IoT Based Cattle Health Monitoring System

Introduction To IoT

Course Project Report

*Submitted by*

Aadi Juvekar - 211010201

Parth Bhandakkar - 211010241

Parth Jhunjhunwala - 211010242

Logo

Description automatically generated

Dr. Shyama Prasad Mukherjee International Institute of

Information Technology, Naya Raipur

July-2022

IoT based cattle health monitoring system

**ABSTRACT**

The Internet of Things is an emerging idea that is transforming real-world objects (things) into smarter devices. This area has started revolutionizing the agriculture industry by providing smart solutions for precision farming, greenhouse management, and livestock monitoring. Good health and wellbeing of animals are essential to dairy cow farms and sustainable production of milk.

Day to day health monitoring of cattle is difficult especially in large farms where workers do not have time to observe each cattle and detect the symptoms of diseases. Our project aims at making such a device which will continuously monitor and track the health of a cattle.

Our project is only applicable if there is limited grazing area because if there is large grazing area then there will be requirement of such a IoT connectivity technology which can cover a large area.

In that system we use sensor technology which maps the special aspects of animal behavior like temperature, heart rate etc. this data is aggregating and reporting to the health care center. This reduces the minimal health inspection and long-term animal healthcare cost.

This research has been undertaken to establish specific sensor technologies as a significant means to monitor animal health and to ensure animal well-being in the fast-changing conditions of automated farms.

**INTRODUCTION**

The influence of technology in recent years and the continued tendency for digitization is being increasingly observed in all areas. Almost every field has digitized parts that enable more efficient and easier management. One of the areas that is being digitized, which is part of the research and subject of the present study, is the field of agriculture, with a particular emphasis on monitoring the health of cattle.

Cattle population is 1009.69 million as of 2022 worldwide. In which, cows take up 85 million. There had been 123.014 thousand CATTLE deaths 75 thousand COW deaths. One of the major reasons for the death of cows is Anaplasmosis. It can be detected in early stages by measuring their temperature, movement pattern and heartrate.

Contributions of this System are as follows:

1. Continuous Health Monitoring

2. IoT-enabled health monitoring devices help in preventing cattle from getting infected.

3. Better yield and prevention of livestock.

**OBJECTIVE**

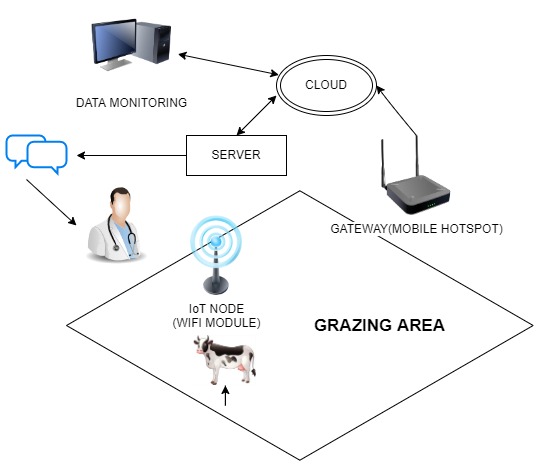
The final objective of this project is to improve the health of cattle thus increase in farm productivity.

**PROPOSED METHODOLOGY**

**A... Architecture Model**

In Health monitoring module, various health sensors collect the required parameters from various body parts of the cattle. The collected information is transmitted to the ThingSpeak via the ESP 8266 WIFI Module.

Then the collected data is compared with the original data and if the collected data lies in the range of original data, then the System continues to work. But if it is not in the range of original data then it gives the notification.

 Fig.0 Architerure

**B... Hardware**

1. In sensor technology, sensors are used to sense the body temperature of the cattle. LM35 is used as the body temperature sensor. The normal body temperature of the cattle is 36.5-39.5 degree Celsius. If the temperature of the cattle is less than the normal temperature the cattle may suffer from diseases like milk fever, indigestion, and poisoning. If the temperature of the cattle is more than the normal temperature the cattle may suffer from diseases like anthrax, influenza, foot, and mouth diseases etc.

