



IJIRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2320-6710



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 10, Issue 7, July 2021

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 7.569



9940 572 462



6381 907 438



ijirset@gmail.com



www.ijirset.com

Design and Implementation of an Advanced System to Measure Steroid Content in Milk and Application in Dairy Forms Using IOT

M.Shobha¹, M.Jeyasravani², A.Pavithra³, M.Likitha⁴, B.Niranjan Reddy⁵, P.Naveen⁶

Assistant Professor, Dept. of ECE, Siddarth Institute of Engineering and Technology (Autonomous), Puttur, India¹

UG Students, Dept. of ECE, Siddarth Institute of Engineering and Technology (Autonomous), Puttur, India²³⁴⁵⁶

ABSTRACT: This project presents the design and development of Arduino controller based system to detect the parameters of milk and meats. The parameters include pH, cholesterol, temperature for milk and freshness and weight for meat. The pH sensor is used to measure the quality, ldr is used to measure the cholesterol and lm-35 is to measure the milk temperature. The sensors are interfaced with the Arduino controller. The software developed enables to read the parameters and display them on the LCD. This is a low cost and efficient tool to measure the content of the milk. Also with the help of IOT (Internet of Things) process the milk industry should be able to send the real time reading information of milk to the government so that it helps to overcome the illegal things such as milk quality during the production of milk pocket. A new analytical method based on ultra-high performance liquid chromatography (UHPLC) coupled to Orbit rap high resolution mass spectrometry (Orbit rap-HRMS) has been developed for the determination of steroid hormones and their metabolite in three meat samples (chicken, pork and beef). Here for measuring the content we are using gas sensor and load cell. Gas sensor is to measure the freshness of meat and load cell to measure the weight of meat. By this we will get to know the quality of meat

KEYWORDS: Ultrasonic Sensor; PH Sensor; Aurduino mega; RFID reader; gas sensor; GPRS; LDR; temperature Sensor; load cell

I. INTRODUCTION

An embedded system is a special-purpose computer system designed to perform one or a few dedicated functions, sometimes with real-time computing constraints. It is usually embedded as part of a complete device including hardware and mechanical parts. In contrast, a general-purpose computer, such as a personal computer, can do many different tasks depending on programming. Embedded systems have become very important today as they control many of the common devices we use. Since the embedded system is dedicated to specific tasks, design engineers can optimize it, reducing the size and cost of the product, or increasing the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale. Physically embedded systems range from portable devices such as digital watches and MP3 players, to large stationary installations like traffic lights, factory controllers, or the systems controlling nuclear power plants. Complexity varies from low, with a single microcontroller chip, to very high with multiple units, peripherals and networks mounted inside a large chassis or enclosure. In general, "embedded system" is not an exactly defined term, as many systems have some element of programmability. An embedded system is some combination of computer hardware and software, either fixed in capability or programmable, that is specifically designed for a particular kind of application device. Industrial machines, automobiles, medical equipment, cameras, household appliances, airplanes, vending machines, and toys (as well as the more obvious cellular phone and PDA) are among the myriad possible hosts of an embedded system. Embedded systems that are programmable are provided with a programming interface, and embedded systems programming is a specialized occupation. Certain operating systems or language platforms are tailored for the embedded market, such as Embedded Java and Windows XP Embedded. However, some low-end consumer products use very inexpensive microprocessors and limited storage, with the application and operating system both part of a single program. The program is written permanently into the system's memory in this case, rather than being loaded into RAM (random access memory), as programs on a personal computer are.

