ISSN 2277-2685
IJESR/Jan-Mar. 2026/ Vol-16/Issue-1/277-285
D.Chaitra *et. al.,* / International Journal of Engineering & Science Research

An Industrial Oriented Mini Project Report On Industrial Automation And Control Based On Cyber-Physical Technologies

Mrs. S. Mayuri[1],D.Chaitra[2],P.Geethasri[3],M.Vaishnavi[4]
[1]Assistant Professor; Department Of Electrical And Electronics Engineering, Bhoj Reddy Engineering College For Women, Hyderabad, India
[2,3,4]B.Tech Students; Department Of Electrical And Electronics Engineering, Bhoj Reddy Engineering College For Women, Hyderabad, India.
dasarichaitra07@gmail.com

Accepted 20-03-2026

Abstract
The rapid evolution of industrial automation is being driven by the integration of cyber and physical systems, leading to the development of Cyber-Physical Technologies (CPTs) that enhance efficiency, safety, and intelligence in industrial operations. This work presents the design and implementation of an Industrial Automation and Control System leveraging CPTs, aimed at achieving real-time monitoring, intelligent decision-making, and automated control of industrial processes. In this system, sensors continuously acquire data from industrial equipment, including temperature, pressure, and motor speed. The collected data is processed using microcontrollers, such as Arduino or STM32 platforms, enabling the generation of control actions to maintain optimal operating conditions. The cyber layer facilitates data communication, remote monitoring, and analysis through digital platforms, allowing operators to supervise operations and make informed decisions. The system integrates a network of sensors and actuators with programmable controllers to monitor physical parameters like flow rate, pressure, and motor speed. Advanced control algorithms maintain system stability, optimize performance, and respond dynamically to process disturbances. By combining cyber and physical layers, the system achieves enhanced reliability, scalability, fault tolerance, and predictive maintenance, forming a foundation for smart manufacturing within Industry 4.0 environments.

Keywords*: Industrial Automation, Cyber-Physical Systems (CPS), Cyber-Physical Technologies (CPTs), Industry 4.0, Real-Time Monitoring, Intelligent Control Systems, Embedded Systems, Sensors and Actuators, Arduino, STM32, Microcontrollers, Process Control, Predictive Maintenance, Smart Manufacturing, IoT (Internet of Things), Data Acquisition, Remote Monitoring, Control Algorithms, System Optimization, Fault Tolerance.*

Introduction

Cyber-Physical Systems (CPS) are interconnected networks of heterogeneous components that integrate computation with physical processes, offering advanced control over industrial operations. The close integration and heterogeneity of components in CPS pose challenges for conventional control, communication, and software design, often rendering traditional approaches inefficient or impractical. Industrial control systems typically employ multiple independent loops to manage subsystems, with interactions facilitated through communication protocols such as Controller Area Networks (CAN). While this modular design improves simplicity and reduces cost, it limits the ability to implement large-scale control loops that span multiple subsystems or distant locations, thereby constraining potential efficiency gains. The rise of CPS addresses these limitations by providing an integrated, heterogeneous control network of sensors, actuators, and processors capable of managing large-scale industrial processes. However, the scale and complexity of such networks require new tools in control, communication, and software theory to ensure real-time, reliable, and low-latency operation.

Literature Survey

The advent of Industry 4.0 has highlighted the role of CPS in connecting machines, sensors, and controllers through digital networks to enhance production efficiency and reliability. Alguliyev et al. (2022) reviewed over 2000 studies, noting that CPS integrates artificial intelligence, big data analytics, and IoT technologies to enable real-time monitoring and automation in manufacturing. Recent smart factory architectures combine CPS with IoT, allowing equipment to exchange information and support flexible, modular production systems capable of dynamic planning, predictive maintenance, and automated decision-making. Time-Sensitive Networking (TSN) has been identified as a key solution for achieving deterministic, low-latency communication between sensors, controllers, and machines in CPS-based industrial networks. Studies by Mutua (2024) and Zaved (2025) emphasize that CPS enhances operational efficiency and decision-making in industrial processes and that its architecture typically comprises three layers—physical, cyber, and communication—to facilitate comprehensive data acquisition and analysis. These findings collectively underscore CPS as a critical enabler for smart manufacturing, predictive maintenance, improved safety, and autonomous production in modern industrial environments.

