INTELLIGENT CARGO MANAGEMENT USING INTERNET OF THINGS

DATE : 31.08.2020

DONE BY : E SAI JASWANTH

1) INTRODUCTION :

1.1 OVERVIEW :

The system aims at focusing of the sensors deployed inside the truck and the gateways which communicate to the monitoring center . The sensors are deployed inside the truck to monitor the freshness of the product.

1.2 PURPOSE :

The main purpose is that the information is revealed at different levels to the truck drivers, suppliers and users. The truck drivers will be provided with a route to deliver the products in the specified warehouse and if there is a change in the specified route due to fall in the freshness levels the change in freight will be intimated to the truck driver. The consumers will be provided with the freshness scale of the product.

2) LITERATURE SURVEY :

2.1 EXISTING PROBLEM :

The problems of the existing system include :

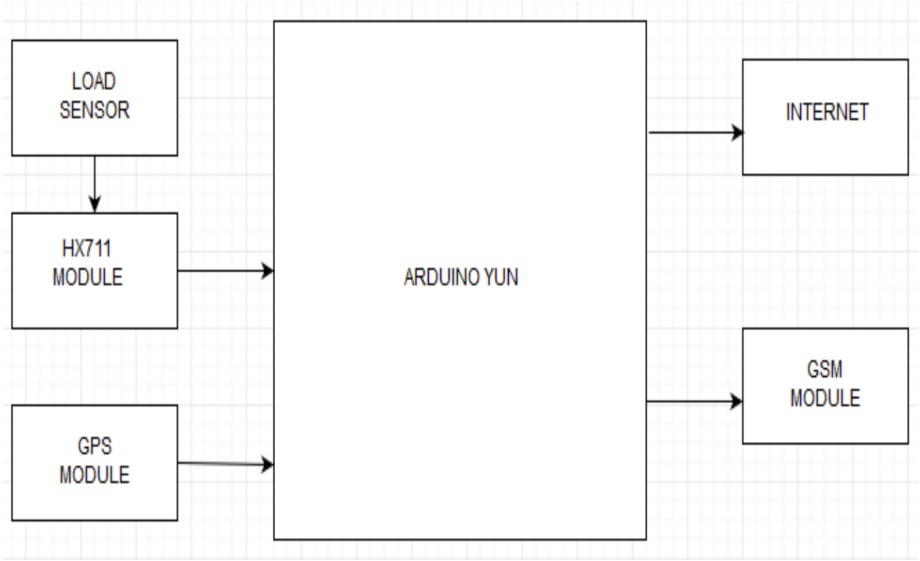
- Damage and file retreival : System of filing , file retrieve problem , file processing and delivery of progress details to clients such as files are exposed to damage and are difficult or impossible to recover.
- Access and information channel : Customer need to constantly check on the progress of their goods and therefore need a reliable channels for monitoring and accessing information on real time or online .

2.2 PROPOSED SOLUTION :

- Damage and file retreival : The files storing the data must be given strong password to make sure no unauthorised user accesses the files of the cargo details .
- Access and information channel : A time-to-time information regarding the goods and the channels that are monitoring that can be done using an mobile application .

3) THEORITICAL ANALYSIS :

3.1 BLOCK DIAGRAM :



3.2 SOFTWARE DESIGN :

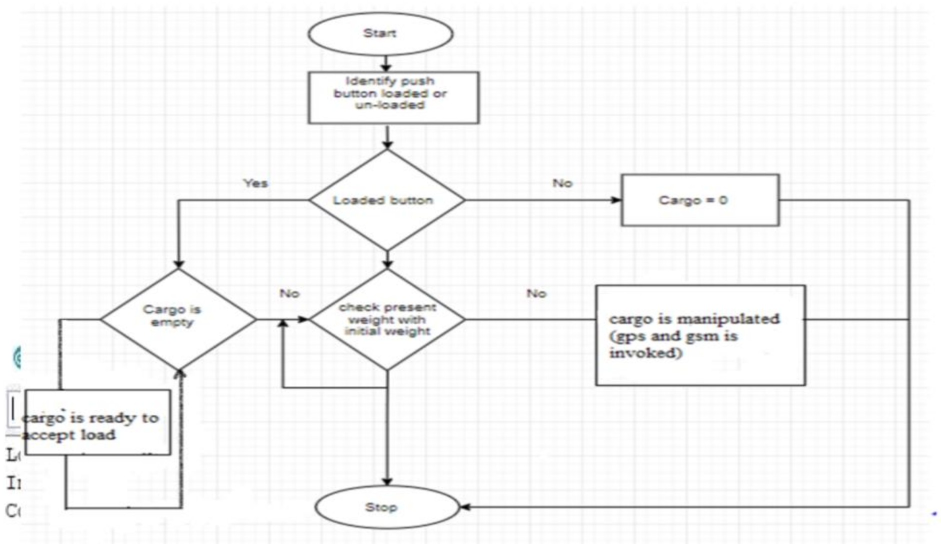
Software components used are : 1.IBM cloud 2.IBM IOT platform 3.IBM Watson 4.node-red 5.Python IDLE 6.MIT app inventor.

