Networked Household Sensor System

User Guide

University of Colorado – Colorado Springs/Team PANDa

December 11, 2015

TABLE OF CONTENTS

1.0 SYSTEM OVERVIEW

The Networked Household Sensor System uses multiple sensors to gather data about environmental factors within a home. The system consists of on board connectors that allow for temperature, humidity, pressure, light, audio, water, volatile organic compound, and motion sensors to be attached to the board. Each connector allows each sensor to be quickly connected/ disconnected; therefore, if a sensor goes bad, it can easily be replaced. Data collected from each sensor is done so through the use of a Raspberry Pi Model B+ which can then be used to display such data on a webpage.

2.0 REFERENCE DOCUMENTS

74HC4052, Toshiba Dual 4-Channel Analog Multiplexer Data Sheet

ADS115, Texas Instruments 16-Bit Analog-to-Digital Converter Data Sheet

BMP180, Bosch Digital Pressure Sensor Data Sheet

HTU21D, TE Connectivity Humidity/ Temperature Sensor, Data Sheet

MQ-135, Gas Sensor Data Sheet

TSL2561, Taos Light-to-Digital Converter Data Sheet

LMV324M, Fairchild Semiconductor Operational Amplifier Data Sheet

MCP23008, Microchip 8-Bit I/O Expander Data Sheet

P82B715, Texas Instruments I2C Bus Extender Data Sheet

Raspberry Pi – B+ Data Sheet

3.0 REQUIREMENTS

The Networked Household sensor system must have the ability to "quick change" sensors provided they fail at any point in time. The data collected from each sensor also needs to be gathered and displayed graphically on a webpage, allowing it to be accessed on an Operating System.

3.1 Sensors

The Networked Household Sensor System must have the ability to sense various elemental factors within a room. These factors include temperature, humidity, pressure, light, air quality (volatile organic compounds), audio, water (leakage). The section below describes requirements for each sensor such as the interface used to communicate between the sensor and the microcontroller, the polling time, and the accuracy of each sensor.

3.1.1 Temperature

The temperature sensor needs to be polled every 1 minute and have an accuracy to about 1$°^{C}$. The sensor needs to communicate with the microcontroller using I^2C.

3.1.2 Humidity

The humidity sensor needs to be polled every 5-10 minutes with an accuracy of 2%RH. The sensor needs to communicate with the microcontroller using I^2C.

3.1.3 Pressure

The pressure sensor needs to poll every 5 minutes with an accuracy of 0.5 kPa. The sensor needs to communicate with the microcontroller using I^2C.

3.1.4 Light

The light sensor needs to poll every 30 seconds and have an accuracy of about 100 lumens. The sensor needs to communicate with the microcontroller using I^2C.

3.1.5 Air Quality

The air quality sensor needs to poll every 5 minutes and be accurate to about 10 ppm NH_3. This sensor is analog and will communicate to the microcontroller using an Analog to Digital Converter.

3.1.6 Audio

The audio sensor needs to be taken as an average over 10 minutes as well as be interrupt driven. The accuracy of this sensor needs to be to about 3dB. This sensor, like the air quality sensor, is analog and will communicate with the microcontroller after being sent to the analog to digital converter.

3.1.7 Water

The waster sensor needs to only be interrupt driven and must sense anytime water comes into contact with the sensor. The sensor needs to communicate with the microcontroller using I^2C.

3.2 Microcontroller

The Microcontroller must have Ethernet capabilities in order to communicate with the database so that data taken from the sensors can be inserted into the database and then displayed in graphical from on a webpage. Must be able to supply 3.3V to sensors that run at the 3.3V with minimal power consumption. The microcontroller must have I^2C protocol capabilities as well in order to interface with the sensors and receive data.

4.0 DESIGN

This section describes the design of the overall system including how the system is powered, the main board design, sensors used, and how the closet board connects with the main board.

4.1 System

The entire Networked Household Sensor System consists of a Raspberry Pi B+, a Main Sensor Board, and one or more Closet Boards. Figure 1 shows a high-level block diagram of the system and Figure 2 shows a high-level block diagram of sensors connecting to the sensor boards.

