OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

INF FACULTY OF COMPUTER SCIENCE

Design and Analysis of IoT Application on SCION Smart Inventory Management

Guided by:
Prof. David Hausheer

Team
Abhijith Remesh (221424)
Manish RamaChandran (221423)

Table of Contents

Business Use Case Context

Current Scenario:

Regular Physical Inspection of inventory status of the products by specific personnel

- Availing Weekly discounts
- Scheduling weekly orders

Proposed Idea :

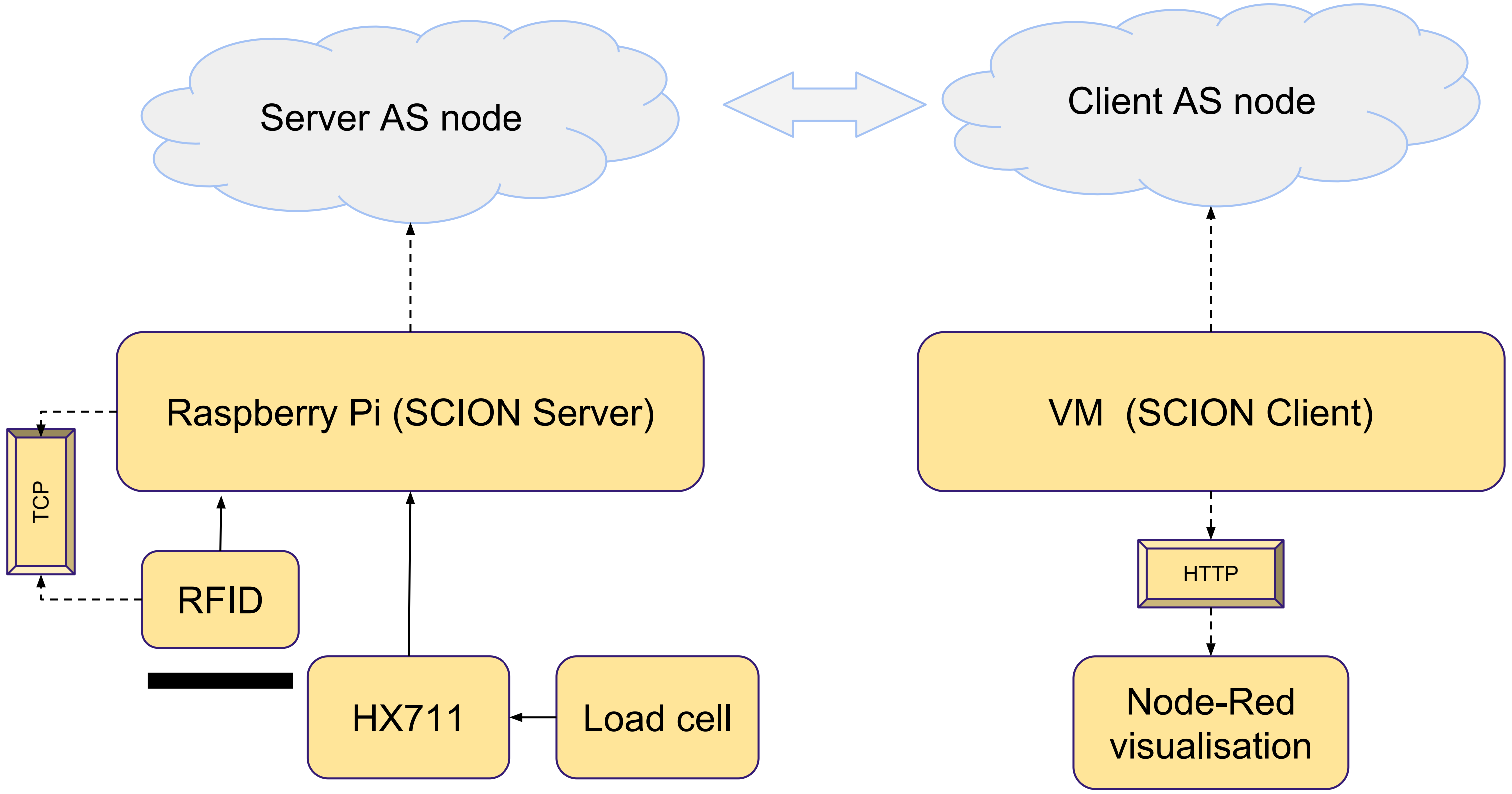
To monitor the real time inventory status of a materials/ products placed in a pallet at a specific location (Retail-Wholesale storage units (stores/warehouses))