II. RELATEDWORK

Now-a-days the milk adulteration is mostly detected using various chemical tests. These methods are tedious, time consuming and costly. Also the knowledge of the tests is necessary. The nutritional value of milk to human health needs no introduction; it also has traditional impact on Indian society. At the same time it is alarming that many vendors adulterating it with water, detergents, caustic soda, starch, formalin, urea, ammonium sulphate, sodium carbonate which have harmful effect on the human health [1, 2]. The greed for money has pushed them to the extent of producing synthetic milk which has no nutritional content. A new analytical method based on ultra-high performance liquid chromatography (UHPLC) coupled to Orbit rap high resolution mass spectrometry (Orbit rap-HRMS) has been developed for the determination of steroid hormones (hydrocortisone, cortisone, progesterone, prednisone, prednisolone, testosterone, melengesterol acetate, hydrocortisone-21-acetate, cortisone-21-acetate, testosterone propionate, 17 α -methyl testosterone, 6 α -methylprednisolone and medroxyprogesterone) and their metabolite (17 α -hydroxyprogesterone) in three meat samples (chicken, pork and beef). Two different extraction approaches were tested (QuEChERS "quick, easy, cheap, effective, rugged and safe" and "dilute and shoot"), observing that the QuEChERS method provided the best results in terms of recovery. A clean-up step was applied comparing several sorbents, obtaining the best results when floristic and aluminum oxide were used. The optimized method was validated, obtaining suitable results for all validation parameters in the three meat matrices evaluated. Recovery values ranged from 70% to 103% (except for prednisone in beef samples), meanwhile repeatability and reproducibility were obtained at values lower than 18% and 21%, respectively. This project presents the design and development of Arduino controller based system to detect the parameters of milk and meats. The parameters include pH, cholesterol, temperature for milk and freshness and weight for meat. The pH sensor is used to measure the quality, ldr is used to measure the cholesterol and lm-35 is to measure the milk temperature. The sensors are interfaced with the Arduino controller. The software developed enables to read the parameters and display them on the LCD. This is a low cost and efficient tool to measure the content of the milk. Also with the help of IOT (Internet of Things) process the milk industry should be able to send the real time reading information of milk to the government so that it helps to overcome the illegal things such as milk quality during the production of milk pocket. A new analytical method based on ultra-high performance liquid chromatography (UHPLC) coupled to Orbit rap high resolution mass spectrometry (Orbit rap-HRMS) has been developed for the determination of steroid hormones and their metabolite in three meat samples (chicken, pork and beef). Here for measuring the content we are using gas sensor and load cell. Gas sensor is to measure the freshness of meat and load cell to measure the weight of meat. By this we will get to know the quality of meat.

III. PROPOSED SYSTEM

To overcome this drawback we proposing this project device to measure steroid content in milk/meat products. Here we are using ph sensor to measure the quality of milk how much water is mix in that milk, ldr is used to measure cholesterol how thick milk is, lm35 is used for measuring the temperature of milk. Load cell is used to measure weight of meat and gas sensor for freshness of meat. After getting value we will display it on LCD. And upload it on cloud using thingspeak.

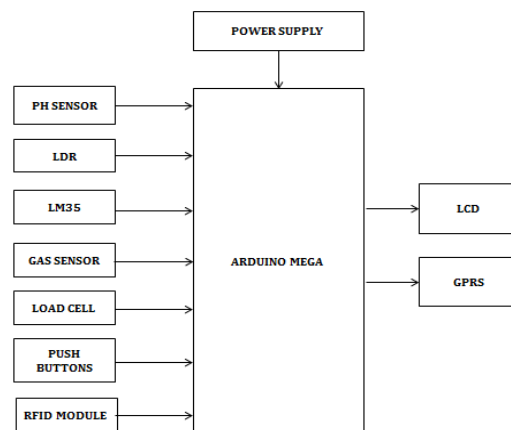
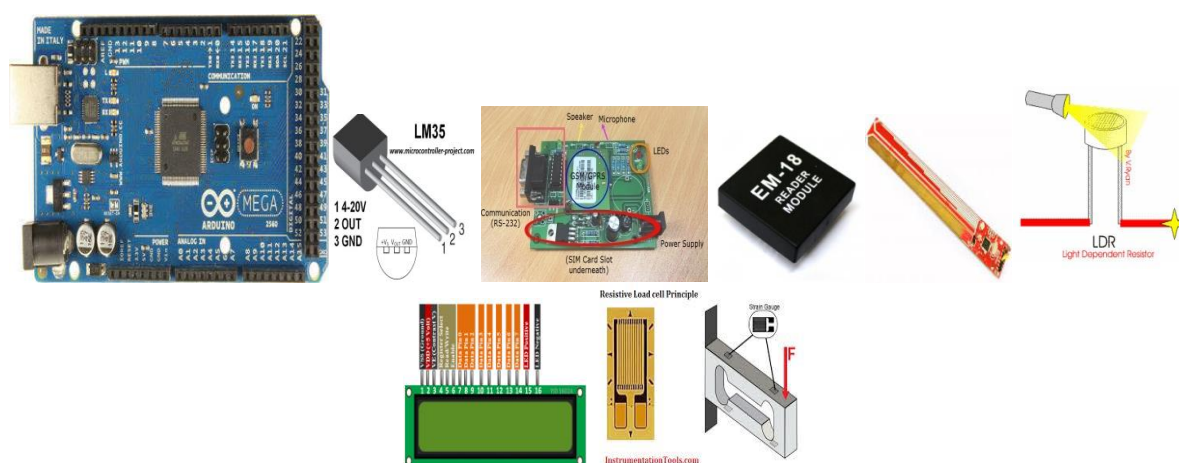