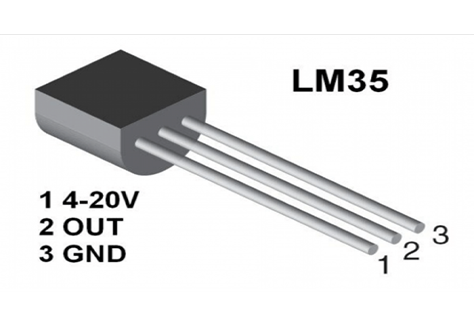


Fig.1 LM35

2. The normal heartbeat of an adult cattle is between 48 and 84 beats per minute. This sensor will detect stress as well as animal's anxiety. The heartbeat sensor generally used is a stethoscope. It is kept behind cow's elbow to listen over the left side of the cow's chest. The elevation of the heart rate can lead to a sign of pain

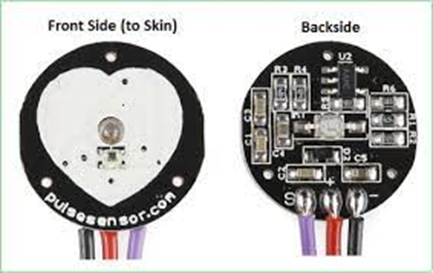


Fig.2 Pulse Sensor

3. Motion sensors use electronic accelerometers to record the lying, walking, and standing behaviour of animals. These sensors are used with aim of monitoring the movement behaviour of cattle for improving animals’ health and production. If the data is automatically collected on large numbers and in continuous period, then the health of the cattle can be improved to a large extent. This data can be used by the stakeholders for management and disease control decision.



Fig.3 Accelerometer

4. We can send the animal health graph to the doctor’s mobile using ESP8266 WIFI module. This WIFI module sends the signals through the IOT technology. So, by observing this graph doctor can talk about the animal health. Arduino UNO has enough memory to transform the signs arriving from the data gaining unit through sensors into a ESP8266 WIFI module for communication and then the signals are given to the software for examining and displaying the data.



Fig.4 Esp 8266

5. The Arduino UNO micro controllers are readily available for a wide variety of applications. The Arduino UNO microcontroller cost is low. Now a days instead of PIC microcontroller Arduino UNO is used because it is more flexible. The signals arriving from the sensors are finally sent to the WIFI module through Arduino and from the WIFI module to ThingSpeak.

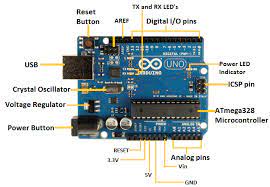


Fig.5 Arduino Uno

**C. Sensor Node**

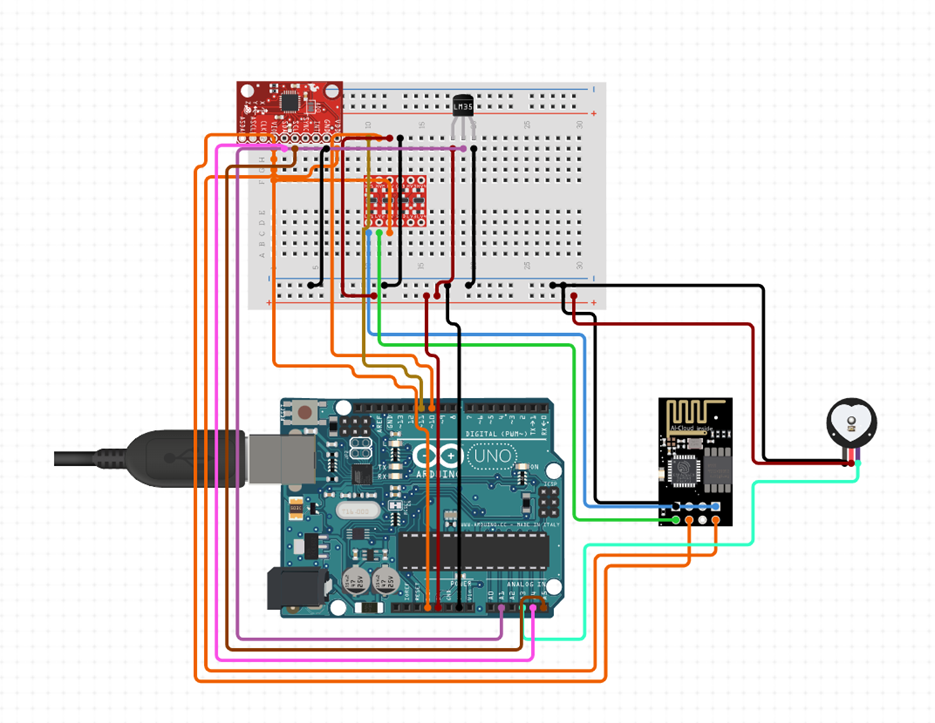
1. The sensor node consists of a Temperature Sensor, Pulse Sensor, Accelerometer, Arduino UNO, and Esp8266 Wi-Fi module.

