Chapter 1

Introduction

SCADA is a type of controlling system that allows us to continuously monitor and control the system that are present among various remote areas. A perfecttly developed SCADA system deliberately has a efficieny for money as well as time discarding the need for labour work to particularly visit each remote area for data sysnthesis or to make any electrical or mechanical adjustments. Till today, the basic measurement techniques using micro-controllers are considered most common.The fully developed and smart embedded system based upon electrical quantity measurement systems has been proposed.In this particular paper, the specific use of LabVIEW for monitoring purpose with a microcontroller for Data Acquisition(DAQ)system is implenented to showcase a real time monitoring of generated mechanical and electrical parameters such as like motors rortating at certain speed and can be controlled at what speed the motor should run.

The LED switching up when there is any high speed of the rotatory functions. The simulation is shown using the proteus8 software where the proteus is linked to the LabVIEW software. For the implementation for this project, the above mechanical and electrical parameters are transfered to the LabVIEW software and remotely controlled via the proteus as well as the LabVIEW.

The ECU system used for sensor failure is deliberately in use for most of the vehicles of this generation. This is used as a practical use in cars mostly if there is any kind of sensor failure. The alarm function switches on if there is any kind of failure proposed by the engine. The Electronic Control Unit(ECU) collects all the data from the ECU memory installed which is acquired by various sensors.

1.1 Classification of SCADA System

SCADA systems are classified into four types which can be considered as SCADA architectures of four dissimilar generations:
1.First Generation: Early SCADA systems or, Monolithic
2.Second Generation: Distributed SCADA systems
3.Third Generation: Networked SCADA systems
4.Fourth Generation: Internet of things (IoT technology), SCADA systems

1.2 Motivation

The Electronic Control Unit(called as ECU) which is used here for vehicle Engine acquires the data from various sensors such as temperature sensor(Cooling Liquid Temperature sensor),pressure sensor(suction manifold sensor),oxygen sensor also called as lambda sensor, throttle position(in degrees) as well as the voltage supply sensor to detect the voltage supplied for the ECU to inhabit various parameters. Also, a small scale SCADA acquires the data from the Atmega16 micro-controller based circuit which is done using the proteus software. The detected values are then transferred into the LabVIEW software for monitoring as well as controlling mechanism. Thus this has motivated us for further stage of SCADA and ECU implementation.

1.3 Problem Statement

ECU Designing and small scale SCADA application uisng LabVIEW and Microcontroller.

1.4 Objectives

The main objectives of our project are :
• To demonstrate the ECU application with sensors and monitoring on the output panel of LabVIEW.
• To demonstrate the results of data transfer of the small scale SCADA system with ATmega 16 microcontroller.

1.5 Scope

The project is developed for monitoring the application of Engine Control Unit(ECU) inhabiting self control or accessing small scale SCADA monitoring system to the circuit simulation software for "to and fro" data transfer. Future SCADA technology can be used for nuclear,thermal and hydro power-plants as well as the industrial automation. On a large scale it will be configuring almost thousands of actuators and sensors repeatedly.

1.6 Outcome of project

The main outcome of the project is to demonstrate the monitoring system for self control from stored data as well as controlling the system with the microcontroller.

1.7 Methodology

• Developing model based on mathematical expressions in the LabVIEW using G programming.
• Simulating the block diagram to check the errors persisting.
• Circuit designing in the proteus8 using the microcontroller coded with assembly language and dumped in hex file.
• Controlling the LabVIEW simulation independently or with the microcontroller.

Chapter 2

Literature survey

The purpose of the literature survey is to identify the information relevant to "LabVIEW Simulation", "ECU designing" and "Microntroller interface".

Shuangshuang Li Baochen Jiang[1] proposed the system which is used for data acquisition, monitoring,proce- dure control for spot devices moreover is a computer based production procedure control and dispatching automation system and how the microcontroller can be interfaced with the provides.As in this format all the research issues that are involved in strengthening the best statistics of SCADA networks, describes the general architecture of SCADA networks and the significant properties of some of the commonly used SCADA communication protocols and the ongoing work in the several SCADA security areas such as improving access control, firewalls and intrusion detection systems, SCADA protocol analyses, cryptography and key management, device and operating system security, concludes with an overview of these standardization efforts.

Igure, V.M., S.A. Laughter, and R.D. Williams[2-3] prposed the most of the SCADA environment in the networking domain C + + language to be used for program development system on Windows NT, Microsoft SQL Server provides some of the strong support of the data in the database to provide effective management and use of beneficial effective measures to achieve data integrity and data security case.

