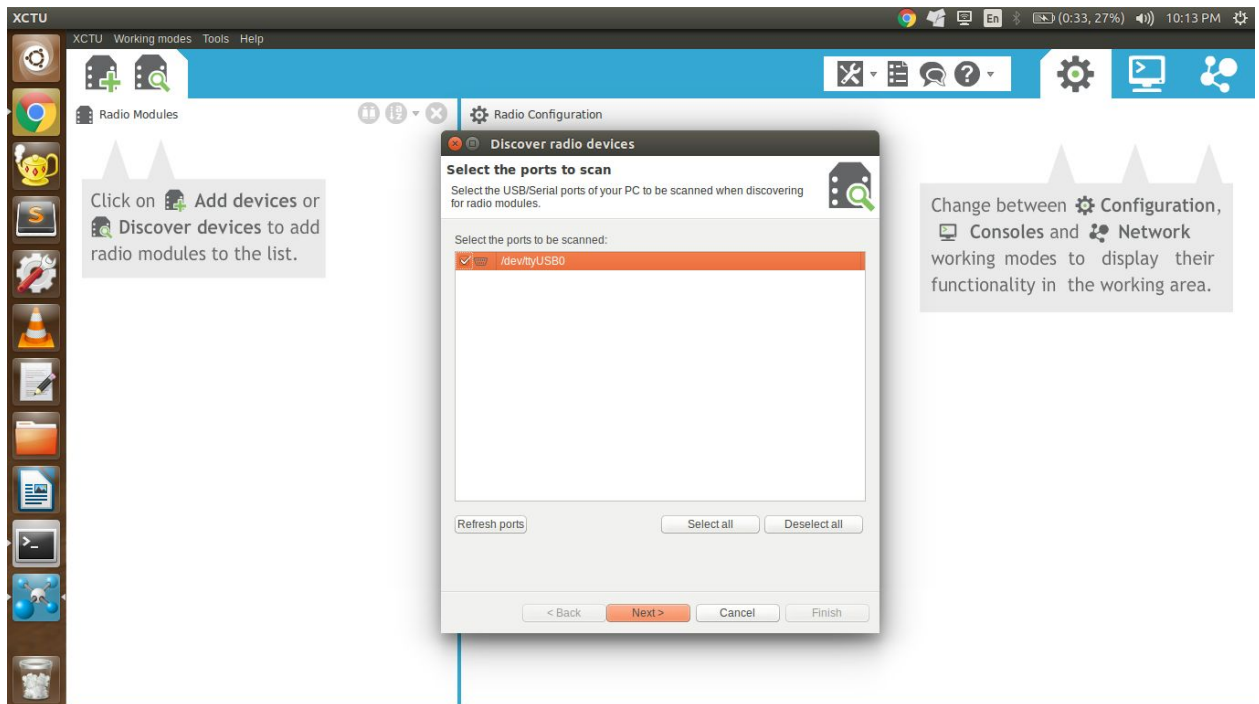
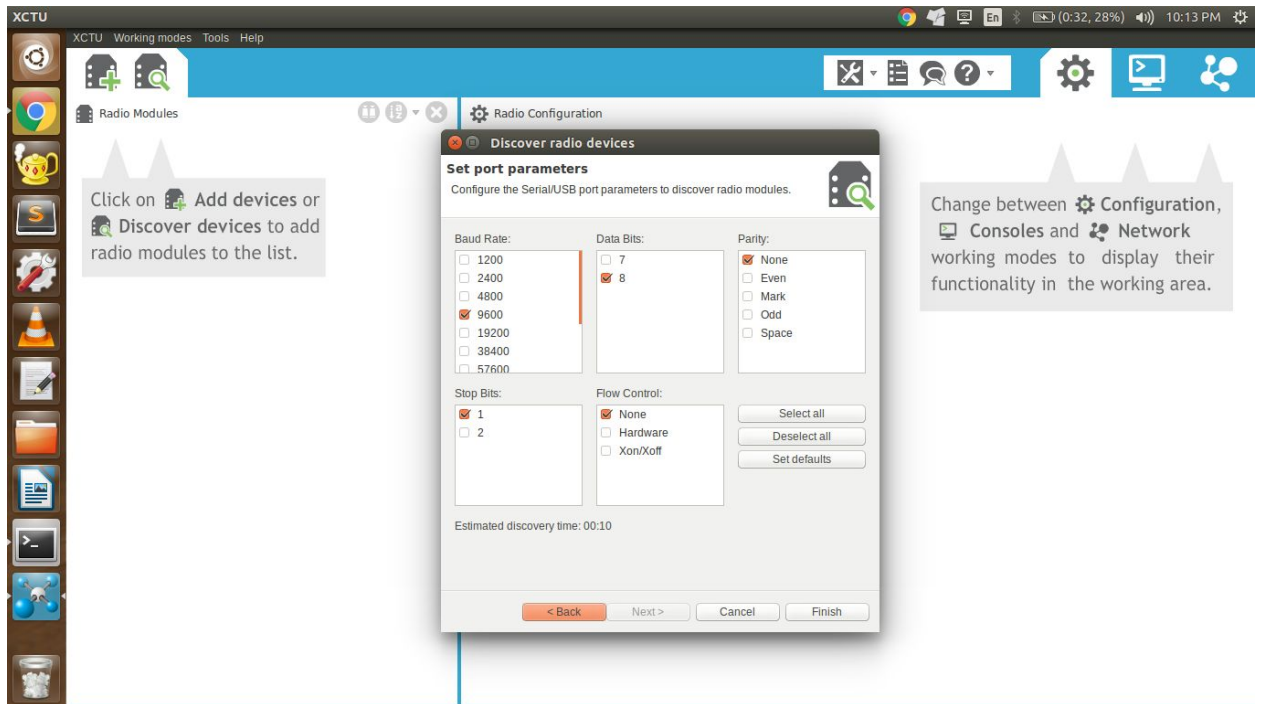


