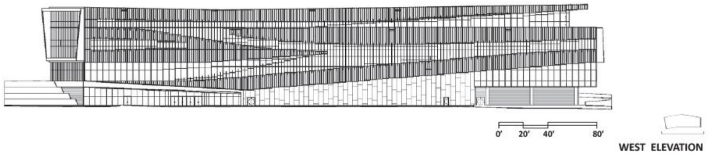
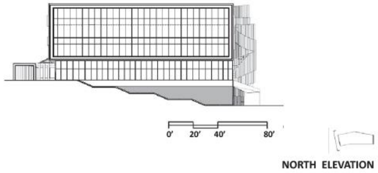
Team 4



Tracking and Predicting the State of a Building



Faculty Mentors
Kostas Christidis
Bobby Compton

Spencer Hitchins	Sensor Control Module Manager, UI Design Specialist, Sponsor Contact
Andrew Kofink	Web Application Manager, Software Specialist, Documentation Specialist
Scott Whalen	PCB Manager, Project Leader, Hardware Specialist

Introduction

- IBM is sponsoring this internet of things (IoT) project
- Track multiple sensor variables about a building
- Display historic data and make forecasts via a web app
- Motivations:
  - Inexpensive electronics
  - Many devices in the IoT
  - Need a common data interface

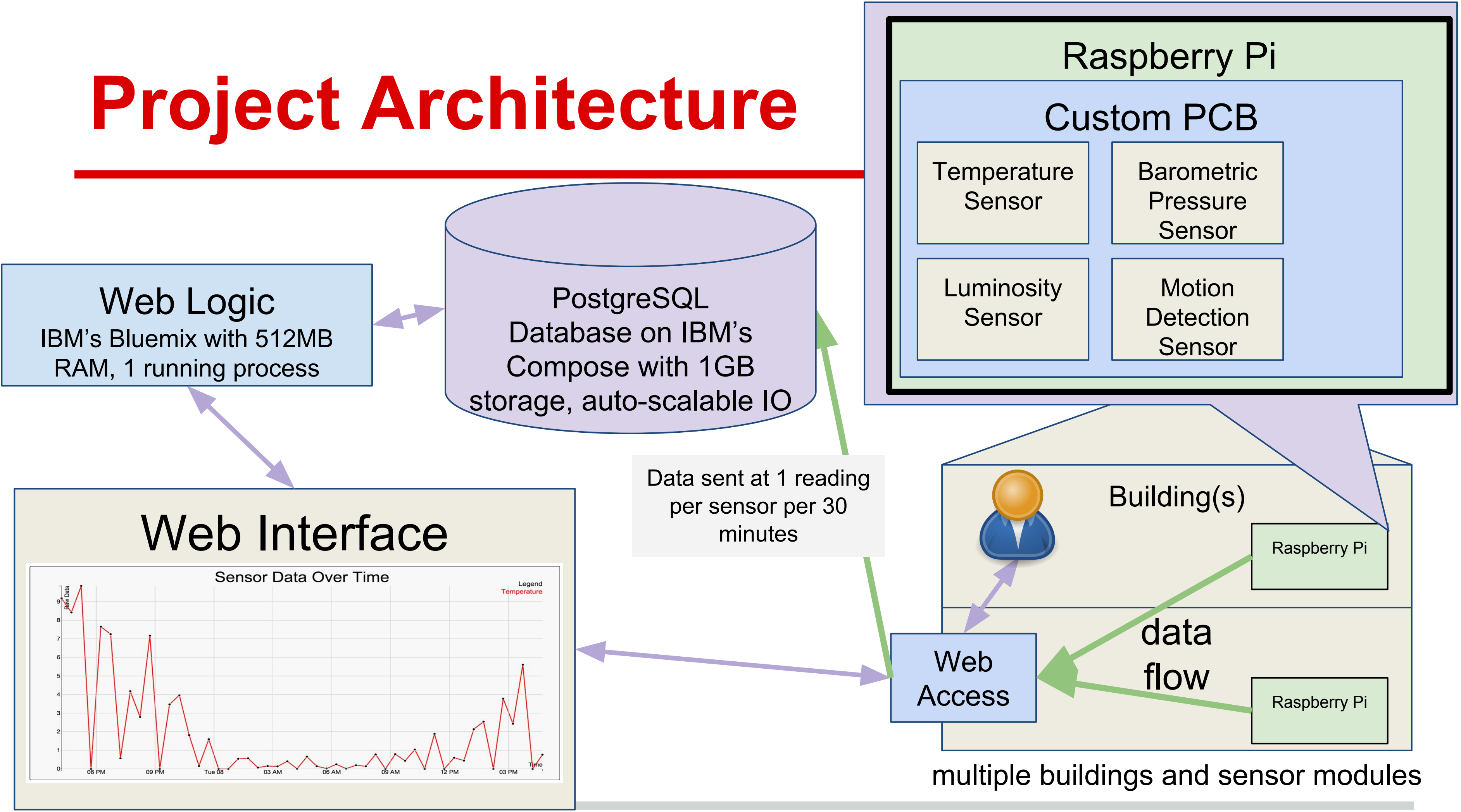
Project Requirements

- IoT devices push data to DB
- web application displays data
- system is easy to use
- IBM technology is showcased
- multiple sensor data per module

Change Summary

- no more detailed prediction algorithms
- web app overhaul
- switched database
- shifted focus from extensibility to ease of use
- sound sensor dropped
- enclosure added

Project Architecture



Raspberry Pi

Custom PCB

Temperature Sensor

Barometric Pressure Sensor

Luminosity Sensor

Motion Detection Sensor

Web Logic

IBM’s Bluemix with 512MB RAM, 1 running process

PostgreSQL Database on IBM’s Compose with 1GB storage, auto-scalable IO

Data sent at 1 reading per sensor per 30 minutes

Web Interface

Building(s)

Raspberry Pi

data flow

Web Access

Raspberry Pi

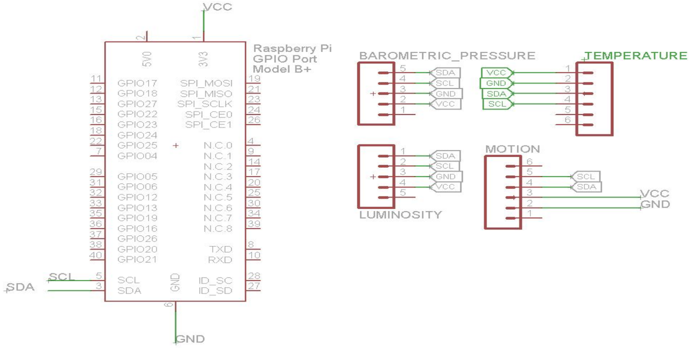
multiple buildings and sensor modules

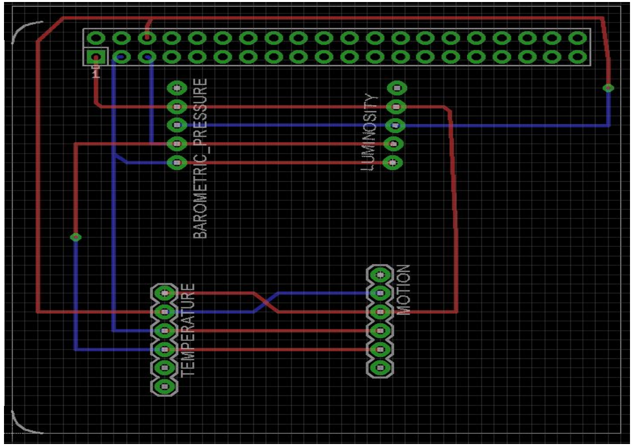
Raspberry Pi

- 5.2 power supply
- 16GB Sandisk microSD card
- Kootek wifi dongle

PCB

- 2 layer board
- 56mm x 58mm(2.2” x 2.3”) dimensions
- daughter board for Raspberry Pi
- .016” routes
- neater than breadboard



