|  |
| --- |
| RFID Application Development Lab  Breno Brezinski, Daniel Chau, Francis Agyapong, Sergio Diaz, Jediah Dizon  April 14, 2016 |

|  |
| --- |
|  |
| EnviroSense: Environmental Sensor Platform |
| RaspberryPi Subsystem User Manual |

Table of Contents

[Preface 3](#_Toc448389280)

[Purpose of Document 3](#_Toc448389281)

[System Overview 4](#_Toc448389282)

[Important Initial Notes 5](#_Toc448389283)

[Adding a New Sensor 5](#_Toc448389284)

[System Startup 5](#_Toc448389285)

[Config.js 5](#_Toc448389286)

[Project Structure 6](#_Toc448389287)

[ libs 6](#_Toc448389288)

[o C 6](#_Toc448389289)

[o Python 6](#_Toc448389290)

[ Server 7](#_Toc448389291)

[o APIserver.js 7](#_Toc448389292)

[o DBserver.js 7](#_Toc448389293)

[ debug.log 7](#_Toc448389294)

[ config.js 7](#_Toc448389295)

[ mainScript.sh 7](#_Toc448389296)

[ package.json 7](#_Toc448389297)

[ testScript.sh 7](#_Toc448389298)

[ util.js 7](#_Toc448389299)

[Setting up a new Pi 7](#_Toc448389300)

[If you already have a working Pi: 7](#_Toc448389301)

[If you don’t have a working Pi: 7](#_Toc448389302)

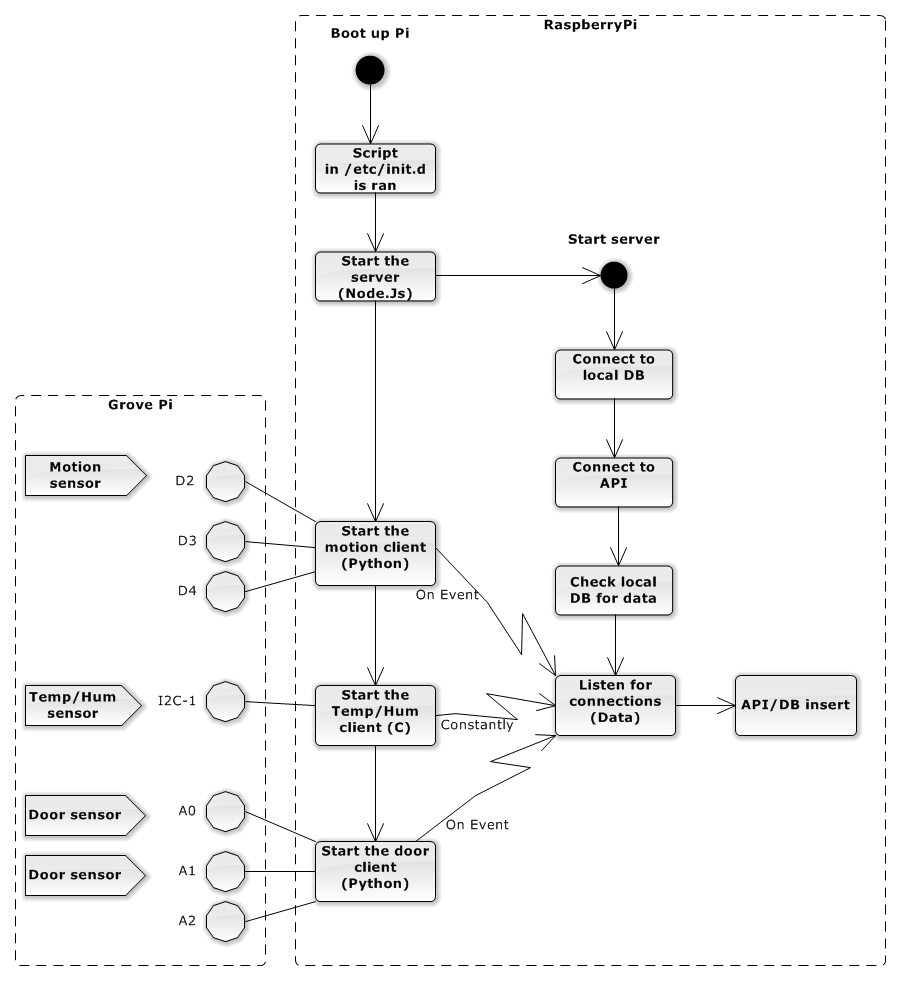
[Installation Map 9](#_Toc448389303)

# Preface

## Purpose of Document

This document presents an overview of the RaspberryPi subsystem, as well as details about the project structure, important things to know about it, and instructions on how to manage the system.