- **Install XCTU**

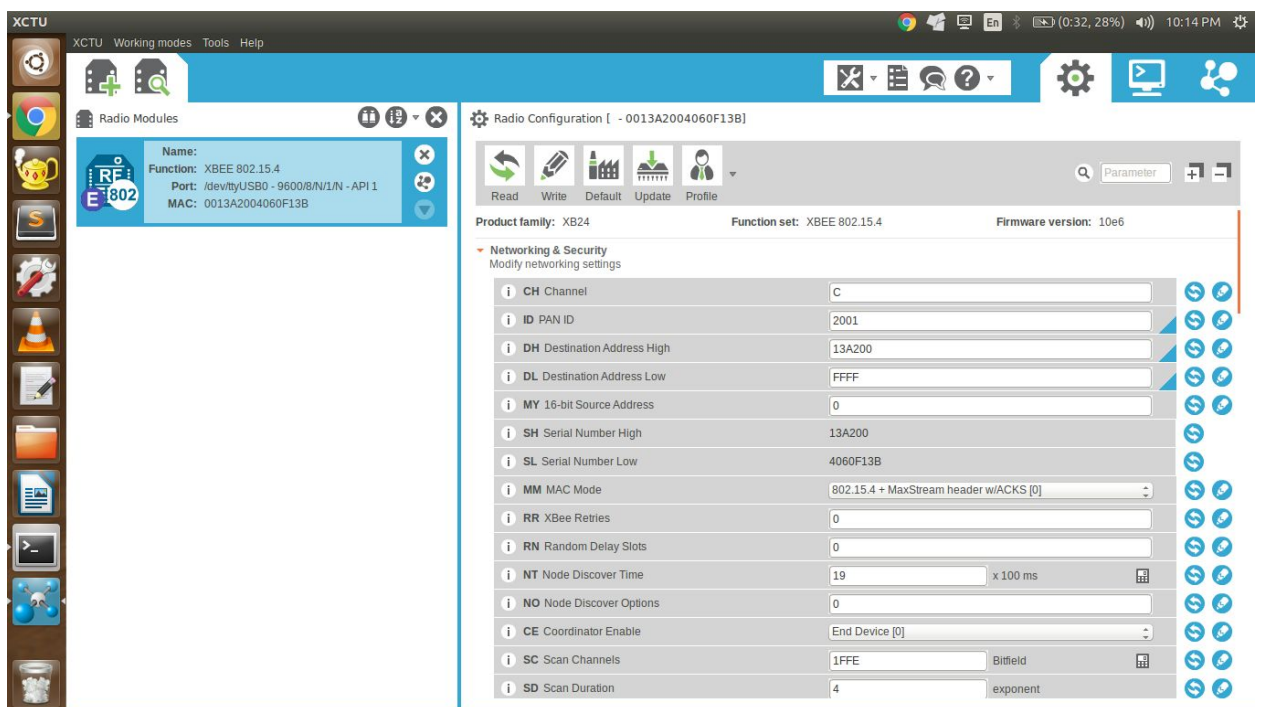
- Download XCTU(for linux) as given in http://knowledge.digi.com/articles/Knowledge_Base_Article/HOW-TO-Install-XCTU-in-Linux
<http://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu#productsupport-utilities>
- [XCTU tutorial](#)
- Three XBees, one in explorer connected to central server and the other two in bots. Identical id's for all bot
- Choose a constant ID (PAN ID) for the all the XBees to be same, for example 2001
- Open XCTU and click on search from the top left to get



