Cold Storage Monitoring Using IOT (internet of things)

**Introduction**

Perishable goods are consumed by almost everyone on a daily basis. These products include fruits, vegetables, dairy products, meat and poultry, fresh food, frozen food, seafood and even pharmaceutical products. Since these goods are temperature-sensitive and their storage time varies from a few days to a few years, these items are stored in temperature-controlled rooms. This is essential to ensure their safety and quality. Moreover, it is also important to ensure that the temperature of the room must never exceed the optimal temperature.

**Purpose:**

**Real-time monitoring:**

Implementation of an IoT solution for cold storage facilities helps to monitor the necessary parameters and adjusts them when deviation occurs from their preset values. This helps to prevent food decay. The solution also sends alerts via SMS text and email whenever an anomaly is detected. Hence, preserving the item and maintaining regulatory compliance becomes easy

**Cold storage temperature monitoring:**

A cold storage temperature monitoring solution includes thermostats and sensors that constantly measure the temperature of a closed system, capture data and send it to a centralized platform over a network. This helps the logistics manager to monitor the shipment remotely and ensure the maintenance of optimum temperature.

The implementation of a cold storage temperature monitoring solution is helpful for climate-sensitive perishable items. Smart warehouse solution is easy to incorporate, convenient to use and ensures that the quality of goods does not degrade in warehouse and shipping.

A cold storage monitoring solution would be most beneficial for:

1)Agriculture industry

2)Blood banks

3)Food & beverage industry etc

**2)Literature Survey**

**Existing Problem:**

a) Problems Only Cold Storage Warehouses Have

b) Different Temperature Zones.

c) Changing Zone Sizes.

d) Cold Temperatures Sap Battery Power.

e) Keeping Workers Warm and Productive.

f) Effect of Cold on Electronics.

g) Maximizing the Cold.

h) Maintaining Product Temperature.

**Proposed Solution**:

a) High humidity cold rooms, Cold Storage for fruit and vegetable storage

b) Low humidity cold rooms, Cold Storage for seed and hygroscopic material storage

c) Humidification and automation systems

d) Unique design for generating low humidity environment using specially designed refrigeration

e) Pre-painted Galvanised Steel (PPGS) or Stainless Steel (SS) metal laminate

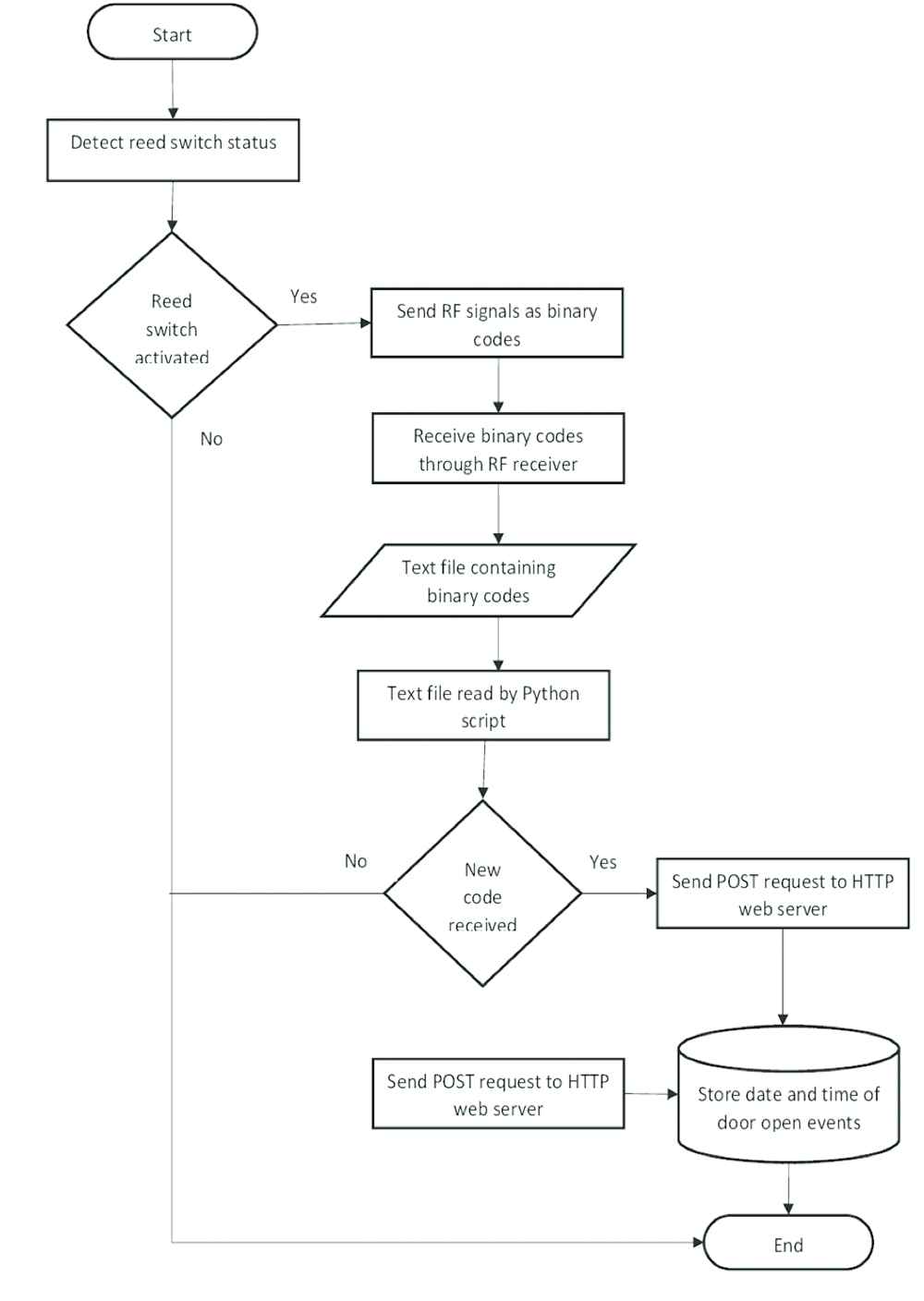
**Hardware / Software designing** :