Objective

This project aims to design and implement a prototype industrial automation system using Cyber-Physical Technologies. The system integrates sensors, microcontrollers, communication networks, and intelligent control to monitor machinery in real-time, enhance operational efficiency, ensure reliability and safety, and enable automated decision-making in modern smart manufacturing contexts.

Proposed Intelligent Cyber-Physical System

The proposed Intelligent Cyber-Physical System (iCPS) follows a three-layer architecture comprising Intelligent Embedded Sensors, Intelligent Coordinators, and an Intelligent Server. The sensors interact directly with the industrial environment to collect data on machines, products, workers, and logistics. Hardware for these sensors includes STM32 Nucleo boards with ARM Cortex-M4 processors and a suite of sensors for temperature, humidity, motion, and pressure. Communication is achieved via WiFi modules, while embedded software implements intelligent mechanisms for model learning, residual generation, change detection, and adaptation. Predictive models, including Auto-Regressive models and Echo State Networks, are trained locally on the sensors to identify discrepancies between predicted and actual measurements. When changes are detected, adaptation mechanisms trigger retraining of models to respond to new environmental or operational conditions.

Intelligent Coordinators, implemented on off-the-shelf embedded platforms such as Odroid boards, manage clusters of sensors. They provide services including MQTT-based communication, dependency graph computation to model inter-sensor relationships, and database storage using NoSQL solutions like CouchDB. An Apache server enables remote monitoring and analysis. The coordinators facilitate distributed intelligence by aggregating sensor data, detecting anomalies, and supporting decision-making for process optimization.

Overview of Embedded Systems

Embedded systems form the backbone of CPS, comprising a CPU, memory (RAM and ROM), input and output devices, communication interfaces, and application-specific circuitry. CPUs may take the form of microcontrollers, microprocessors, or digital signal processors, depending on computational requirements. Memory stores the firmware, which executes the application logic, while input devices range from simple keypads to sensor inputs, and output devices include LEDs and LCDs for status indication. Communication interfaces allow interaction with other systems or computers, using protocols such as RS232, USB, or Ethernet. Application-specific circuitry, including sensors, transducers, and control hardware, enables the embedded system to perform its designated industrial tasks efficiently while minimizing power consumption.

DESIGN OF SOFTWARE

Introduction to ARDUINO IDE:

This is free software (evaluation version) which solves many of the pain points for an embedded system developer. This software is an Integrated Development Environment (IDE), which integrated text editor to write program, a compiler and it will convert your source code into HEX file. Here is simple guide to start working with Arduino IDE Vision which can be used for:

- Writing programs in Arduino IDE
- Compiling and assembling programs
- Debugging programs

Embedded C Language

Embedded C is a programming language used for developing software for embedded systems such as microcontrollers. It is an extension of the C programming language that provides additional features for controlling hardware devices. In this project, Embedded C is used to write the program for the Arduino microcontroller. The program controls different sensors and output devices such as the LDR sensor, gas sensor, temperature sensor, relay module, fan, and light. Embedded C allows the system to read sensor values, process the data, and perform the required actions automatically. Therefore, it plays an important role in implementing the automation and control functions of the system.

Arduino IDE

The Arduino Integrated Development Environment (IDE) is the software used to write, compile, and upload programs to the Arduino board. It provides a simple and user-friendly interface for developing embedded applications. In this project, the Arduino IDE is used to write the Embedded C program that controls the sensors and actuators used in the automation system.

The IDE allows the user to select the Arduino board type, configure the communication port, verify the program, and upload the code to the microcontroller. It also provides tools such as the Serial Monitor to observe sensor data and system outputs during testing and debugging.

Arduino IDE Compiler

The Arduino IDE compiler is responsible for converting the program written in Embedded C into machine code that can be executed by the microcontroller. When the user clicks the verify or upload button in the Arduino IDE, the compiler checks the program for errors and translates it into binary code. If the program is free of errors, the compiled code is then uploaded to the Arduino board through the USB connection. The compiler ensures that the program runs correctly on the microcontroller and controls the hardware components efficiently.