Fig:Block Diagram

HARDWARE REQUIREMENTS:

- ARDUINO MEGA 2560
- LCD
- LOAD CELL
- LDR
- RFID READER
- LM35 SENSOR
- GPRS MODULE
- GAS SENSOR
- PH SENSOR



HARDWARE DESCRIPTION:

we are using ph sensor to measure the quality of milk how much water is mix in that milk, ldr is used to measure cholesterol how thick milk is, lm35 is used for measuring the temperature of milk. Load cell is used to measure weight of meat and gas sensor for freshness of meat. After that it will send it to arduino mega. First we will swipe the rfid card on rfid reader if that card is register then it will ask to choose milk or meat. If we press the switch1 then it will show the content of milk and if we press the switch2 then it will show the content of meat. After getting value we will display it on LCD

SOFTWARE REQUIREMENTS:

- Arduino IDE
- Embedded C
- Fritzing

SOFTWARE DESCRIPTION:

NodeMCU and Arduino UNO uses Lua scripting language to program NodeMCU. Here, we are going to learn how to program NodeMCU using Arduino C++ language. Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board. When you use the NodeMCU with the Arduino IDE, it will write directly to the firmware, of NodeMCU erasing the original firmware, so if you want back the Lua SDK, and use the “flasher” to re-install the firmware. For installing the flasher you need to follow the steps discussed in 4.5.6 NODEMCU ESP8266. The NodeMCU automatically draw power from either, the USB connection to the computer or an external power supply. Connect the NodeMCU board to your computer using the USB cable. The power LED (labeled PWR) should glow. After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.



CODE:

```
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>
//SoftwareSerial mySerial(9, 10);
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
const int lm35_pin = A1; /* LM35 O/P pin */
int smoke = A2;
int ldr = A3;
int loadcell = A4;
char a;
float val1;
float ldr_val;
float ph_value;
float lc_val;
float smokesensor;
char pH_array[30], b[30];
int i=0, j;
String final_ph="";
bool complete_flag=0, start_flag=0;
#define sw1 8
#define sw2 9
int K;
int S;
String final="";
//void GPRS_send(unsigned int ,unsigned int,unsigned int ,unsigned int );
void setup() {
    lcd.begin(16, 2);
    lcd.setCursor(1, 1);
    final.reserve(13);
    Serial.begin(9600);
    Serial1.begin(9600);
    Serial3.begin(9600);
    pinMode(sw1, INPUT_PULLUP);
    pinMode(sw2, INPUT_PULLUP);
    pinMode(smoke, INPUT);
    pinMode(ldr, INPUT);
    pinMode(loadcell, INPUT);
    lcd.print("hello.....");
}
void loop() {
    //Serial.println("card please");
    while(Serial3.available() > 0)
    {
        // Reset count to zero// Keep reading Byte by Byte from the Buffer till the Buffer is empty
        delay(10);
        char input = Serial3.read();
        if (input != '\0')
        {
            break;
        }
        final += input;
    }
    // Serial.println("card please");
    if (final.length() > 0)
    {
```

```

lcd.print("place card:");
if(((final)=="3600A5D21C5D") || ((final)=="3600A5C0CE9D"))
{
  lcd.clear();
  lcd.print("choose:");
  lcd.setCursor(0,1);
  lcd.print("1.milk 2.meat");
  Serial.println(final);
  delay(6000);
  K= digitalRead(sw1);
  S= digitalRead(sw2);
  if(K==0)
  {
    lcd.clear();
    lcd.print("milk...");
    Serial.print("milk.....");
    temp();
    ldr_fat();
    ph();
    GPRS_sendmilk();
    // GPRS_send(val1,ldr_val,ph_value);
  }
  if(S==0)
  {
    lcd.clear();
    lcd.print("meat...");
    Serial.print("meat.....");
    weight();
    gas();
    GPRS_sendmeat();
    //GPRS_send(lc_val,smokesensor);
  }
  //lcd.clear();
  // GPRS_send(val1,ldr_val,ph_value,lc_val,smokesensor);
}
else
{
  // lcd.clear();
  lcd.setCursor(0,1);
  Serial.println(final);
  Serial.println("not matching");
  lcd.print("not matching");
  delay(1000);
}
final="";
}
}
void temp()
{
int adc_val;
float val1;
  adc_val = analogRead(lm35_pin); /* Read Temperature */
  val1 = (adc_val * 4.88); /* Convert adc value to equivalent voltage */
  val1 = (val1/10); /* LM35 gives output of 10mv/°C */
  Serial.print("Temperature = ");
  Serial.print(val1);

