

Motor and Sensor Lab Review

Sida Wang (sidaw)

Individual progress

I was responsible for GUI part of the project. I used QT C++ for the GUI. This is the first time that I used QT GUI. In the past, I have used MFC, Matlab GUI, and Java GUI. QT can be supported by multi-platforms, and C++ is my favorite language. So I choose QT C++.

Failed attempt

I did something very interesting. I tried to code the whole GUI without plotting it. I thought that it was not difficult because just several classes, several .cpp and .h files. However, there is always big difference between the real life and ideal life. I've made millions of mistakes in the process. Actually, it was not very different, but very detailed. And I think I am not quite good at object-oriented programming. And I plan to study <C++ Primer> during the winter break. After I failed millions of time, I gave up making the GUI with pure coding C++.

The art of GUI Design

I love GUI design, not only because the fancy technology, but also because I love the design ideas. For example, golden ratio (≈ 0.618) is very important in GUI design. Sometimes design is more important than functionalities. I appreciate pretty, and convenient GUI. I hope to learn more about design idea, and art.

TTY

I am also very interested in "tty". As a freshman for Ubuntu, I wonder what it is at the very beginning. And I searched on Google, and I learned several commands about tty.

```
sida@sida-ThinkPad-T450s:~$ ls /dev
autofs          loop5           ram9            tty24           tty57           ttyS30
block           loop6           random          tty25           tty58           ttyS31
bsg             loop7           rfskill         tty26           tty59           ttyS4
btrfs-control   loop-control    rtc             tty27           tty6            ttyS5
bus             mapper          rtc0            tty28           tty60           ttyS6
char            mcelog          sda             tty29           tty61           ttyS7
console         media0          sda1            tty3            tty62           ttyS8
core            mei0            sda2            tty30           tty63           ttyS9
cpu             mem             sda5            tty31           tty7            uhid
cpu_dma_latency memory_bandwidth sda6            tty32           tty8            uinput
cuse            net             sg0             tty33           tty9            urandom
disk            network_latency shm              tty34           ttyprintk       v4l
dri             network_throughput snapshot          tty35           ttyS0           vcs
encryptfs       null            snd              tty36           ttyS1           vcs1
fb0             nvram           stderr          tty37           ttyS10          vcs2
fd              port            stdin           tty38           ttyS11          vcs3
full            ppp             stdout          tty39           ttyS12          vcs4
fuse            psaux           tpm0            tty4            ttyS13          vcs5
hpet            ptmx            tty             tty40           ttyS14          vcs6
i2c-0           ptp0            tty0            tty41           ttyS15          vcs7
i2c-1           pts             tty1            tty42           ttyS16          vcsa
i2c-2           ram0            tty10           tty43           ttyS17          vcsa1
i2c-3           ram1            tty11           tty44           ttyS18          vcsa2
i2c-4           ram10           tty12           tty45           ttyS19          vcsa3
i2c-5           ram11           tty13           tty46           ttyS2           vcsa4
i2c-6           ram12           tty14           tty47           ttyS20          vcsa5
i2c-7           ram13           tty15           tty48           ttyS21          vcsa6
i2c-8           ram14           tty16           tty49           ttyS22          vcsa7
input           ram15           tty17           tty5            ttyS23          vfio
kmsg            ram2            tty18           tty50           ttyS24          vga_arbiter
log             ram3            tty19           tty51           ttyS25          vhci
loop0           ram4            tty2            tty52           ttyS26          vhost-net
loop1           ram5            tty20           tty53           ttyS27          video0
loop2           ram6            tty21           tty54           ttyS28          watchdog
loop3           ram7            tty22           tty55           ttyS29          watchdog0
loop4           ram8            tty23           tty56           ttyS3           zero
```

There are many ttys. And I wonder where I am. So I also learned a command:

```
sida@sida-ThinkPad-T450s:~$ tty
/dev/pts/0
sida@sida-ThinkPad-T450s:~$
```

And if I want to use pure command lines, I can just “Ctrl + Alt + F1 / F2/ F3/ F4/ F5/ F6”, and go into tty1 / 2 / 3 / 4 / 5 / 6. But I still wonder where is tty20 ... etc.

```
sida@sida-ThinkPad-T450s:~$ who
sida    tty1          2015-10-16 23:51
sida    :0            2015-10-16 23:13 (:0)
sida    pts/0        2015-10-16 23:20 (:0)
sida@sida-ThinkPad-T450s:~$
```