2. The power supply is 5v which is given by a power bank.

3. IoT Gateway is a mobile hotspot or Wi-Fi-router.

4. The cloud service used is Thing Speak Cloud, which is an exclusive cloud service provider for IoT Applications.

**Sensor Node Diagram**

 Fig.6 Circuit

**Block Diagram**

Diagram

Description automatically generated

Fig.7 Sensor NODE

**D... Working Principle**

1. The sensor node collects Temperature, Heartbeat and x, y, z Coordinates in real time in analog format.

2. The sensed data is then processed to their specific values i.e., Voltage changes to temperature, Amount of reflected light to bpm, input signals to coordinates by using an Arduino UNO.

3. Then this processed data is sent to ThingSpeak cloud.

4. Using online website, a health specialist can analyse real time temperature, Heart rate and coordinates of the cattle.

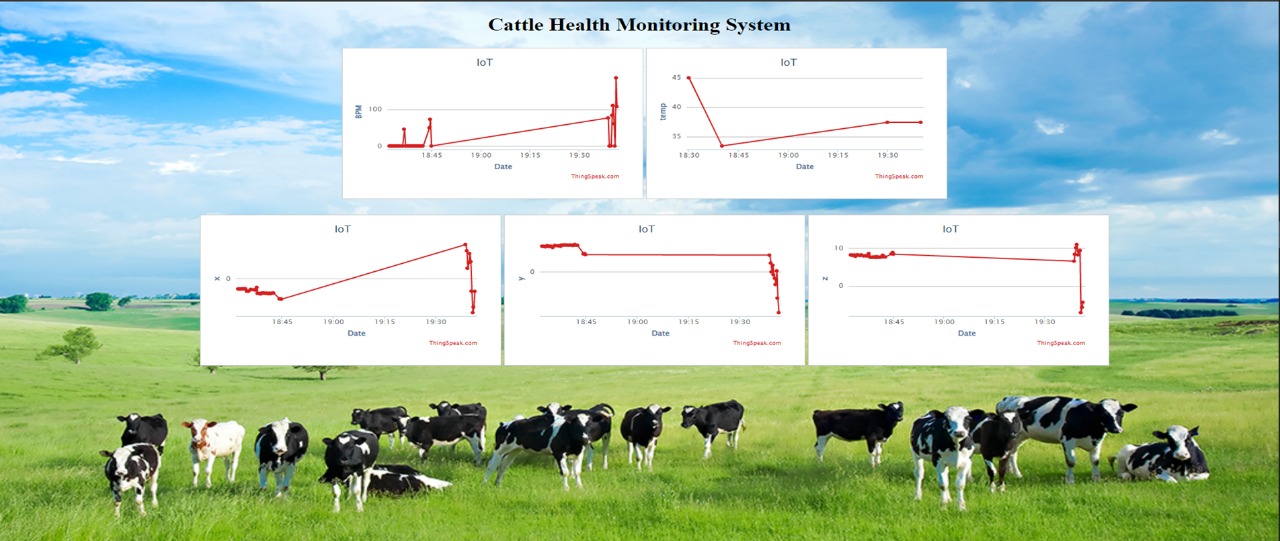


Fig.8 Webpage Interface

**E... Flow Chart**

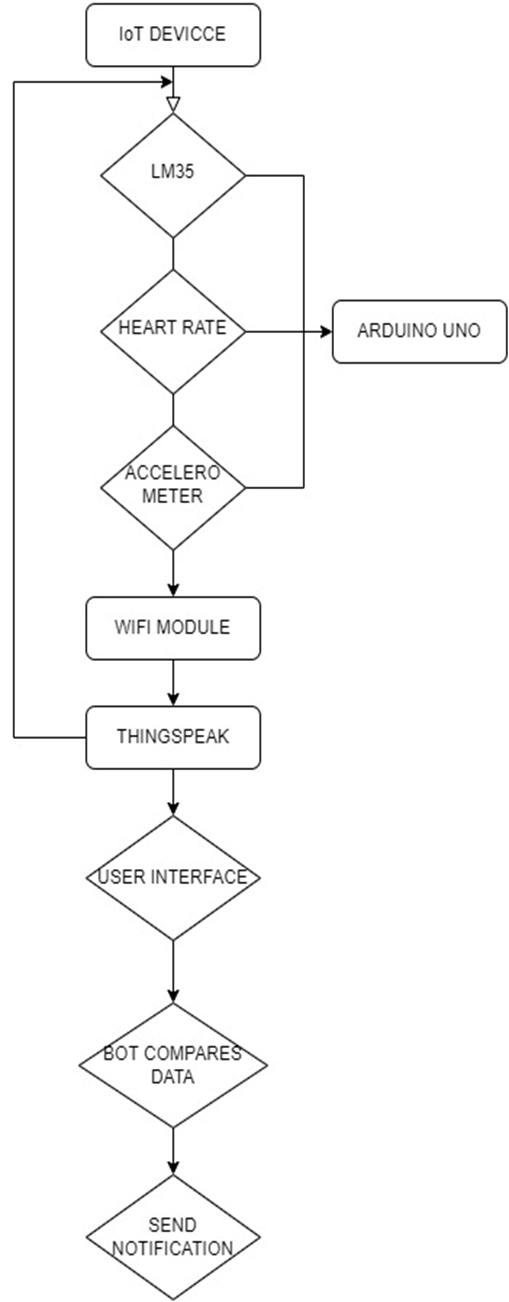


Fig.9 Working

**DATA ANALYSIS**

We have created a small bot in python using pywhatkit library of python. It basically collects the data from ThingSpeak and analyse it.

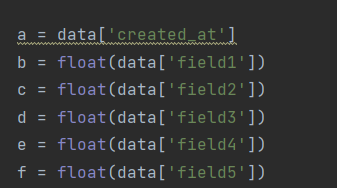