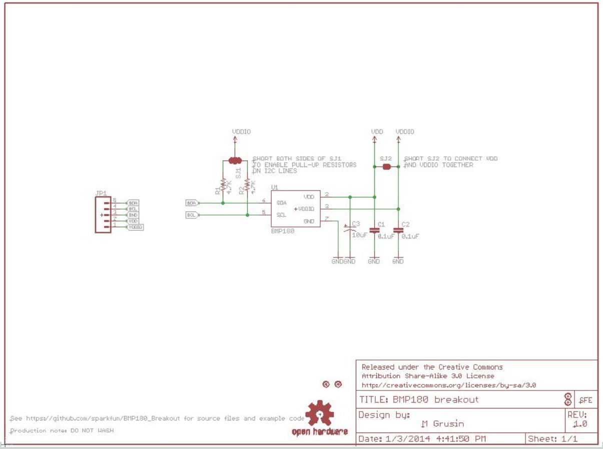


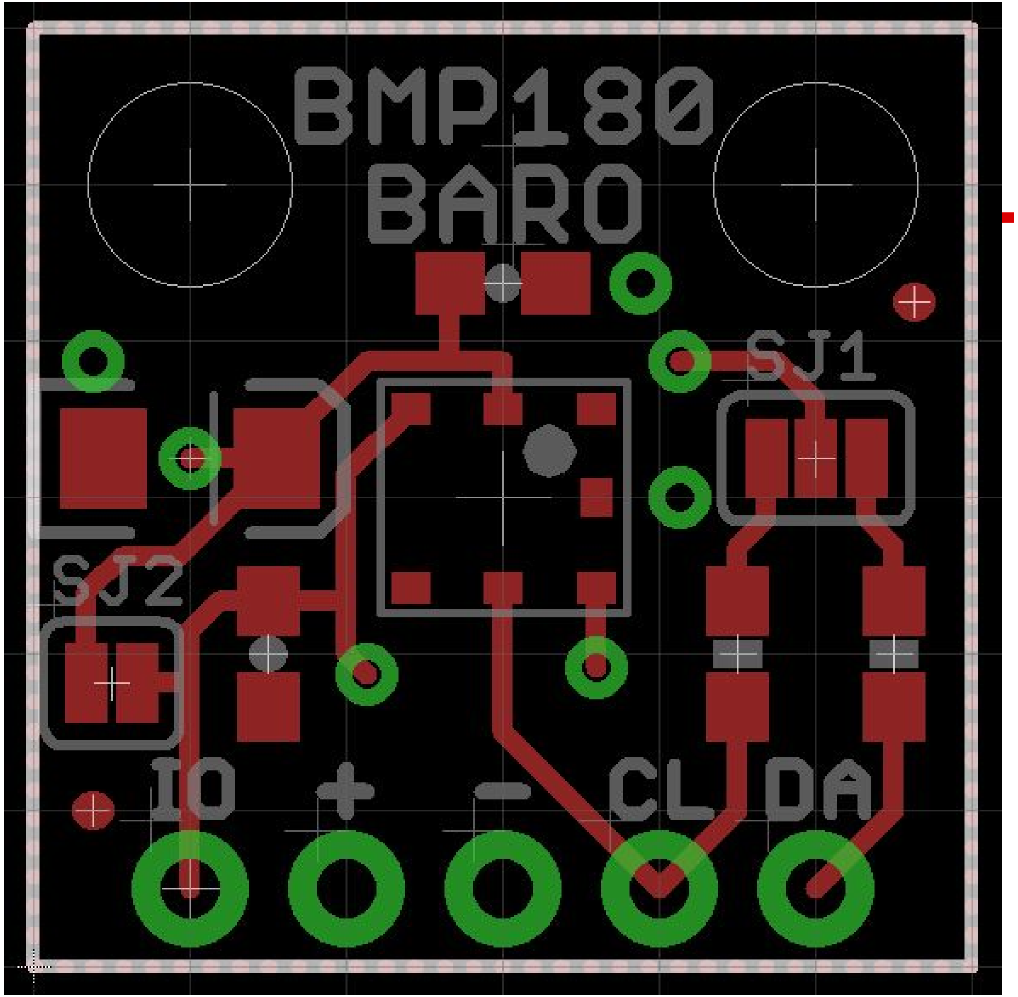
Sensors

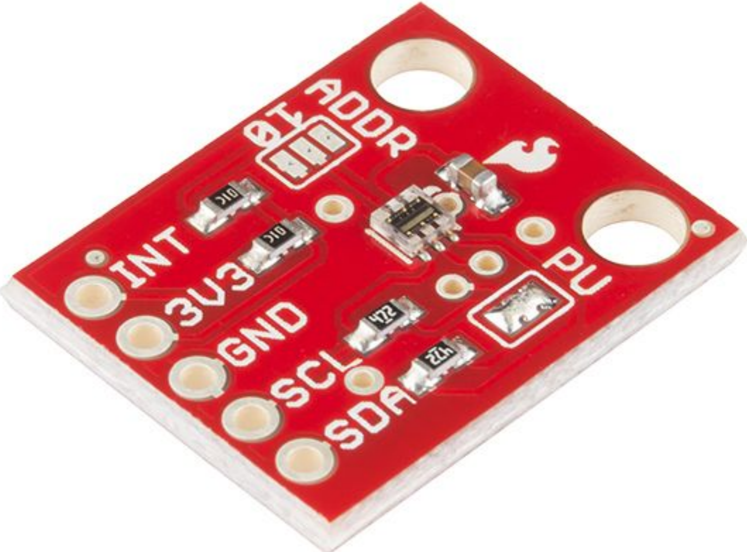
- have light, temperature, barometric pressure, motion
- I2C
- 3.3v power supply
- sparkfun
  - easy breakout board

Light

- range: of 0.1 - 40k+ Lux
  - troxler - ~380 Lux
- .6” x .6” dimensions
- URL: https://www.sparkfun.com/products/12055

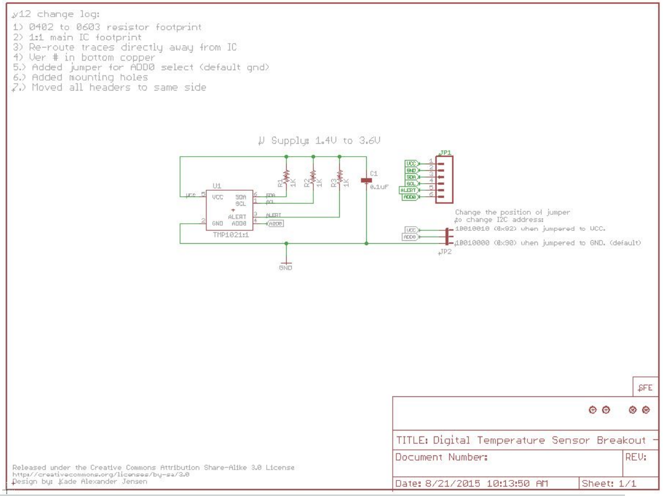

VDDIO
SHORT BOTH SIDES OF SJ1
TO ENABLE PULL-UP RESISTORS
ON I2C LINES
SJ1
R1
4.7K
R2
4.7K
VDD
VDDIO
SJ2
SHORT SJ2 TO CONNECT VDD
AND VDDIO TOGETHER
JP1
5
4
3
2
1
SDA
SCL
GND
VDD
VDDIO
SDA
SCL
U1
6
5
SDA
SCL
VDD
VDDIO
GND
2
3
7
BMP180
C3
10uF
C1
0.1uF
C2
0.1uF
GND
GND
GND
GND
Released under the Creative Commons
Attribution Share-Alike 3.0 License
http://creativecommons.org/licenses/by-sa/3.0
TITLE: BMP180 breakout
SFE
Design by:
M Grusin
REV:
1.0
Date: 1/3/2014 4:41:50 PM
Sheet: 1/1
See https://github.com/sparkfun/BMP180_Breakout for source files and example code
Production note: DO NOT WASH
open hardware


BMP180
BARO
SJ1
SJ2
IO
+
-
CL
DA


ADDR
01
INT
3V3
GND
SCL
SDA
PU

Temperature

- .0625° C resolution, .5° C accuracy
- range: -25° C(-13° F) - 85° C(185° F)
- .6” x .6” dimensions
- URL: https://www.sparkfun.com/products/11931



