Article title: IOT BASED INDUSTRIAL PARAMETERS MONITORING AND CONTROLLING SYSTEMS
Authors: Atharva Ghodvaidya[1], Hritik Biswas[2], Utkarsh Kadu[3], Madan Gughe[4]
Orcid ids: 0000-0002-4923-9368[1]
Contact e-mail: atharvawork05@gmail.com
License information: This work has been published open access under Creative Commons Attribution License http://creativecommons.org/licenses/by/4.0/, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. Conditions, terms of use and publishing policy can be found at https://www.scienceopen.com/.
Preprint statement: This article is a preprint and has not been peer-reviewed, under consideration and submitted to ScienceOpen Preprints for open peer review.
DOI: 10.14293/S2199-1006.1.SOR-.PPHN7BE.v1
Preprint first posted online: 17 May 2022
IOT BASED INDUSTRIAL PARAMETERS MONITORING AND CONTROLLING SYSTEMS

Department of Electrical Engineering Marathwada Mitra Mandal’s College Of Engineering, Karvenagar, Pune - 52

Guided by:
Prof. Akshay Suryawanshi

Authors:
Hritik Biswas
Atharva Ghodvaidya
Madan Gughe
Utkarsh Kadu
ABSTRACT

Safety is exceptionally principal in any industry, particularly with manufacturing, producing businesses and numerous others. Therefore we mean to help these issues in businesses by fostering a safety boundaries monitoring and controlling system, also, making it more proficient and easy to understand by consideration of IoT. That's what we trust: a system ought to be all around automated with the goal that another client or another worker who has no related knowledge in controlling a unit ought to have the option to get familiar very without any problem.

With the assistance of IoT, administrators will come to know the live status of a unit on when a boundary monitoring and controlling system is introduced, it very well may be finished by means of sends, or on the other hand on the off chance that an individual is available there they can notice themselves. For instance we can draw temperature to a specific line and assuming temperature surpasses past the set limit, the fans or other cooling system will begin consequently. This system will likewise have gas sensors, flame sensors as well as radiation sensors. We are utilizing Arduino UNO ATmega 328 as a controller for this system.

Finally the administrator will have records or logs of the boundaries variance also, different exercises at a specific time so it will be prepared for reference in future Also, this will assist the administrator with going to security lengths.
INTRODUCTION

Technology development is an unending process and hence it is necessary as far as we're concerned to be well equipped and aware of the new upgrades in technology. These technological changes have in this way acquired ease everyday human existence. Automation has become the need of the day. Today every one of information is available on the internet and web technology is developing very quickly. Embedded systems with web technology provide remote management and controlling of embedded devices by means of network interface, Internet of Things (IoT) etc. Devices are controlled by a web controller or E-controller, which is the most renowned method for web development over the world. Remote login and monitoring by building a distributed web control system with the help of web pages worked in web applications is presently used instead of involving enormous server systems for monitoring, overseeing and taking care of information.

These sorts of web control systems with IoT are characterized by: Energy Saving, Solace, and Efficiency. Our essential objective is to apply the Internet control system to the Internet of things, with the end goal that the customers can use the application from any place all over the planet with the help of Internet facilities.
# COMPONENTS

<table>
<thead>
<tr>
<th>COMPONENTS</th>
<th>TYPES</th>
<th>SPECIFICATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gas Sensor</td>
<td>MQ135</td>
<td>5V, 20°C 10-300ppm for NH3 and Alcohol. 10-1000ppm Benzene</td>
</tr>
<tr>
<td>Flame Sensor</td>
<td>LM393</td>
<td>5V, 15mA, 760-1100nm wavelength</td>
</tr>
<tr>
<td>Radiation Sensor</td>
<td>LDR</td>
<td>100mW, 540 nm, 10 Lux</td>
</tr>
<tr>
<td>Temperature Sensor</td>
<td>LM35</td>
<td>5V, 10mA (-60° to 150° C)</td>
</tr>
<tr>
<td>Controller</td>
<td>Arduino UNO AT mega 328</td>
<td>5V, 50mA, 16MHz, 6 pins</td>
</tr>
<tr>
<td>LED</td>
<td></td>
<td>3V, 25mA</td>
</tr>
<tr>
<td>Buzzer</td>
<td></td>
<td>3V, 25mA</td>
</tr>
<tr>
<td>5V DC Fan</td>
<td></td>
<td>5V</td>
</tr>
</tbody>
</table>
LabVIEW, short for Laboratory Virtual Instrument Engineering Workbench, is a programming climate where you make programs using a graphical documentation (interfacing useful nodes by means of wires through which data streams); in such manner, it varies from standard programming dialects like C, C++, or Java, in which you program with text. Regardless, LabVIEW is considerably more than a programming language. It is an intuitive program advancement and execution framework intended for people, like researchers and designers, who need to program as a feature of their positions.

