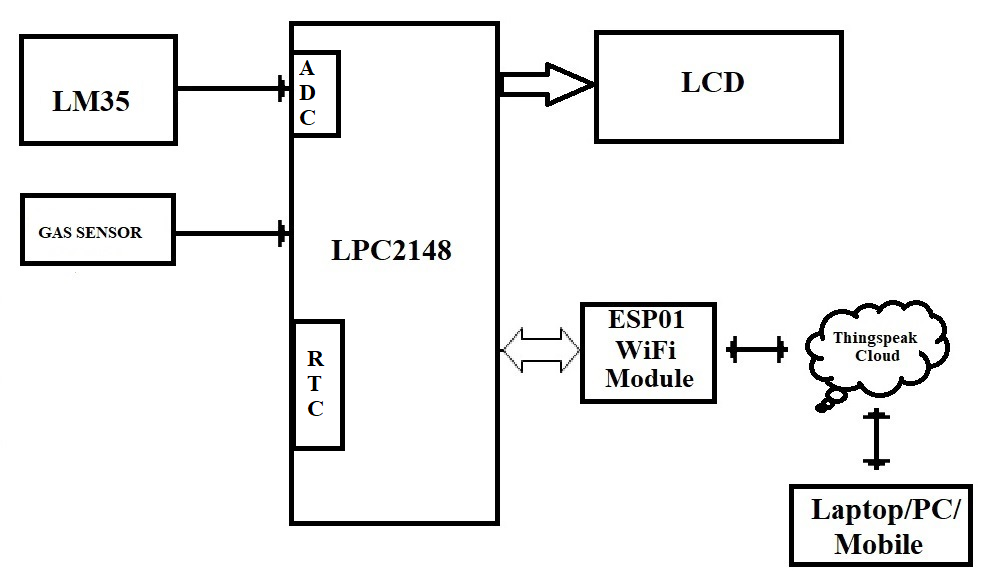
**SMART INDUSTRIAL PROCESS MONITORING SYSTEM USING IOT**

**AIM:** The main aim of the project is to monitor the industrial parameters (Temperature and gas level) using IOT.

# INSIGHT:

* Good knowledge of C-Programming.
* Knowledge of Embedded-C programming
* Thorough with the LPC2148 Architecture, General purpose I/O Interface
* Understanding of Wi-Fi module usage.
* Understanding of ADC.
* Understanding of LCD.
* Understanding of on-chip RTC utilization.

**Block diagram:**

****

**REQUIREMENTS:**

**HARDWRAE REQUIREMENTS:**

* LPC2148
* LM35
* MQ-2 GAS SENSOR
* LCD
* WI-FI MODULE (ESP01)
* DB-9 CABLE/USB-UART CONVERTER

**SOFTWARE REQUIREMENTS:**

* KEIL C Compiler
* PROGRAMMING IN EMBEDDED C
* Flash Magic

**Steps to be followed to complete your project:**

* Create New Folder in your laptop/PC and save that folder with your project name.
* Then copy what you done files -lcd.c, lcd.h, delay.c, delay.h, uart.c&uart.h, into project folder.
* Individually can check each and every module.
* First check LCD to display character constant, string constant and integer constant.
* Next Check UART peripheral by transmitting string constant on hyper terminal.
* Then download the UART interrupt code and check the working on hardware board.
* Next connect lm35 sensor to LPC2148 and develop the driver for reading LM35 information (temperature value) using on-chip ADCand display on LCD. And check the MQ-2 gas/smoke sensor with microcontroller interface. Take the digital result to know gas leakage occurred or not from MQ-2 gas sensor.

**Note: refer on-chip LPC2148 ADC code from LMS**

* Next check the working condition of ESP01 with the help of flash magic terminal. (Which commands are required to use for your project refer the screen shots under reference data folder)
* Next connect the ESP01 module to LPC2148. (You can refer the data sheet of ESP01 to know how to connect)
* Then develop the driver for ESP01 for LPC2148. By using ESP01 driver, send one constant data to thingspeak clod.
* After checking sensors and ESP01 driver then implement the main logic. Inside main initialize all required peripherals.
* In continuous loop, for every second read current temperature value and gas level status on LCD along with real time clock. Then send the same information (only temperature value) to thingspeak cloud with respect to sometime-gap (for every 3 mins). Use in-built RTC for specific time interval. And whenever gas leakage occurred, then only send gas sensor information to thingspeak.
* If you’re getting this output then your project is completed.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* **ALL THE BEST** \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**DIFFERENT IOT APPLICATION PLATFORMS: (FYR)**

**AWS IoT:** Amazon Web Services (AWS) IoT is a cloud-based platform that allows users to connect and manage IoT devices and applications. It provides a wide range of services for data collection, storage, analysis, and visualization.

