then I did the bringup on the netbook:

```
$ roscore  
$ source /opt/ros/hydro/setup.bash  
$ roslaunch turtlebot_bringup minimal.launch
```

To configure the network settings on the workstation, I wrote the following commands:

```
$ echo export ROS_MASTER_URI=http://192.168.1.120:11311 >> ~/.bashrc
```

```
$ echo export ROS_HOSTNAME=192.168.1.101 >> ~/.bashrc
```

To verify the connection, I stopped the roscore on netbook, sshed into it through my laptop and again started the roscore. Then I wrote this command:

```
$ rostopic pub -r10 /hello std_msgs/String "hello"
```

On turtlebot, I wrote this:

```
$ rostopic echo /hello
```

The message "hello" started printing on it about 10 times a second.

After doing all this, I closed the lid of the netbook and placed it on the turtlebot and just wrote the following command for calibration:

```
$ roslaunch turtlebot_calibration calibrate.launch
```

I placed the turtlebot against the wall. As a result, instead of rotating about a point for calibration, it started moving and avoiding obstacles through the Kinect.

II. PDS PCB Conceptual Design

To write the conceptual design, I first found out what were the various components in our project that required electrical power. Then I arranged them in a table (figure 3) and found out their different power parameters like source voltage, regulation, drawn current, etc. by studying their datasheets. Once I was done with analyzing the needs of our system, I moved on to defining the design parameters of the power distribution board corresponding to each of the relevant need (2 servo motors and 1 arduino as they require external power; other things run on rechargeable sources). I arranged them in a table (figure 4) along with various parts that we will need for overvoltage/reverse voltage protection, voltage regulation, connector, high current protection, etc. To select these parts, I studied their datasheets and matched them to our project requirements. I further got it verified from our sponsor Professor Katia Sycara.

Table 1 : Needs Description(Per Turtlebot)								
S.No	Need(Qty)	Subsystem	Power Source Type Required	Voltage Range Required	Regulation Required	Continuous current drawn	Peak current drawn	Comments
1	Hi-Tech HS 311 servos(2)	Pan Tilt Unit	3-5 Volt Peak to Peak Square Wave	4.8-6V	yes	7.4-7.7mA/idle; 160-180mA no load operating	700-800mA(Stall current)	from power distribution board
2	Arduino Uno(1)	Pan Tilt Unit	7-12V regulated DC supply	7-12V(recommended); 6-20V(limit)	yes	20mA(per I/O pin); 50mA(3.3V pin)	500mA	from power distribution board
3	Logitech HD Webcam C615(1)	Detection	through USB 2.0 or above equipped laptop	-	-	-	-	from Netbook
4	MS Kinect(1)	Planning and Navigation	12V 1.5Amp Software Enabled Power Supply	-	-	-	-	from TurtleBot Power Board
5	TurtleBot		3000 mAh Ni-MH Battery Pack	-	-	-	-	from TurtleBot Power Board; plugged into main supply for charging
6	Netbook		6cells Li-ion Battery Pack	-	-	-	-	plugged into main supply for charging

Figure 4 Needs Description

Design Description(per Turtlebot)																	
S.No.	Need	Source	Supply required	Connector Type	Number of Connectors	Max Current	Current Capacity of connector	Monitor/ Switch	Over voltage Protection	Voltage Regulation and Reverse Voltage Protection	DC-DC Converter	DC-DC Converter Vout	DC-DC Converter Iout(max)	DC-DC Converter Efficiency (typical)	Power Control	Voltage Control	Visual Indication
1	Hi-Tech HS 311 servos (pan/tilt)(2)	24V Rechargeable Battery	5 Volt regulated DC supply	Minifit Jr power connectors	2	700-800mA	3A	no	0.8A TES Fuse 395	Uni-TVS Diode P4SMA6.8A	Murata UWS-5/4.5-Q48	5V	10	91% no	no	no	LED
2	Hi-Tech HS 311 servo (elevation-optional)(1)	24V Rechargeable Battery	5 Volt regulated DC supply	Minifit Jr power connectors	1	700-800mA	3A	yes	0.8A TES Fuse 395	Uni-TVS Diode P4SMA6.8A	Murata UWS-5/4.5-Q48	5V	10	91% no	no	no	LED
3	Arduino Uno(1)	24V Rechargeable Battery	12V regulated DC supply	Minifit Jr power connectors	1+1(extra)	500mA	3A	no	0.5A TES Fuse 395	Uni-TVS Diode P4SMA15A	Murata UWS-12/4.5-Q48	12V	4.5	91% no	no	no	LED

Figure 3 Design Description

III. Presentation for Progress Review

For the Progress Review 2, I was the presenter for my team. So I presented the following accomplished goals in the following order. After that, every team member presented any extra details of the goals allotted to him/her.

1. CAD Model of PTZ unit + Turtlebot Assembly
2. Working demo of Intraface features
3. Collaboration Sub-system Fall and Spring work breakdown schedule and the modified Functional Architecture
4. Working model of the 3D printed pan-tilt unit
5. Demo of the running turtlebots avoiding obstacles

b. Challenges

- I. The first challenge that I faced this time was to find the chip number on the power board of the turtlebot. I was doing this task with Jimit. At first, we tried to use the flash light of the phone by inserting it inside the turtlebot assembly. But it didn't work for us. So finally we decided to disassemble the turtlebot so that we could also learn about the hardware. Rohan Thakker also gave his inputs in this.
- II. When I tried to launch the ros node for calibration for the first time, I found out that the robot doesn't get charged if it is being used side by side. Also it takes a lot of time to get charged and then works for just 20-25 minutes. This problem is a recurrent challenge for me.
- III. We brought the turtlebot down to the MRSD lab and nothing worked in the first go. Then we realized that this problem occurred as the network changed for the netbook. So I had to reconfigure all the network settings again for both the netbook and my laptop. I had to relaunch all the bringup files on the netbook. The major problem while doing this was that the netbooks are very slow. The connection took a lot of time to establish.
- IV. During the network reconfiguration, some problems occurred with the Microsoft permissions and agreements on my laptop even though I was working on Ubuntu. Due to this, I was unable to install anything through sudo apt-get. So I surfed the web for this problem and found a command to reconfigure all those permissions. That turned out to be successful.
- V. After the network was setup, I launched the ros node for calibration of the turtlebot. But it started to avoid obstacles. I will rectify that problem by this weekend.

c. Team Work

For this progress review, I worked with Jimit to run the turtlebot. Rohit worked on the CAD model of the pan-tilt unit. He 3D printed the unit and assembled it on the turtlebot. He also ran a code on the Arduino for running the pan and tilt servo motors

simultaneously. Sida and Tiffany talked to the Intraface developers to understand the software and showed a working demonstration of it. They have also obtained the permissions to modify some parts of the code to meet the project needs. Jimit worked on the collaboration sub-system work schedule. He also revised our functional architecture and produced a more efficient one.

d. Future Plans

Before the next progress review, I will be working on the following tasks:

- I. Complete the Calibration of the IMU and wheel odometry of the TurtleBots.
- II. Work on the development of local EKF on ROS.
- III. Caliberate single april tag with Kinect.
- IV. Finish the PDS PCB Task with my team.
- V. Update the team website.

e. References

1. Turtlebot ROS tutorials

wiki.ros.org