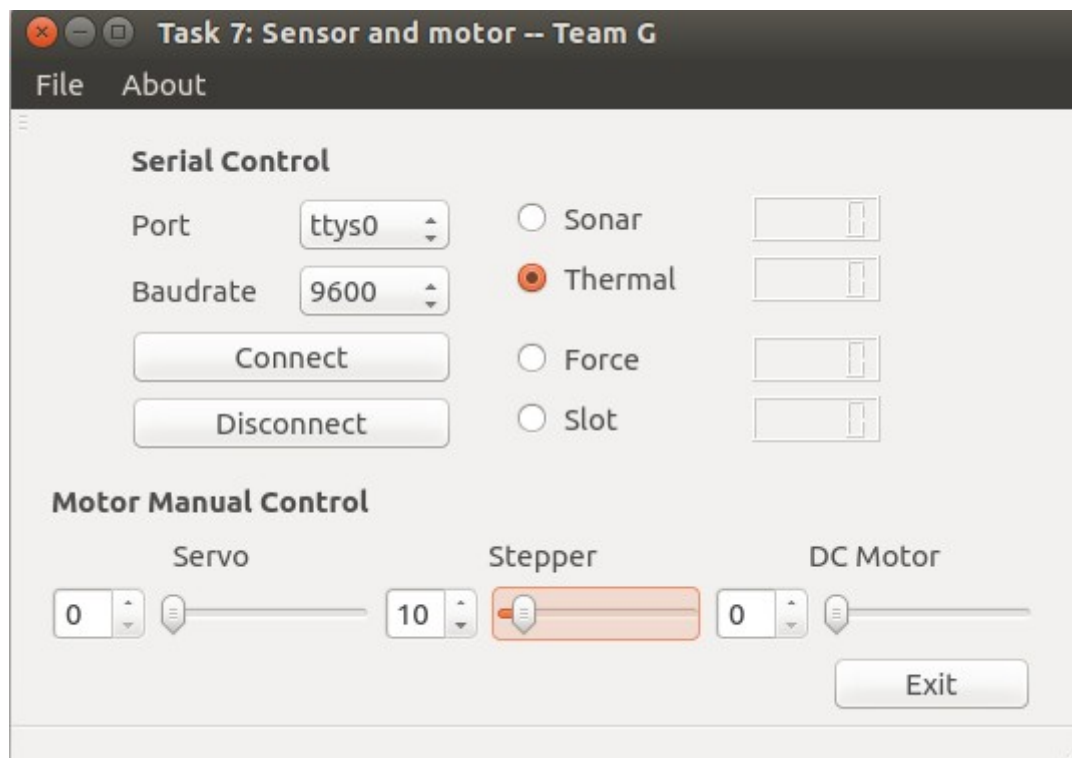
I think this is also very interesting. In order to learn more, I also typed:

```
sida@sida-ThinkPad-T450s:~$ whatis who
who (1)          - show who is logged on
sida@sida-ThinkPad-T450s:~$ whereis who
who: /usr/bin/who /usr/bin/X11/who /usr/share/man/man1/who.1.gz
```

OK, I think this is enough for tty.

My GUI:

My GUI is like:



Hardware

I found it interesting until this part, because up to now is pure software. For the next part, I don't think I can do it very well, because I have no background for circuit. The other 4 students in our group are responsibility for circuit design and sensor and motor realization. After the lab, I called my friend for help in the field of circuit. He told me that circuit is just boolean classes, and I can use 1 or 0 to determine which one is on and which one is off. I think in this way, I can understand the circuit better.

Merge the code:

I am also participated in merging the code. First, I changed each students' format of coding. There is pep8 coding format for Python, which has specific rules to write clean and pretty code. I think there must be similar rules for C++. And merging code is kind of art, and I classified different functions and make the code prettier.

Connect qt and arduino

Furthermore, I connected qt and arduino. There is youtube video to show how to connect it. And I successfully connected, and I also transferred values between qt and arduino.

Failures

However, I failed to change the format of the signals between qt and arduino. So I failed to read the value of sensor. I feel sorry to our group...

Future plan:

I have used Ubuntu for several weeks, and I do love this fancy OS!!! And I also installed IntraFace, openCV, etc. I hope to explore the field of Detection of Turtlebots!!!

Furthermore, I will communicate more with my teammates. To talk, always listen to them, and hope we will have great year in CMU MRSD!