Python Idle, IBMCloud , MIT app inventor , Nodered,Cloudant DB

**Experimental Investigation:**

The Internet of things(IoT) aims at connecting different objects, things using internet. The rapid development of the Internet of Things motivate use to apply for the food preservation domain such as maintain the quality of fruits and vegetable. In this project a system has been proposed to analyze the ambient conditions under which the food item is being stored. The proposed solution senses the temperature, moisture, light parameters of surrounding environment as these parameters affect nutritional values of food items. In this project we have designed and implemented Raspberry Pi which works as a sensor node for the fruit and vegetable storage house as well as central base station is connected to cloud where MySQL open source database server to support data storage functionalities. The sensor values are stored in the cloud and sent to the base station by connecting to database using its IP address. Then a data fusion model is experimented which takes multiple sensed data as input and produces single fused information or action to be taken as the output.

**Flowchart:**



**Result:**

1. We have analyzed continuous monitoring of temperature, humidity and light intensity and Switching ON the motor fan in case of bad condition of temperature by using mobile app,Storing the sensor data in the database
2. Alerts will be sent to the admin if the sensor parameters exceeds the threshold values.
3. Very less latency in communication from device to cloud with MQTT

**Advantages and Disadvantages:**

Advantages:

A cold storage room can keep fruits and vegetables at the correct temperature while controlling the moisture level to help extend the life of the fresh produce longer until it can be used.

Disadvantages:

a) limited amount of genetic diversity conserved

b)highmaintenancecosts

**Applications:**

a)distribution/logistic center

b)frozen produce

c)processing room

**Conclusion:**

IoT based temperature and humidity detecting device provides an efficient and definitive system for monitoring agricultural parameters. The system also provides a corrective movement or decision-making system. IoT based monitoring of area is a handiest, but it also allows the consumers to research the correct modifications within the surroundings and for taking possible action. It is inexpensive and consumes much less electricity. The Gross Domestic Product (GDP) per capitals in agriculture can be multiplied and helps to add our need parameters

**Future Scope:**

This set up can also control the DC fan, motor, and water levels for supporting farmers. Then the measured values of humidity and temperature values from the Arduino MCU are uploaded to the cloud. Then the collected data are transferred to the farmers live through the GSM to their cell phones. Based on the water level measuring system, the collected data are sending to the farmer‟s cell phone continuously. They can switch on or off their motor based on the collected data from the water level measuring system.

**Appendix:**

A Source Code :

import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

#Provide your IBM Watson Device Credentials

organization = "a9z35z"

deviceType = "raspberrypi"

deviceId = "123456"

authMethod = "token"

authToken = "123456789"

# Initialize GPIO

def myCommandCallback(cmd):

print("Command received: %s" % cmd.data)

try:

deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken}

deviceCli = ibmiotf.device.Client(deviceOptions)

#..............................................

except Exception as e:

print("Caught exception connecting device: %s" % str(e))

sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times

deviceCli.connect()

while True:

hum=random.randint(10,40)

#print(hum)

temp =random.randint(30,80)

light=random.randint(10,50)

#Send Temperature & Humidity to IBM Watson

data = { 'Temperature' : temp, 'Humidity': hum,'lightintensity' : light }

#print (data)

def myOnPublishCallback():

print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % hum,"lightintensity= %s C" % light,"to IBM Watson")

success = deviceCli.publishEvent("DHT11", "json", data, qos=0, on\_publish=myOnPublishCallback)

if not success:

print("Not connected to IoTF")

time.sleep(2)

deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud

deviceCli.disconnect()

By

**Pittu Venkat Reddy**

**P.sainithin**

**Balla Dani Hemanth**

**Kallagunta Ajay Kumar**