Arduino Mobile Control

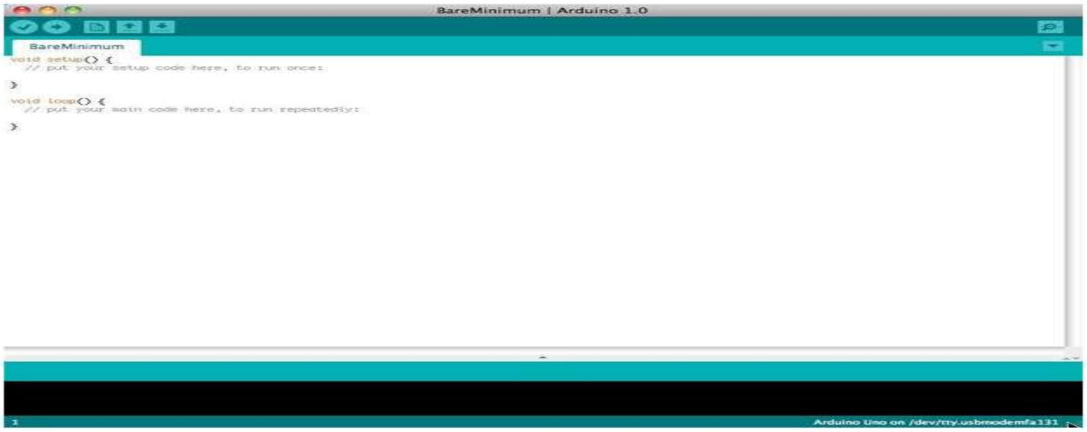
Arduino mobile control refers to the ability to monitor and control the automation system using a mobile device. In industrial automation systems, mobile control improves flexibility and remote monitoring capability. In this project, the Arduino communicates with external modules such as the GSM module to send information or alerts to a mobile phone. When abnormal conditions such as gas leakage or high temperature are detected, the system can notify the user through a mobile network. This feature helps in improving safety, monitoring, and control of the industrial environment.

Program

The program is the main part of the system that controls the overall operation of the project. The program is written in Embedded C using the Arduino IDE. It contains two main functions called setup() and loop(). The setup() function is executed once when the system starts and is used to initialize sensors, communication, and output devices. The loop() function runs continuously and reads the sensor data, processes it, and controls the outputs such as relay, fan, buzzer, and light. Based on the sensor readings, the Arduino automatically performs the required actions to maintain safe and efficient operation of the industrial automation system.

SOFTWARE STEPS:

Before you can start doing anything with the Arduino, you need to download and install the Arduino IDE (integrated development environment).



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Fig. Arduino Programming Window

After the opening IDE the settings are changed in order to connect to the Arduino.

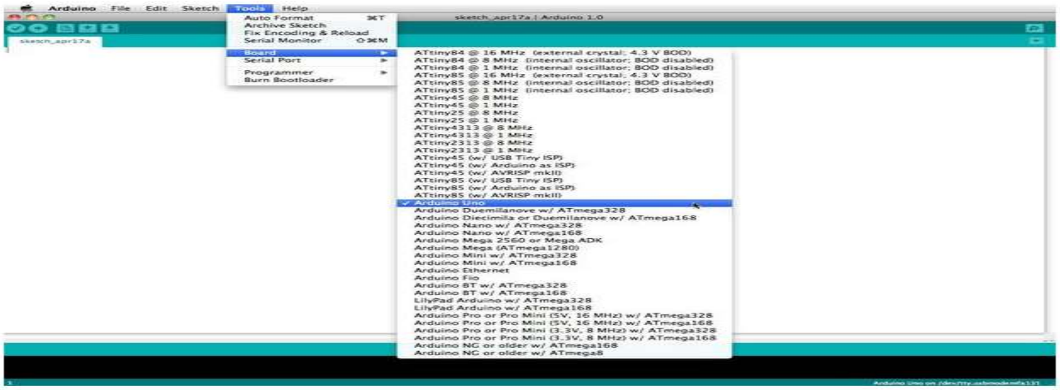


Fig: Arduino IDE Board Setup Window

This above image shows the Arduino IDE Tools menu where the board type is being selected. The user opens the Tools → Board option and chooses the appropriate board, such as Arduino Uno, to ensure that the program is compiled and uploaded for the correct microcontroller. Selecting the correct board is an important step before uploading the code to the Arduino device.

