KnowThings

Hackathon No. 2

Your Mission

- Connect LIDAR to a Raspberry Pi
- Report its data to an OM2M server
- Virtualize it

SETUP: SD Card Network Settings

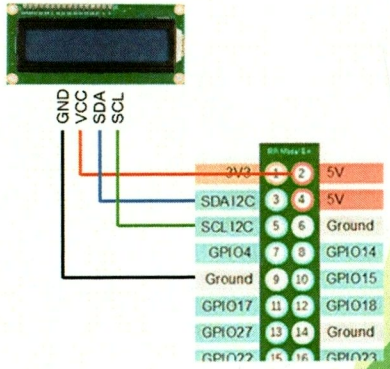
- Add wpa_supplicant.conf to the boot partition.
  - Use “KnowThings_717E” or “KnowThings_1CB2” or “knowthingsblue”

```
update_config=1

network={
    ssid="KnowThings_717E"      (or KnowThings_1CB2)
    psk="knowthings"
    scan_ssid=1
}
```

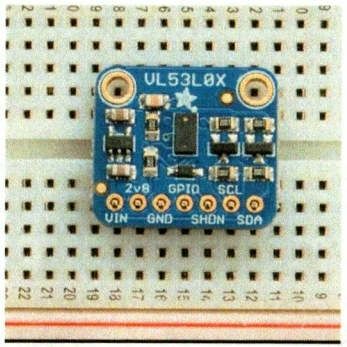

SETUP: KS1602 I2C LCD Display

- Vcc to RPi pin 2 (5VDC)
- GND to RPi pin 9
- SDA (I2C Data) to RPi pin 4
- SCL (I2C Clock) to RPi pin 5



VL53L0X Time of Flight ranging sensor

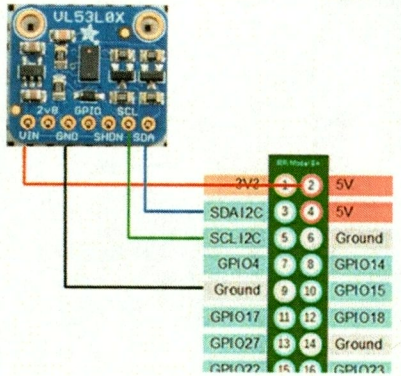
- VIN: 3-5VDC in
- 2v8: 2.8VDC@100ma supply
- GND
- GPIO: data-ready
- SHDN: shutdown (3-5V ok)
- SCL: I2C Clock
- SDA: I2C Data



https://learn.adafruit.com/adafruit-vl53l0x-micro-lidar-distance-sensor-breakout

Connect to the RPi

- VIN to RPi pin 2 (or 1)
- GND to RPi pin 9 (or 6)
- SCL to RPi pin 5
- SDA to RPi pin 4



https://github.com/johnbryanmoore/VL53L0X_rasp_python

Read the Sensor (sample code)

```
$ ssh pi@192.168.xxx.xxx
pi@192.168.xxx.xxx's password: raspberry

$ cd sens/vl53l0x/
$ ls
core_vl53l0x.py  loop_vl53l0x.py
```


Downloading the OM2M Server

- Install Java 8 (Java 10 doesn't work)
  http://www.oracle.com/technetwork/java/javase/downloads
- Download OM2M server from http://wiki.eclipse.org/OM2M/Download
  - Unzip the download someplace like /hackathon/om2mserver
- Download the OM2M monitor jar:
  https://www.laas.fr/projects/IOT/sites/www.laas.fr.projects.IOT/files/u77/monitor.jar

Starting the OM2M Server and Gateway

- Go to /in-cse and execute start.bat or start.sh
- Go to /mn-cse and execute start.bar or start.sh
- Launch the Monitor: java -jar monitor.jar
- localhost:8080/webpage (admin/admin)

- REST API: http://wiki.eclipse.org/OM2M/one/REST_API

Parent URL: http://wiki.eclipse.org/OM2M/one

Register Your Sensor

- POST http://localhost:8080/~/in-cse

```
X-M2M-Origin: admin:admin
Content-Type: application/json;ty=2

{
   "m2m:ae":
   {
     "api": "app-sensor",
     "rr": "false",
     "lbl": ["Type/sensor", "Category/distance",
"Location/hackathon"],
     "rn": "VL53L0X"
   }
}
```


Sending Data

- Create a Data Container
- Then log data
- Optionally, the Monitor can listen for data events

Extra Credit: Virtualize

- https://knowthings.io/eap
  - Early Adopter download (yeah, there's a registration step)
- Documentation: https://knowthings.gitbook.io/ktug/v/0.3.0/
- Process:
  - Capture packets
  - Create virtual device
  - "Play back" device
- Result: OM2M server should act as if real sensor data were being transmitted

Contact

- Thank you for participating!
- I'm Rob.
- Robert.Eaglestone@ca.com
- birdrock@knowthings.io

Appendix: OM2M get sensor list

- Get a list of registered sensors.
- GET http://localhost:8080/~/in-cse?fu=1&lbl=Type/sensor

```
X-M2M-Origin: admin:admin
Accept: application/json
```

```
(no content)
```


Appendix: OM2M create data container

- POST http://localhost:8080/~/in-cse/in-name/VL53L0X

```
X-M2M-Origin: admin:admin
Content-Type: application/json;ty=3
```

```
{
   "m2m:cnt":
   {
      "rn": "MY_DATA"
   }
}
```


Appendix: OM2M log data

- POST http://localhost:8080/~/in-cse/in-name/VL53L0X/MY_DATA

```
X-M2M-Origin: admin:admin
Content-Type: application/json;ty=4
```

```
{
   "m2m:cin":
   {
      "rn": "my_data_instance_1",
      "cnf": "application/json",
      "con": "{\"distance\":214,\"timestamp\":1517912099}"
   }
}
```

Appendix: Subscribe to OM2M container

- POST http://localhost:8080/~/in-cse/in-name/VL53L0X/MY_DATA

```
X-M2M-Origin: admin:admin
Content-Type: application/json;ty=23

{  "m2m:sub": {
       "rn": "subscribe_test",
       "nu": ["http://localhost:1400/monitor"],
       "nct": 2,
       "enc": {
        "net": 3
       }}}
```


Appendix: OM2M data retrieval

- GET http://localhost:8080/~/in-cse/in-name/VL53L0X/MY_DATA/my_data_instance_1

```
X-M2M-Origin: admin:admin
Accept: application/json

(no content)
```