Fig.10 Bot Code

Here b, c, d, e, f are fields - heart rate, x, y, z coordinate, temperature respectively.

Text

Description automatically generated

Fig.11 Bot Code

If heart rate is not equal to zero, then it will go further and analyse the temperature and Heart rate. If heart rate is in between 48 and 84 bpm and temperature is between 36 and 40 degrees Celsius then it will send a notification to the doctor (on any given phone number) and system will play a sound “Cow is Not Well”. After implementation system will go on sleep for 5 minutes.

Text

Description automatically generated

Fig.12 Bot Code

If field B (i.e., Heart Rate) is not detected then System will send the notification as “Cow is Dead”.

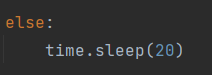


Fig.13 Bot Code

And if both the above cases are not satisfied then it will continue to analyse the data after every 20 seconds.

**Result**

The prototype is used to measure the continuous Temperature, Heart Rate and Coordinates. The prototype was deployed on a cow and the measurements were recorded.



Fig.14 Result

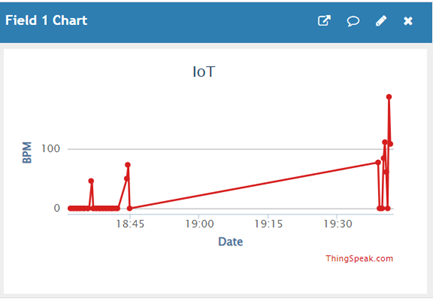
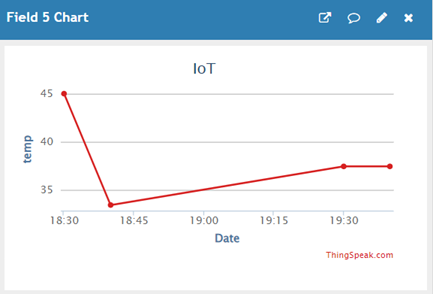
 