```



```

lcd.setCursor(0,1);
lcd.print("temp:");
  lcd.setCursor(5,1);
  lcd.print(val1);
  Serial.print(" Degree Celsius\n");
  delay(1000);
}
void ldr_fat()
{
unsigned int ldr_val = analogRead(ldr);
  Serial.print("ldr:");
  Serial.println(ldr_val);
  lcd.setCursor(0,1);
  lcd.print("colestral:");
  lcd.setCursor(10,1);
  lcd.print(ldr_val);
  delay(1000);
}
void ph()
{
  final_ph="";
  while(complete_flag==0)
  {
    if(Serial1.available())
    {
      a=Serial1.read();
      //Serial.print(
      if(a=='P')
      {
        start_flag=1;
      }
      if(start_flag==1)
      {
        pH_array[i]=a;
        i++;
        if((a==',') || (i>=30))
        {
          complete_flag=1;
          start_flag=0;
          j=i;
          i=0;
        }
      }
    }
  }
  if(complete_flag==1)
  {
    //int k=0;
    for(int i=0;i<j;i++)
    {
      if(isDigit(pH_array[i])||(pH_array[i]=='.'))
      {
        final_ph=final_ph+String(pH_array[i]);
        // k++;
      }
    }
  }
}

```



```

j=0;
//lcd.clear();
lcd.setCursor(0,0);
complete_flag=0;
//lcd.print("pH= ");
//lcd.print(final_ph);
Serial.println("ph:"+final_ph);
unsigned int ph_value=final_ph.toInt();
Serial.println(ph_value);
if(ph_value < 7 || ph_value > 8)
{
    Serial.println("milk is impure");
    lcd.setCursor(0,1);
    lcd.print("milk is impure");
}
else if(ph_value==7)
{
    Serial.println("Pure milk");
    lcd.setCursor(0,1);
    lcd.print("ph:");
    lcd.setCursor(4,1);
    lcd.print(ph_value);
    delay(2000);
    lcd.setCursor(0,1);
    lcd.print("Pure milk");
}
else
{
    Serial.println("nil");
    lcd.setCursor(0,1);
    lcd.print("Nil ");
}
}
delay(1000);
}
void weight()
{
    Serial.print("meat weight:");
    unsigned int lc_val=analogRead(loadcell);
    Serial.print("meat weight:");
    Serial.println(lc_val);
    lcd.setCursor(0,1);
    lcd.print("weight:");
    lcd.setCursor(7,1);
    lcd.print(lc_val);
    delay(2000);
}
void gas()
{
    int smokesensor = analogRead(smoke);
    if( smokesensor>400)
    {
        Serial.println(smokesensor);
        Serial.print("meat not fresh");
        lcd.setCursor(0,1);
        lcd.print("meat not fresh");
    }
}