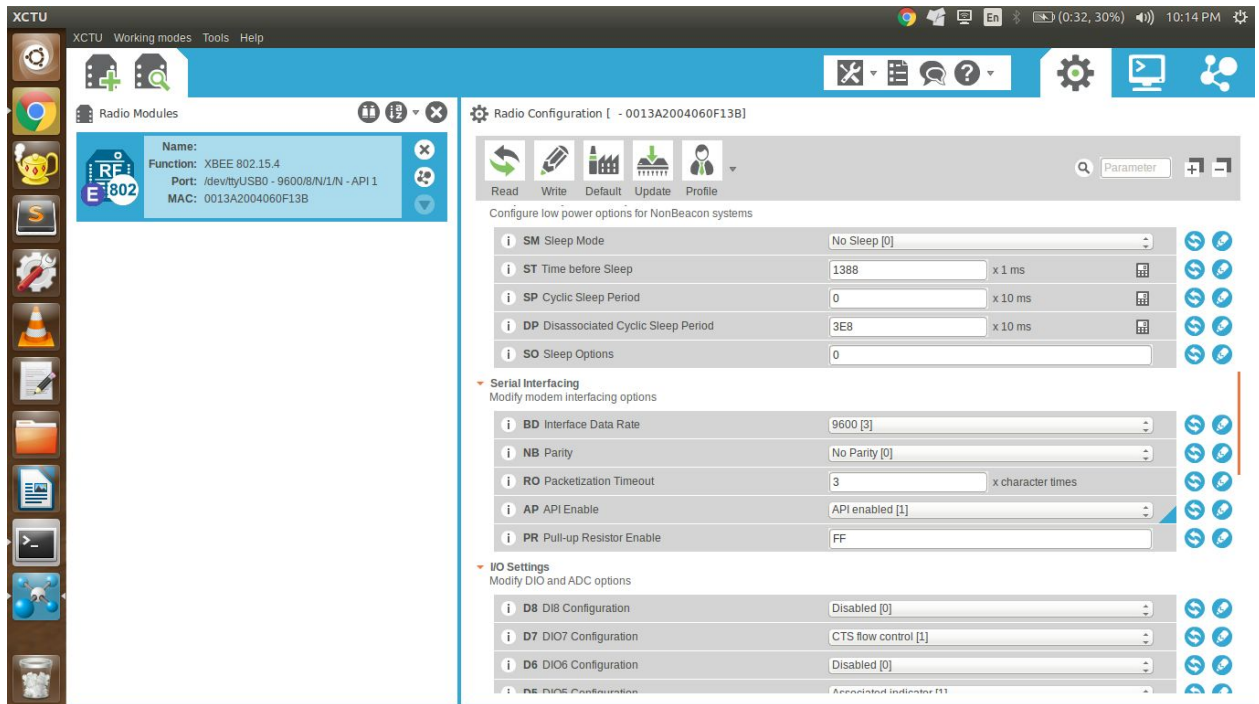
- Click on Next



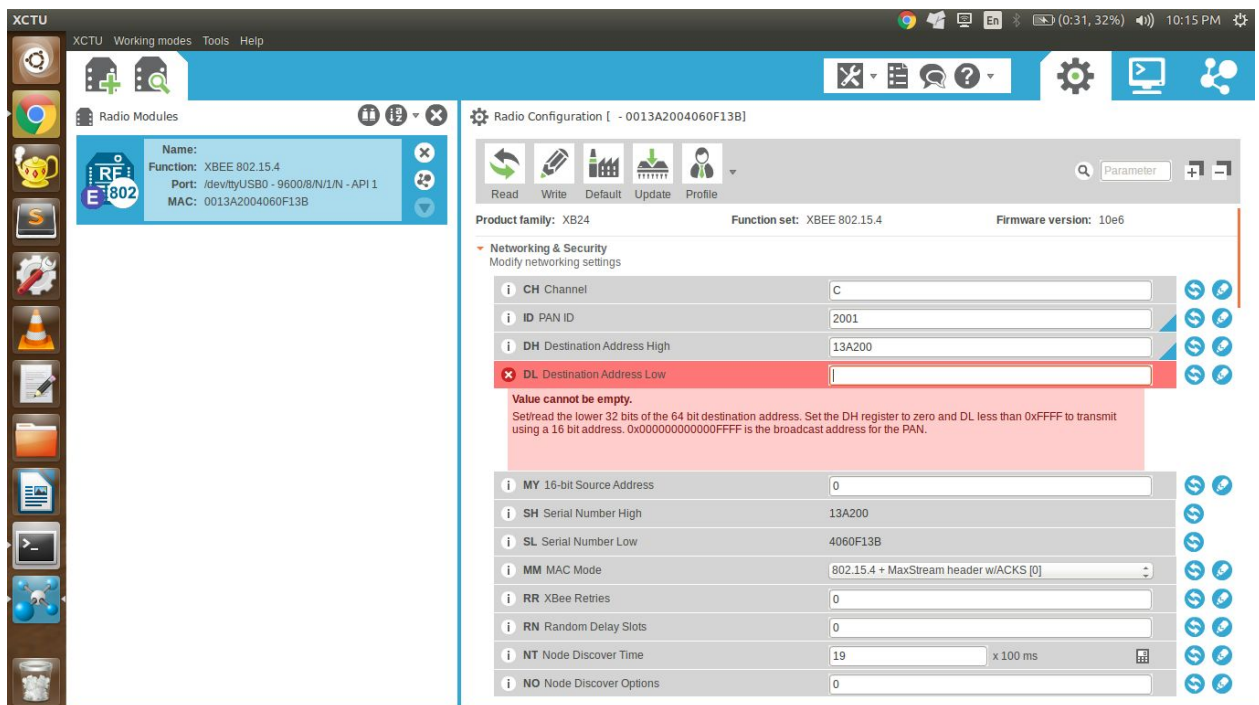
- Click Finish



- Choose different parameters and save them by clicking on a button to the right of the corresponding row



- Set DH to 13A200 (generally 13A200 for XBees or look