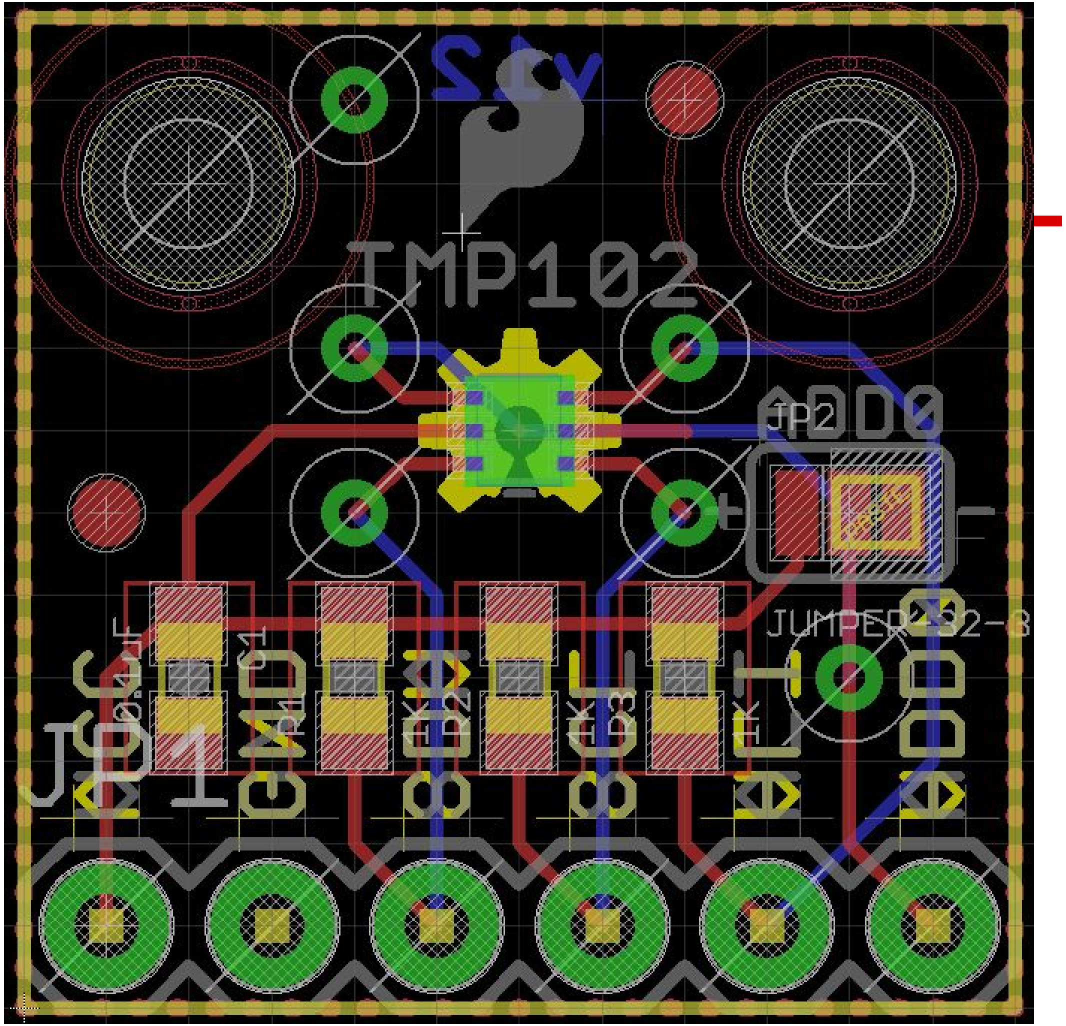
v12 change log:
1) 0402 to 0603 resistor footprint
2) 1:1 main IC footprint
3) Re-route traces directly away from IC
4) Ver # in bottom copper
5.) Added jumper for ADD0 select (default gnd)
6.) Added mounting holes
7.) Moved all headers to same side

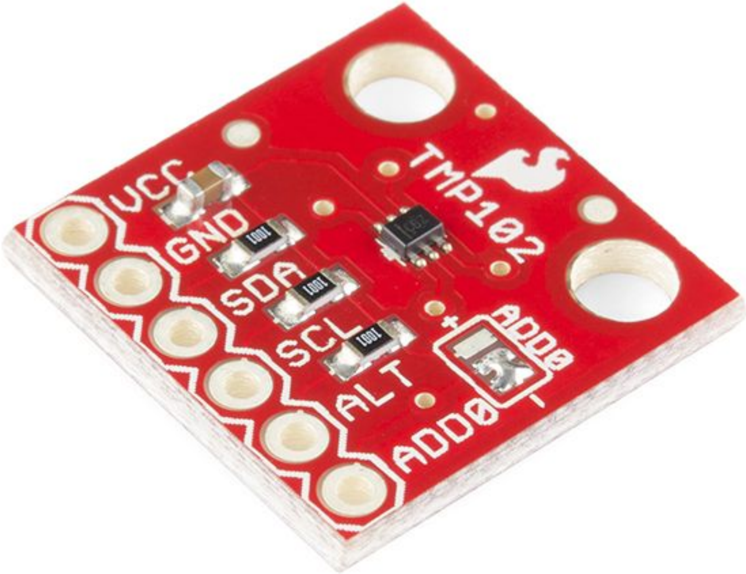
V Supply: 1.4V to 3.6V

Change the position of jumper to change I2C address:
10010010 (0x92) when jumpered to VCC.
10010000 (0x90) when jumpered to GND. (default)

Released under the Creative Commons Attribution Share-Alike 3.0 License
http://creativecommons.org/licenses/by-sa/3.0
Design by: Kade Alexander Jensen

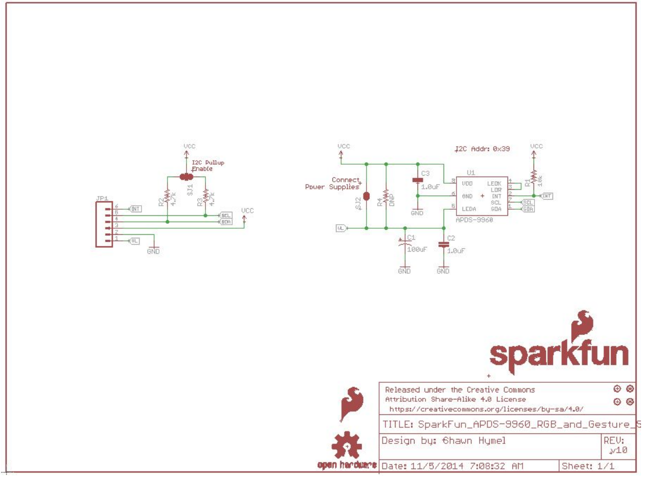
TITLE: Digital Temperature Sensor Breakout
Document Number:
REV:
Date: 8/21/2015 10:13:50 AM
Sheet: 1/1