**Microsoft Azure IoT:** Microsoft Azure IoT is a comprehensive cloud-based platform that provides tools and services for IoT device management, data analysis, and application development. It supports a wide range of devices and platforms, including Windows and Linux.

**Google Cloud IoT:** Google Cloud IoT is a cloud-based platform that provides services for IoT device management, data processing, and application development. It supports a wide range of devices and platforms, including Android and iOS.

**IBM Watson IoT Platform:** IBM Watson IoT Platform is a cloud-based platform that provides services for IoT device management, data analysis, and application development. It supports a wide range of devices and platforms, including Raspberry Pi and Arduino.

**ThingSpeak:**ThingSpeak is an open-source Internet of Things (IoT) application platform and API (Application Programming Interface) that allows users to collect, analyze and act on data from various sources. It was developed by MathWorks, a leading provider of technical computing software. ThingSpeak provides an easy-to-use interface for users to create and manage IoT devices and applications. It supports a wide range of data sources, including sensors, web services, and social media. The platform also includes features for data visualization, data analysis, and real-time alerts.

**Cayenne:** Cayenne is a cloud-based IoT platform that provides services for device management, data visualization, and automation. It supports a wide range of devices and platforms, including Arduino and Raspberry Pi.

**Kaa IoT Platform:** Kaa IoT Platform is an open-source IoT platform that provides tools and services for IoT device management, data processing, and application development. It supports a wide range of devices and platforms, including Android and iOS.

**Ubidots:**Ubidots is a cloud-based IoT platform that provides services for data collection, visualization, and analysis. It supports a wide range of devices and platforms, including Arduino and Raspberry Pi.

**Losant:**Losant is a cloud-based IoT platform that provides services for device management, data visualization, and automation. It supports a wide range of devices and platforms, including MQTT and HTTP.

**Xively:**Xively is a cloud-based IoT platform that provides services for device management, data analytics, and application development. It supports a wide range of devices and platforms, including Arduino and Raspberry Pi.

**Blynk:** Blynk is a mobile app and cloud-based IoT platform that provides services for device management, data visualization, and control. It supports a wide range of devices and platforms, including Arduino and Raspberry Pi.

**Particle:** Particle is an IoT platform that provides tools and services for device management, data visualization, and firmware development. It supports a wide range of devices and platforms, including Wi-Fi and cellular.

**Thinger.io:** Thinger.io is a cloud-based IoT platform that provides services for device management, data collection, and analysis. It supports a wide range of devices and platforms, including ESP8266 and ESP32.

**DeviceHive:**DeviceHive is an open-source IoT platform that provides services for device management, data analytics, and application development. It supports a wide range of devices and platforms, including Java and .NET.

**Mongoose OS:** Mongoose OS is an open-source IoT platform that provides services for device management, data analytics, and firmware development. It supports a wide range of devices and platforms, including ESP32 and STM32.

**Hologram:** Hologram is a cloud-based IoT platform that provides services for device management, data collection, and visualization. It supports a wide range of devices and platforms, including Raspberry Pi and BeagleBone.

**ioBroker:**ioBroker is an open-source IoT platform that provides services for device management, data visualization, and automation. It supports a wide range of devices and platforms, including Node.js and Docker.

**Cayenne:** Cayenne is a cloud-based IoT platform that provides services for device management, data visualization, and automation. It supports a wide range of devices and platforms, including Arduino and Raspberry Pi.

**Altair SmartWorks:** Altair SmartWorks is a cloud-based IoT platform that provides services for device management, data analytics, and application development. It supports a wide range of devices and platforms, including industrial sensors and machines.

**Helium:** Helium is a decentralized IoT platform that provides services for device management, data collection, and application development. It supports a wide range of devices and platforms.