at the back of the XBees and write the first half of the number written on the back).



- For the XBee in the explorer:
DL: FFFF (Broadcast Mode)
AP: 1
- For the XBee in the bots:

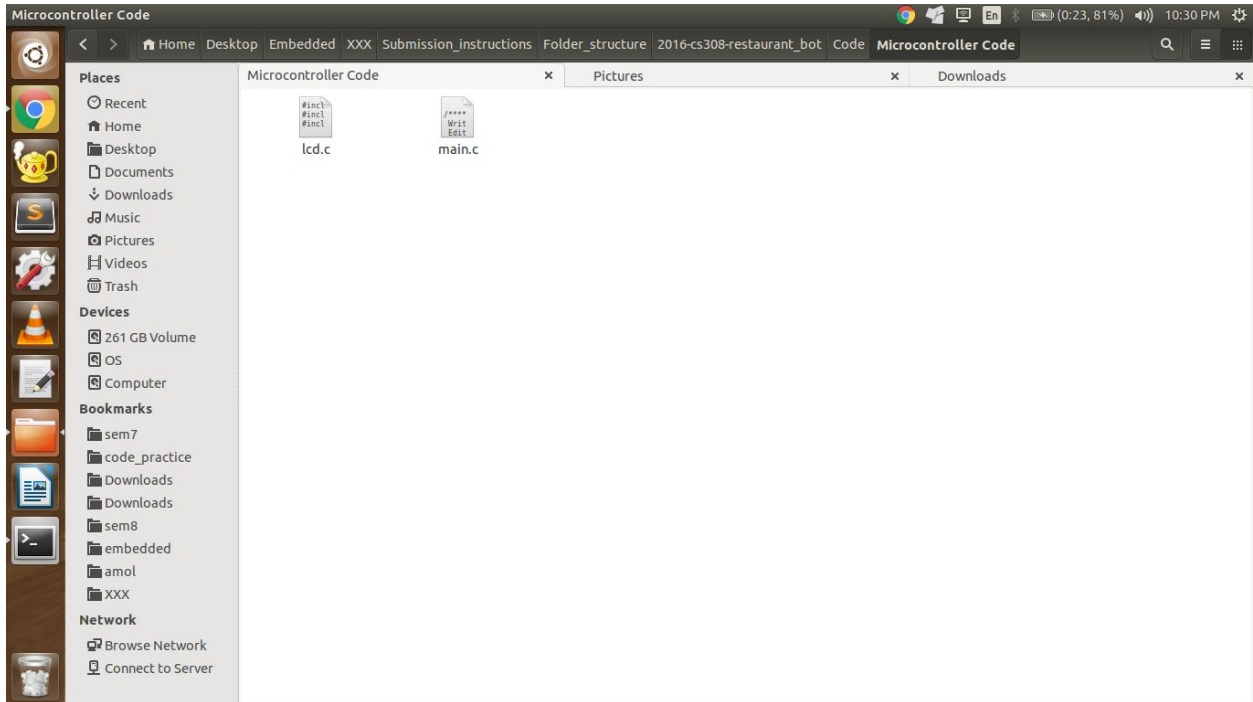
DL: Low address of Xbee in explorer (given on the back)
AP: 0