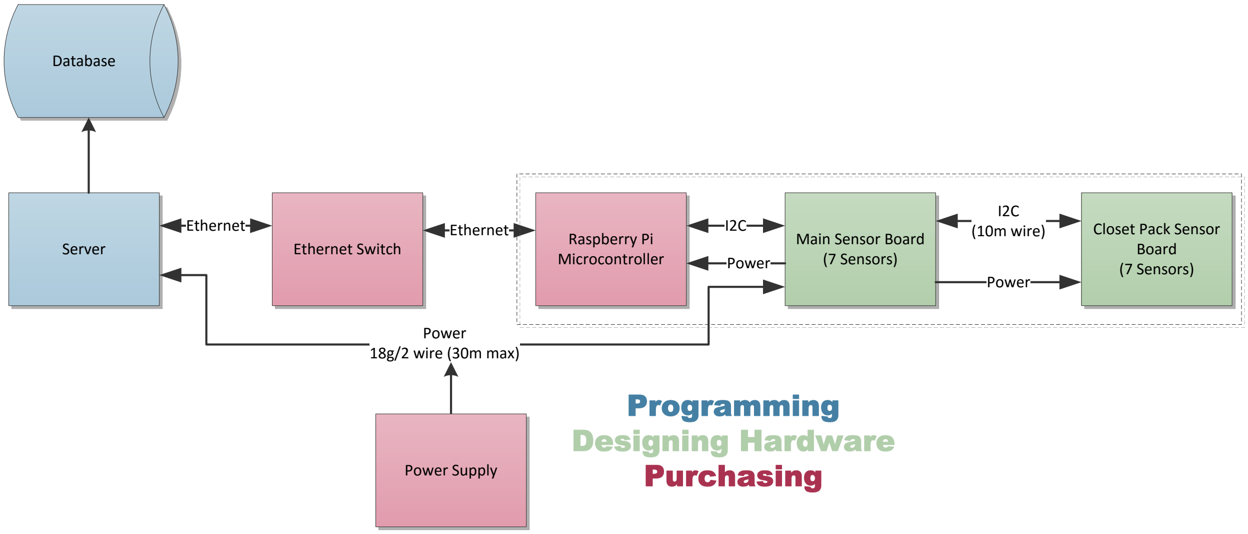


Figure 1: System block diagram of the Networked Household Sensor System.

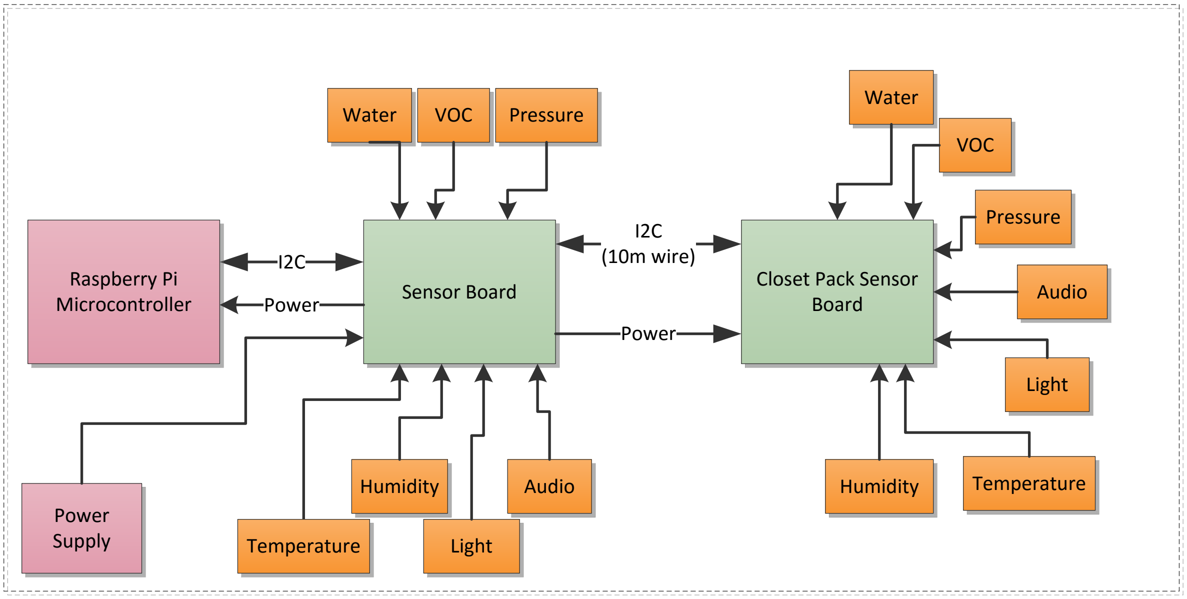


Figure 2: Block diagram of sensors connecting to the Main Sensor Board and the Closet Board.

4.1.1 Power

From Figure 1, the entire system is powered by a single power supply which will run power to both the server of the system and the Main Senor Board. The Main sensor board contains an adjustable Synchronous Rectified Step-Down Converter which takes a power supply input of 4.76V – 23V* and reduces it down to 5V to power sensors which are powered at 5V (water, audio, and air quality) and the Raspberry Pi B+. The rest of the sensors and IC's on board each run at 3.3V which is supplied through a 3.3V pin the Raspberry Pi B+.

***Note that the regulators used on the sensor boards will need to be adjusted to the correct output voltage**.

4.1.2 Sensors

All but the Air Quality sensor is mounted on its own breakout board and connects to the sensor boards through pins that plug into connectors mounted on the board. The Air Quality sensor connects to the sensor board through MQ series sockets mounted on the board. Due to the Air Quality sensor not having any polarity any MQ series sensor could work with the socket. Suggested sensors to use with the board are listed in Table 1.

Sensor Type	**Suggested Sensor**
Temperature/ Humidity	HTU21D
Pressure	BMP180
Light	TSL2561
Air Quality	MQ-135
Audio	LMV324M – Sparkfun Sound Detector

Table 1: Suggested sensors to be used on Sensor Boards.

