IoT based Smart Factory System

Kalki S

126006056@sastra.ac.in

9-27-2024

TABLE OF CONTENTS

ABSTRACT

The **IoT-based Smart Factory System** automates factory operations using temperature, ultrasonic, and PIR sensors. The system controls a servo motor, DC motors, and lighting based on sensor readings. When the ultrasonic sensor detects a distance less than 30 cm, the servo motor rotates. If the temperature exceeds 30°C, the DC motor activates, simulating cooling mechanisms. The PIR sensor detects human movement, triggering the DC motor and turning on the lights via an SPDT relay. An **Arduino Uno** microcontroller processes the sensor data and sends it to the Thinkspeak Cloud for real-time monitoring and analysis, **Adafruit IO** for real-time monitoring and analysis.

OBJECTIVE

The objective of this project is to design and implement an IoT-based smart factory system that:

- Automate factory processes using sensor data for motor and lighting control.
- Use the **Arduino Uno** to process sensor inputs and send data to Thinkspeak Cloud , Adafruit IO for remote monitoring.
- Enhance factory efficiency and safety by reducing manual intervention.
- React to distance, temperature, and motion using servo motors, DC motors, and lighting automation.

INTRODUCTION

Industry 4.0 introduces smart systems that utilize IoT to improve the efficiency of factory operations. In this project, an Arduino Uno-based IoT smart factory system is implemented using three primary sensors:

- Ultrasonic Sensor for proximity detection.
- Temperature Sensor for monitoring the ambient temperature.
- PIR Sensor for detecting human presence.

The system responds as follows:

- A Servo Motor rotates when the ultrasonic sensor detects an object closer than 30 cm.
- A DC Motor activates, and a buzzer sounds when the temperature exceeds 30°C.
- The PIR sensor triggers the DC motor and turns on the factory lights using an SPDT relay when motion is detected.

Sensor data is sent to Thinkspeak Cloud and Adafruit IO for remote monitoring and analysis, enabling factory operators to track conditions in real-time.

METHADOLOGY

The methodology for building the IoT-based smart factory system involves the following steps:

1. **System Design**:
    - **Ultrasonic Sensor (HC-SR04)**: Measures distance to detect objects. If an object is closer than 30 cm, the servo motor rotates.
    - **Temperature Sensor (LM35)**: Monitors temperature. If the temperature rises above 30°C, the DC motor starts, and the buzzer sounds as a warning.
    - **PIR Sensor**: Detects human movement and sends a signal to activate the DC motor and turn on the lights via an SPDT relay.
2. **Microcontroller - Arduino Uno**:
    - The **Arduino Uno** processes inputs from the sensors and controls the servo motor, DC motor, lights, and buzzer. It also sends the collected sensor data to Thinkspeak Cloud and Adafruit IO for remote monitoring.
3. **Sensor Integration**:
    - **Ultrasonic Sensor**: Measures the distance between the sensor and an object. When the distance is less than 30 cm, the servo motor rotates.
    - **Temperature Sensor**: Monitors the ambient temperature. When the temperature exceeds 30°C, the DC motor starts, and the buzzer activates.

- **PIR Sensor**: Detects human presence and activates the DC motor and lights via an SPDT relay for improved safety.

4. **Actuation and Alerts**:
    - **Servo Motor**: Rotates when the ultrasonic sensor detects an object within 30 cm.
    - **DC Motor**: Activated when either the temperature exceeds 30°C or the PIR sensor detects motion.
    - **Buzzer**: Sounds when the temperature rises above 30°C to signal high temperature conditions.
    - **Lights**: Turn on when the PIR sensor detects motion using an SPDT relay for safety lighting around machinery.
5. **Data Communication**:
    - The Arduino Uno sends real-time data from all sensors to **Thinkspeak Cloud** and **Adafruit IO** via a Wi-Fi module. This enables remote monitoring of factory conditions such as temperature, object proximity, and motion detection.
6. **Thinkspeak Cloud and Adafruit IO Setup**:
    - Separate channels are configured on **Thinkspeak Cloud** and **Adafruit IO** to receive and visualize sensor data. Factory managers can access both platforms to monitor sensor readings in real-time and set up alerts for critical conditions.
7. **Automation and Monitoring**:
    - The system operates autonomously based on sensor inputs, controlling motors, lights, and the buzzer. Alerts are sent to operators through Thinkspeak or Adafruit IO if the system detects high temperatures or object proximity.