```

```

}
else
{
    Serial.println("fresh");
    lcd.setCursor(0,1);
    lcd.print("meat not fresh");
}
delay(1000);
}
void GPRS_sendmilk()
{
    unsigned int ldr_val = analogRead(ldr);
    int adc_val;
    float val1;
    adc_val = analogRead(lm35_pin); /* Read Temperature */
    val1 = (adc_val * 4.88); /* Convert adc value to equivalent voltage */
    val1 = (val1/10); /* LM35 gives output of 10mv/°C */
    ph();
    //unsigned int ph_value=final_ph.toInt();
    Serial.println("AT");
    delay(1000);
    Serial.println("AT+CPIN?");//check for sim
    delay(1000);
    Serial.println("AT+CREG?");// checking sim registration
    delay(1000);
    Serial.println("AT+CGATT?");//checking if MS is connected to GPRS
    delay(1000);
    Serial.println("AT+CIPSHUT");
    delay(1000);
    Serial.println("AT+CIPSTATUS");// current connection status
    delay(2000);
    Serial.println("AT+CIPMUX=0");// start multiconnection
    delay(2000);
    Serial.println("AT+CSTT=\"airtelgprs.com\"");// APN of the sim
    delay(1000);
    Serial.println("AT+CIICR");// start wireless connection with GPRS
    delay(3000);
    Serial.println("AT+CIFSR");//get local IP address
    delay(1000);
    Serial.println("AT+CIPSPRT=0");
    delay(3000);
    Serial.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", \"80\");// start TCP connection
    AT+HTTPPARA=\"URL\", \"api.thingspeak.com/update\"
    delay(5000);
    Serial.println("AT+CIPSEND");// send data through TCP/UDP connection
    delay(5000);
    //String str="GET https://api.thingspeak.com/update?api_key=LH1P22B5YNMOPMHU&field1="
    +String(a)+"&field2=" +String(b);
    String str="GET https://api.thingspeak.com/update?api_key=48I9A99591B0WLGO&field1="
    +String(val1)+"&field2=" +String(ldr_val)+"&field3=" +String(final_ph);
    Serial.println(str);// The SMS text you want to send
    delay(100);
    Serial.println((char)26);// ASCII code of CTRL+Z
    //mySerial.println(str);
    delay(300);
    Serial.println();

```



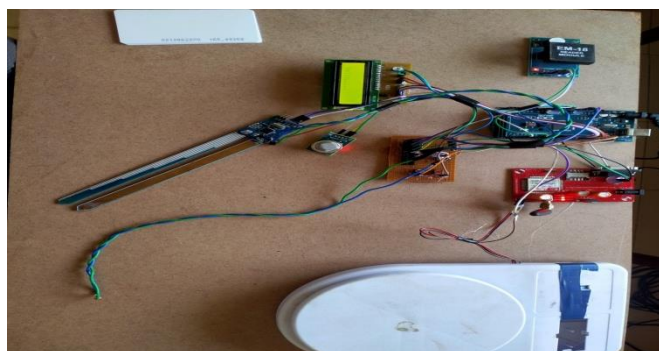
```

delay(3000);
Serial.write(26);
delay(1000);
  Serial.println();
  Serial.println("AT+CIPSHUT");
  delay(1000);
}
void GPRS_sendmeat()
{
  unsigned int lc_val=analogRead(loadcell);
  int smokesensor = analogRead(smoke);
  Serial.println("AT");
  delay(1000);
  Serial.println("AT+CPIN?");//check for sim
  delay(1000);
  Serial.println("AT+CREG?");// checking sim registration
  delay(1000);
  Serial.println("AT+CGATT?");//checking if MS is connected to GPRS
  delay(1000);
  Serial.println("AT+CIPSHUT");
  delay(1000);
  Serial.println("AT+CIPSTATUS");// current connection status
  delay(2000);
  Serial.println("AT+CIPMUX=0");// start multiconnection
  delay(2000);
  Serial.println("AT+CSTT=\"airtelgprs.com\"");// APN of the sim
  delay(1000);
  Serial.println("AT+CIICR");// start wireless connection with GPRS
  delay(3000);
  Serial.println("AT+CIFSR");//get local IP address
  delay(1000);
  Serial.println("AT+CIPSPRT=0");
  delay(3000);
  Serial.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", \"80\"");// start TCP connection
  AT+HTTPPARA=\"URL\", \"api.thingspeak.com/update\"
  delay(5000);
  Serial.println("AT+CIPSEND");// send data through TCP/UDP connection
  delay(5000);
  //String str="GET https://api.thingspeak.com/update?api_key=LH1P22B5YNMOPMHU&field1="
  +String(a)+"&field2=" +String(b);
  String str="GET https://api.thingspeak.com/update?api_key=48I9A99591B0WLGO&field4="
  +String(lc_val)+"&field5=" +String(smokesensor);/+ "&field3=" +String(c)+"&field4=" +String(d);
  Serial.println(str);// The SMS text you want to send
  delay(100);
  Serial.println((char)26);// ASCII code of CTRL+Z
  //mySerial.println(str);
  delay(300);
  Serial.println();
  delay(3000);
  Serial.write(26);
  delay(1000);
  Serial.println();
  Serial.println("AT+CIPSHUT");
  delay(1000);
}