4.2 Main Board

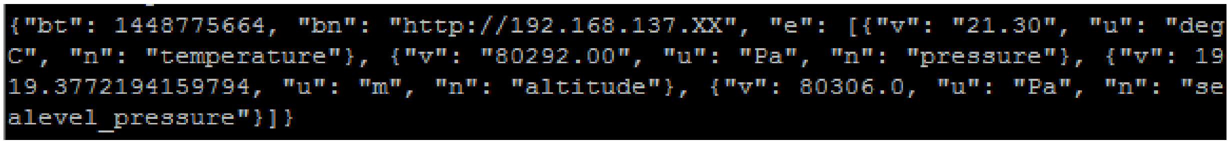
The main sensor board is designed to work with the Raspberry Pi B+ and allows for three closet boards to connect to the main board to use in other rooms while still only using one Raspberry Pi. The board is also designed to house multiple connectors around the board to allow each sensor to be connected and disconnected when needed. Section 6.3 describes the sensor pin assignments. This allows for each sensor to be quickly replaced if a sensor fails for any reason. It also allows room for expansion beyond using only the 7 sensors suggested as well as connections for Bluetooth, a Neopixel LED, EEPROM, and EM Micros' EM7180 SENtral Fusion Platform if desired.

5.0 SOFTWARE

This section describes the software that was used to take data from sensors, insert the data into the database, security, and the database itself. All software is written in either Python, PHP, SQL, JavaScript, JSON, or HTML.

5.1 Sensor Firmware

The sensor firmware to control when each sensor takes data is written in Python. Each sensor has its own corresponding I^2C library which is used to collect data that is parsed into JSON and then stored into a variable, as seen in Figure 3.



```
{"bt": 1448775664, "bn": "http://192.168.137.XX", "e": [{"v": "21.30", "u": "deg
C", "n": "temperature"}, {"v": "80292.00", "u": "Pa", "n": "pressure"}, {"v": 19
19.3772194159794, "u": "m", "n": "altitude"}, {"v": 80306.0, "u": "Pa", "n": "se
alevel_pressure"}]}
```

Figure 3: JSON representation of sensor data collection.

Each variable value is then inserted into the Database using a Python SQL call. Immediately after, the program goes into a subroutine that polls each sensor as defined in the requirements, records the data obtained by the sensors, and then inserts the data into the Database. In addition to taking the sound sensor data as an average over 10 minutes, the sound sensor is handled as an interrupt. The program for the sound sensor constantly polls for a loud noise depending on the limit it is set to, in this case around 0.625 volts out of 2.5V. If the voltage detected by the sound sensor is higher than the limit, it gets captured and inserted into the Database.

5.2 Database

The database is built using a database management program, in this case PhpMyAdmin. All SQL tables are created manually within PhpMyAdmin. Each table has a unique ID, date, and 7 values for the sensors and after table creation, the data is inserted from the Python script collecting data. This is done automatically every 5 minutes unless a loud noise or water is detected. The initial set up for the database can be found under section 6.2.

5.3 Graphs

All of the data, database, and GUI are stored on the SD located on the Raspberry Pi B+. The GUI was created using PHP, JavaScript, and HTML. All of the sensor data is stored within a table and the graphs become visible once the JavaScript parses the data and creates the graph.

5.4 Security

A user log in has been implemented so only those with a valid username and password can view the collected data from the sensors.

6.0 <u>BASIC SETUP</u>

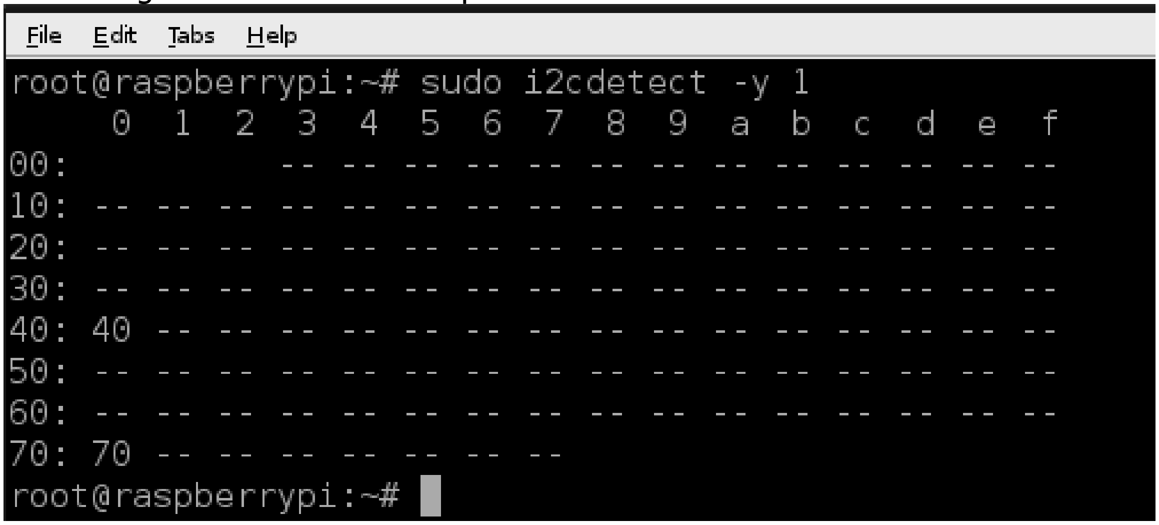
This section describes the basic set up of the Raspberry-Pi and how the sensors connect to the Networked Sensor Board.