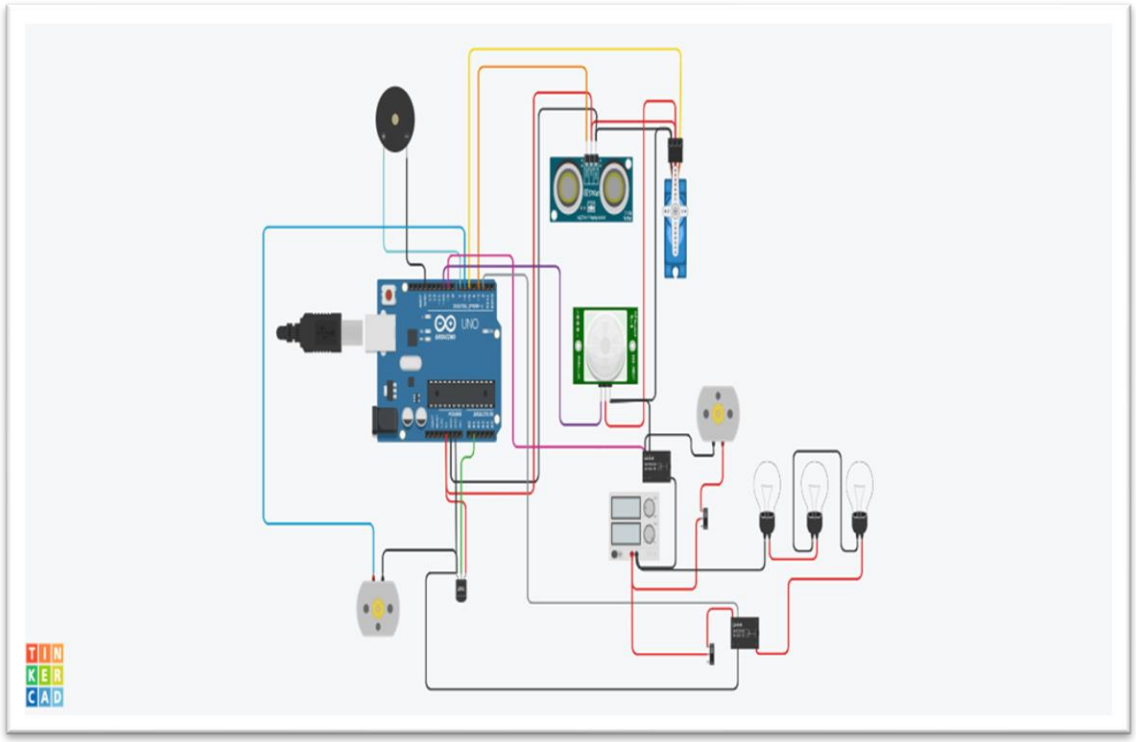
CODE:

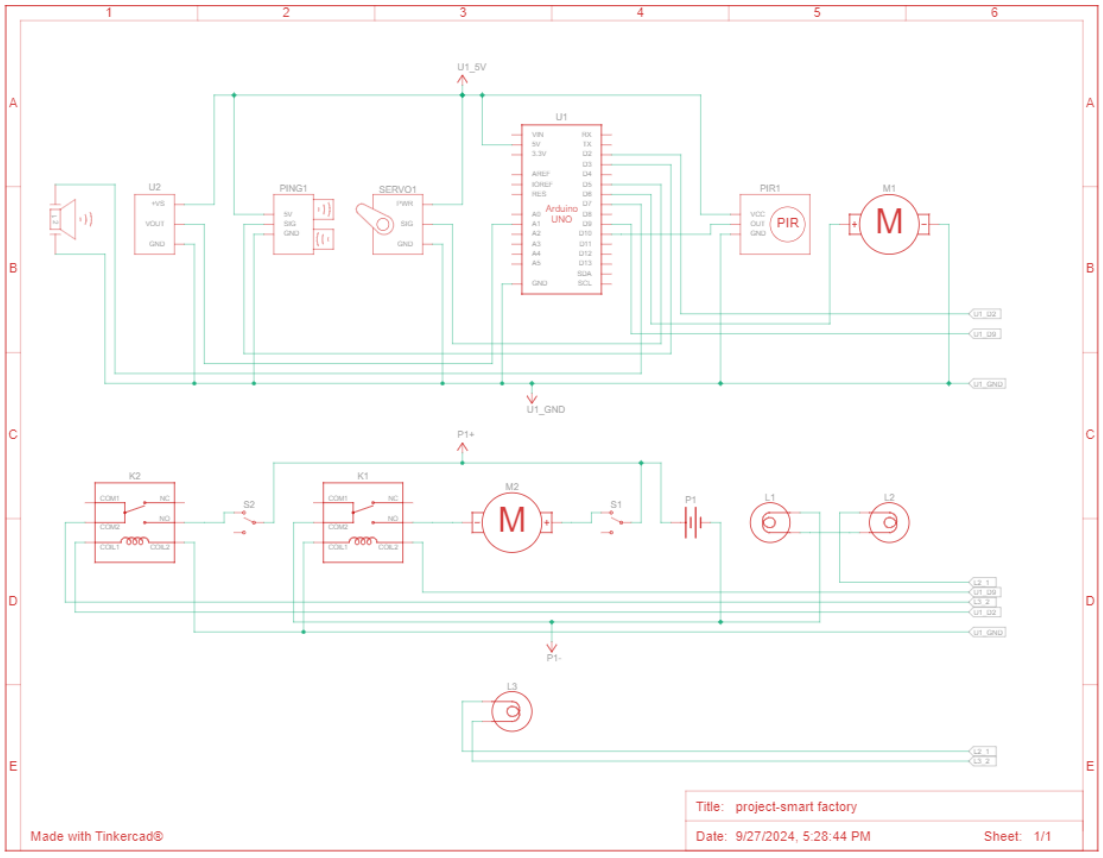
```
#include <Servo.h>

#include <WiFi.h>

#include <ThingSpeak.h>

#include <AdafruitIO.h>

#include <AdafruitIOArduino.h>


#define IO_USERNAME  "kalki_18"

#define IO_KEY   "aio_OtWb23nQYPTlSFIkmpnVdu8xDSFn"


const char *ssid = "Oppo A16"; // Your WiFi SSID

const char *password = "Kalpana@5"; // Your WiFi password

unsigned long myChannelNumber = 2672588; // Replace with your ThingSpeak channel number

const char *myWriteAPIKey = "U5PZYMLJUDFPOUZU"; // Replace with your ThingSpeak Write API key


Servo servo_5;


const int pingUltra = 3;

const int buzzPin = 7;

const int tempsen = A1;

const int fan = 6;

const int pirSen = 10;

const int pirMotor = 9;

const int pirBulb = 2;


int temp = 0;

int pir = 0;

AdafruitIO_Arduino io(IO_USERNAME, IO_KEY)
```

```
void setup() {
   Serial.begin(9600);
   servo_5.attach(5);
   pinMode(pingUltra, OUTPUT);
   pinMode(buzzPin, OUTPUT);
   pinMode(tempsen, INPUT);
   pinMode(fan, OUTPUT);
   pinMode(pirSen, INPUT);
   pinMode(pirMotor, OUTPUT);
   pinMode(pirBulb, OUTPUT);
   WiFi.begin(ssid, password);
   while (WiFi.status() != WL_CONNECTED) {
      delay(1000);
     Serial.println("Connecting to WiFi...");
   }
   Serial.println("Connected to WiFi");
   ThingSpeak.begin(WiFi);
}

void loop() {
  io.run();
  long duration,distance;
  pinMode(pingUltra,OUTPUT);
  digitalWrite(pingUltra,LOW);
  delay(200);
  digitalWrite(pingUltra,HIGH);
  delay(500);
  digitalWrite(pingUltra,LOW);
```

```
pinMode(pingUltra,INPUT);
duration=pulseIn(pingUltra,HIGH);
distance=microSecondsToCentimeters(duration);
if(distance<30){
 servo_5.write(90);
 delay(2);
}
else{
 servo_5.write(0);
}

 if (distance < 30) {
    servo_5.write(90);
 } else {
    servo_5.write(0);
 }

 temp = (-40 + 0.488155 * (analogRead(tempsen) - 20));
 delay(1000);
 if (temp >= 30) {
    digitalWrite(fan, HIGH);
    digitalWrite(buzzPin, HIGH);
 } else {
    digitalWrite(fan, LOW);
    digitalWrite(buzzPin, LOW);
 }

 pir = digitalRead(pirSen);
 if (pir == LOW) {
```

```
      digitalWrite(pirMotor, LOW);
      digitalWrite(pirBulb, LOW);
    } else {
      digitalWrite(pirMotor, HIGH);
      digitalWrite(pirBulb, HIGH);
    }

    distanceFeed->save(distance);
    tempFeed->save(temp);
    pirFeed->save(pir);

    ThingSpeak.setField(1, distance);
    ThingSpeak.setField(2, temp);
    ThingSpeak.setField(3, pir);
    ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

    Serial.print("Distance: ");
    Serial.println(distance);
    Serial.print("Temperature: ");
    Serial.println(temp);
    Serial.print("PIR Detection: ");
    Serial.println(pir);
    Serial.print("*******");
    delay(200);
}

long microSecondsToCentimeters(long microseconds) {
    return microseconds / 29 / 2;
}
```