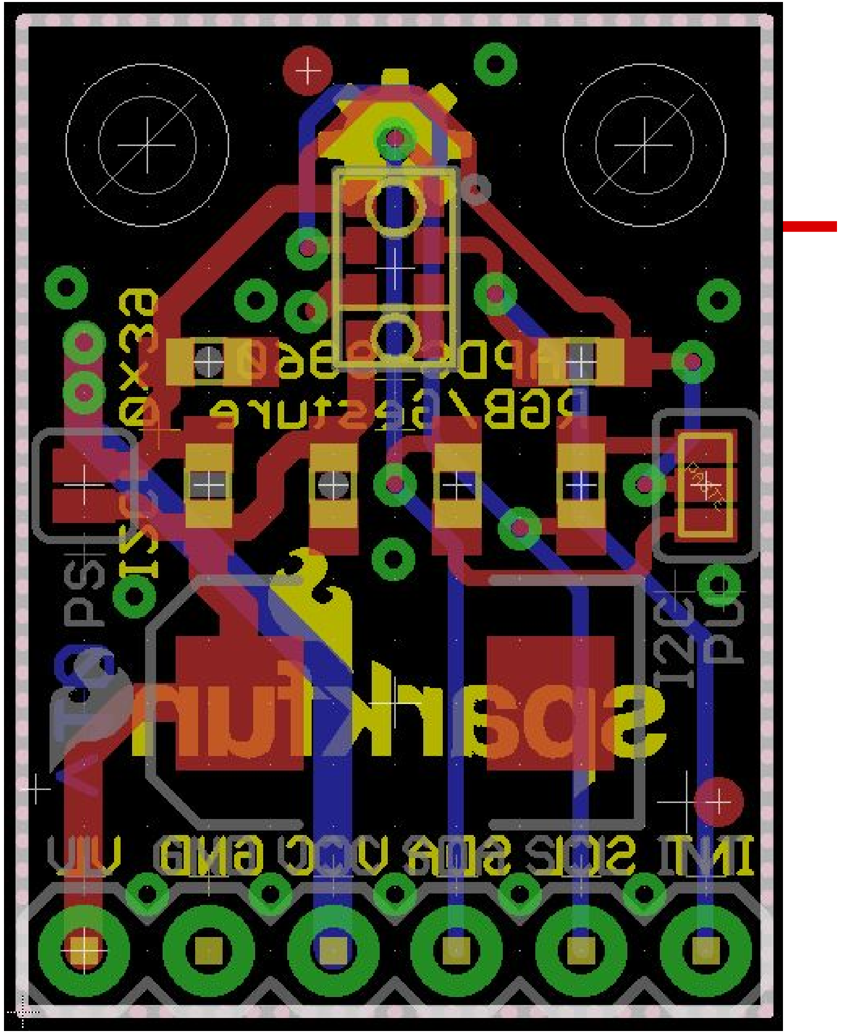

TMP102
ADD0
JP2
JP1
VCC
GND
SDA
SCL
ALT
ADD0
C1
0.1uF
R1
R2
R3

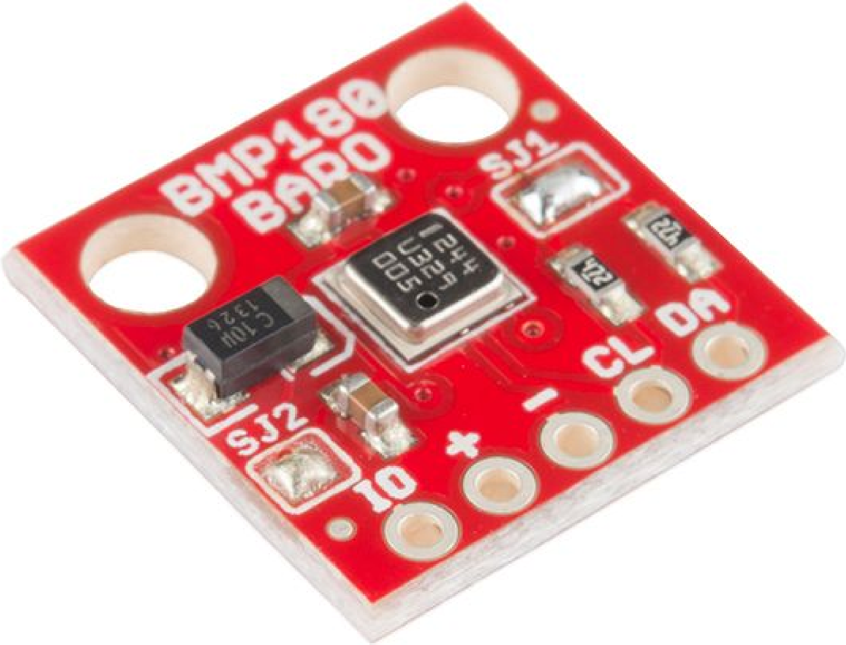

TMP102
VCC
GND
SDA
SCL
ALT
ADD0
ADD0
+
-

Barometric Pressure

- .02 hPa accuracy
- range: 300-1100 hPa
  - standard atmosphere - 1013.25 hPa
- .6” x .8” dimensions
- URL: https://www.sparkfun.com/products/11824

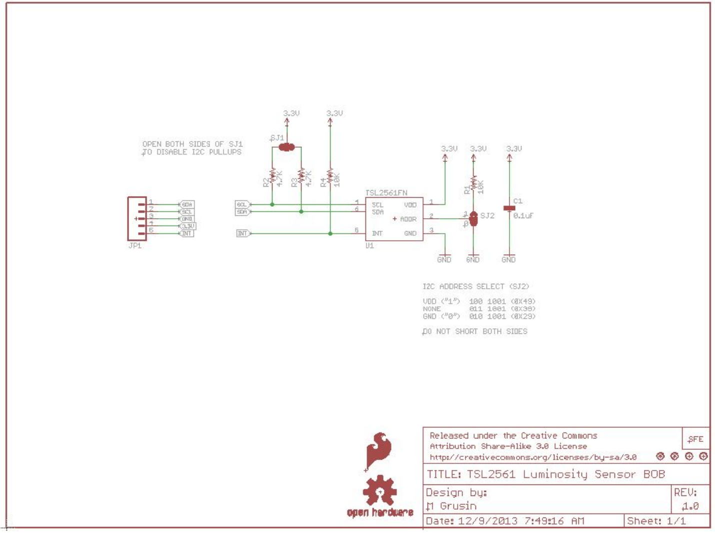



APDS-9960
RGB/Gesture
sparkfun
PS
I2C PU
VL
GND
VCC
SDA
SCL
INT


BMP180
BARO
SJ1
SJ2
IO
+
-
CL
DA

Motion

- 4”-8” detection range
- gesture detection, proximity sensing
- .6” x .75” dimensions
- URL: https://www.sparkfun.com/products/12787



OPEN BOTH SIDES OF SJ1
TO DISABLE I2C PULLUPS

I2C ADDRESS SELECT (SJ2)

VDD ("1")	100 1001	(0X49)
NONE	011 1001	(0X39)
GND ("0")	010 1001	(0X29)