```

V. SIMULATION RESULTS

The quality measurement was found to be accurate and precise within acceptable variation. However the system is susceptible to position of sensor and shape of the container. Hence more perfect setup is needed to get accurate measurement.



VI. CONCLUSION AND FUTUREWORK

This method is also accurate and precise with respect to other traditional methods, but the entire system must be closed space. This must be calibrated before putting it into practice by considering the fat content with other traditional methods. If the depositor cheats the machine by pouring other than milk the products like honey the system will not work properly until it is serviced. The system can be further developed to separate the different fat content milks into different containers and to identify the user who tries to spoil the machine. In this paper, we developed an IoT based system which gives faster and more accurate results. In our proposed system; Microbial activity is determined using gas sensor. The high-quality milk should have no salinity, so salinity of the milk is measured by using a salinity sensor and level of the milk will be measured by using a level sensor. In addition to that customer should have their own card for accessing the milk diaries. The milk collection parameters such as weight, fat and temperature are measured by this system gives fast and more accurate results than the existing systems which are more costly than the developed one. As a future work, it is intended to implement IOT and DBMS for billing system, in which each user will have a data base of their own, in which the data is recorded for the amount of milk taken, in this payment can be done using debit or credit cards, payment can be done on monthly basis. Further this system will be used by the management for tracking the milk production and marketing, all the information from milk production to marketing will be stored in management's website which can be accessed by any user having account in that firm.

REFERENCES

- [1] Pedinti Sankaran Venkateswaran; Abhishek Sharma; Santosh Dubey; Ajay Agarwal; Sanket Goel "Rapid and Automated Measurement of Milk Adulteration Using a 3D Printed Optofluidic Micro Viscometer (OMV)" IEEE Sensors Journal, Year: 2016, Volume: 16, Issue: 9, DOI: 10.1109/JSEN.2016.2527921
- [2] Lucas de Souza Ribeiro; Fábio Augusto Gentilin; José Alexandre de França; Ana Lúcia de Souza Madureira Felício; Maria Bernadete de M. França "Development of a Hardware Platform for Detection of Milk Adulteration Based on NearInfrared Diffuse Reflection" IEEE Transactions on Instrumentation and Measurement, Year: 2016, Volume: 65, Issue: 7, DOI: 10.1109/TIM.2016.2540946
- [3] Maurício Moreira; José Alexandre de França; Dari de Oliveira Toginho Filho; Vanerli Beloti; Alberto Koji Yamada; Maria Bernadete de M. França; Lucas de Souza Ribeiro "A Low-Cost NIR Digital Photometer Based on InGaAs Sensors for the Detection of Milk Adulterations with Water" IEEE Sensors Journal, Year: 2016, Volume: 16, Issue: 10, DOI: 10.1109/JSEN.2016.2530873.
- [4] Jinying Yin; Siqi Zhang; Hongyan Yang; Lijie Wang; Zhen Zhou "Influence of Fat Particle Size on Light Scattering Properties in Milk Quality Testing" Year: 2014, DOI: 10.1109/IMCCC.2014.157
- [5] Pallavi Gupta; Anwar Sadat; Mohd Jamilur Rahman Khan "An Opto electro mechanical Sensor for Detecting Adulteration in Anhydrous Milk Fat" IEEE Sensors Journal, Year: 2014, Volume: 14, Issue: 9, DOI: 10.1109/JSEN.2014.2319113 International Journal of Pure and Applied Mathematics Special Issue 30 11



**Impact Factor:
7.569**



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 **9940 572 462**  **6381 907 438**  **ijirset@gmail.com**



www.ijirset.com

Scan to save the contact details