The real time variations in weight > inventory status > stakeholders take business decisions

- Replenish stock in real time
- Grant discounts in real time

Merits : Saves time, Saves money (Time is money)

Overview : Block Diagram

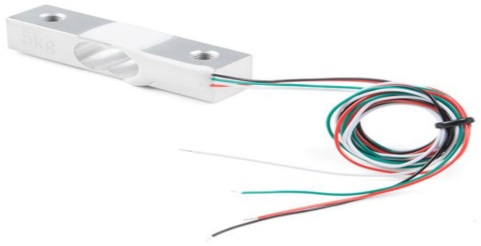


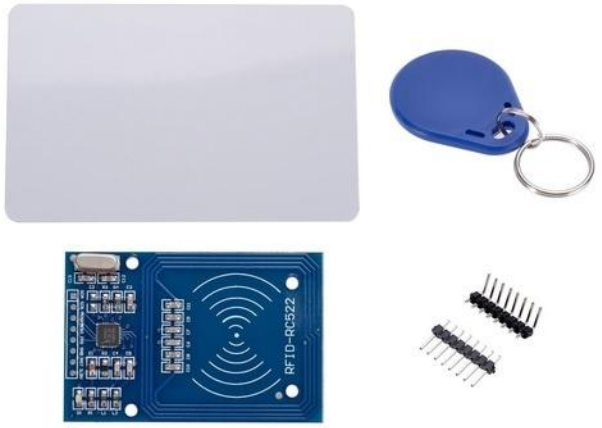
Overview : Key Components

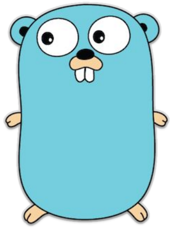
- Load cell amplified with HX711 ADC module is interfaced with Pi, which fetches the weight values.
- RFID MRFC522 module interfaced with the Pi which scans the respective RFID tags and hosts the RFID data over a TCP socket.
- A SCION Server AS node instance running on Raspberry Pi gets the RFID tag data via TCP along with the respective weight data and sends to the SCION client when requested.
- A SCION client AS node running on the Virtual Machine on request gets the data packet as response from the SCION Server AS node and hosts the data over a HTTP socket.
- Node-Red Dashboard, a flow based visualisation tool gets this data packet with a HTTP GET request.