DO NOT SHORT BOTH SIDES

SFE

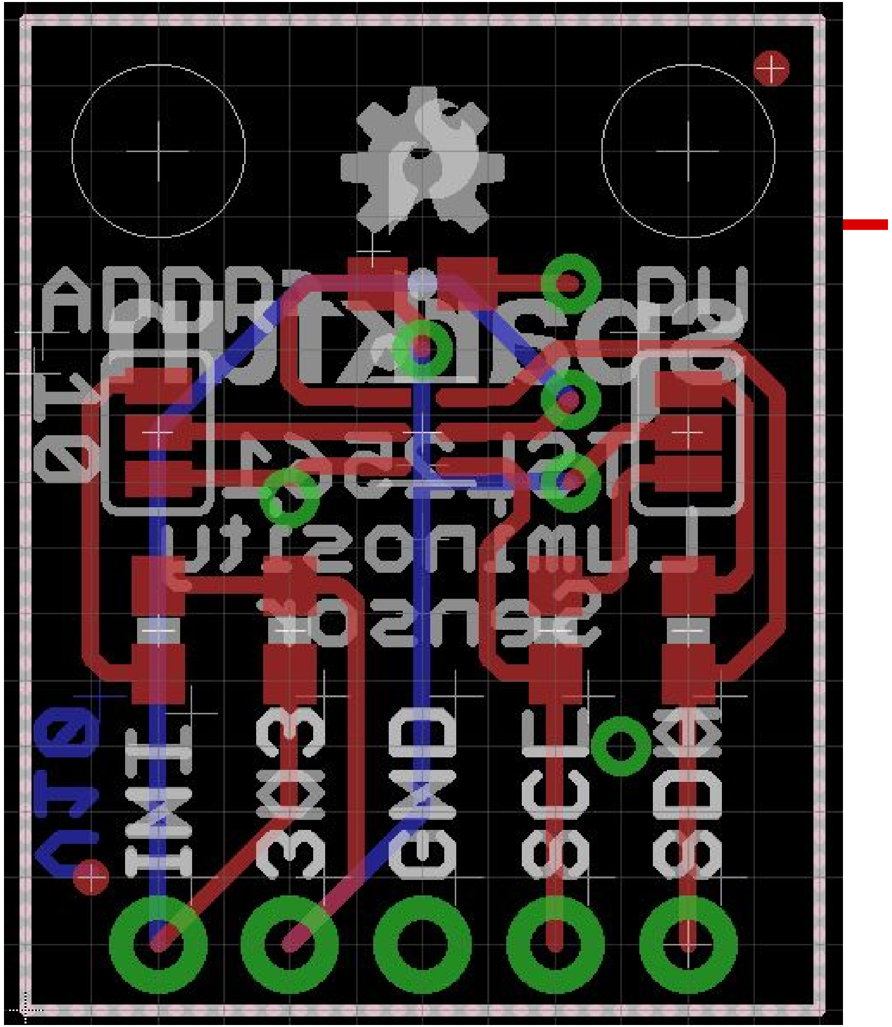
TITLE: TSL2561 Luminosity Sensor BOB

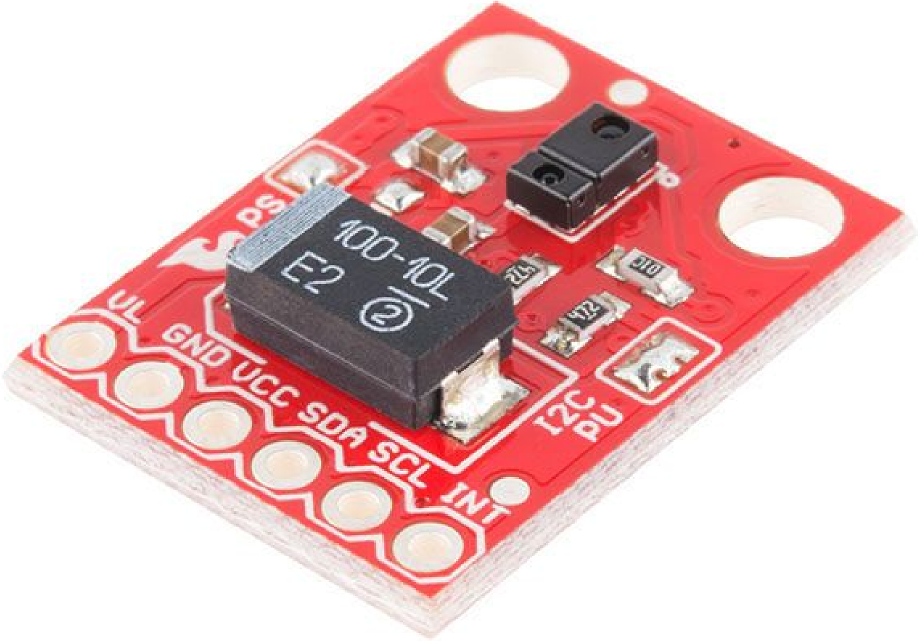
Design by:
M Grusin

REV:
1.0

Date: 12/9/2013 7:49:16 AM

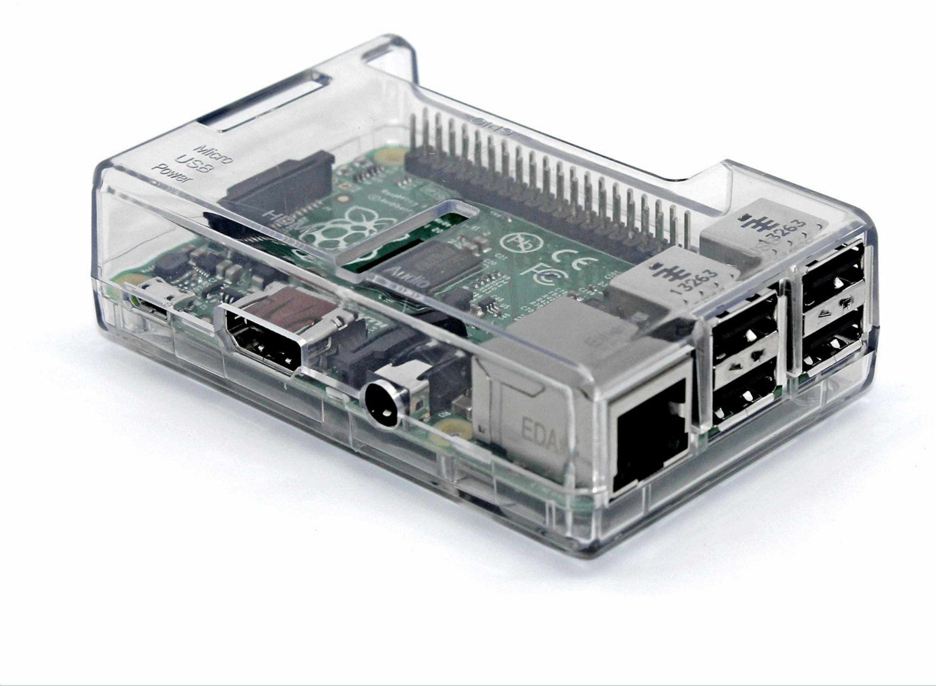
Sheet: 1/1


ADDR
PU
INT0
INT
3V3
GND
SCL
SDA


PS
100-10L
E2
VL
GND
VCC
SDA
SCL
INT
I2C
PU

Enclosure for PCB and Raspberry Pi

- http://www.amazon.com/dp/B00MQLB1N6/



Power Budget

- Raspberry Pi takes up to 50mA per pin
- temperature sensor uses up to 10µA
- BMP sensor uses up to 5µA
- motion sensor uses up to 250µA
- luminostiy sensor up to uses .6mA
- total = 10µA + 5µA + 250µA + .6mA = .865mA power budget (well below 50mA)

Hardware Testing

- solder on headers to PCB
  - easy connect/disconnect for sensors and from Raspberry Pi
- check connections are good with PCB
  - make sure Raspberry Pi can read from sensors and get valid data
- use multimeter/oscilloscope to debug issues
  - have vias and other holes that can be used as test points

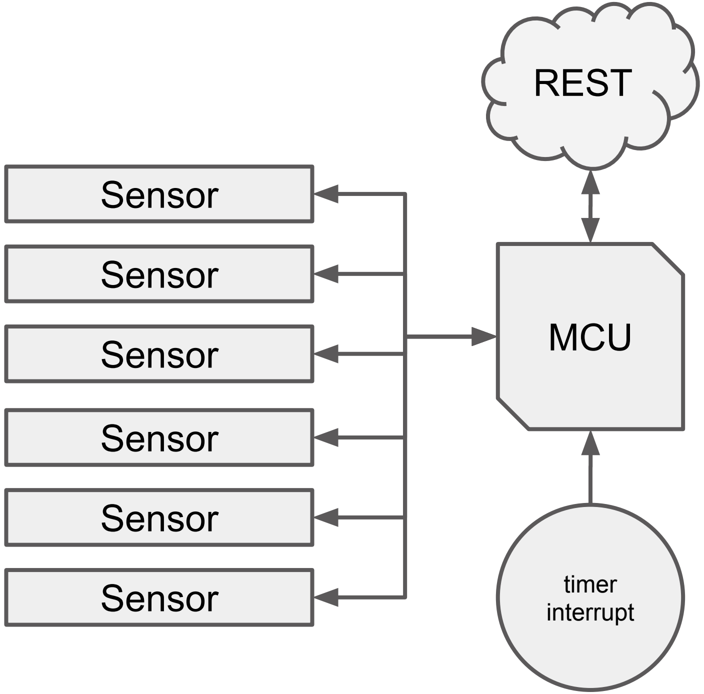
Hardware Testing (cont)

- double check that solder points aren’t soldered together by accident

Design Challenges

- sound sensor
- Raspberry Pi vs. Beaglebone Black
- why not 3D-printed enclosure

Raspberry Pi Software Architecture



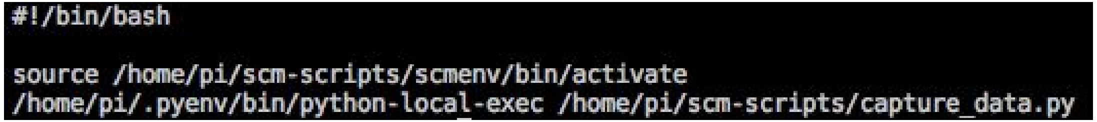
Sensor Control Module

- Consists of Raspberry Pi and daughter board with sensors
- Preloaded with unique module ID and IDs for each sensor
- Ships with microSD card preloaded with golden image
  - Software and packages pre-installed

Sensor Software Requirements

- Raspbian
- Python v2.7.9
- Pip v1.5.6
- Python Packages
  - psycopg2 v2.6.1
  - smbus-cffi v0.5.1
- Git v1.7.10

Sensor Software



```
#!/bin/bash

source /home/pi/scm-scripts/scmenv/bin/activate
/home/pi/.pyenv/bin/python-local-exec /home/pi/scm-scripts/capture_data.py
```

- Bash Scripts
  - Run on system startup
  - Activate Python scripts
- Python Scripts
  - Communicate with sensors
  - Push data to web application
- Python Configuration Files
  - Provide information about sensors

Sensor Software Test Plan

- Unit tests for functions using PyUnit
- Test integration with database
  - Manually ensure data actually sent to database
- Test integration with PCB
  - Ensure PCB connections correctly allow I2C communication with each sensor

