

```
if(parameters.contains("name")){
    hql += " and p.name = :name";
}
if(parameters.contains("age")){
    hql += " and p.age = :age";
}
TypedQuery<Person> query = em.createQuery(hql);
if(parameters.contains("name")){
    query.setParameter("name", values[0].valueOf());
}
if(parameters.contains("age")){
    query.setParameter("age", Integer.valueOf(values[1].valueOf()));
}
```

## Open Source Rover

### Software Install Instructions



**Jet Propulsion Laboratory**  
California Institute of Technology

## Contents

<b>1</b>	<b>Downloading the App</b>	<b>3</b>
<b>2</b>	<b>Setting up the Raspberry Pi 3</b>	<b>3</b>
2.1	Getting the Pi online . . . . .	3
2.1.1	Installing a new OS on the Pi . . . . .	3
2.1.2	Getting the Rover code: . . . . .	3
2.1.3	Installing packages . . . . .	4
2.2	Setting up Bluetooth . . . . .	5
2.2.1	Naming the Bluetooth Device . . . . .	6
2.2.2	Pairing Bluetooth from Commandline . . . . .	6
2.3	Init Script . . . . .	7
2.4	Setting up serial communication . . . . .	8

# **1 Downloading the App**

Need android app install information

# **2 Setting up the Raspberry Pi 3**

In this section we'll go over setting up the Raspberry Pi and how to get all the code to run the rover going, as well as setting up the bluetooth pairing from the android device.

## **2.1 Getting the Pi online**

### **2.1.1 Installing a new OS on the Pi**

For instructions on getting the raspberry pi OS downloaded and installed visit the following link:

- <https://www.raspberrypi.org/documentation/installation/noobs.md>

### **2.1.2 Getting the Rover code:**

The following are the instructions on how to get the rover code for the beta groups, this will change for the final release.

- Download id\_rsa text file from the google drive, and move it to /.ssh on
  - On a Raspberry pi open up a terminal (ctrl + alt + t) and type:  
– cp /home/pi/Downloads/id\_rsa /.ssh
- Open up a terminal (ctrl + alt + t) and type the following commands:
  - cd /home/pi/Desktop
  - mkdir osr
  - cd osr

- git init
- git pull git@github.com:ericjunkins/OSR.git
- if prompted type "yes"
- Passkey is: rover

### 2.1.3 Installing packages

Below are the packages you will need to use Bluetooth communication through the raspberry pi. Open up a terminal on the Raspberry Pi (**ctl + alt + t**) and type the following commands:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
sudo apt-get install python-bluez
```

```
sudo apt-get install python-pip python-dev ipython
```

```
sudo apt-get install bluetooth libbluetooth-dev
```

```
sudo pip install pybluez
```

```
sudo systemctl start bluetooth
```

```
sudo nano /lib/systemd/system/bluetooth.service
```

- Add "-C" after 'bluetoothd', the file should look like the following:

```
[Unit]  
Description=Bluetooth service  
Documentation=man:bluetoothd(8)  
ConditionPathIsDirectory=/sys/class/bluetooth
```

```
[Service]
Type=dbus
BusName=org.bluez
ExecStart=/usr/lib/bluetooth/bluetoothd -C
NotifyAccess=main
#WatchdogSec=10
#Restart=on-failure
CapabilityBoundingSet=CAP_NET_ADMIN CAP_NET_BIND_SERVICE
LimitNPROC=1
ProtectHome=true
ProtectSystem=full
```

```
[Install]
```

```
WantedBy=bluetooth.target
Alias=dbus-org.bluez.service
```

– exit and save (ctrl + x, y, enter)

```
sudo reboot
```

```
sudo sdptool add SP
```

## 2.2 Setting up Bluetooth

There are a few ways to set up the Bluetooth pairing with your android device. For additional help you can follow steps

- <https://www.cnet.com/how-to/how-to-setup-bluetooth-on-a-raspberry-pi-3/>

### 2.2.1 Naming the Bluetooth Device

Create a file that will name the bluetooth device your desired name. **\*NOTE\*** This name must also be updated in the Android App such that the App finds the correct device. First on the RPi type `sudo nano /etc/machine-info` and name your device by the following, where `deviceName` is what you want the name of your Raspberry Pi to be:

```
PRETTY_HOSTNAME=deviceName
```

Now in `sudo nano /etc/bluetooth/main.conf` Uncomment the "Name=" line and add your `deviceName`. The first couple lines should read:

```
[General]
```

```
# Default adapter name  
# Defaults to 'BlueZ X.YZ'  
Name = deviceName  
  
# Default device class. Only the major and minor device class bits are  
# considered. Defaults to '0x000000'.  
#Class = 0x000100
```

### 2.2.2 Pairing Bluetooth from Commandline

We're now ready to pair the device with your android device. Have your android phone discoverable and searching for Bluetooth (for individual phones look up instructions if necessary to do this)

`sudo bluetoothctl` and do the following commands:

```
agent on
```

```
default-agent
```

scan on

Once you find your device add it from its address, pair XX:XX:XX:XX:XX  
and then type yes

You'll be prompted on the phone asking to accept the pairing, select okay

Type quit to exit the bluetoothctl

### 2.3 Init Script

We will want the code on the rover to run when the pi boots up, so we will add that to an init script, which on Raspberry Pi can be found at /etc/rc.local. In a terminal type sudo nano /etc/rc.local and then add the following lines after the if statement and before exit, it should look like the following.

```
#!/bin/sh -e

#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
```

```
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi

cd /home/pi/Desktop
sleep 10
sudo python -m osr.rover.main &
python -m osr.led.screen &

exit 0
```

## 2.4 Setting up serial communication

In this project we will be using serial communication to talk to the motor controllers. This is a communication protocol that describes how bits of information will be transferred between one device and another. More information on this can be found at:

- <https://learn.sparkfun.com/tutorials/serial-communication>

Do the following to setup/enable the serial communication for the RoboClaws:

```
sudo raspi-config
```

*Select -> InterfaceOptions -> Serial*

- Would you like a login shell to be accessible over serial? – > No
- Would you like the serial port hardware to be enabled? – > Yes
- Would you like to reboot now? – > Yes

```
ls -l /dev/serial*
```

- Make sure that serial0 -> ttyS0

```
sudo nano /boot/cmdline.txt
```

- Edit **ONLY** the part with "console = ....", change it to be "console=tty1" and remove any other instance where it references console. The first bit of that line looks like the following:

```
dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p7\$
```

Here is a link to help with this process if necessary.

- <https://spellfoundry.com/2016/05/29/configuring-gpio-serial-port-raspbian-jessie-including-pi-3/>.

Once this is all done then reboot the pi.

```
sudo reboot
```

There are two files in test\_files that will allow you to test the serial connection, use a jumper wire between pin 14 and 15 (see GPIO pinout if necessary). In two different terminal windows navigate to the desktop and then run the test files by:

```
python -m osr.test_files.serial_write
```

```
python -m osr.test_files.serial_read
```

If the serial communication is working properly then in one terminal you will be printing out a counting number, and in the other you will be reading that number over the serial port.