Mayur Avhad, Vinit Divekar and Harshad Golatkar [4] proposed the increasing need of wireless SCADA system, one major and critical application could be in the detection of leakages in the underground water pipelines. In such a system, wireless sensor network can be used to effectively monitor the data related to pressure and flow at various locations in the pipelines.The advantage of using SCADA system are Higher productivity, superior quality of system end product, Efficient usage of materials and energy improved and Safety in working condition. This Paper has clarified itself to study the integral parts of the entire process which is involved, their implementation and the problems that may show up has also been given their due importance.

Both the real system and its model process the same input signals.[5,6,7,8] The output of the system is compared with the expected output signal anticipated on the basis of the model. The difference between those signals, the so called residuum R, is directed to the diagnostic system DIAG. When the residuum equals zero (with particular set exactness), it means that the current behaviour of the system does not diverge from the expected

behaviour obtained on the basis of the familiar model; in such a case it should be stated that the system works properly.

Modern systems of controlling internal combustion engines with spark ignition are very complex units [9,10]. On the one hand, it means a danger of lower reliability resulting from a growing number of components. However, on the other hand, diagnosing by means of ECU is becoming possible. An electronic control unit (so called ECU) gathers information from various sensors: cooling liquid temperature sensor, inlet air temperature sensor, sensor of absolute pressure in the suction manifold, engine rotational speed sensor or sensor of oxygen in fumes.

Chapter 3

Methodology

3.1 Design Methodology And Implementation

The chapter deals with the method used for implementing ECU Sensor Designing with LabVIEW and a small scale SCADA application with LabVIEW and ATMega16 microcontroller via Proteus8. The algorithm used is:
• G(Graphical) Modelling

3.2 Practical Model of Electronic Control Unit – ''Effective ECU Model"

The basic simulation model of the system was delivered in Graphical language via the LabVIEW software. The main part of the system is to demonstrate the monitoring system on the front panel which enables the final output for the fuel injection time depending on input data/parameters(signals).

The input signals are as follows:
• Temperature sensor for cooling liquid temperature
• Engine Rotational Speed sensor for detecting the rpm of the vehicle
• Throttle angle to check the evasion for bypassing
• Presurre sensor for the Absolute Pressure for the suction manifold
• Oxygen/Lambda Sensor to detect the oxygen level in volts.
• Voltage supply for the injection-ingnition system.

Final output is represented on the monitoring panel consisting of the basic injection time(ms) as well as the failure control showcased as "Check Engine". To simulate the above mentioned sensor failures,logical buttons are also introduced.

The following failures are simulated in the developed system:
• Failure of the Oxygen/Lambda sensor
• Failure of Cooling Liquid Temperature sensor sensor,
• Failure of Engine Rotational Speed sensor
• Failure of Throttle position sensor

Figure 3.1 shows the monitoring and control panel of spark ignition system Engine Driver. The whole setup is developed using the G program. Every sensor has been designed by mathematical formulas used to construct sensors so that it acquires the data in terms of bits from the ECU memory which is installed in the main memory which is a three dimensional mapping unit.The data is synthesized in a array Index function used. The array is perfectly scaled and is operated for every sensor data to be grasped from the memory. The main tested array is connected to the n-dimensional array system for input and by means of 0 to n-1 inputs that can be determined which element is ready to read the perfect calculated values.