Before you can start doing anything in the Arduino programmer, you must set the board-type and serial port.

To set the board, go to the following: Tools --> Boards

Select the version of board that you are using. Since I have an Arduino Uno plugged in, I obviously selected"ArduinoUno."To set the serial port, go to the following: Tools --> Serial Port

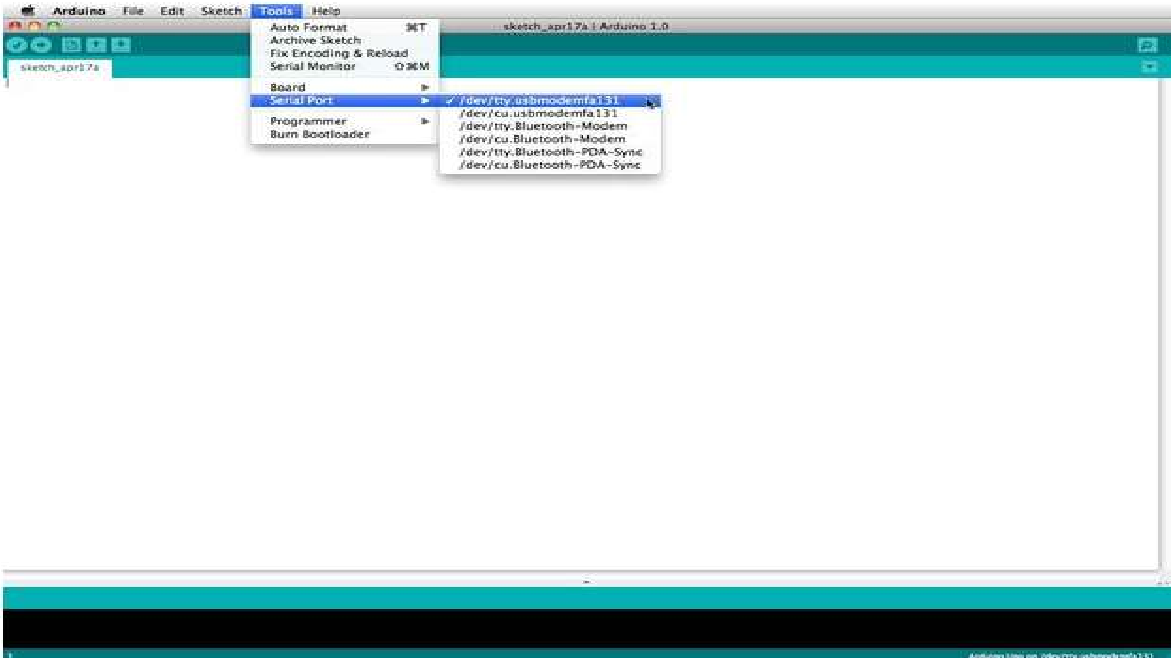
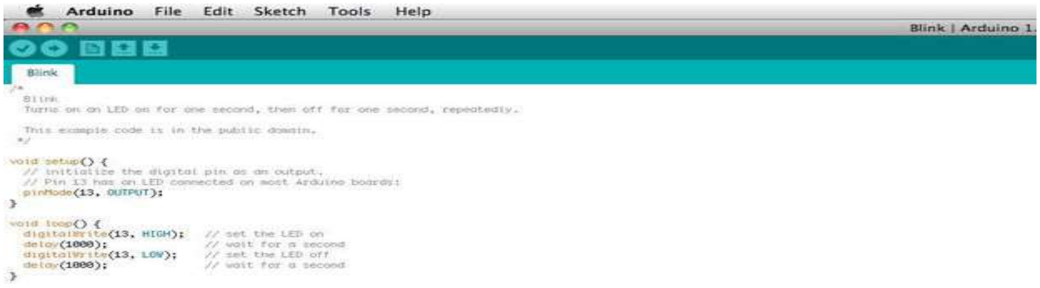