- **Install CodeBlocks**

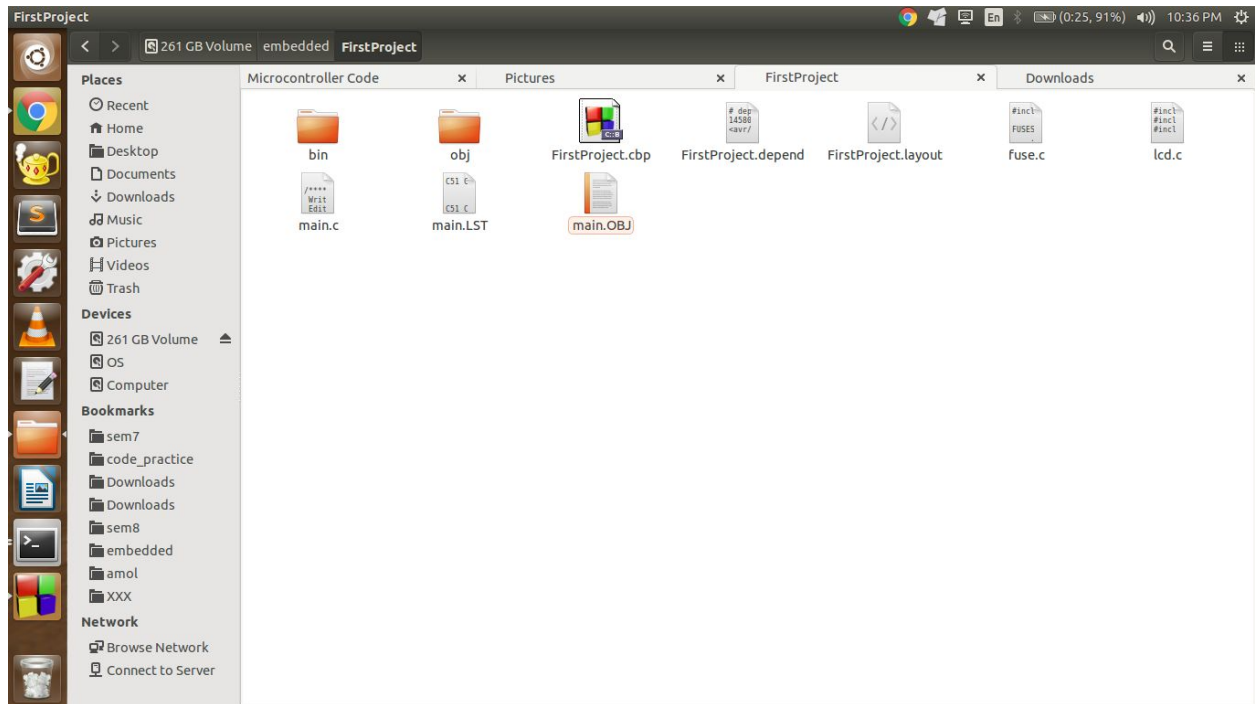
- Check http://wiki.codeblocks.org/index.php/Installing_Code::Blocks

- **To burn the code in the bot**

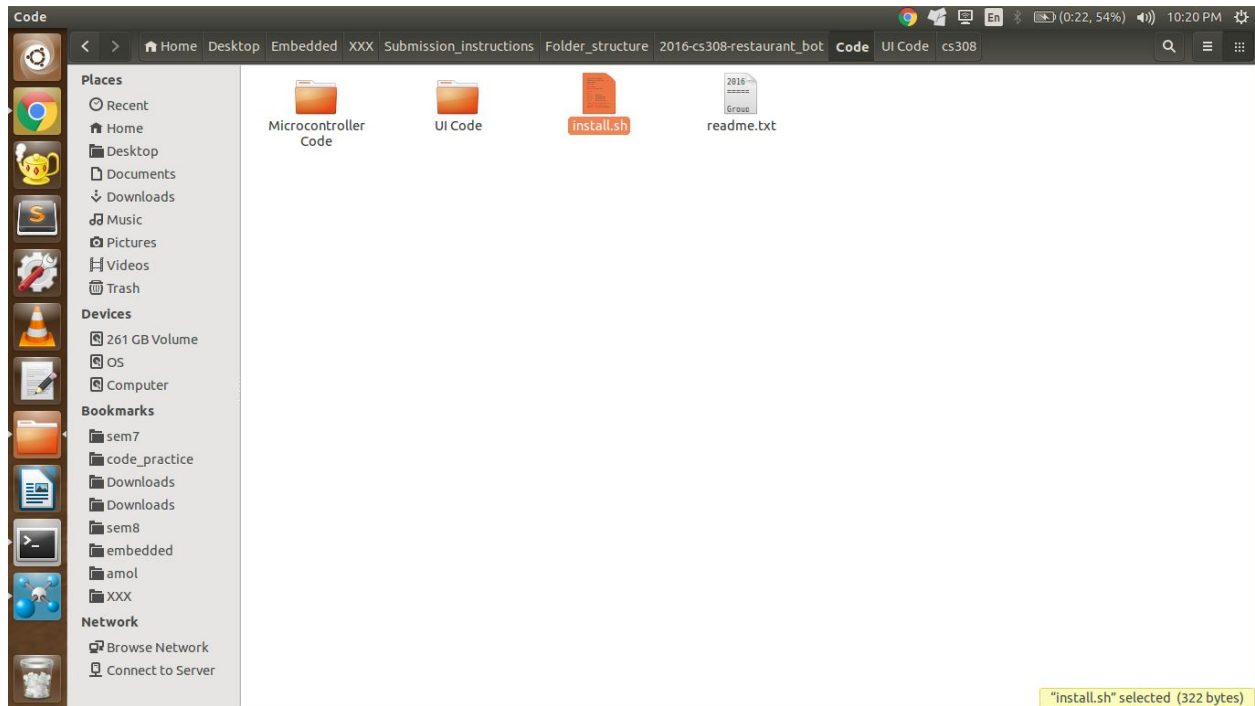
- Go to 2016-cs308-restaurant_bot/Code/Microcontroller Code