3.3 Spark Ignition Engine Virtual Driver

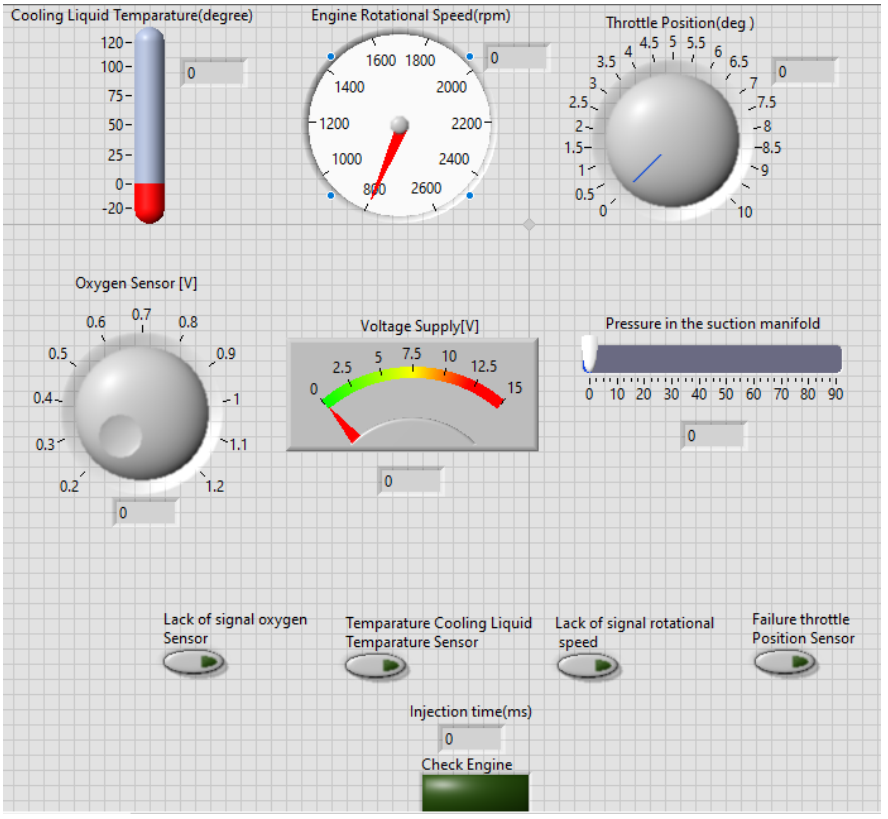

Figure 3.1: Monitoring and Control panel of the ECU using LabVIEW program

The array index of the 3 dimensional(3D) map of the injection which is shown in the Fig 3.2 .Parameters are set in this block so that sensor will be acquiring the data and will give the simultaneous output for the front panel monitoring so that we can set the data at particular range. The data here is as :Cooling Liquid Temperature as well as the Engine Rotational Speed and Absolute Injection in the Suction Manifold. This combined demonstrates the mapping of the ECU system.

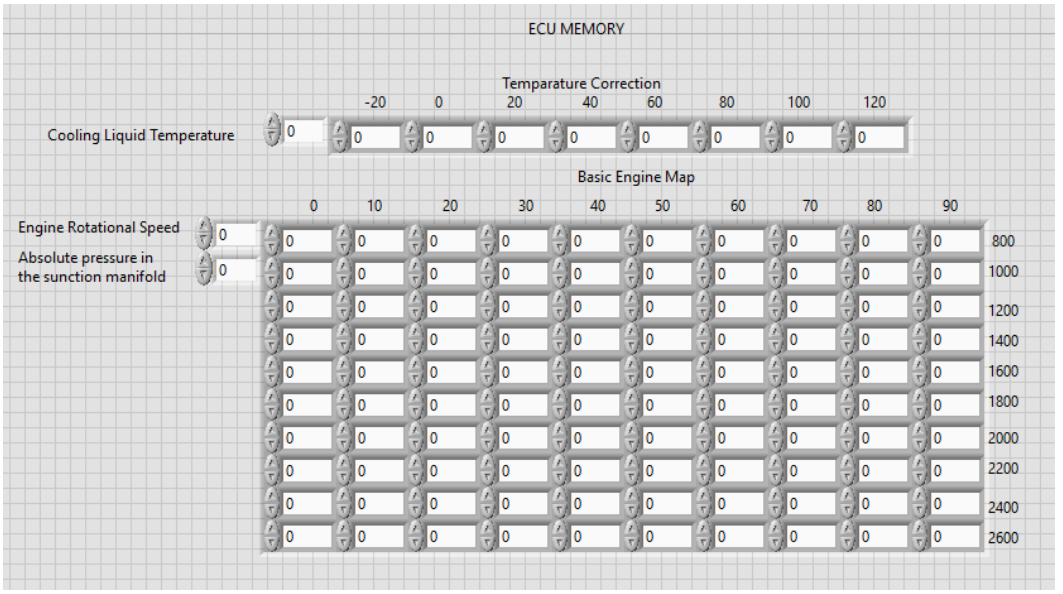

Figure 3.2: Three-dimensional map of basic injection

3.3.1 Array introduced for the Injection time

The block diagram shown in Figure 3.3 consists of engine rotational speed which symbolises the rpm of the engine. This variable is set to its maximum value(2200 rpm) and is connected to an array index which collects the data even from the pressure variable(pressure in the suction manifold)and with mathematical formulas shows the basic injection time(ms) on the front panel. The basic injection time gets input also from the case structure of the temperature module.The rpm of the engine and the pressure is controlled by the user.