Shipping

- Before
  - Unique module and sensor IDs loaded in database
  - Unit testing to ensure module integrity
- After
  - Ships with instructions and sticker with unique ID
    - Plug in Pi
    - Visit URL in instructions
    - Register with unique ID on sticker
- Data immediately begins pushing to DB

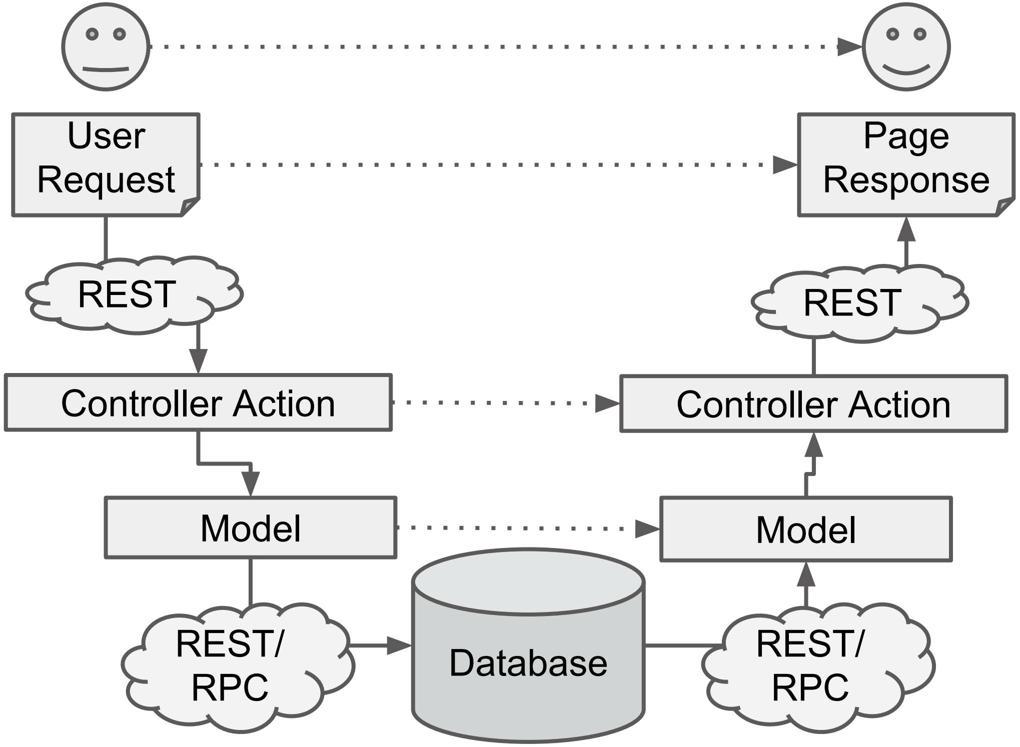
Sensor Software Timeline

Item Description	Expected Completion
Integrate temperature sensor with database	2015-09-07
Integrate luminosity sensor with database	2015-09-16
Integrate barometric pressure sensor with database	2015-09-30
Test PCB integration with temperature sensor, luminosity sensor, and barometric pressure sensor	2015-10-05 (Alpha Demo)
Test PCB integration with proximity sensor	2015-10-14
Integrate proximity sensor with database	2015-10-21
Complete registration workflow for module	2015-11-04 (Beta Demo)
Design Day, Version 1.0.0 locked	**2015-12-01**

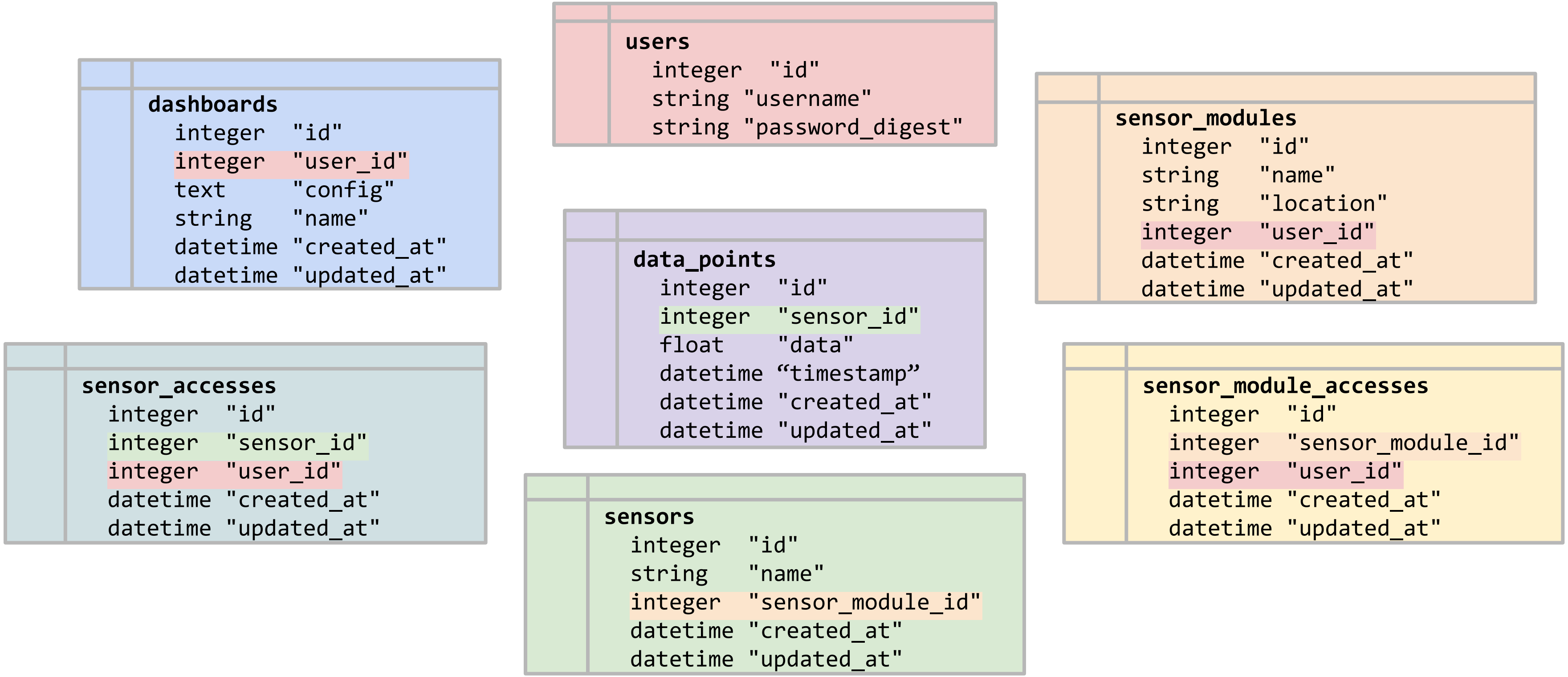
Design Challenges

- Ease of use vs. Extensibility
- Differences in sensors
  - Data addresses
  - Control signals
  - Power signals
- Data format

MVC Software Architecture



Web Application DB Schema



```
users
  integer  "id"
  string "username"
  string "password_digest"
```

```
dashboards
  integer  "id"
  integer  "user_id"
  text     "config"
  string   "name"
  datetime "created_at"
  datetime "updated_at"
```

```
sensor_modules
  integer  "id"
  string   "name"
  string   "location"
  integer  "user_id"
  datetime "created_at"
  datetime "updated_at"
```

```
data_points
  integer  "id"
  integer  "sensor_id"
  float    "data"
  datetime “timestamp”
  datetime "created_at"
  datetime "updated_at"
```

```
sensor_accesses
  integer  "id"
  integer  "sensor_id"
  integer  "user_id"
  datetime "created_at"
  datetime "updated_at"
```

```
sensor_module_accesses
  integer  "id"
  integer  "sensor_module_id"
  integer  "user_id"
  datetime "created_at"
  datetime "updated_at"
```

```
sensors
  integer  "id"
  string   "name"
  integer  "sensor_module_id"
  datetime "created_at"
  datetime "updated_at"
```