Hardware and Software Utilities

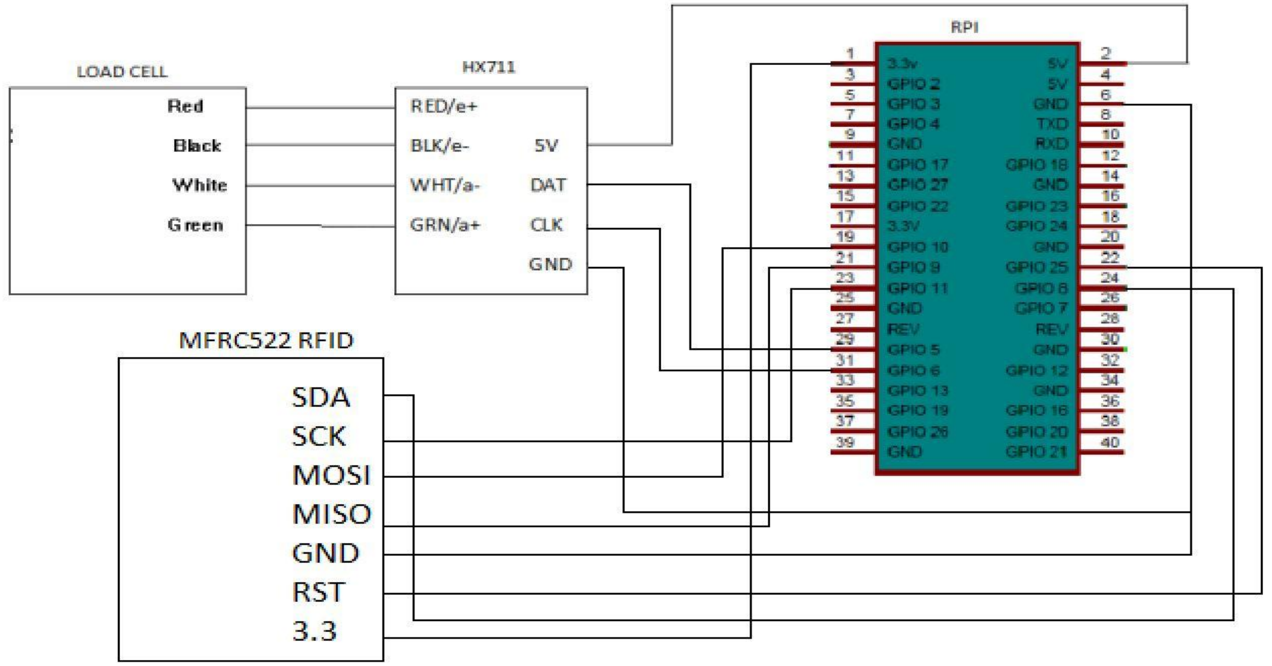
Raspberry Pi 3 B+ Model


Load cell 1 Kg


MFRC522 RFID module kit



Implementation : Hardware Pin Diagram



Implementation : Hardware Interfacing

As per the PIN diagram, the application involves two sensor interfaces.

- Load cell > HX711 > Raspberry Pi
  - Ensuring that the load cell is producing raw analog values proportionally based on the applied pressure upon it
  - Calibrating the load cell with some known weights
  - Reading the raw weight values.

- MFRC522 reader > Raspberry Pi
  - Detecting the UID of the tag when tag is swiped across.
  - No Tag swiped > No UID detected.

Implementation : SCION installation on VM & Pi

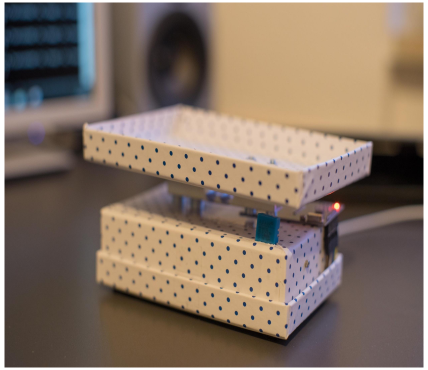
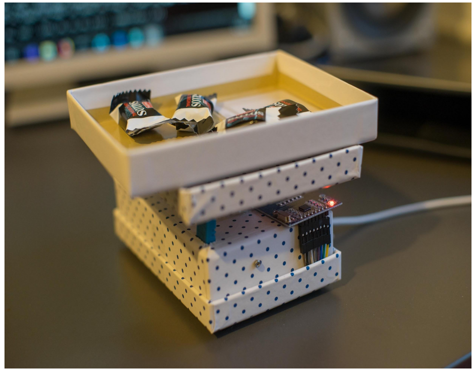
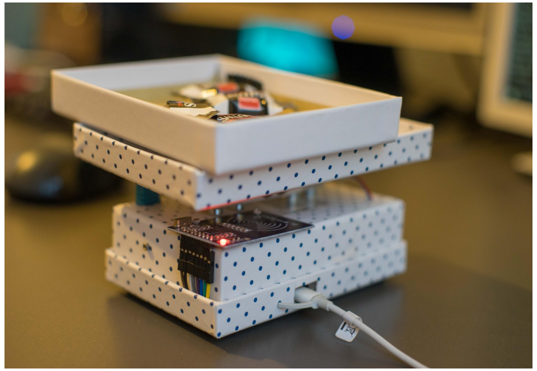
Client Side : SCION AS installed on VM with the following parameters