Fig. Selecting Serial Port in Arduino IDE

Arduino programs are called sketches. The Arduino programmer comes with a ton of example sketches preloaded. This is great because even if you have never programmed anything in your life, you can load one of these sketches and get the Arduino to do something.



```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

Fig. Blink Program in Arduino IDE

The serial monitor allows your computer to connect serially with the Arduino. This is important because it takes data that your Arduino is receiving from sensors and other devices and displays it in real-time on your computer. Having this ability is invaluable to debug your code and understand what number values the chip is actually receiving. For instance, connect center sweep (middle pin) of a potentiometer to A0, and the outer pins, respectively, to 5v and ground. Next upload the sketch shown below:

File --> Examples --> Basics --> Analog Read Serial Click the button to engage the serial monitor which looks like a magnifying glass. You can now see the numbers being read by the analog pin in the serial monitor. When you turn the knob the numbers will increase and decrease. The numbers will be between the range of 0 and 1023. The reason for this is that the analog pin is converting a voltage between 0 and 5V to a discreet number.

TECHNICAL ARCHITECTURE

5.1. BLOCK DIAGRAM:

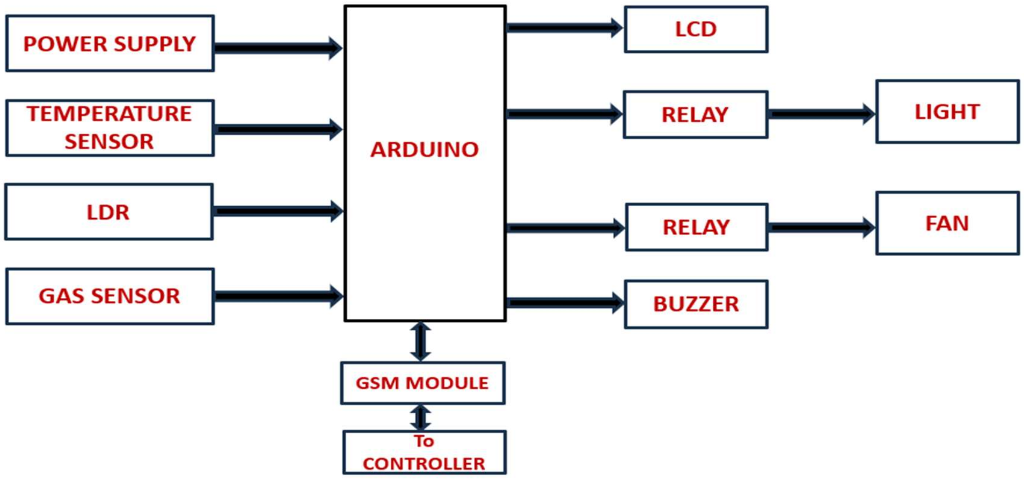


Fig. Block diagram

This block diagram represents a cyber-physical industrial automation system where sensors collect real-time data, the Arduino processes it, actuators control industrial devices, and a communication module enables remote monitoring and control.

5.2 HARDWARE REQUIREMENTS:

- Arduino Uno
- Power Supply
- Gsm Module
- Ldr
- Buzzer
- Light
- Relay Module
- Lcd
- Gas Sensor
- Tempareture Sensor
- Fan

WORKING:

The project Industrial Automation and Control Based on Cyber-Physical Technologies works by integrating physical industrial devices with digital monitoring and control systems. The system mainly consists of sensors, a microcontroller, communication modules and actuators that together form a Cyber-Physical System for industrial automation.

Initially, different sensors are installed in the system to monitor important parameters such as temperature, light intensity, gas level, or machine status in the industrial environment. These sensors continuously collect real-time data from the physical environment and convert it into electrical signals. The collected sensor data is then sent to the microcontroller, which acts as the central processing unit of the system.

The microcontroller processes the received data and analyses whether the values are within the predefined safe limits. If the sensor readings indicate abnormal conditions such as high temperature or gas leakage, the controller automatically triggers the appropriate control action. This may include activating alarms, switching on cooling systems, or controlling industrial equipment through relay modules. The processed information can also be transmitted through communication modules such as GSM to a monitoring system or mobile device. This enables remote monitoring and control of industrial processes. The operators can observe system parameters in real time and take necessary actions if required.

In this way, the cyber system continuously interacts with the physical system. This interaction creates a smart automation system capable of monitoring, analysing and controlling industrial operations efficiently. Thus, the proposed system demonstrates how Cyber-Physical Technologies can be used to achieve real-time monitoring, automated control, improved safety and efficient industrial operation in modern smart manufacturing environments.