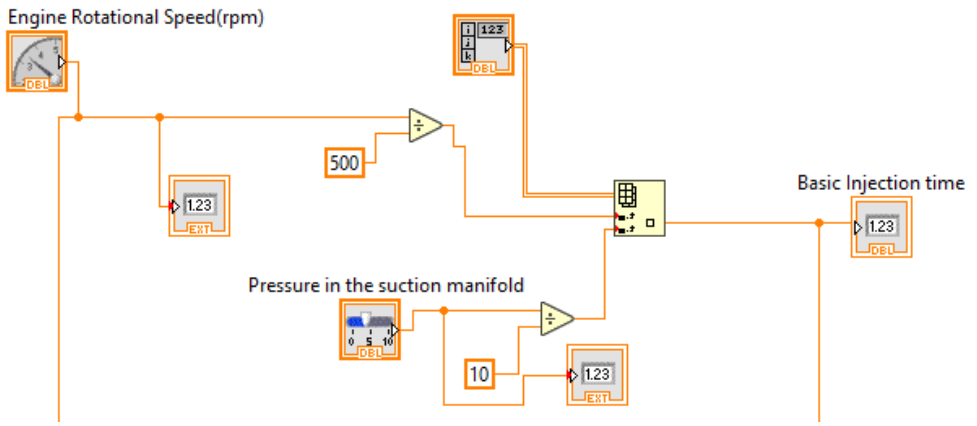

Figure 3.3: Monitoring Engine speed and the pressure in the suction manifold

3.3.2 Injection time depending on Temperature

The block diagram contains the model of a temperature sensor which is connected to a logical case structure having true/false for the data flow. If the case is true then the data flows from the temperature sensor given values by the user and this is showcased on the monitor of the control panel.There is a temperature correction array which stores the data collected from the array. And it shows the correction of the temperature for the injection time. And if the logical case structure is false then there will not be any flow of data.It is also connected with a boolean logic which will show the failure of the data collection.

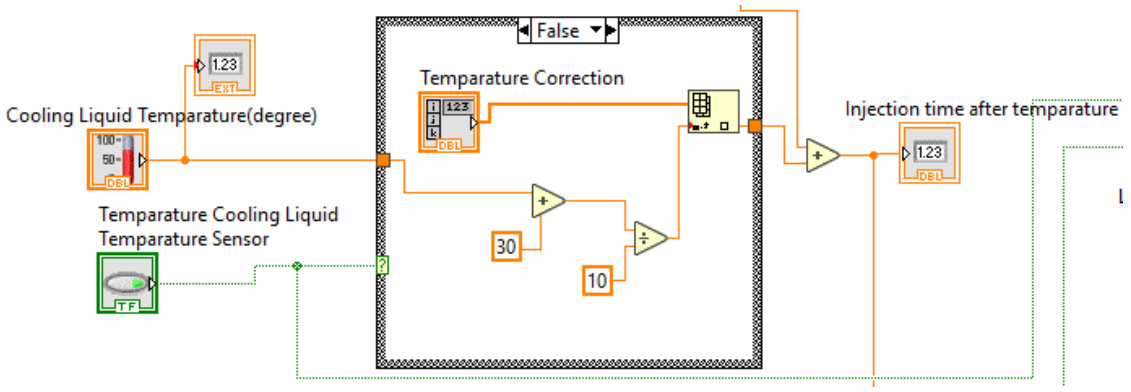

Figure 3.4: Correcting the injection time(ms) depending upon the liquid temperature

3.3.3 Time correction in relation with voltage supply and oxygen sensor

As shown in the Figure 3.5 Voltage supply[V] is calculated by mathematical formulas with the constants. It is multiplied by the constant 2 and then added to another constant to give out the injection time after voltage correction. Also the Oxygen sensor/Lambda sensor[V] is connected to a logical case structure giving the final injection time(ms) as the output. The boolean logic for the lack of oxygen sensor is also applied to the case structure so that when the range of the sensor doesn't be in position then it will showcase the sensor failure.
Thus the result obtained before the injection time correction is slightly different from the injection time after the temperature correction. And the data of the basic injection time is further classified as the injection time after cc. This intern monitors the temperature of the whole oxygen sensor while it gets the voltage supply from the voltage meter. The constants multiplied by -1 is set to nullify the ignition-injection system and again to take the actual reading for the times specified.In such a case, almost 30 percent of the ait to fuel mixture ration is added further to the injection time. If there is a lambda/oxygen sensor failure then the output has its constant value equalling almost 20 percent of the calculated injection time.