- Attachment point : 19-ffaa:0:1303
- Port : 5000, Check “Install inside a Virtual machine”
- SCION AS node : 19-ffaa:1:bfa
- Set up the virtual environment with vagrant and VirtualBox
- Run SCION network
- Successful reception of beacons indicates successful connection to the SCION network.

Server Side : SCION AS installed on the Pi

- Attachment point : 19-ffaa:0:1303
- Port : 5000, Check “Install on a dedicated SCION system”
- SCION AS node : 19-ffaa:1:161
- Download Ubuntu MATE for Raspberry Pi 3B+
- Copy certain files from Raspbian OS to Ubuntu MATE.
- Flash this OS to the Pi,follow SCION tutorials and run the SCION network.
- Successful reception of beacons indicates successful connection to the SCION network.

Implementation : Assembling as a StandAlone Unit

Implementation : Client-Server schema

Implementation : Client-Server communication

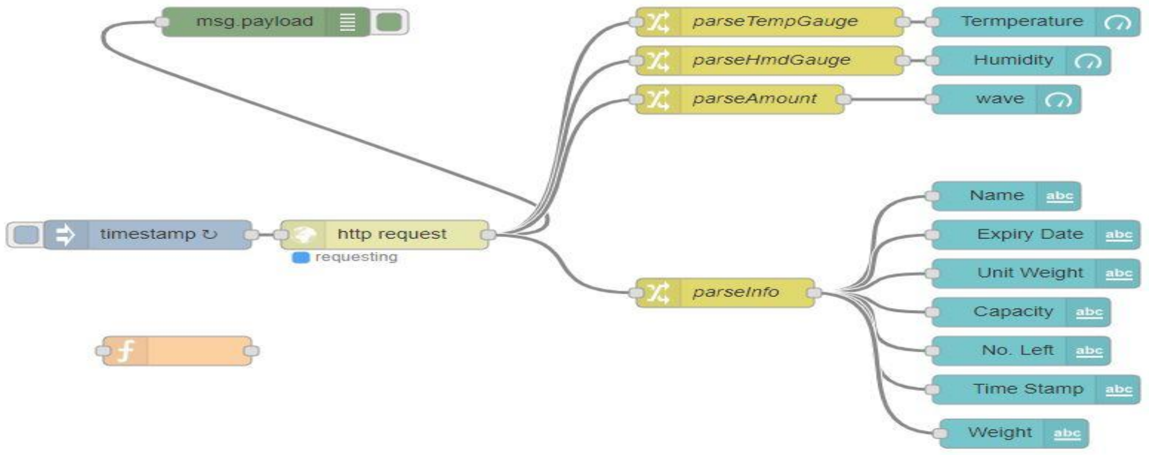
Background :

- The Raspberry Pi initiates the RFID reader to read the UID of the scanned RFID tag and hosts this UID data over a TCP socket which starts listening for any request.
- The Pi connects to the SCION network as a server AS node , establishes a UDP connection and starts listening for any request.
- The SCION server AS node establishes a TCP connection over the defined port to access the UID data and also reads the weight values from the sensor unit on any request from the SCION client AS node.
- The VM connects to the SCION network as a SCION client node, hosts a HTTP connection over the port 4000 and starts listening for any request.
- The Node-Red Dashboard tool initiates the request process flow by making a HTTP GET request to this defined port.