- Create a new AVR project in CodeBlocks with the two files main.c and lcd.c as shown in the following figure

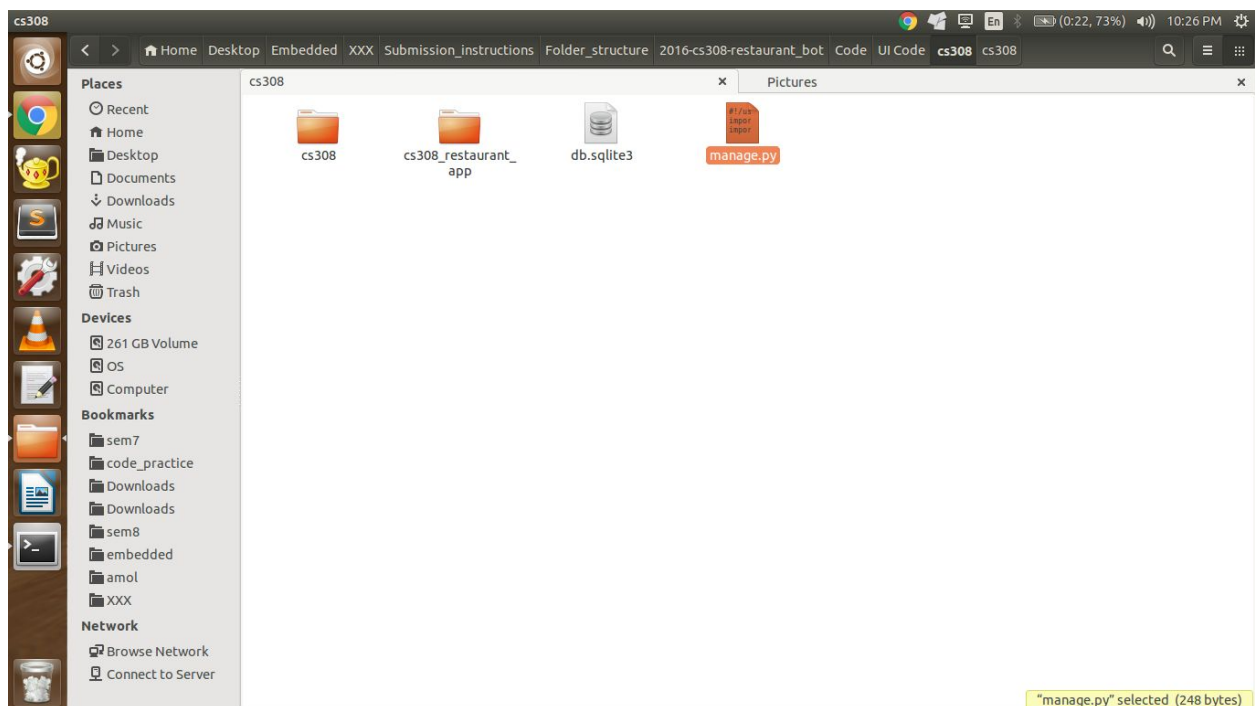


- Change the variable “botId” in main.c to 0 and 1 for bot 1 and bot 2 respectively
 - Burn the hex file after building the project successfully in the bot as shown in Firebird V tutorials
-
- **Installations required for Web server**
 - Go to 2016-cs308-restaurant_bot/Code
 - Run ./install.sh