LabVIEW improvement climate deals with PCs running Windows, Mac OS X, or Linux. LabVIEW can make programs that sudden spike in demand for those stages, as well as Microsoft Pocket PC, Microsoft Windows CE, Palm OS, and an assortment of inserted stages, including Field Programmable Gate Arrays (FPGAs), Digital Signal Processors (DSPs), and chip. LabVIEW program advancement climate is not the same as standard C or Java improvement frameworks in one significant regard: While other programming frameworks use text-based dialects to make lines of code, LabVIEW utilizes a graphical programming language, frequently called "G," to make programs in a pictorial structure called a square chart. Graphical programming kills a ton of the linguistic subtleties related with text-based dialects, like where to put your semicolons and wavy supports. Graphical programming permits you to focus on the flow of data inside your application on the grounds that its basic grammar doesn't obscure what the program is doing.
The system works in 4 phases of operation. The first phase is DATA ACQUISITION. The sensors used for this purpose are GAS SENSOR, TEMPERATURE SENSOR, FLAME SENSOR, LIGHT SENSOR. These sensors are used to collect real time data on the respective parameters which are then feeded to the controller. The second phase is DATA PROCESSING. The data acquired by the sensors are compared to the set
safety limits by the users depending on the area of utilization. The third phase is DATA LOGGING.

The data collected in real time is then sent to the user by means of email. The user has moreover got the office to screen the sensor acquired data in real time by means of IOT. The fourth phase is SAFETY ALERT. In this phase the user or the people working in the area of deployment is made aware of the emergency by means of different arrangements and fundamental preventive measures are triggered too.

The real time monitoring can be likewise done by means of a DIGITAL MONITORING PANEL made utilizing NI-LabVIEW. The panel has likewise got a secondary panel called SETTING. This panel has a feature which gives the immediate user flexibility to set the safety limit according to the circumstances in the area of deployment. The DIGITAL MONITORING PANEL and SETTING panel are explained in detail in the following paragraphs.

SIMULATION
The above image is the front end of the presentation panel. The panel has parameters shows like TEMPERATURE, LIGHT INTENSITY, HAZARDOUS GAS LEVEL and FIRE DETECTOR. The panel has additionally got a beginning/stop button to run the systems at required time to screen the given attributes. It has additionally got a TIME and DATE for SYSTEMATIC DATA LOGGING. The system is likewise capable of sending the user with notification through email. The panel has another window called SETTING. The setting window gives the flexibility to the user to SET THE LIMIT to activate the safety/disturbing systems.

The SETTING window is explained in detail in the following paragraphs. The first display (from left to right):

**TEMPERATURE:**
The temperature display shows the real time temperature in the deployed area. LM35 is used to monitor the temperature. The limit for safe temperature is set on the setting page. Once the temperature exceeds the given temperature limit, the safety fan is activated automatically to cool down the affected area. Once the temperature goes below the safety limit the safety fan is turned off.
RADIATION/LIGHT INTENSITY:
The light intensity sensor is used as a power saving smart sub-system. This sub-system is designed in such a way that the light would be turned on only when the ambient light would be less than the specified limit. Thus resulting in power saving. LDR is used for this application.

HAZARDOUS GAS SENSOR:
Hazardous gas leaks are one of the common occurrences in manufacturing, metallurgy, and chemical industries etc. We have used an MQ-135 sensor to detect the hazardous gas level in the working area and as soon as the safety limit is crossed, the safety fan is turned on to provide adequate ventilation.

FLAME/FIRE SENSOR:
LM 393 is the flame sensor used in our system. The working of this part is quite simple as soon as the sensor detects fire, a buzzer which acts as a safety alarm to alert the individuals in the vicinity.

This is the settings panel of the system. The parameters like TEMPERATURE, HAZARDOUS GAS, and LIGHT INTENSITY can be set utilizing this panel. This feature gives the user the flexibility to set the parameters as they desire in accordance with the
equipment that this device is being implemented to. The way to save the data logging file can likewise be set utilizing this panel.

CONCLUSION
The prototype systems worked true to form all through the parameters. The feature to permit the limits to be set by the user seems to be giving huge operational flexibility for the system. This flexibility makes the system deployable in a number of scenarios according to requirement. The project has additionally proven that it has persuaded potential to be scaled up to cater the demand and that's just the beginning and more sensors also, sub-systems can be integrated too.

REFERENCE


[6] Connecting Arduino UNO to PC to use with LabVIEW Connecting Arduino Uno to PC:

Connecting Arduino Uno to PC to Use with LabVIEW - NI

[7] Introduction to LabVIEW programming environment:

Introduction to the LabVIEW Programming Environment - NI

[8] Monitor and Log Data with LabVIEW Real-Time:

Monitor and Log Data With LabVIEW Real-Time - NI

[9] Using sensors with Arduino:

Using Sensors with Arduino - Arduino Project Hub

[10] Arduino cloud IoT:
https://docs.arduino.cc/cloud/iot-cloud/