Chronological sequence of events :

- The Node-Red dashboard tool sends a HTTP GET request to the defined port 4000.
- The SCION Client AS node on receiving this HTTP request, sends a request to the SCION Server AS.
- The SCION Server AS node on receiving this request from the SCION client AS node, sends a sample data packet as a request to the TCP connection and gets back the UID data packet as the TCP response if a RFID tag is detected.
- If the UID obtained is valid, then the SCION Server AS reads the real time weight values from the sensor unit and appends the same to the UID data packet which is then made as JSON format.
- The SCION Server AS sends this JSON data packet as the response to the respective SCION Client AS
- The SCION Client AS node on receiving this JSON data packet as response, exposes this JSON data packet over the port 4000.
- The Node-Red Dashboard tool now accesses this JSON data packet and it then processes, parses and displays this information using several node functions.

Implementation : Node-Red Dashboard as GUI



Implementation : Node-Red Dashboard as GUI

Developing a Node-Red flow based diagram which visualises the real time inventory status at the client side.

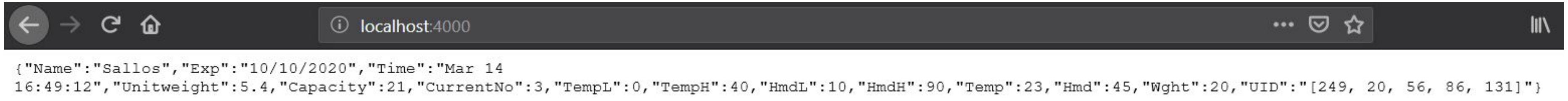
Node Red accepts data packet in JSON Object format as message payloads. Several function nodes are used to parse and display the data segments as follows :

- Timestamp node : to stamp the sequence of time.
- Http input type node : sends a HTTP GET request to access the JSON object hosted over the port 4000.
- Change type nodes : parse the several product information from the message payload.
- Dashboard type nodes : displays these parsed product information using different graphical representations.

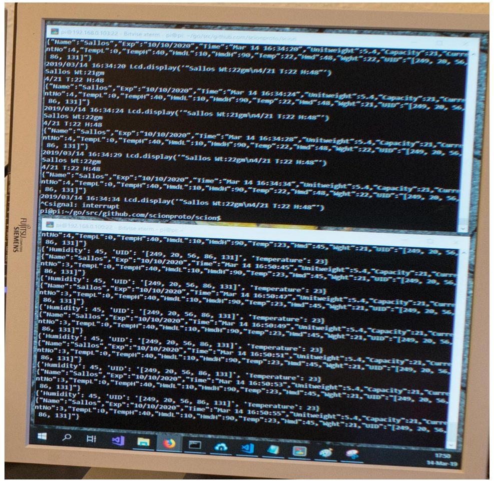
Challenges faced

- Installation of SCION on raspberry Pi 3 B+ model as the scion image for the latest version of the pi was not available.
- Building a platform for the sensor unit comprising the load cell, HX711 amplifier, Raspberry Pi and the RFID reader module.
- SCION Beacons reception failures due to time mismatch and revocation issues.
- Programming with Go.
- Learning to create flow based diagrams in Node-Red.

Results

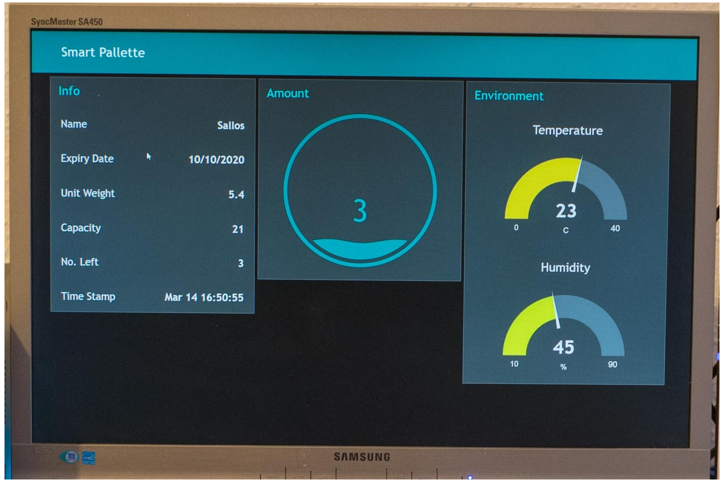


```
{"Name":"Sallos","Exp":"10/10/2020","Time":"Mar 14
16:49:12","Unitweight":5.4,"Capacity":21,"CurrentNo":3,"TempL":0,"TempH":40,"HmdL":10,"HmdH":90,"Temp":23,"Hmd":45,"Wght":20,"UID":"[249, 20, 56, 86, 131]"}
```