4) EXPERIMENTAL INVESTIGATIONS :

An initiation of our project we will check and tear weight of a weighing panel using load cell & HX711 module. Identifying exact weight of cargo or package which is placed above weighing panel using load cell &hx711 module. The identified weight of cargo is stored in database which is present remotely through internet. Speed of cargo movement is detected using gps module. As and when the cargo is in movement with a minimum speed limit, the weight of cargo will be recursively checked with exact value stored in database through programming logic. . Recursively keep on detecting speed using gps module. As on when the speed gets into minimum limit, keep tracking the weight of cargo. . The monitored values are sent in 2 forms:-

- Sms using gsm module.
- Database which is present remotely using internet

5) FLOWCHART :


Start
Identify push button loaded or un-loaded
Yes
Loaded button
No
Cargo = 0
No
Cargo is empty
check present weight with initial weight
No
cargo is manipulated (gps and gsm is invoked)
cargo is ready to accept load
Stop

6) RESULT :

This was built using IBM watson assistant, IOT sensor simulator to check the cargo management .

7) ADVANTAGES & DISADVANTAGES

ADVANTAGE:

- Manipulation of cargo can be easily tracked.
- Cargo will be in surveillance during the transition between source and destination.

DISADVANTAGE:

- Internet connectivity is mandatory.

8) APPLICATIONS :

- Allows exchanging data from one end to other .
- Tacking the passage of the goods on mobile interface .

9) CONCLUSION :

The cargo management app is available on your phones. Any time, any place it can be accessible. It is very feasible and eco-friendly. This helps to detect any problem in the cargos and can be rectified when necessary .

10) FUTURE SCOPE :

With more High-end hardware and software, the cargo management can be customized and can be upgraded and improved for more efficiency and success rate in future .

11) BIBLIOGRAPHY :

- Github
- Youtube
- Smart internz

12) APPENDIX

A .SOURCE CODE :

1. DHT11 SENSOR TO MEASURE HUMIDITY AND TEMPERATURE:

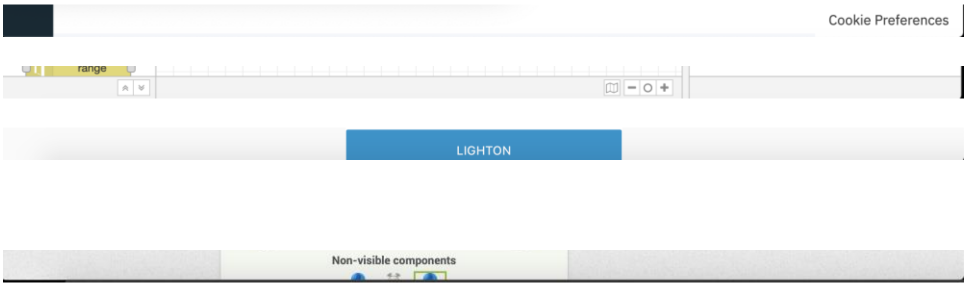
```
#include <dht.h>
 #define dht_apin A0
 dht DHT;

void setup(){

Serial.begin(9600); delay(500);

Serial.println("DHT11 Humidity & temperature Sensor\n\n"); delay(1000);

}

void loop(){ DHT.read11(dht_apin);

Serial.print("Current humidity = ");

Serial.print(DHT.humidity);

Serial.print("% ");

Serial.print("temperature = ");

Serial.print(DHT.temperature);

Serial.println("C "); delay(5000);

}
```