HARDWARE MODULE, TESTING AND RESULTS

The system was thoroughly tested to evaluate its performance, reliability and accuracy. Different testing methodologies were adopted to ensure that each component and the overall system functioned as intended.

The system works in two modes:

- Automatic Mode
- Manual Mode

In Automatic Mode, the system takes the wheel, operating entirely on its own without needing any manual input. The Arduino acts as the "brain," constantly polling data from the DHT11 temperature sensor, the LDR (light sensor) and the gas sensor to stay updated on its surroundings.

Whereas in Manual Mode, the system shifts control from the sensors directly to you. This mode is perfect for those times when you want to override the automatic settings or manage your environment based on your personal preference rather than pre-set data.

AUTOMATIC MODE

In automatic mode, the system operates independently without requiring any human intervention. The Arduino continuously reads data from all sensors such as the temperature sensor (DHT11), LDR and gas sensor. Based on predefined threshold values, the system makes decisions automatically and controls the connected devices.

When the temperature exceeds the preset limit, the Arduino processes this input and activates the relay module to turn ON the fan. This helps in reducing the temperature and maintaining a safe environment.

Similarly, the LDR sensor monitors the light intensity. When the surrounding light level falls below a certain threshold, the system automatically switches ON the light to ensure proper visibility.

In the case of gas leakage, the gas sensor detects the presence of harmful gases. The Arduino immediately triggers the buzzer to provide an audible alert. At the same time, the GSM module sends a warning message to the user, enabling quick action even if the user is not physically present near the system.

All these operations occur continuously in real time. The system keeps monitoring the environment and updates its actions dynamically based on changing conditions.

AUTO_MODE

AUTO MODE

GAS DETECTED:185

HIGH TEMPERATURE:36

HIGH TEMPERATURE:36

HIGH TEMPERATURE:37

HIGH TEMPERATURE:36

HIGH HUMIDITY:81

Fig. : Automatic mode operation in mobile

WORKING MODEL:

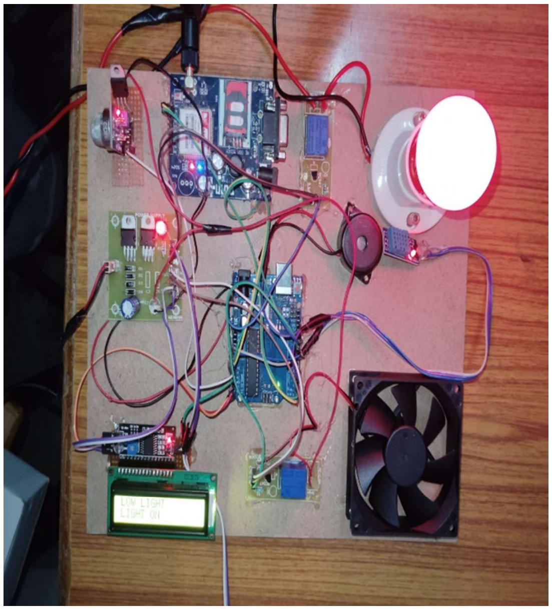


Fig. : Woking model kit

Result (Automatic Mode):

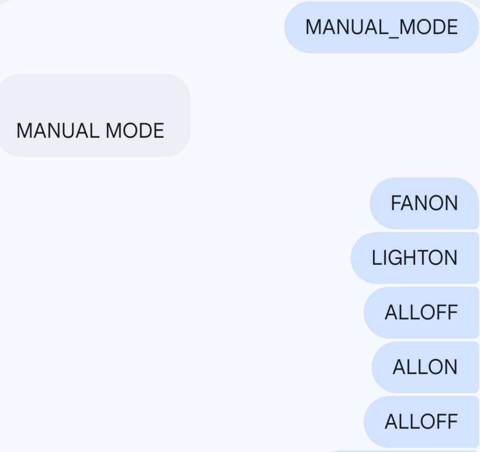
The system demonstrated fast response and high accuracy in detecting environmental changes. It successfully activated the required devices without delay and maintained safe operating conditions. This mode ensures efficient automation and reduces the need for manual supervision.

The automatic mode is set in the working model by default. The below shows the automatic mode operation of commands in the system. This information is sent to the controller through GSM module.

MANUAL MODE

In manual mode, the system allows the user to directly control the devices according to their requirements. This mode is useful when the user wants to override automatic operation or manually manage the system under specific conditions.

The user can send commands through the GSM module or control interface to switch ON or OFF the fan and light.



MANUAL_MODE

MANUAL MODE

FANON

LIGHTON

ALLOFF

ALLON

ALLOFF

Fig. : Command input in Manual mode

Once a command is received, the Arduino processes it and activates the corresponding relay to control the device. Unlike automatic mode, the system does not depend on sensor values for decision-making in this mode.

For example, the user can turn ON the fan even if the temperature is normal, or switch OFF the light regardless of light intensity. This provides flexibility and better control over the system.