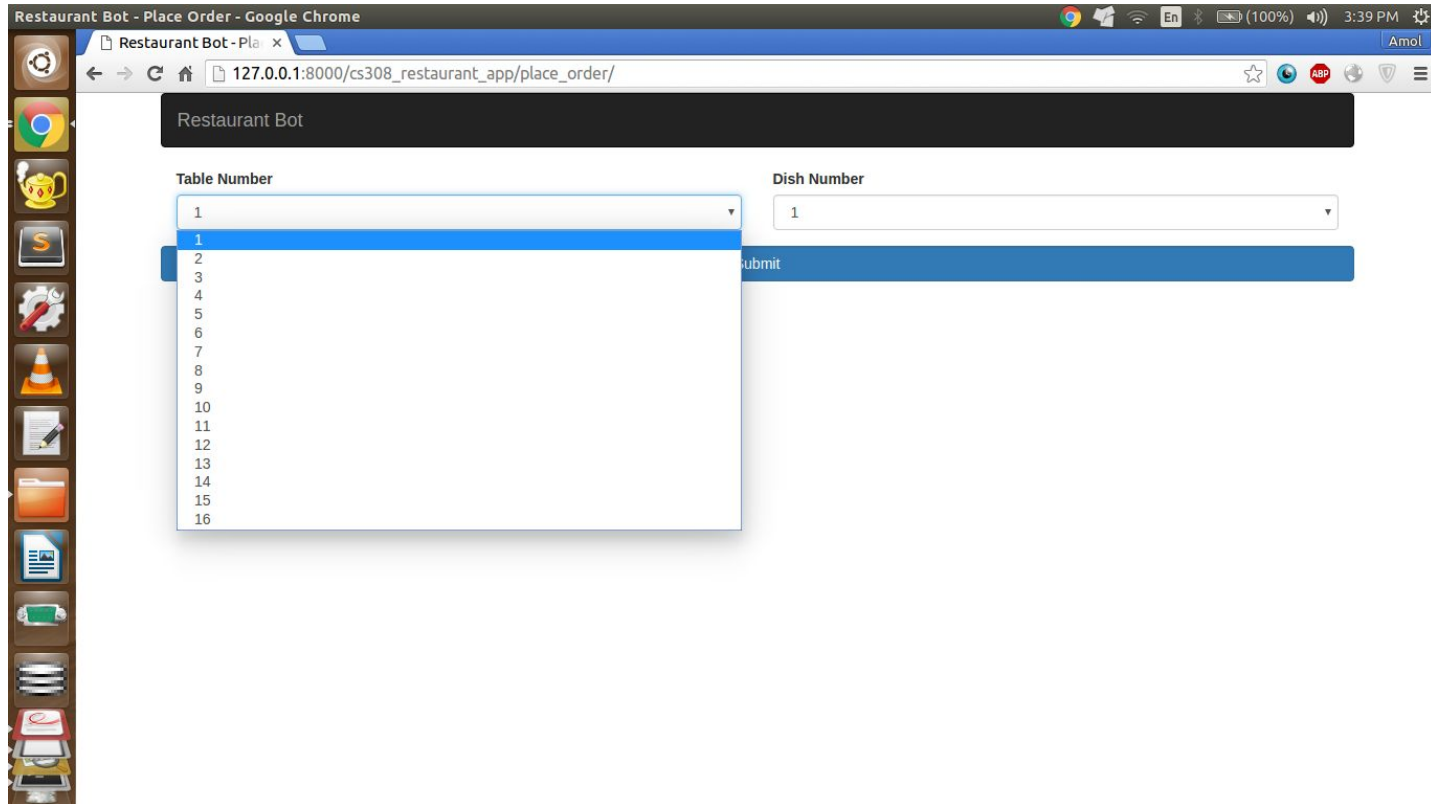


- **To Run the web server**

- Go to 2016-cs308-restaurant_bot/Code/UI Code/cs308
- Command: `./manage.py runserver`
- This will get the server started. Once the server is started. Follow instruction below



- To Place order follow the following URL after running the server:
http://127.0.0.1:8000/cs308_restaurant_app/place_order/



The screenshot shows a web browser window titled "Restaurant Bot - Place Order - Google Chrome". The address bar displays the URL "127.0.0.1:8000/cs308_restaurant_app/place_order/". The page content includes a header "Restaurant Bot" and two dropdown menus: "Table Number" and "Dish Number". The "Table Number" dropdown is open, showing a list of numbers from 1 to 16. The "Dish Number" dropdown is set to 1. A blue "submit" button is visible below the dropdowns.

