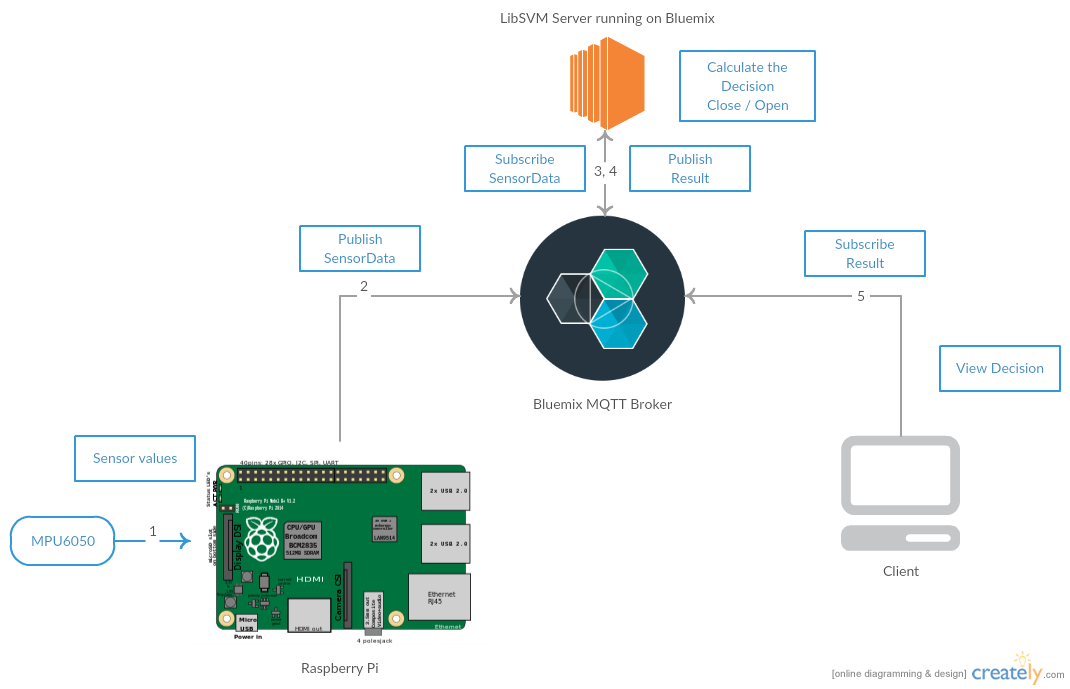
­­Project report

When we visualize these data, we can see only two columns of these data are change accordingly. So, to solve this problem, we choose the last two columns of data.

Using threshold to divide the whole data sequence into sub-sequence that corresponds to different events that correspond to open and close.

After averaging these values, we choose the last columns of values.

****

**Flow diagram of the project**

**Flow of the application:**

1 . MPU6050 is mounted on the breadboard and the readings are continuously sent to SDA and SCL ports which is polled by the python program that is running on Raspberry pi every 100ms

2 . According the specifications, we have scaled each of the values ie. the three axial values of gyroscope and accelerometer

3. The gyroscope operates between the range of -250 to 250 degrees per second and the sensitivity scale factor is 131 degree/sec

4. The accelerometer operates between the range of += 2 g and sensitivity scale factor for it is 16384 lsb/g

5. We keep a threshold for the values that need to be sent to the bluemix. I.e we publish the values to the broker only when there is a sizable change in any of the axial values. This is because, the sensor keeps generating a large amount of data even when there is no motion. This keeping a threshold saves unnecessary data flow and processing needed by the broker.

→ Topic name : SensorData and type : Json

6. The server running the svm predict listens to the topic SensorData and calls the svm predict module in it's event call back.

7. The server then publishes the result to the broker to the topic “Result”. The result comprises of the latest axial values and decision (close or open)

8. Any End user application can subscribe to the topic “Result” and display the data. In our case, the same server which is running the classifier also has the UI which uses AJAX calls once every second to poll for the result.

**Design Choices**

**How we implemented Live update to the web application?**

We decided to use AJAX to keep the web page up to date with the decision from the classifier. Using AJAX helped us show live feed without having to reload the web page.

**How we automated the training ?**

We mounted the Sensor on our door and captured the axial values of both gyroscope and accelerometer. We also developed a UI module from where the trainer can select open or close each time the user opens or close the door and all these 6 values along with the class are stored in the training file.

We then supply this training file to a SVM model builder which generates a svm.model file which is then pasted into the server that makes the decision

**How we aggregated the continuous values ?**

Every door open and close event generates a series of values . We need only one data for the classifier. So we just took the values that are above the threshold (for each event) and found the average each of the values. The same mechanism to used for both testing and training data.

**How we built trained and built the model?**

Before generating the model, we rigorously collected training data set by moving the door with varying intensity , velocity and sweep. This helped us to cover a wide range of test cases.