However, safety features remain active even in manual mode. If a critical condition such as gas leakage is detected, the system will still trigger the buzzer and send alert messages to the user. This ensures that safety is not compromised.

Result (Manual Mode):

The system responded accurately to user commands and executed operations without delay. It provided flexible control while maintaining essential safety functions. This mode enhances user interaction and ensures reliable performance under manual operation.

TESTING:

The system was tested under various environmental conditions to evaluate its performance, accuracy and reliability. Each sensor and component was examined individually to ensure proper functioning. The testing process included both automatic as well as manual modes to verify system response under different scenarios. The results confirm that the system operates efficiently and responds correctly to all conditions.

1. Temperature Condition

The temperature sensor (DHT11) was tested by increasing the surrounding temperature above the set threshold. When the temperature exceeded the limit, the Arduino activated the relay and turned ON the fan. The system responded quickly and reduced the temperature effectively.

2. Light Condition

The LDR sensor was tested by reducing the light intensity (covering the sensor). When the light level dropped below the threshold, the system automatically switched ON the light. When normal light conditions were restored, the light turned OFF.

3. Gas Leakage Condition

The gas sensor was tested by exposing it to gas. When gas concentration exceeded the safe limit, the buzzer was activated and a warning message was sent through the GSM module. This confirmed proper detection and alert functionality.

4. Normal Condition

Under normal environmental conditions, all sensor values remained within safe limits. The system maintained stable operation without activating any devices unnecessarily.

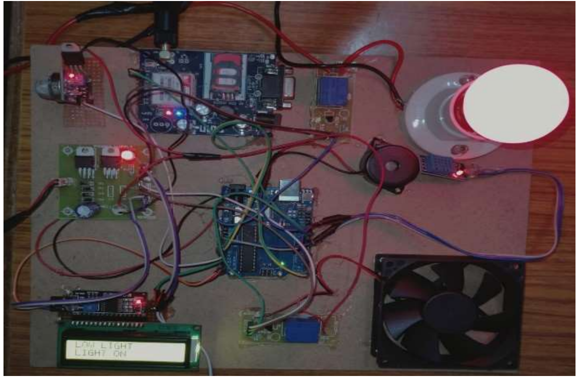
5. Manual Control Condition

The system was tested in manual mode by sending commands to control devices. The fan and light responded correctly to ON and OFF commands, confirming proper manual operation.

The below shown figures depict the working of the kit in different conditions.

- All the parameter readings are shown:

- Output in different coditions:



- Output display in low light condition:

COMPARISON BETWEEN AUTOMATIC MODE AND MANUAL MODE:

S.no	**Parameter**	**Automatic Mode**	**Manual Mode**
1	Operation Type	Fully automatic	User-controlled
2	Human Intervention	Not required	Required
3	Decision Making	Based on sensor data and predefined conditions	Based on user commands
4	Control Method	Sensors+ Arduino logic	GSM / user interface commands
5	Response to Environment	Automatically adjusts to changes	Does not depend on environment
6	Flexibility	Less flexible	More flexible
7	Accuracy	High (sensor-based)	Depends on user input
8	Safety	High (automatic alerts and actions)	High (alerts still active)
9	Device Control	Automatic ON/OFF based on thresholds	Manual ON/OFF by user
10	Usage	Continuous monitoring systems	Situations requiring user control

Conclusion

The project on **Industrial Automation and Control Using Cyber-Physical Technologies** demonstrates the transformative potential of integrating physical industrial systems with digital intelligence. By employing sensors to continuously collect data from the physical environment and processing this data through microcontrollers, the system enables real-time monitoring and control of industrial operations. This close coupling of cyber and physical components enhances accuracy, reduces manual intervention, and increases overall productivity.