6.1 Raspberry Pi – B+

- ❑ Download the Raspberry Pi image Raspibian from https://www.raspberrypi.org/downloads/ and follow the onsite instructions on how to write the image to an SD card (suggested size for SD card is 8GB or larger).
- ❑ Insert the SD card into the Raspberry Pi and plug the pi into internet using and Ethernet cable.
- ❑ Connect to the Raspberry Pi using SSH via Putty or plug in a keyboard, mouse, and monitor/TV.
- ❑ Log into the Raspberry Pi using default username and password:

  Login name: pi
  Password: raspberry
- ❑ Set up I^{2}C by following Adafruit's guide at https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/configuring-i2c.

- □ The I^{2}C setup can be tested after reboot by running “sudo i2cdetect –y 1”. Something like below should be printed to the terminal:



```
root@raspberrypi:~# sudo i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: 40 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: 70 -- -- -- -- -- -- --
root@raspberrypi:~#
```

Figure 4: Taken from Adafruit’s I^{2}C setup tutorial at
https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/configuring-i2c

- □ Next add a default user “pi” or any other name to the I^{2}C user group so it can run I^{2}C scripts without using sudo, run “sudo usermod –a –G i2c pi”. For a name other than “pi” just replace “pi” with chosen name.
- □ Next set up the Raspberry Pi date and time.

  Run “sudo raspi-config”
  Select “Internationalization Options”
  Select “Change Time Zone”
  Choose correct geographical area
  Choose correct city/region
  Reboot the Raspberry Pi.

6.2 Database Set Up

This section describes how to set up the database used to store collected sensor data. This set up is to be completed AFTER Raspbian has been installed and is running on the Raspberry Pi. All sensor logging code can be found at: https://github.com/panda4899.

6.2.1 Installing the Local Web Server

- □ Within the Raspberry Pi terminal run:
  - ○ **sudo apt-get update**
  - ○ **sudo apt-get install apache2 php5 libapache2-mod-php5**
- □ Override’s now need to be allowed, this is done by editing the 000-default file, do so by running the following command:
  - ○ **sudo nano /etc/apache2/sites-enabled/000-default**
- □ Now the edit the following lines:
  - ○ **Change “AllowOverride None” to “AllowOverride ALL”**
- □ To restart apache with the new settings execute:

  - **sudo service apache2 restart**
- □ At this point the site should be up and running. Now go to /var/ and change the permissions on www to make it writable. This can be done by entering the lines below into the terminal.
  - **cd /var/**
  - **sudo chmod 777 /www**
- □ Changing the permissions will now allow HTML pages to be uploaded to the site.
- □ To view the default page, open a browser and point to 192.168.xx.xx (the ip address of the Raspberry Pi).

6.2.2 Installing PhpMyAdmin

- □ From the Raspberry Pi terminal run:
  - **sudo apt-get install phpmyadmin**
- □ The package will begin installing and when prompted to choose which server to reconfigure automatically choose:
  - **apache2**
- □ Next configure PhpMyAdmin's database by choosing **<Yes>** when prompted.
- □ Now configure Apache to work with PhpMyAdmin by entering the command:
  - **nano /etc/apache2/apache2.conf**
- □ This will load the configuration file in Nano. Navigate to the bottom of the file and add the following line at the bottom:
  - **Include /etc/phpmyadmin/apache.conf**
- □ Save the file by pressing **CTRL + X** and enter **Y** when prompted to save.
- □ Restart Apache by entering:
  - **/etc/init.d/apache2 restart**
- □ At this point PhpMyAdmin will have been installed.
- □ To test, open a browser and point to 192.168.xx.xx (the ip address of the Raspberry Pi).

6.2.3 Build the Site

- □ Using SFTP or UNIX install all of the "WEBSITE" files into the new Apache server.

6.2.4 Create Tables

- □ Tables will need to be created for each sensor intended to be used. This can be done by using the following commands in PhpMyAdmin:
  - **USE DBNAME…;**
  - **CREATE TABLE TABLENAME…(UserID int (10) NOT NULL AUTO_INCREMENT, COLUMN1… varchar(225) NOT NULL, COLUMN2… varchar(255) NOT NULL, PRIMARY KEY (UserID))**

6.2.5 Inserting Values

- □ Once the database has been properly set up and is displaying correctly you can run:
  - **python BMP180.py** (or any python script of the sensors being used)
- □ On the command line you will see:

- **Data Committed**

- ❑ This will be written every two minutes to notify successful insertion.
- ❑ Refreshing the browser will show the new points on the graph.