pi@192.168.0.103:22 - Bitvise xterm - pi@pi: ~/go/src/github.com/scionproto/scion
pi@192.168.0.100:22 - Bitvise xterm - pi@pi: ~
FUJITSU SIEMENS
17:50
14-Mar-19

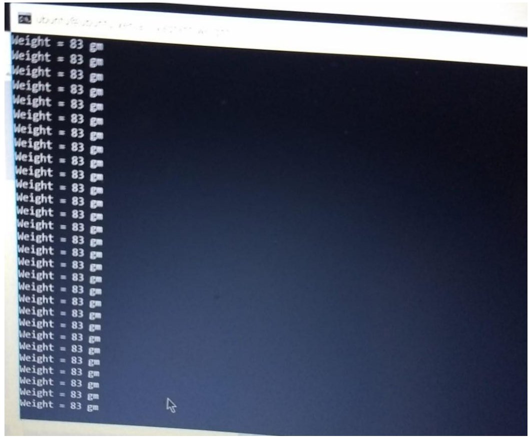
```
{"Name":"Sallos","Exp":"10/10/2020","Time":"Mar 14 16:34:20","Unitweight":5.4,"Capacity":21,"Curre
ntNo":4,"TempL":0,"TempH":40,"HmdL":10,"HmdH":90,"Temp":22,"Hmd":48,"Wght":22,"UID":"[249, 20, 56,
 86, 131]"}
2019/03/14 16:34:20 Lcd.display('"Sallos Wt:22gm\n4/21 T:22 H:48"')
Sallos Wt:21gm
4/21 T:22 H:48
{"Name":"Sallos","Exp":"10/10/2020","Time":"Mar 14 16:34:24","Unitweight":5.4,"Capacity":21,"Curre
ntNo":4,"TempL":0,"TempH":40,"HmdL":10,"HmdH":90,"Temp":22,"Hmd":48,"Wght":21,"UID":"[249, 20, 56,
 86, 131]"}
2019/03/14 16:34:24 Lcd.display('"Sallos Wt:21gm\n4/21 T:22 H:48"')
Sallos Wt:22gm
4/21 T:22 H:48
{"Name":"Sallos","Exp":"10/10/2020","Time":"Mar 14 16:34:28","Unitweight":5.4,"Capacity":21,"Curre
ntNo":4,"TempL":0,"TempH":40,"HmdL":10,"HmdH":90,"Temp":22,"Hmd":48,"Wght":22,"UID":"[249, 20, 56,
 86, 131]"}
2019/03/14 16:34:29 Lcd.display('"Sallos Wt:22gm\n4/21 T:22 H:48"')
Sallos Wt:22gm
4/21 T:22 H:48
{"Name":"Sallos","Exp":"10/10/2020","Time":"Mar 14 16:34:34","Unitweight":5.4,"Capacity":21,"Curre
ntNo":4,"TempL":0,"TempH":40,"HmdL":10,"HmdH":90,"Temp":22,"Hmd":48,"Wght":22,"UID":"[249, 20, 56,
 86, 131]"}
2019/03/14 16:34:34 Lcd.display('"Sallos Wt:22gm\n4/21 T:22 H:48"')
^Csignal: interrupt
pi@pi:~/go/src/github.com/scionproto/scion$
```

```
ntNo":4,"TempL":0,"TempH":40,"HmdL":10,"HmdH":90,"Temp":23,"Hmd":45,"Wght":21,"UID":"[249, 20, 56,
 86, 131]"}
{'Humidity': 45, 'UID': '[249, 20, 56, 86, 131]', 'Temperature': 23}
{"Name":"Sallos","Exp":"10/10/2020","Time":"Mar 14 16:50:45","Unitweight":5.4,"Capacity":21,"Curre
ntNo":3,"TempL":0,"TempH":40,"HmdL":10,"HmdH":90,"Temp":23,"Hmd":45,"Wght":21,"UID":"[249, 20, 56,
 86, 131]"}
{'Humidity': 45, 'UID': '[249, 20, 56, 86, 131]', 'Temperature': 23}
{"Name":"Sallos","Exp":"10/10/2020","Time":"Mar 14 16:50:47","Unitweight":5.4,"Capacity":21,"Curre