Fig.15 Heart Rate Fig.15 Temperature

Chart, line chart

Description automatically generated Graphical user interface

Description automatically generated

Fig.16 X coordinates Fig.17 Y coordinates

Graphical user interface, chart, line chart

Description automatically generated with medium confidence

Fig.18 Z coordinates

 A group of people posing for a photo

Description automatically generated with medium confidence

Fig.19 Data Collection

Graphical user interface, application

Description automatically generated

Fig.20 Real Data Collected

As we can see here two data is collected with heart rate as 77 and 83 beats per minute and temperature as 36.7 and 37.1 degree Celsius which lies in the normal range.

**Conclusion**

Our project presents an IoT architecture for a real-time cattle health monitoring system composed of a hardware prototype. In our prototype, we have used a Temperature Sensor, Heart Rate Sensor, Accelerometer, ESP. 8266 Wi-Fi module, Arduino Uno Board. The Temperature, Heart rate and Accelerometer sensor senses the data and pushes it to the Arduino Uno microcontroller. In the microcontroller, we process the data and using the Wi-Fi module we push it to the Thing Speak cloud server. From there we access the data using an Interface.

**Future Scope**

This is a basic level prototype of the proposed idea. More advancements can be done on this idea by using more accurate sensors and implementing it on an android software.

We can also make it more compact by using a compact micro controller board like Seeduino boards.

The prototype is powered with a power bank, in future we can also use a 5-9 v battery.

In future this project can also help in sensing more severe diseases just by adding more sensors and changing the analysing model and can also give the cure by itself by completely analysing the data.

**Learning Outcomes**

1. Basic building blocks of IoT Infrastructure.
2. The value that IoT provides in our day-to-day life.
3. We learned how to solve real-life problems while dealing with an IoT Solution.
4. Learned how to program a microcontroller board according to our specific need
5. Dealing with various types of sensors and basic electronic components such as LCD display, breadboard, etc.
6. Learned how to push data to a cloud server and process it to a useful source of information.
7. Basics of developing a mobile application for IoT.
8. Real-life deployment of an IoT solution

**References**

1.https://docs.google.com/document/d/1MeGZmgvrHx-X\_TpXTbDC5K9PKk7Sxtkl9w-M01o986c/edit

2.https://docs.google.com/document/d/1dxhpPEA5Yx8ZJ2lrxHlYRCScSSKksWTEsUeYqpIq71Y/edit

3.https://www.researchgate.net/publication/337197924\_An\_IoT\_Solution\_for\_Cattle\_Health\_Monitoring

4.https://www.mdpi.com/2073-431X/11/5/79/pdf?version=1652351

**Source Code**

#include <SoftwareSerial.h> //Software Serial library

SoftwareSerial espSerial(3, 4); //Pin 2 and 3 act as RX and TX. Connect them to TX and RX of ESP8266

#define DEBUG true

#define USE\_ARDUINO\_INTERRUPTS true

String mySSID = "IoT"; // WiFi SSID

String myPWD = "1234qwer"; // WiFi Password

String myAPI = "6XOBGRZVNNWBTV64"; // API Key

String myHOST = "api.thingspeak.com";

String myPORT = "80";

String myFIELD = "field1";

String myFIELD2 = "field2";

String myFIELD3 = "field3";

String myFIELD4 = "field4";

String myFIELD5 = "field5";

#include <OneWire.h>

#include <DallasTemperature.h>

#include <PulseSensorPlayground.h>

#include <Wire.h>

#include <Adafruit\_Sensor.h>

#include <Adafruit\_ADXL345\_U.h>

#define ONE\_WIRE\_BUS 2

#define SensorPin A0

OneWire oneWire(ONE\_WIRE\_BUS);

const int LED13 = 13;

int Threshold = 550;

PulseSensorPlayground pulseSensor; // Creates an object

Adafruit\_ADXL345\_Unified accel = Adafruit\_ADXL345\_Unified();

//unsigned long millisCurrent;

//unsigned long millisLast = 0;

//unsigned long millisElapsed = 0;