# System Overview



# Important Initial Notes

* Right now this system is designed
* The Raspberry Pi system was designed to work exclusively with the following GrovePi sensors:
  + HDC1000 Temperature & Humidity Sensor (Max 1 per Pi)
  + PIR Motion Sensor (Max 3 per Pi)
  + Analog door sensors (Max 3 per Pi)
* The relationship between sensors and ports on the GrovePi must be respected in order for it to work properly. These are:
  + Door sensors: A0, A1, and/or A2
  + PIR Motion Sensor: D2, D3, and/or D4
  + HDC1000 (Temperature and Humidity sensor): I2c-1
* HDC1000 sensors (Temperature and Humidity) must be divided in 2 different rows in the main DB, therefore with 2 different ids. When inserting this type of sensor to the DB, they MUST be inserted in order (Temperature first, then Humidity. e.g. Temperature id=13, Humidity id=14)

# Adding a New Sensor

Before adding a new sensor, make sure to read through the Important Initial Notes section.

* As soon as a sensor is physically installed on a GrovePi, the relationship between this sensor (that must exist in the main DB) and the port number must be specified in the config.js (configuration file of the system in the Pi).

Example:

* + - We have one HDC1000 (Temp/Hum sensor) with id 13 and 14
    - We have two door sensors with ids 16 and 23 respectively
    - Door sensors are installed on the ports A0 and A2
    - Temp/Hum sensor is installed on the I2C-1

In this case, the following information must exist in the config.js:

config.THIS\_PI.PORT\_SENSORID = {'I2C-1': '13,14',

'A0': '16',

'A1': '23'};

# System Startup

Every time the Pi boots up, an initialization script is executed. This script is located at /etc/init.d/ (e.g. /etc/init.d/mainScript.sh)

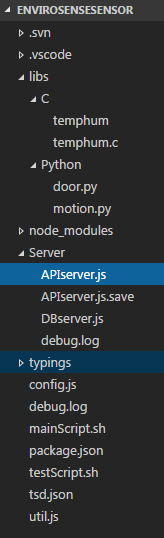
This script is responsible for starting the main server in the Pi (APIserver.js – NodeJs) and respective clients responsible for collecting data from the sensors and sending to the APIserver. The clients started are: (motion.py – responsible for reading motion data, temphum – responsible for temperature and humidity reading, and door.py – responsible for door reading).

# Config.js

This is the file that contains all custom information in order for this system to work properly. Some of the information listed there that needs to be changed depending on the context:

* THIS\_PI.PORT\_SENSORID: Specifies the relationship between the sensor installed (its id in the DB) and the port that it is connected to.
* netServer.PORT: The port number that the APIServer within this system will listen for connections coming from the local clients (the ones collecting data from the sensors).
* API.LOGINPAGE: The login page and port number in the main server to which the API connection shall be made (GET).
* API.NEWDATA: The page and port number in the main server to which new data will be sent (POST).
* API.EMAIL and API.PASSWORD: The credentials used to connect the Pi to the main server.
* DB.\*: Credentials used to connect to the local MySQL database (the one hosted in the Pi).