Restaurant Bot

Table Number

Dish Number

submit

- To assign order to a particular bot follow the following URL after running the server:

http://127.0.0.1:8000/cs308_restaurant_app/orders_left/

Restaurant Bot - Pending Orders - Google Chrome

Restaurant Bot - Pending Orders

127.0.0.1:8000/cs308_restaurant_app/orders_left/

Restaurant Bot

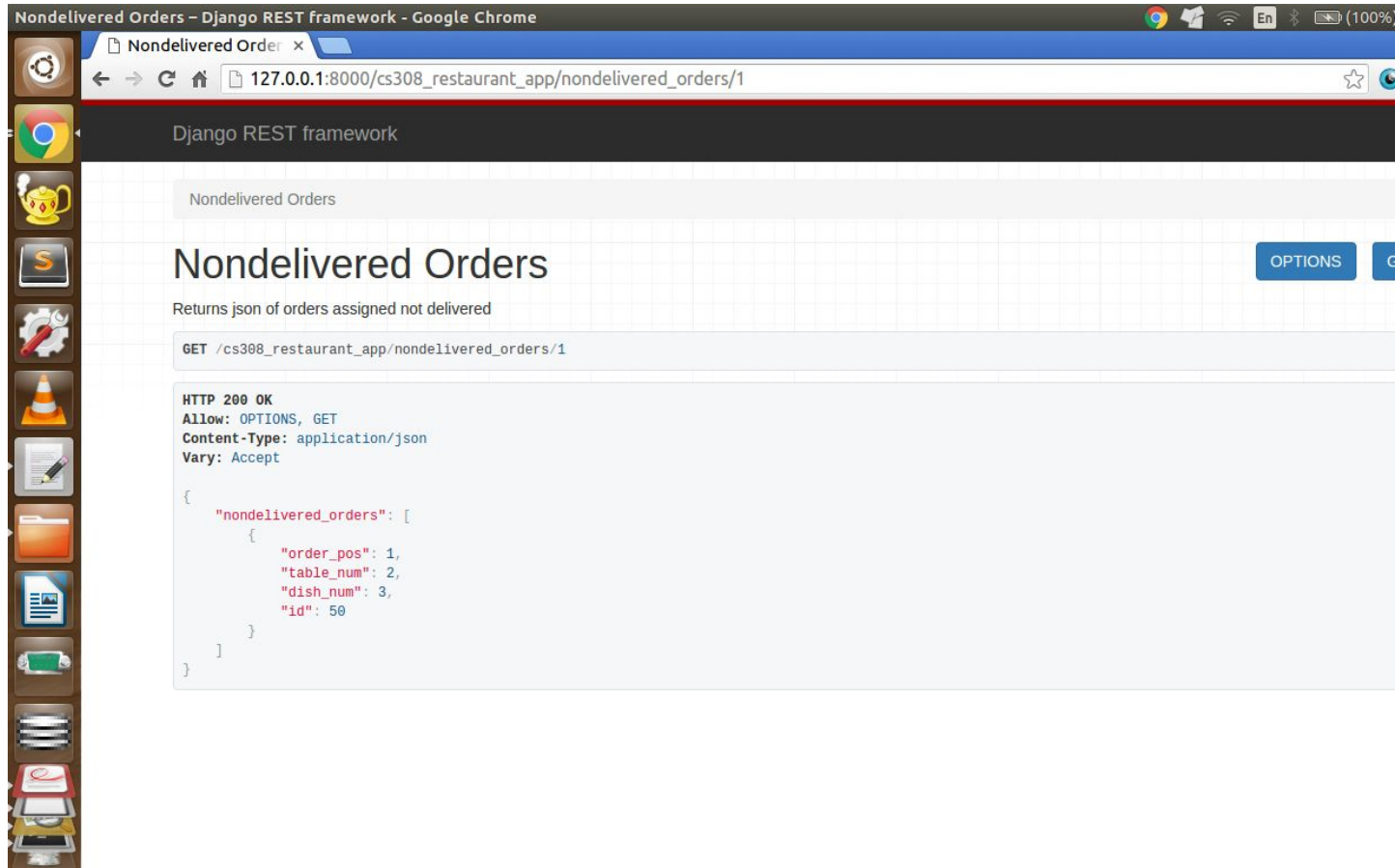
Table Number	Dish Number	Bot Number	Order Position	Order made
10	1	1	1	Order Made
10	3	1	1	Order Made
14	2	1	1	Order Made
3	1	1	2	Order Made
2	3	1	1	Order Made
1	1	1	2	Order Made

- To see non-delivered orders to a particular bot follow the following URL after running the server:

http://127.0.0.1:8000/cs308_restaurant_app/nondelivered_orders/botId

Where

botId = {1, 2}



- **Start the servicing by running the python code**
 - Go to 2016-cs308-restaurant_bot/Code/UI Code
 - Place the bots at the initial location on the red part of the arena, just before the white lines where all three IR sensors show white
 - Then run the command: `python main.py`
 - If there is a port error change the value of PORT to `ttyUSB1` or some other number