Setting up Raspberry Pi, Camera, and iRobot
Create

Follow the following instructions to set up your system.

Available at: https://github.com/zyrgit/cs424/tree/master/Setting%20up%20the%20lab

Note:

On laptop means using the terminal logged in to your own laptop; On RPi means using the terminal logged in to your Raspberry Pi.

You can also read the reference to last year’s PDF: https://courses.engr.illinois.edu/cs424/fa2016/mp/init.pdf. But don’t copy paste commands from that PDF since there are invalid encodings.

Setup RPi:

1. Format SD card.

Download SD Formatter 4.0 for either Windows or Mac https://www.sdcard.org/downloads/formatter_4/index.html

Install it on your laptop and format the SD card using default option.

2. Flash SD card.

On laptop (Mac), insert SD card (assume appearing as BOOT) and run:

```
cd /Volumes/BOOT
wget https://downloads.raspberrypi.org/NOOBS_latest
unzip -a NOOBS_latest
rm -f NOOBS_latest
```

If not using Mac, see https://www.raspberrypi.org/documentation/installation/noobs.md

Once flashed, eject the SD card and plug it in RPi, connect RPi to a screen (HDMI), a mouse and a keyboard. Power it up.
3. Install OS.

Once RPi powers up, select raspbian OS to install. You can choose to connect to WiFi before installation via the screen connected to RPi.

4. Setup networking configurations.

Suppose you are interacting with RPi via a screen, mouse, and a keyboard. Open a terminal on RPi.

On RPi, run `ifconfig` to see RPi’s IP (e.g. 10.194.113.96). Run `ssh-keygen -t rsa` and `sudo service ssh restart` on RPi. If there is connection refused error, run `sudo apt-get purge openssh-server` and `sudo apt-get install -y openssh-server`.

On laptop, SSH into RPi using `ssh pi@10.194.113.96` with password `raspberry`.

On RPi, run `echo -n 'Your_UIUC_password' | iconv -t UTF-16LE | openssl md4` and copy the string after `(stdin)=`, this is to replace the password field in `wpa_supplicant.conf`. Run `history -cw` to clear cmd history.

On RPi, run `sudo nano /etc/wpa_supplicant/wpa_supplicant.conf`, change to the following (if you do not use home wifi, omit the network config of "Your_home_wifi"):

```plaintext
country=GB
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="Your_home_wifi"
    psk="Your_password"
    key_mgmt=WPA-PSK
}

network={
    ssid="IllinoisNet"
    key_mgmt=WPA-EAP
    identity="Your_netid"
    password=hash:string_from_previous_cmd
}
```

Use Ctrl-o, Enter, Ctrl-x to save and exit nano.

You also need to use Ethernet on RPi to communicate with your laptop. Because once you get rid of the screen for RPi, you have no way of knowing its IP. You need to SSH into RPi from your laptop via Ethernet (or USB-Ethernet adaptor) to know RPi’s IP on wireless interface. So connect your laptop with RPi via Ethernet cable now.

If you use Ubuntu, run `ifconfig eth0 up` on laptop to bring up Ethernet interface `eth?`, if you use Mac, plug in cable and `en?` should be up automatically. Find the IP of your laptop Ethernet interface
using `ifconfig`. On Ubuntu it may be `192.168.0.1`, on Mac it may be `169.254.109.86`, something like that.

Your RPi and laptop needs to be on the same subnet. If you run `ifconfig` on your laptop, you see the netmask on your laptop, may be `netmask 0xffff0000 broadcast 169.254.255.255` on Mac or `netmask 0xffffff00 broadcast 192.168.0.255` on Ubuntu. If broadcast is `?.?.255.255` it means the lowest two bytes are masked, if broadcast is `?.?.?.255` it means the lowest one byte is masked.

If on your laptop, it's `netmask 0xffff0000 broadcast 169.254.255.255` then the netmask on your RPi should be `netmask 255.255.0.0`. If it's `netmask 0xffffff00 broadcast 192.168.0.255` then the netmask on your RPi should be `netmask 255.255.255.0`.