The developed prototype illustrates that Cyber-Physical Systems (CPS) can significantly improve industrial efficiency while supporting safer operational conditions. Furthermore, the project highlights the critical role of communication networks and data analytics in modern smart manufacturing. Overall, the study confirms that CPS-based industrial automation offers substantial advantages and can serve as a foundation for broader industrial adoption.

Despite increasing interest in Industry 4.0 solutions, current research remains fragmented, often focusing on isolated aspects of automation. To address this, the concept of **Intelligent Cyber-Physical Systems (iCPS)** integrates autonomy, adaptive intelligence, and energy-efficient mechanisms. iCPS frameworks enable fault prediction, self-adaptation, and autonomous decision-making, positioning them as key enablers for next-generation smart manufacturing.

Scope of the Project

The project's scope extends to several advanced industrial applications. Future developments may include:

1. **Integration with IoT and Cloud Computing:** Connecting machines and sensors via industrial networks enables centralized monitoring and control of entire production lines.
2. **Artificial Intelligence Enhancements:** Machine learning algorithms can be applied to predict maintenance needs, optimize processes, and reduce downtime.
3. **Wireless Connectivity:** Technologies such as Wi-Fi and 5G can facilitate remote monitoring and control of industrial operations.
4. **Smart Factory Implementation:** Fully networked machines can autonomously adjust operational parameters in real-time based on collected data, creating a self-optimizing production environment.

In conclusion, this project provides a robust foundation for developing advanced Industry 4.0 automation systems. It offers significant opportunities for future research in CPS, AI-driven industrial intelligence, and smart manufacturing technologies.

References

1. Edward A. Lee, “Computing Foundations and Practice for Cyber-Physical Systems: A Preliminary Report,” Technical Report UCB/EECS-2007-72, May 2007.
2. Roy H. Campbell, Guy Garnett, Robert E. McGrath, “CPS Environments,” NSF Workshop on Cyber-Physical Systems, Austin, TX, 2006.
3. Brian D. Noble, Jason Flinn, “Wireless, Self-Organizing Cyber-Physical Systems,” NSF Workshop on Cyber-Physical Systems, Austin, TX, 2006.
4. A. W. Colombo, S. Karnouskos, O. Kaynak, Y. Shi, S. Yin, “Industrial Cyber-Physical Systems: A Backbone of the Fourth Industrial Revolution,” IEEE Industrial Electronics Magazine, vol. 11, no. 1, 2017.
5. R. R. Rajkumar, I. Lee, L. Sha, J. Stankovic, “Cyber-Physical Systems: The Next Computing Revolution,” Proc. 47th Design Automation Conf., ACM, 2010, pp. 731–736.
6. S. Sarma, N. Dutt, P. Gupta, N. Venkatasubramanian, A. Nicolau, “Cyber-Physical System-on-Chip (CPSOC): A Self-Aware MPSoC Paradigm with Cross-Layer Virtual Sensing and Actuation,” DATE Conf., March 2015, pp. 625–628.
7. F. D. Macías-Escrivá, R. Haber, R. del Toro, V. Hernandez, “Self-Adaptive Systems: A Survey of Current Approaches, Research Challenges, and Applications,” Expert Systems with Applications, vol. 40, no. 18, 2013.
8. C. Alippi, M. Roveri, “The (Not) Far-Away Path to Smart Cyber-Physical Systems: An Information-Centric Framework,” Computer, vol. 50, no. 4, pp. 38–47, 2017.
9. C. Krupitzer, F. M. Roth, S. VanSyckel, G. Schiele, C. Becker, “A Survey on Engineering Approaches for Self-Adaptive Systems,” Pervasive and Mobile Computing, vol. 17, pp. 184–206, 2015.
10. C. Alippi, V. D’Alto, M. Falchetto, D. Pau, M. Roveri, “Detecting Changes at the Sensor Level in Cyber-Physical Systems: Methodology and Technological Implementation,” IJCNN, 2017, pp. 1780–1786.