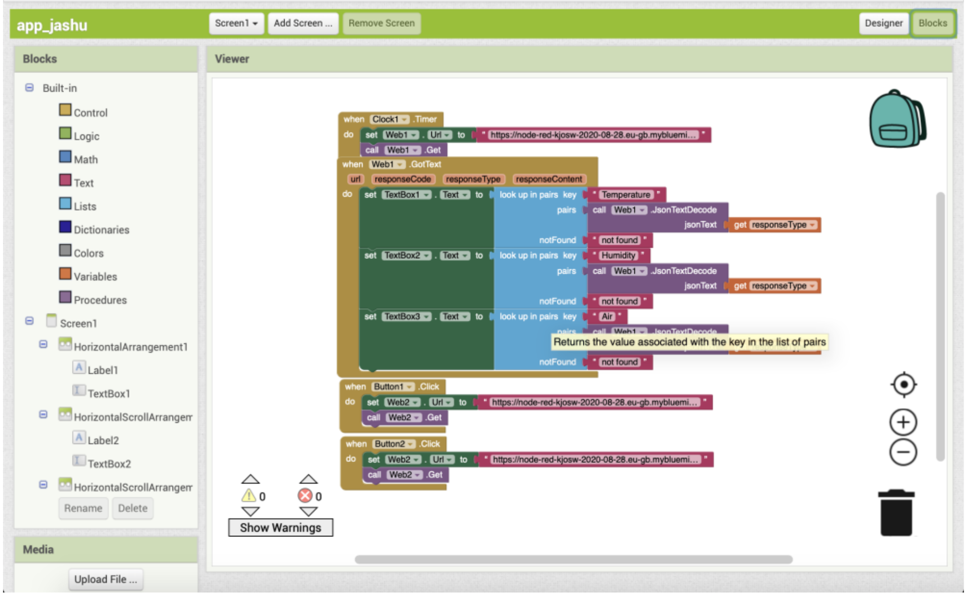
2. AIR QUALITY MONITORING :

```
// Air Quality monitoring

 #include <YunClient.h>

#include <ThingerYun.h>

// The Thinger.io cloud

#define USERNAME "YOUR_USER_NAME"

#define DEVICE_ID "YOUR_DEVICE_ID"
```

```
#define DEVICE_CREDENTIAL "YOUR_CREDENTIAL"
ThingerYun thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);
void setup() {
pinMode(A0, INPUT);
  pinMode(A1, INPUT);
pinMode(A2, INPUT);
  pinMode(A3, INPUT);
// initialize bridge
  Bridge.begin();
// resource output example (i.e. reading a sensor value, a variable, etc)
thing["LPG"] >> outputValue(analogRead(A0)); // optional
thing["CO2"] >> outputValue(analogRead(A1));
thing["CO"] >> outputValue(analogRead(A2));
thing["AIRQ"] >> outputValue(analogRead(A3));
}
void loop() {
thing.handle();
}
```

B. UI OUTPUT SCREENSHOT :





```
when Clock1 .Timer
do  set Web1 . Url to " https://node-red-kjosw-2020-08-28.eu-gb.mybluemi... "
    call Web1 .Get

when Web1 .GotText
  url  responseCode  responseType  responseContent
do  set TextBox1 . Text to  look up in pairs  key  " Temperature "
                                              pairs  call Web1 .JsonTextDecode
                                                                    jsonText  get responseType
                                           notFound  " not found "
    set TextBox2 . Text to  look up in pairs  key  " Humidity "
                                              pairs  call Web1 .JsonTextDecode
                                                                    jsonText  get responseType
                                           notFound  " not found "
    set TextBox3 . Text to  look up in pairs  key  " Air "
                                              pairs  call Web1 .JsonTextDecode
                                           notFound  " not found "

when Button1 .Click
do  set Web2 . Url to " https://node-red-kjosw-2020-08-28.eu-gb.mybluemi... "
    call Web2 .Get

when Button2 .Click
do  set Web2 . Url to " https://node-red-kjosw-2020-08-28.eu-gb.mybluemi... "
    call Web2 .Get
```

Returns the value associated with the key in the list of pairs