Web Application Routes

User
#new
#create
#edit
#update
#destroy
#index

Dashboard
#new
#create
#edit
#update
#destroy
#index

Session
#new
#create
#destroy

Sensor Module
#new
#create
#edit
#update
#destroy
#index

Sensor Module Access
#new
#create
#edit
#update
#destroy
#index

Sensor
#new
#create
#edit
#update
#destroy
#index

Sensor Access
#new
#create
#edit
#update
#destroy
#index

Data Point
#new
#create
#edit
#update
#destroy
#index

Web UI Libraries

- Bootstrap v.3.3.5
- D3.js v.3.5.6
- HAML v.4.0.6
- Font Awesome v.4.4.0
- SASS v.3.2.19
- jQuery v.2.1.4
- Uglifier v.2.7.0
- Turbolinks v.2.5.3

Web Management Libraries/Tools

- Bundler v.1.10.6
- Rake v.10.4.2
- Cloud Foundry v.6.12.2
- Pry v.0.10.1
- Git v.2.2.2

Web Testing/Analysis Libraries

- Capybara v.2.4.4
- RSpec v.3.1.0
- Rspec Mocks v. 3.1.3
- SimpleCov v.0.9.1
- Guard v.2.11.1

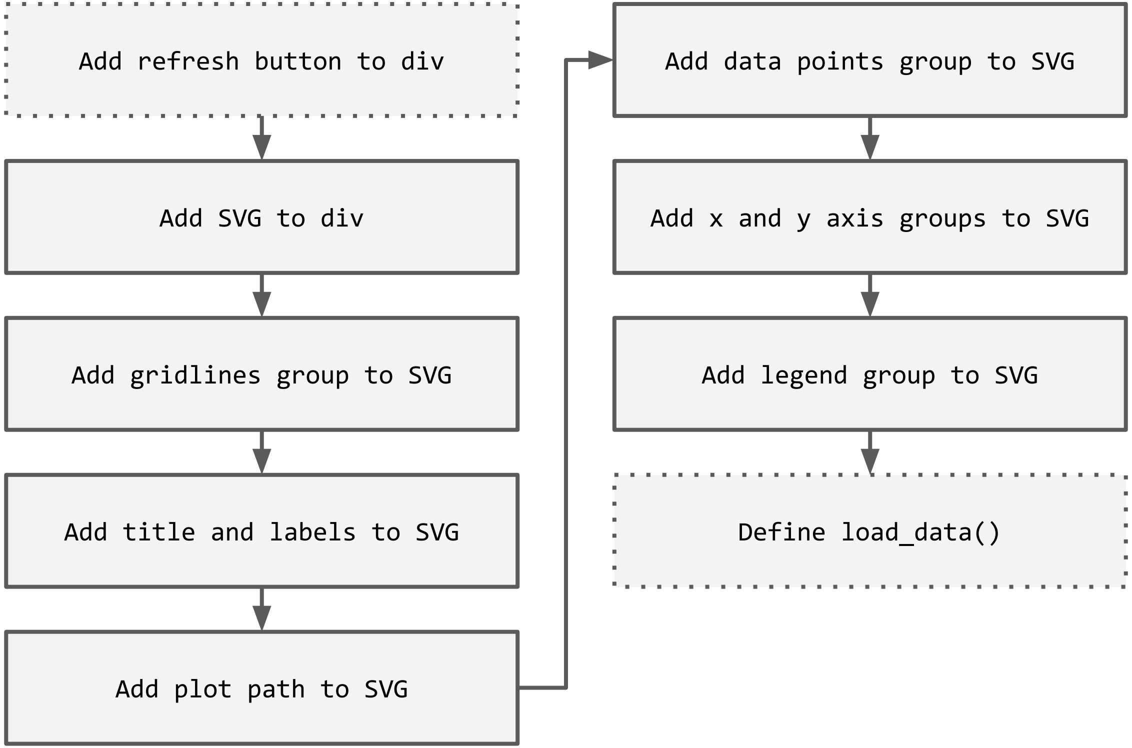
Web Backend Libraries

- Bcrypt v.3.1.9
- ActiveRecord v.4.0.13
- ActiveModel v.4.0.13
- AREL v.4.0.2
- PostgreSQL v.9.4.4
- Rack v.1.5.2
- Unicorn v.4.9.0

Resource Hosting

- Bluemix by IBM (Ruby on Rails)
- Compose by IBM (PostgreSQL)
- Users must purchase internet service
- Users must have an 802.11abg 2.5Ghz AP

D3 Flowchart (FC) - simple_plot()