U1_5V
U1
U2
PING1
SERVO1
PIR1
M1
Arduino UNO
PIR
U1_GND
P1+
K2
K1
M2
S2
S1
P1
L1
L2
L3
P1-
Made with Tinkercad®
Title: project-smart factory
Date: 9/27/2024, 5:28:44 PM
Sheet: 1/1

PROJECT FILE:

Simulation with tinkercad:
https://www.tinkercad.com/things/18xzN3muPy0-project-smart-factory/editel?sharecode=3dZqzLF0REuTcfwEm083GqQvwllhNneRl6AmUVrLJGo

RESULT

The system operates under the following conditions:

- Ultrasonic Sensor: When an object is detected within 30 cm, the servo motor rotates.
- Temperature Sensor: If the temperature exceeds 30°C, the DC motor and buzzer activate, signalling high temperatures.
- PIR Sensor: When motion is detected, the DC motor starts, and the lights are turned on using an SPDT relay for safety.
- Thinkspeak and Adafruit IO: Sensor data is uploaded to both platforms for real-time monitoring, allowing factory operators to track conditions remotely.

CONCLUSION

The IoT-based Smart Factory System demonstrates how automation and IoT can improve factory operations. By integrating an Arduino Uno microcontroller with temperature, proximity, and motion sensors, the system controls motors, lights, and a buzzer based on real-time data. Sensor readings are transmitted to both Thinkspeak Cloud and Adafruit IO, providing factory managers with continuous, real-time monitoring of critical factory parameters. This project exemplifies the use of IoT to enhance safety, reduce manual intervention, and improve overall factory efficiency.