# Project Structure



* libs -> All the clients responsible for reading from sensors and sending data to the APIserver
  + C -> Clients written in C
    - temphum.c -> HDC1000 sensor reader. Currently reads only from port I2C-1
  + Python -> Clients written in Python
    - door.py -> Analog port reader. Reads from ports A0, A1, and A2 and sends the data to the APIserver
    - motion.py -> Digital port reader. Reads from ports D2, D3, and D4 and sends the data to the APIserver
* Server -> Contains the main server that listens for data from clients to send them to the main server through a REST API
  + APIserver.js -> Main server that maintains no direct connection to the main server DB
  + DBserver.js -> Direct DB connection version of the serve
* debug.log -> Any errors are written in here
* config.js -> Custom configuration information (as main server address and port, relationship between sensors and port numbers in the Pi, etc.)
* mainScript.sh -> Main script that must be ran at startup
* package.json -> Project dependencies
* testScript.sh -> Same as mainScript.sh, but output results to terminal and can be aborted easily
* util.js -> Contains useful functions used around the application

# Setting up a new Pi

## If you already have a working Pi:

1. Shutdown the Pi
2. Remove the MicroSD card from the Pi and use a SD adapter to insert to a PC
3. Read the contents of the MicroSD to a .img file (Win32DiskImager)
4. Insert a new MicroSD
5. Write the .img to the this new MicroSD (Win32DiskImager)
6. Insert the new MicroSD to the new Pi and customize the config.js

## If you don’t have a working Pi:

1. Download Raspbian Jessy Lite (<https://www.raspberrypi.org/downloads/raspbian/>)
2. Insert a new MicroSD to a PC
3. Write the .img downloaded to the MicroSD (Win32DiskImager)
4. Insert the MicroSD in the Pi (No GrovePi board at this point)
5. Connect the Ethernet cable to the Pi
6. Connect the camera (if any)
7. Connect the power supply (it should power up and connect to the network)
8. Find out the Pi IP (Advanced IP Scanner)
9. SSH into it
10. Change time on the Pi (sudo dpkg-reconfigure tzdata)
11. Enable camera support/expand filesystem/enable I2c:
    1. sudo raspi-config
    2. Go to the camera option and select enable
    3. In the main menu, go to the camera option and enable it
    4. In the main menu, go to the Expand Filesystem option and enable it
    5. In the main menu, go Advanced Options and enable I2C
    6. Exit and reboot the Pi
12. sudo apt-get update
13. sudo apt-get upgrade
14. Add the package repository (apt.adafruit.com) to the Pi:
    1. curl -sLS https://apt.adafruit.com/add | sudo bash
15. Install node: sudo apt-get install node
16. Install npm: sudo apt-get install npm
17. Install node-grovepi: sudo npm install –g node-grovepi
18. Install MySQL:
    1. sudo apt-get update
    2. sudo apt-get install mysql-server
19. Optimize MySQL:
    1. sudo mv /etc/mysql/my.cnf /etc/mysql/my.cnf.bak
    2. sudo cp /usr/share/doc/mysql-server-5.5/examples/my-small.cnf /etc/mysql/my.cnf
    3. sudo nano /etc/mysql/my.cnf
       1. Edit the max\_allowed\_packet to 8M
    4. sudo service mysql restart
20. Run the database scripts for the Pi (no need for sample data)
21. sudo apt-get install git
22. Go to /home/pi/Desktop (Make sure there is a Desktop directory. If there isn’t, create it)
23. sudo git clone https://github.com/DexterInd/GrovePi
24. Go to /home/pi/Desktop/GrovePi/Script
25. Make the install.sh bash script executable: sudo chmod +x install.sh
26. Start the script: sudo ./install.sh
27. Pi will automatically restart when installation is finished
28. When the Pi powers off, stack the Grove Pi on the top of the Pi
29. Power it on
30. Make sure Pi can detect Grove Pi: sudo i2cdetect –y 1 (If you see a “04”, means it worked).
31. Import EnvirosenseSensor project to /Desktop
32. Install project dependencies:
    1. Go to /EnvirosenseSensor
    2. npm install
33. Give permissions to user pi: sudo chown –R pi ~/Desktop
34. Make the main script un on boot:
    1. Copy the mainScript.sh (located in /home/pi/Desktop/EnvirosenseSensor) to /etc/init.d
    2. Configure the init system to run the script at startup:
       1. sudo update-rc.d mainScript.sh defaults (Ignore warnings)
    3. Reboot and check if it works

# Installation Map