ntNo":3,"TempL":0,"TempH":40,"HmdL":10,"HmdH":90,"Temp":23,"Hmd":45,"Wght":21,"UID":"[249, 20, 56,
 86, 131]"}
{'Humidity': 45, 'UID': '[249, 20, 56, 86, 131]', 'Temperature': 23}
{"Name":"Sallos","Exp":"10/10/2020","Time":"Mar 14 16:50:49","Unitweight":5.4,"Capacity":21,"Curre
ntNo":3,"TempL":0,"TempH":40,"HmdL":10,"HmdH":90,"Temp":23,"Hmd":45,"Wght":21,"UID":"[249, 20, 56,
 86, 131]"}
{'Humidity': 45, 'UID': '[249, 20, 56, 86, 131]', 'Temperature': 23}
{"Name":"Sallos","Exp":"10/10/2020","Time":"Mar 14 16:50:51","Unitweight":5.4,"Capacity":21,"Curre
ntNo":3,"TempL":0,"TempH":40,"HmdL":10,"HmdH":90,"Temp":23,"Hmd":45,"Wght":21,"UID":"[249, 20, 56,
 86, 131]"}
{'Humidity': 45, 'UID': '[249, 20, 56, 86, 131]', 'Temperature': 23}
{"Name":"Sallos","Exp":"10/10/2020","Time":"Mar 14 16:50:53","Unitweight":5.4,"Capacity":21,"Curre
ntNo":3,"TempL":0,"TempH":40,"HmdL":10,"HmdH":90,"Temp":23,"Hmd":45,"Wght":21,"UID":"[249, 20, 56,
 86, 131]"}
{'Humidity': 45, 'UID': '[249, 20, 56, 86, 131]', 'Temperature': 23}
{"Name":"Sallos","Exp":"10/10/2020","Time":"Mar 14 16:50:55","Unitweight":5.4,"Capacity":21,"Curre
ntNo":3,"TempL":0,"TempH":40,"HmdL":10,"HmdH":90,"Temp":23,"Hmd":45,"Wght":21,"UID":"[249, 20, 56,
 86, 131]"}
```



Info

Name	Sallos
Expiry Date	10/10/2020
Unit Weight	5.4
Capacity	21
No. Left	3
Time Stamp	Mar 14 16:50:55



```
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
Weight = 83 gm
```

References

Images

- Raspberry Pi : https://venturebeat.com/2018/03/14/the-new-raspberry-pi-3-model-b-microcomputer-promises-more-speed/
- Load cell-1kg : https://www.sparkfun.com/products/14729
- Python Logo : https://www.python.org/
- Go Logo : https://hackernoon.com/the-beauty-of-go-98057e3f0a7d
- Node-Red : https://hackernoon.com/the-beauty-of-go-98057e3f0a7d
- Sample pin diagram : https://www.researchgate.net/figure/Fig1-shows-the-block-of-a-Proposed-System-which-consists-of-Load-Cell-HX711-Raspberry_fig3_322653199

Thanks for your attention