On RPi, run `ifconfig -a` and see the interface name for Ethernet, it may not be `eth0`, it can be something like `enxb827eb04aeaf`. Choose a different IP for your RPi's `eth?`, like `192.168.0.123` or `169.254.109.123`, only the last byte is different so they are on the same local network.

On RPi, run `sudo nano /etc/network/interfaces` and change to the following:

```bash
source-directory /etc/network/interfaces.d
auto lo
iface lo inet loopback

auto enxb827eb04aeaf
iface enxb827eb04aeaf inet static
    address 169.254.109.123
    netmask 255.255.0.0

allow-hotplug wlan0
iface wlan0 inet dhcp
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Note that the address for `enxb827eb04aeaf` is what you've chosen for RPi, and the netmask is either `255.255.0.0` or `255.255.255.0`. Save and exit nano.

Restart by running `sudo shutdown -r now`. If everything goes well, on RPi, run `sudo apt-get update`, `sudo apt-get upgrade -y`, restart if necessary.

After restart, RPi's Ethernet should be automatically up. On laptop, SSH into RPi using `ssh pi@169.254.109.123`. On RPi you can check `ifconfig` and see the IPs:

```bash
pi@robotpi1:~ $ ifconfig
enxb827eb04aeaf: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 169.254.109.123 netmask 255.255.255.0 broadcast 169.254.109.255
    inet6 fe80::ba27:ebff:fe04:aeaf prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:04:ae:af txqueuelen 1000 (Ethernet)
    RX packets 42 bytes 13151 (12.8 KiB)
    TX packets 34 bytes 4782 (4.6 KiB)
```
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1 (Local Loopback)
    RX packets 2 bytes 78 (78.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2 bytes 78 (78.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.194.102.108 netmask 255.252.0.0 broadcast 10.195.255.255
    inet6 fe80::ba27:ebff:fe51:fbfa prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:51:fb:fa txqueuelen 1000 (Ethernet)
    RX packets 117 bytes 21303 (20.8 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 97 bytes 16527 (16.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

5. How to login to your RPi from now on.

From now on you don't need to screen or keyboard for your RPi. You need to connect your laptop to your RPi via Ethernet, and then power on RPi, run `ifconfig eth0 up` on Ubuntu if necessary. Your subnet IP (IP after masking) of the eth0 or en0 on your laptop should not be changing (unless you use another computer). Your RPi will always use the IP you've chosen for it. Therefore, on your laptop, run `ssh pi@169.254.109.123` or `ssh pi@192.168.0.123` to login your RPi (you may need to wait for 1 minute for your laptop to bring up Ethernet interface).

Then you need to find your RPi's wireless IP. On RPi run `ifconfig` and see the IP for wlan0, e.g., 10.194.102.108. Then you can disconnect your laptop and RPi now. Exit the terminal and unplug the Ethernet cable, login into your RPi using wireless IP: `ssh pi@10.194.102.108`.

On laptop, run `ssh-copy-id pi@10.194.102.108` so that you don't need to type password to login.


Suppose you're group 1, the name of your RPi should be robotpi1, change accordingly.

Change hostname by running `sudo nano /etc/hosts` on RPi and change 127.0.1.1 raspberrypi to 127.0.1.1 robotpi1

On RPi, run `sudo nano /etc/hostname` and change raspberrypi to robotpi1. Restart using `sudo reboot now` or `sudo shutdown -r now`. To power off RPi, run `sudo shutdown -h now`.

7. Additional configurations.

On RPi, run `sudo dpkg-reconfigure tzdata` to set the timezone (US/Central). Run `sudo apt-get install ntp` to install network time protocol for synchronization.
Install VNC (optional): For remote access, so far we have connected to and controlled the Pi through ssh. Sometimes we need to access the Raspberry Pi desktop GUI. You can install VNC for this purpose. Follow https://www.raspberrypi.org/documentation/remote-access/ to install VNC if you need it.

Install Serial Communications Libraries and Dependencies:

```bash
sudo apt-get install libboost-all-dev
sudo apt-get install python-sip
sudo apt-get install python-sip-dev
sudo apt-get install python-pip
sudo pip install pyserial

wget http://downloads.sourceforge.net/project/libserial/libserial/0.6.0rc2/libserial-0.6.0rc2.tar.gz
tar -zxvf libserial-0.6.0rc2.tar.gz
cd libserial-0.6.0rc2/
./configure
make
sudo make install
sudo ldconfig
```

8. Camera and OpenCV.

Install libraries:

```bash
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install -y build-essential cmake pkg-config
sudo apt-get install -y libjpeg-dev libtiff5-dev
sudo apt-get install -y libavcodec-dev libavformat-dev libswscale-dev
sudo apt-get install -y libv4l-dev libxvidcore-dev libx264-dev
sudo apt-get install -y libgtk2.0-dev
sudo apt-get install screen
sudo apt-get install python2.7-dev python3-dev
sudo apt-get install python-pip
sudo pip install numpy
```

Install OpenCV. Using `screen` makes sure that even if you close the terminal, the compiling is still running.

```bash
screen

cd ~
wget -O opencv.zip https://github.com/Itseez/opencv/archive/3.1.0.zip
unzip opencv.zip
wget opencv_contrib.zip
```
https://github.com/Itseez/opencv_contrib/archive/3.1.0.zip
unzip opencv_contrib.zip
cd ~/opencv-3.1.0/
mkdir build
cd build
cmake -D CMAKE_BUILD_TYPE=RELEASE \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D INSTALL_PYTHON_EXAMPLES=ON \
-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-3.1.0/modules \
-D ENABLE_PRECOMPILED_HEADERS=OFF \
-D BUILD_EXAMPLES=ON ..
make -j4

This will take 1.5 hour. Run:

```
sudo make install
sudo ldconfig
```

Install RaspiCam.

```
git clone https://github.com/cedricve/raspicam
cd raspicam
mkdir build
cd build
cmake ..
```

Make sure `CREATE_OPENCV_MODULE=1` is displayed somewhere in the command line output.

```
make
sudo make install
sudo ldconfig
```

Enable the camera module, on RPi, run

```
sudo raspi-config
```

Select `Interfacing Options, P1 Camera, and enable it. Reboot.`

At this point, the entire assembly should be `mobile`. Power Raspberry Pi from the provided external `battery`. Make sure iRobot's charger is not connected to it, and the system is `free` to move.

To test the entire system, SSH into RPi and execute:

```
cd ~
wget https://courses engr illinois edu/cs424/fa2016/mp/irobot-example.tar.gz
```
If the compilation succeeds, run the program \texttt{./robotest}.

The program will at first initialize camera, robot, etc. Once ready it will send a command to the robot to continuously stream (1) Bump and Wheel Drop sensor, (2) Wall Signal, (3) Buttons. After that, it will command the robot to drive straight at 200 mm/s. Then the robot will continuously look for bumps (with 100ms sleep in between). If the robot has bumped, it will back up straight, rotate, and continue running. At every time, robotest will also print at the console what it is doing. If the robot has seen a "wall", it will take a photo using the raspberry pi camera and save it to a file named \texttt{irobot_image.jpg}. If you are using a Mac or a Linux machine, you can copy the photo to your computer using the following command. If you are using windows, you can use WinSCP or similar programs to transfer the photos (suppose your RPi's wlan0 IP is 10.194.102.108):

\begin{verbatim}
scp pi@10.194.102.108::~/irobot-example/irobot_image.jpg ./
\end{verbatim}

Note the location of the wall sensor. The sensor works by transmitting a signal and measuring the strength of the received signal. This type of positioning allows it to detect a wall that is on the side. You can artificially check the wall sensor by a bringing a dark colored paper near it or taking it away.

If the "Advance" button is pressed, the robot will start rotating in place. Subsequent presses of the advance button results in reversing the direction of rotation. It will also change the colors of LED. Study the code to make sure you understand it. The program will continue running in the aforementioned manner until the play button is pressed. Once the play button is pressed, the robot will stop and the robotest program will exit gracefully. You can also use the power button to turn off the iRobot at any time, but the program robotest does not detect such event and as a result it will continue running. But iRobot will not respond as the power has been turned off.