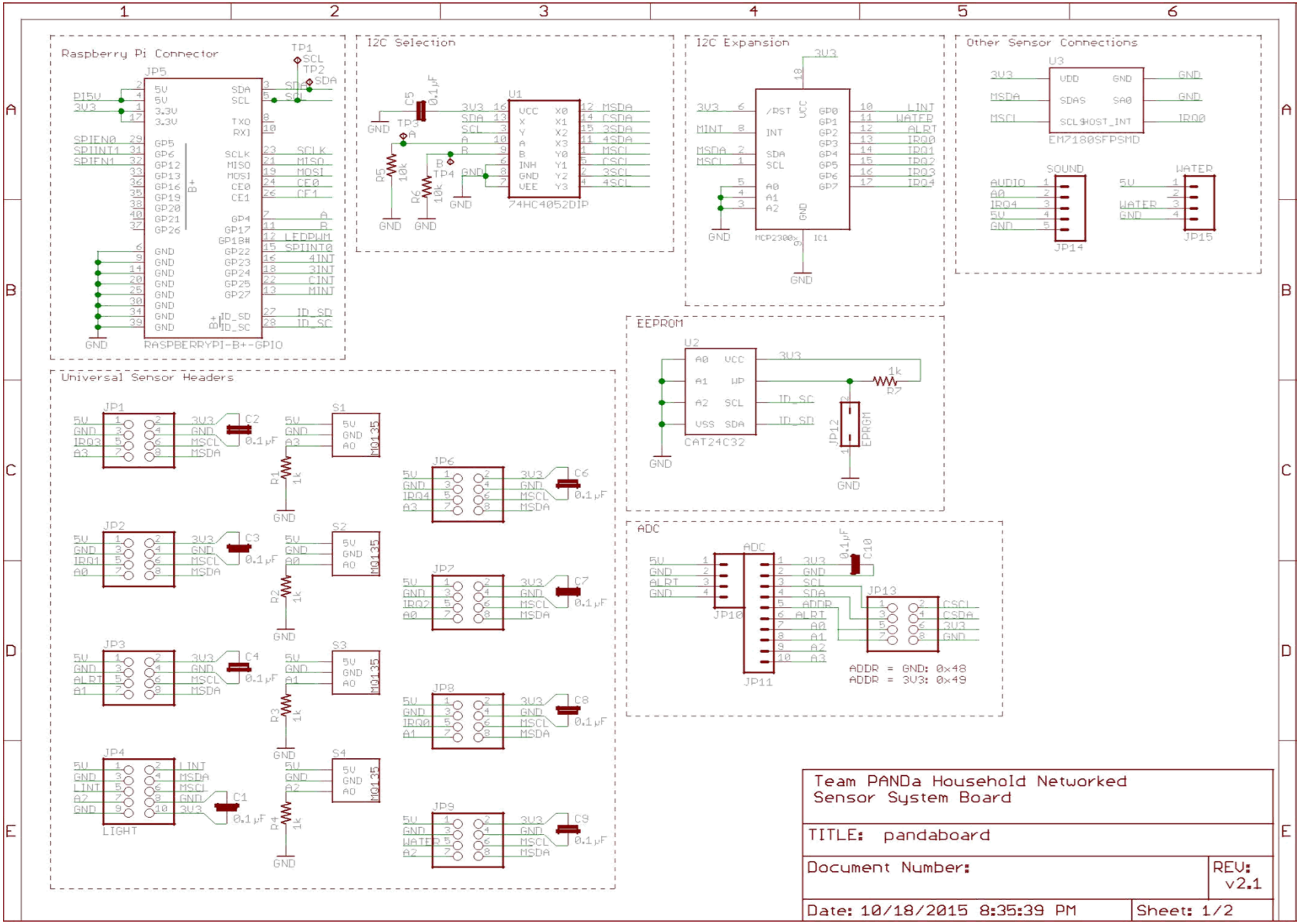
6.3 Board and Sensor Pin Assignments

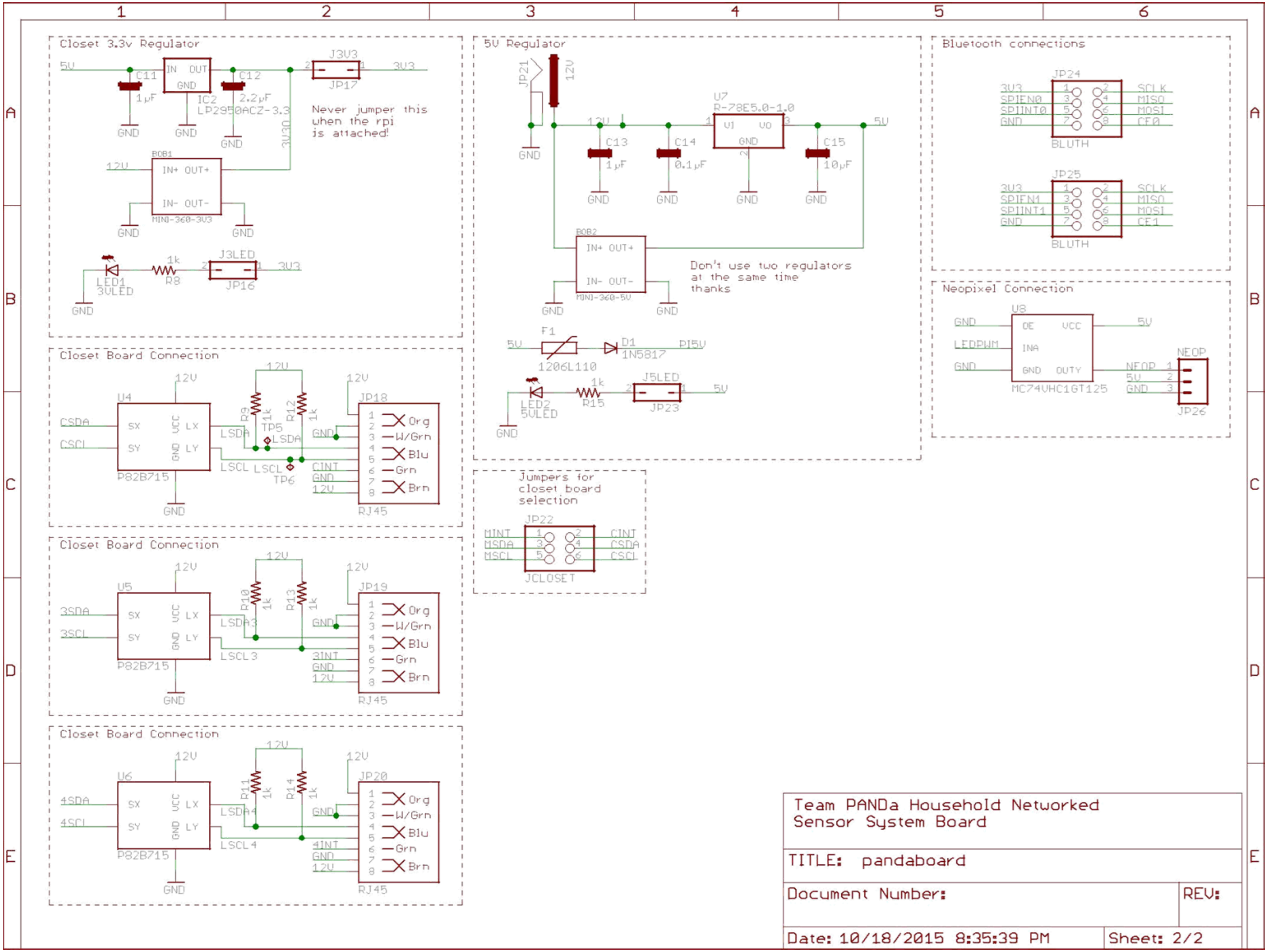
- ❑ Table 2 describes initial and basic pin set up of the Networked Household Sensor Board. Note this description is general due to the fact different sensors of each type can be used.

Pin Number	Description	Suggested Sensors To Use
JP0	Left row of pins is designed to work with analog sensors (preferably a water sensor) and the right row of pins is to be used with a sensor that works with I^{2}C Protocol.	HTU21D BMP180
JP1	Built to work with light sensors. Left row of pins with an analog light sensor and right row with and I^{2}C light sensor.	TSL2561
JP2, JP3	Left row of pins is designed to work with analog sensors and the right row of pins is to be used with a sensor that works with I^{2}C Protocol.	HTU21D BMP180
JP6, JP7, JP8, JP9	Left row of pins is to be used with a sensor that works with I^{2}C Protocol and the right row of pins is designed to work with analog sensors.	HTU21D BMP180
JP14	Designed to work with audio sensors.	LMV324M – Sparkfun Sound Detector
JP15	Designed to work with various water sensors.	
S0, S1, S2, S3	Designed to work with MQ Series sockets and any of the MQ Series Air Quality sensors.	MQ-135

Table 2: Sensor Pin Assignments with Suggested Sensor

7.0 SCHEMATICS





Team PANDa Household Networked
Sensor System Board

TITLE: pandaboard

Document Number:

REV:

Date: 10/18/2015 8:35:39 PM

Sheet: 2/2

8.0 LAYOUT

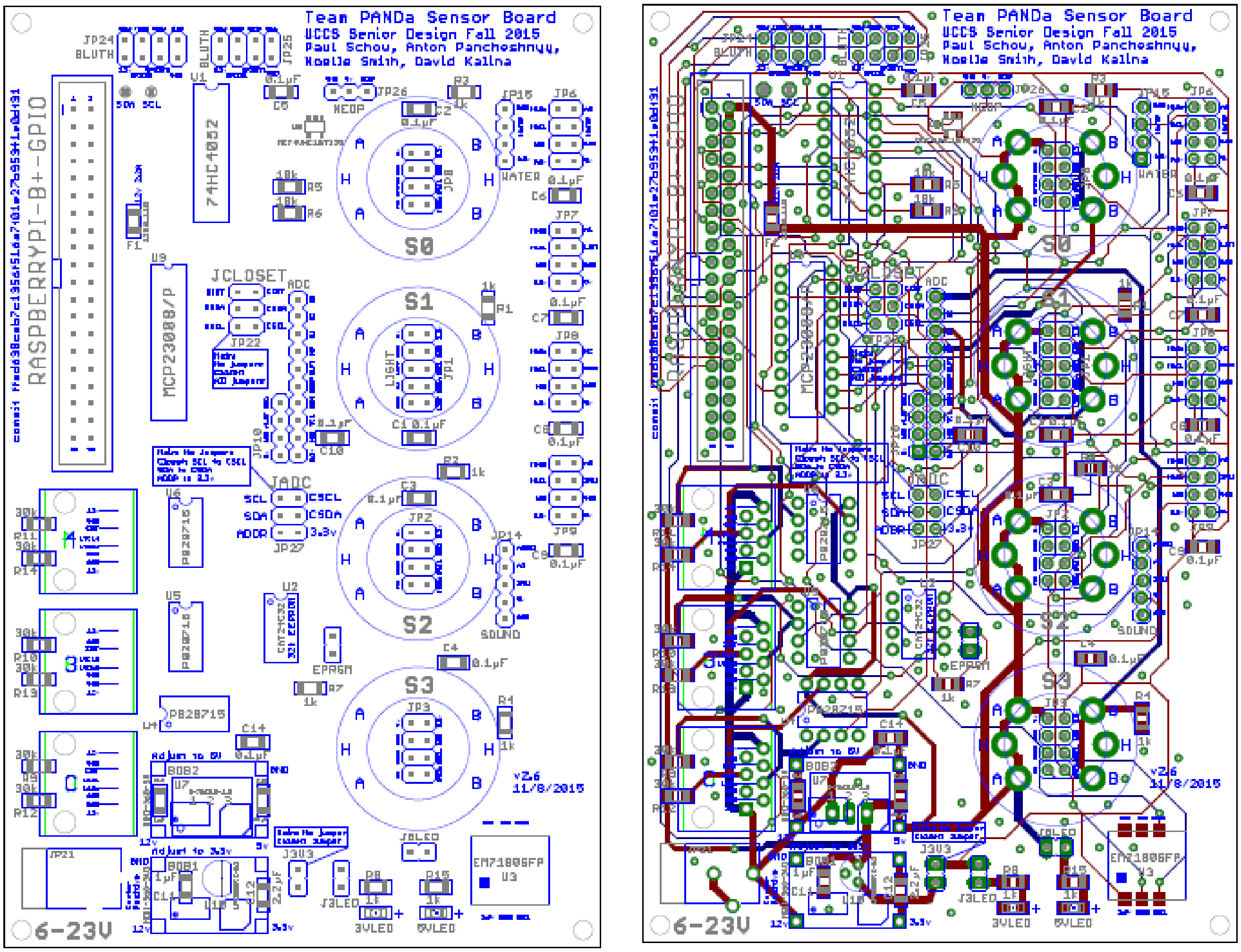


Figure 5: PCB Layout of the Networked Houshold Sensor Board

9.0 BILL OF MATERIALS

9.1 Main Board

Quantity	Reference	Manufacturer	Manufacturer Part Number	Description
3	U4, U5, U6	TEXAS INSTRUMENTS	P82B715P	I2C EXTENDER – IC REDRIVER I2C 1CH 8DIP
1	U1	TOSHIBA SEMICONDUCTOR AND STORAGE	TC74HC4052APF	ANALOG IC MUX/DEMUX DUAL 4X1 16DIP
1	U9	MICROCHIP TECHNOLOGY	MCP23008-E/P	IC I/O EXPANDER I2C 8B 18DIP
1	F1	LITTELFUSE INC (VA)	1206L110/12SLYR	FUSE – PTC RESETBLE 12V 1.1A LOW-R 1206
1	5VLED	STANLEY ELECTRIC CO (VA)	BR1101W-TR	LED RED CLEAR 1206 SMD
1	3VLED	STANLEY ELECTRIC CO (VA)	PG1101W-TR	LED GREEN CLEAR 1206 SMD
6	R1, R2, R3, R4, R8, R15	PANASONIC ELECTRONIC COMPONENTS (VA)	ERJ-8ENF1001V	RES SMD 1K OHM 1% 1/4W 1206
2	R5, R6	PANASONIC ELECTRONIC COMPONENTS (VA)	ERJ-8ENF1002V	RES SMD 10K OHM 1% 1/4W 1206
6	R9, R10, R11, R12, R13, R14	PANASONIC ELECTRONIC COMPONENTS (VA)	ERJ-8GEYJ303V	RES SMD 30K OHM 5% 1/4W 1206
11	C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C14	KEMET (VA)	C1206C104K5RAC7867	CAP CER 0.1UF 50V X7R 1206
1	BOB2	MONOLITHIC POWER SYSTEMS INC.	MP2307DN	IC REG BUCK ADJ 3A SYNC
3	4, 3, C	MOLEX, LLC	955032881	CONN MOD JACK 8P8C VERT UNSHLD
1	Raspberry Pi – B+ GPIO	ON SHORE TECHNOLOGY INC	302-S401	CONN HEADER VERT 40POS GOLD
2		ON SHORE TECHNOLOGY INC	101-406	CONN SOCKET IDC 40POS W/KEY GOLD
1	JP21	CUI INC	PJ-102AH	CONN PWR JACK 2.1X5.5MM HIGH CUR
4	JP0, JP6, JP7, JP8, JP9	SULLINS CONNECTOR SOLUTIONS	PPPC042LFBN-RC	CONN HEADER FMAL 8PS .1" DL GOLD
1	JP1	SULLINS CONNECTOR SOLUTIONS	PPPC052LFBN-RC	CONN HEADER FMAL 10PS.1" DL GOLD
1	JP14	SULLINS CONNECTOR SOLUTIONS	PPPC051LFBN-RC	CONN HEADER FEMALE 5POS .1" GOLD
2	JP10, JP15	SULLINS CONNECTOR SOLUTIONS	PPPC041LFBN-RC	CONN HEADER FEMALE 4POS .1" GOLD
1	ADC	SULLINS CONNECTOR SOLUTIONS	PPPC101LFBN-RC	CONN HEADER FMALE 10POS .1" GOLD