Add refresh button to div

Add SVG to div

Add gridlines group to SVG

Add title and labels to SVG

Add plot path to SVG

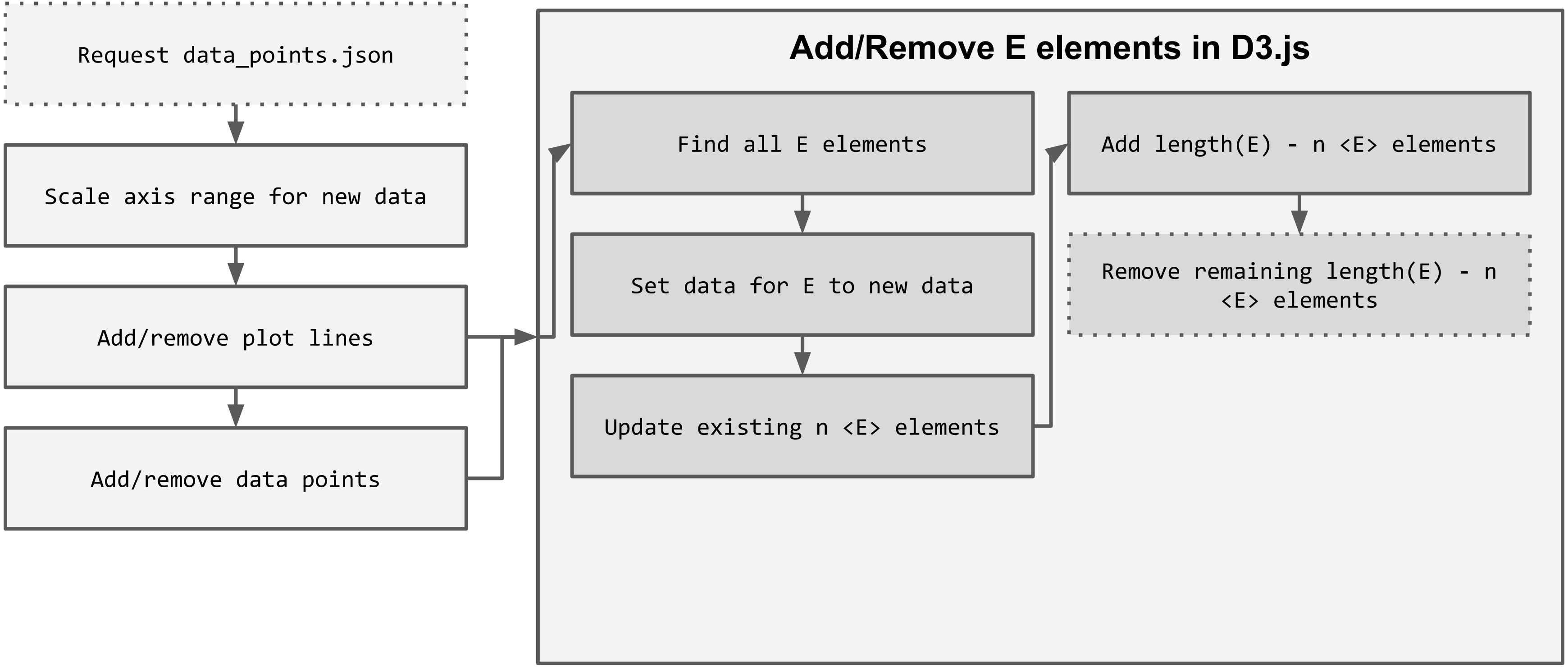
Add data points group to SVG

Add x and y axis groups to SVG

Add legend group to SVG

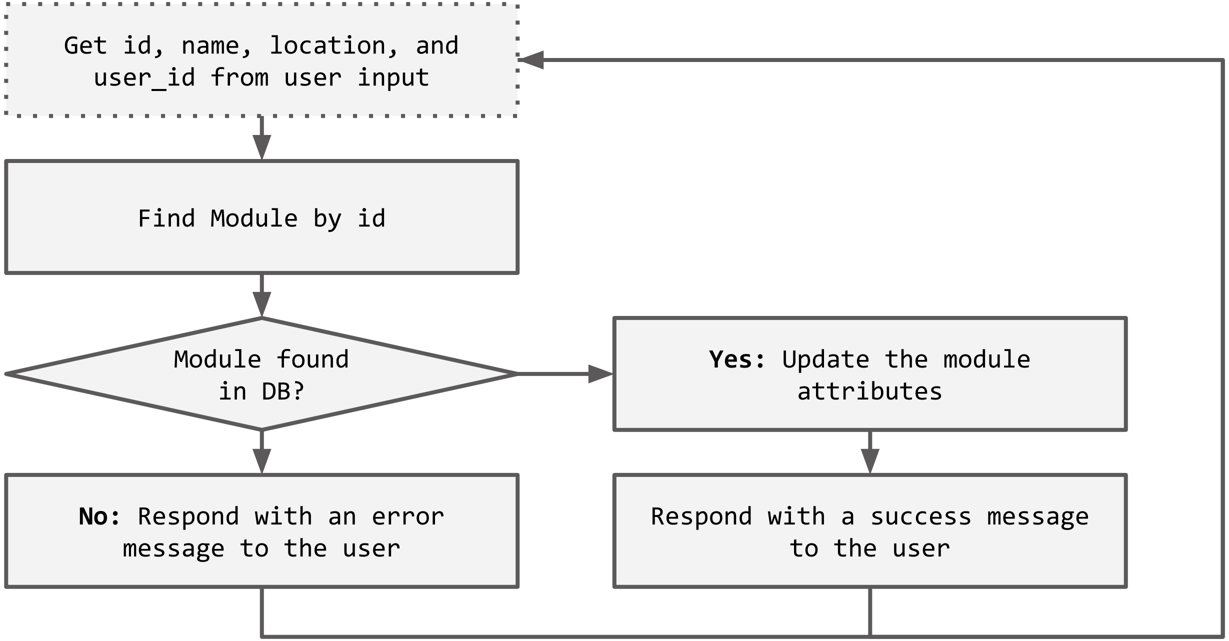
Define load_data()

D3 FC - load_data()



Request data_points.json
Scale axis range for new data
Add/remove plot lines
Add/remove data points
Add/Remove E elements in D3.js
Find all E elements
Set data for E to new data
Update existing n <E> elements
Add length(E) - n <E> elements
Remove remaining length(E) - n <E> elements

Control Module Identification FC



Get id, name, location, and user_id from user input

Find Module by id

Module found in DB?

Yes: Update the module attributes

Respond with a success message to the user

No: Respond with an error message to the user

Web Application Test Plan

- Unit tests written for all models using RSpec
  - `data_point, sensor_access, sensor_module_access, dashboard, sensor, sensor_module, user`
- Integration tests written for all controllers/views using RSpec Mocks
  - `data_points, sensor_accesses, sensor_module_accesses, dashboards, sensors, sensor_modules, users`
- Acceptance Test list for manual exercising requirements

Black Box Acceptance Tests

- Users can create a user account
- Users can log in
- Users can add sensor modules
- Users can edit sensor metadata
- Users can delete data points
- Users can grant/revoke access to their sensors
- Users can grant/revoke access to their control modules
- Users can create new dashboards
- Users can edit and delete their dashboards

Web Application Timeline

Item Description	Expected Completion
Adding strict authorization	2015-09-16
Adding web encryption	2015-09-23
Adding an administrative system	2015-09-30
Allowing users to switch sensors shown on the dashboard plot	2015-10-05
Allowing users to add multiple plots per graph	2015-10-05 (Alpha Demo)
Allow displaying of historical data (user-specified date range)	2015-10-14
Allowing users to save/edit the dashboard configuration	2015-10-14
Allowing users to add multiple graphs on the dashboard	2015-11-04 (Beta Demo)
Adding basic prediction mechanism	2015-11-11
Design Day, Version 1.0.0 locked	**2015-12-01**

Manufacture/Running Cost

- Luminosity: https://www.sparkfun.com/products/12055 $5.95
- Temperature: https://www.sparkfun.com/products/11931 - $5.95
- Barometric Pressure: https://www.sparkfun.com/products/11824 - $9.95
- Motion: https://www.sparkfun.com/products/12787 - $14.95
- Raspberry Pi B+: ~$30.00
- PCB: ~$30.00
- MicroSD card: $10
- wifi dongle: $8
- Bluemix Hosting (1 instance at 512MB RAM) - free (scalable)
- Compose PostgreSQL Hosting - $17.50/month minimum (scalable)

Total: ~$114.80 + $17.50 / month

Alpha Demo (2015-10-05)

- 3 sensors
- Authorization on the web application
- TLS on the web application
- Multiple plots per graph

Beta Demo (2015-11-04)

- All sensors
- PCB mounted to control module
- Control module housed in the enclosure
- Add UUID sticker for the Pi
- Multiple graphs on a dashboard
- Save/edit dashboard configuration
- Complete user unboxing web workflow

Final Design Day Demo (2015-12-04)

- Example sensor control module
- Web application demo
- Data from three modules tracking Troxler lab for the week prior to design day
  - One by each door
  - One near a window

Demo

http://tpsb.mybluemix.net

Q & A - IBM Team 4

Team Members:

Scott Whalen

Spencer Hitchins

Andrew Kofink

Bill of Materials

- Luminosity: https://www.sparkfun.com/products/12055 $5.95
  - Datasheet: http://cdn.sparkfun.com/datasheets/Sensors/LightImaging/TSL2561.pdf
- Temperature: https://www.sparkfun.com/products/11931 - $5.95
  - Datasheet: http://www.sparkfun.com/datasheets/Sensors/Temperature/tmp102.pdf
- Barometric Pressure: https://www.sparkfun.com/products/11824 - $9.95
  - Datasheet: http://cdn.sparkfun.com/datasheets/Sensors/Pressure/BMP180.pdf
- Pi Camera: http://www.adafruit.com/products/1367?gclid=CNW1zuesickCFYc8gQod3dAKOQ - $29.95
  - Datasheet: http://www.element14.com/community/servlet/JiveServlet/downloadBody/54413-102-1-273177/Unofficial%20guide%20to%20getting%20up%20and%20running%20with%20the%20Raspberry%20Pi%20Camera.pdf
- Raspberry Pi B+: http://www.amazon.com/Raspberry-Pi-Model-512MB-Computer/dp/B00LPESRUK/ref=sr_1_3?ie=UTF8&qid=1449783772&sr=8-3&keywords=raspberry+pi+b%2B - $30.00
- PCB: http://www.4pcb.com/pcb-student-discount.html - $33.00
- MicroSD card: http://www.amazon.com/SanDisk-microSDHC-Standard-Packaging-SDSQUNC-032G-GN6MA/dp/B010Q57T02/ref=sr_1_2?s=pc&ie=UTF8&qid=1449783848&sr=1-2&keywords=microSD+card - $10.95
- WiFi dongle: http://www.amazon.com/Edimax-EW-7811Un-150Mbps-Raspberry-Supports/dp/B003MTTJOY/ref=sr_1_1?s=pc&ie=UTF8&qid=1449783920&sr=1-1&keywords=wifi+dongle - $9.99
- Raspberry Pi B+ case: http://www.amazon.com/Premium-Clear-Case-Raspberry-Model/dp/B00MQLB1N6/ref=sr_1_1?ie=UTF8&qid=1446562138&sr=8-1&keywords=raspberry+pi+b%2B+enclosure - $8.92
- Bluemix Hosting (1 instance at 512MB RAM) - free (scalable)
- Compose PostgreSQL Hosting - $17.50/month minimum (scalable)

Total: $144.66 + $17.50 / month

Final Prototype Cost

- 4x Sensor Modules ($144.66)
- 2x Extension Cords ($15.99)
- 4x Sound Detectors ($10.95, unused)
  - Datasheet: http://cdn.sparkfun.com/datasheets/Sensors/Sound/LMV324.pdf
- 2x Long Range IR sensor ($14.95, unused)
  - Datasheet: https://www.sparkfun.com/datasheets/Sensors/Infrared/gp2y0a02yk_e.pdf
- Adafruit ADS1015 12-Bit ADC ($9.95, unused)
  - Datasheet: http://adafruit.com/datasheets/ads1015.pdf

Total: $609.26

Not all items had to be purchased because the department owned them already.