//int sampleBufferValue = 0;

void setup(){

Serial.begin(9600); //init serial 9600

espSerial.begin(9600);

espData("AT+RST", 1000, DEBUG); //Reset the ESP8266 module

espData("AT+CWMODE=1", 1000, DEBUG); //Set the ESP mode as station mode

espData("AT+CWJAP=\""+ mySSID +"\",\""+ myPWD +"\"", 1000, DEBUG); //Connect to WiFi network

}

void loop(){

long measurement1 = getPulseValue();

float measurement2 = getxaccelerationValue();

float measurement3 = getyaccelerationValue();

float measurement4 = getzaccelerationValue();

long measurement5 = getTempValue();

delay(50);

// Serial.print("measurement = ");

Serial.println(measurement1);

Serial.println(measurement2);

Serial.println(measurement3);

Serial.println(measurement4);

Serial.println(measurement5);

/\*if (measurement > 1000){

digitalWrite(LED\_Pin, HIGH);

}

else{

digitalWrite(LED\_Pin, LOW);

}\*/

String sendData = "GET /update?api\_key="+ myAPI +"&"+ myFIELD +"="+String(measurement1)+"&"+ myFIELD2 +"="+String(measurement2)+"&"+ myFIELD3 +"="+String(measurement3)+"&"+ myFIELD4 +"="+String(measurement4)+"&"+ myFIELD5 +"="+String(measurement5);

espData("AT+CIPMUX=1", 1000, DEBUG); //Allow multiple connections

espData("AT+CIPSTART=0,\"TCP\",\""+ myHOST +"\","+ myPORT, 1000, DEBUG);

espData("AT+CIPSEND=0," +String(sendData.length()+4),1000,DEBUG);

espSerial.find(">");

espSerial.println(sendData);

}

long getPulseValue(){

delay(10);

pulseSensor.analogInput(SensorPin);

pulseSensor.blinkOnPulse(LED13);

pulseSensor.setThreshold(Threshold);

if (pulseSensor.begin()) {

Serial.println("PulseSensor object created!");

}

int myBPM = pulseSensor.getBeatsPerMinute(); // Calculates BPM

if (pulseSensor.sawStartOfBeat()) { // Constantly test to see if a beat happened

Serial.println("♥ A HeartBeat Happened ! "); // If true, print a message

Serial.print("BPM: ");

Serial.println(myBPM); // Print the BPM value

}

return myBPM;

}

float getxaccelerationValue(){

delay(10);

if(!accel.begin())

{

Serial.println("No ADXL345 sensor detected.");

while(1);

}

sensors\_event\_t event;

accel.getEvent(&event);

float xacceleration = event.acceleration.x;

Serial.print("X: "); Serial.print(event.acceleration.x); Serial.print(" ");

return xacceleration;

}

float getyaccelerationValue(){

delay(10);

if(!accel.begin())

{

Serial.println("No ADXL345 sensor detected.");

while(1);

}

sensors\_event\_t event;

accel.getEvent(&event);

float yacceleration = event.acceleration.y;

Serial.print("Y: "); Serial.print(event.acceleration.y); Serial.print(" ");

return yacceleration;

}

float getzaccelerationValue(){

delay(10);

if(!accel.begin())

{

Serial.println("No ADXL345 sensor detected.");

while(1);

}

sensors\_event\_t event;

accel.getEvent(&event);

float zacceleration = event.acceleration.z;

Serial.print("Z: "); Serial.print(event.acceleration.z); Serial.print(" ");

return zacceleration;

}

float getTempValue(){

delay(10);

int val;

int tempPin = 1;

val = analogRead(tempPin);

float mv = ( val/1024.0)\*5000;

float cel = mv/10;

float farh = (cel\*9)/5 + 32;

Serial.print("TEMPRATURE = ");

Serial.print(cel);

return cel;

}

String espData(String command, const int timeout, boolean debug)

{

Serial.print("AT Command ==> ");

Serial.print(command);

Serial.println(" ");

String response = "";

espSerial.println(command);

long int time = millis();

while ( (time + timeout) > millis())

{

while (espSerial.available())

{

char c = espSerial.read();

response += c;

}

}

if (debug)

{

//Serial.print(response);

return response;

}