2	JP22, JP27	HARWIN INC	M20-9980346	DIL VERTICAL PC TAIL PIN HEADER
1	J5LED, J3LED, J3V3	SULLINS CONNECTOR SOLUTIONS	PEC36SAAN	CONN HEADER .100 SINGL STR 36POS
2	J5LED, J3LED	TE CONNECTIVITY AMP	382811-8	SHUNT ECON PHBR 5AU BLACK

9.2 Closet Board

Quantity	Reference	Manufacturer	Manufacturer Part Number	Description
1	U4	TEXAS INSTRUMENTS	P82B715P	I2C EXTENDER – IC REDRIVER I2C 1CH 8DIP
1	U9	MICROCHIP TECHNOLOGY	MCP23008-E/P	IC I/O EXPANDER I2C 8B 18DIP
1	5VLED	STANLEY ELECTRIC CO (VA)	BR1101W-TR	LED RED CLEAR 1206 SMD
1	3VLED	STANLEY ELECTRIC CO (VA)	PG1101W-TR	LED GREEN CLEAR 1206 SMD
6	R1, R2, R3, R4, R8, R15	PANASONIC ELECTRONIC COMPONENTS (VA)	ERJ-8ENF1001V	RES SMD 1K OHM 1% 1/4W 1206
2	R9, R12	PANASONIC ELECTRONIC COMPONENTS (VA)	ERJ-8GEYJ303V	RES SMD 30K OHM 5% 1/4W 1206
10	C1, C2, C3, C4 , C6, C7, C8, C9, C10, C14	KEMET (VA)	C1206C104K5RAC7867	CAP CER 0.1UF 50V X7R 1206
2	BOB1, BOB2	MONOLITHIC POWER SYSTEMS INC.	MP2307DN	IC REG BUCK ADJ 3A SYNC
1	C	MOLEX, LLC	955032881	CONN MOD JACK 8P8C VERT UNSHLD
4	JP0, JP6, JP7, JP8, JP9	SULLINS CONNECTOR SOLUTIONS	PPPC042LFBN-RC	CONN HEADER FMAL 8PS .1" DL GOLD
1	JP1	SULLINS CONNECTOR SOLUTIONS	PPPC052LFBN-RC	CONN HEADER FMAL 10PS.1" DL GOLD
1	JP14	SULLINS CONNECTOR SOLUTIONS	PPPC051LFBN-RC	CONN HEADER FEMALE 5POS .1" GOLD
2	JP10, JP15	SULLINS CONNECTOR SOLUTIONS	PPPC041LFBN-RC	CONN HEADER FEMALE 4POS .1" GOLD
1	ADC	SULLINS CONNECTOR SOLUTIONS	PPPC101LFBN-RC	CONN HEADER FMALE 10POS .1" GOLD
2	JP22, JP27	HARWIN INC	M20-9980346	DIL VERTICAL PC TAIL PIN HEADER
1	J5LED, J3LED, J3V3	SULLINS CONNECTOR SOLUTIONS	PEC36SAAN	CONN HEADER .100 SINGL STR 36POS
10	J5LED, J3LED, J3V3, JADC, JCLOSET	TE CONNECTIVITY AMP	382811-8	SHUNT ECON PHBR 5AU BLACK

10.0 REVISION HISTORY

Date	Revision #	Change Description
11/2/2015	A	Initial Draft
12/4/2015	B	Changed schematics and layout to reflect changes in PCB, updated BOM