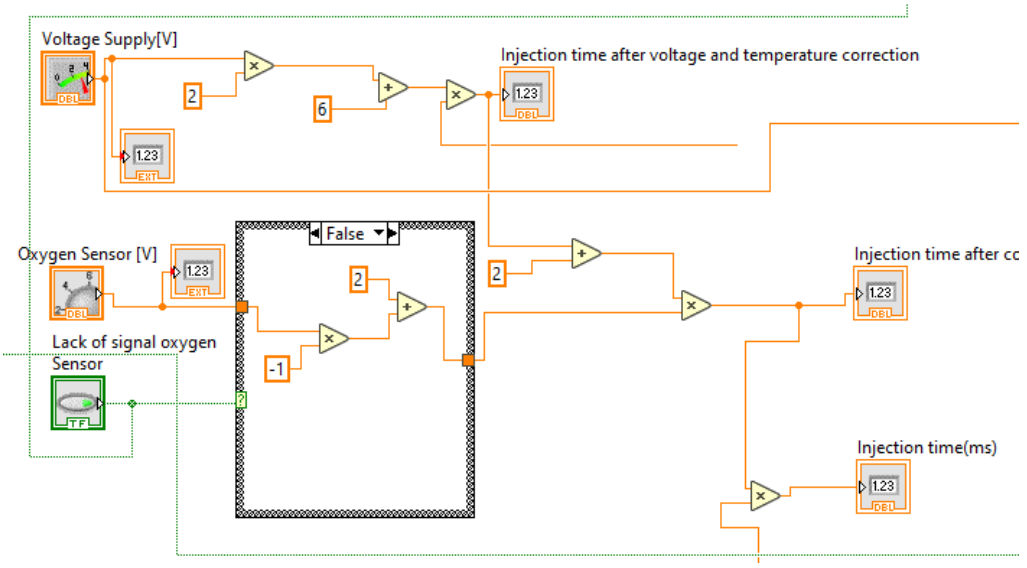

Figure 3.5: Injection time correction with respect to voltage supply and lambda sensor

3.3.4 Fuel cut-off during engine braking

Figure 3.6 shows the system during an Engine breaking, the throttle position of any vehicle is in action by the engine rotational speed(rpm).In this case, if the rpm of the engine exceeds 2200 revs/min then it will compare with the zero constant. As the rpm of the engine will at zero position in the first place of intact so, it attempts to close the connection with the case structure as the ECS is assembled with the logic sign to compare. And we have set the range of the engine speed maximum to 2200 revs/min.Thus,when the breaking is in action the system will check the flaws and eliminate the erros obtained. Such a concept is used to check the failure of the speed system and here if such problem occurs then it will showcase the entire engine failure by monitoring the Engine hault message. The comparison check is further detailed by the nested loop logic case. As the engine will check if the problem persists while applying the breaks, it will sequence the failure for further check in a loop. If the failure attempts on every step then it shows the failure on the control panel.

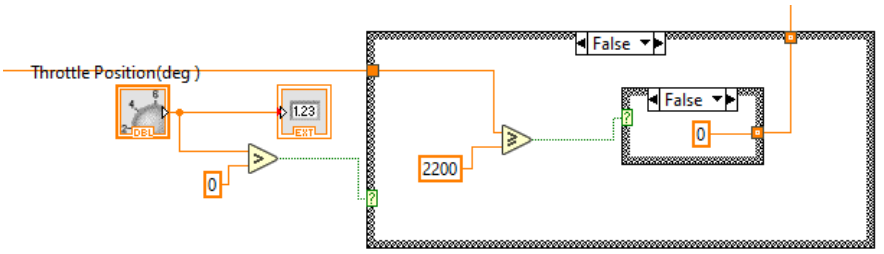

Figure 3.6: Throttle positioning at the time of engine breaking to realize fuel cut-off

3.3.5 Message display for failure of the signals

This block diagram explains of how a message is displayed when failure of the sensor occurs. As we have several sensors like Temperature coolant sensor, pressure sensor to check pressure in the suction manifold,voltage supply system,oxygen sensor as well as the throttle positioning sensor module and the engine rotational speed. When there is a lack of signal in the rotational speed of the engine then it will show the message to stop the engine. As here every signals of the sensors are connected to the OR logic which will finalize what kind of failure has occured. And if the voltage supply is less than 10V the lack of oxygen sensor is displayed. In case of the all the sensors failure then there will an alarm showing the "Check Engine" message.
As the entire system in running inside a while loop then it will continuously operate the parameters unless and until the engine comes to a stop either by any sensor failure or by the physical stop. The sensors failures like lack of signal for rotational speed as well as the failure of the throttle will also be monitored. The stop signal depends upon the logic case structures. Here the shift registers also plays an important role as it is used to store the temporary data for displaying the specific message on the panel.This reflects the ECU memory and can define at what stage of the cut-off the message should be alarmed.In this simulation as shown in the Figure 3.7, the boolean logic compares every parameter with the OR logic and also depends upon the case structure.

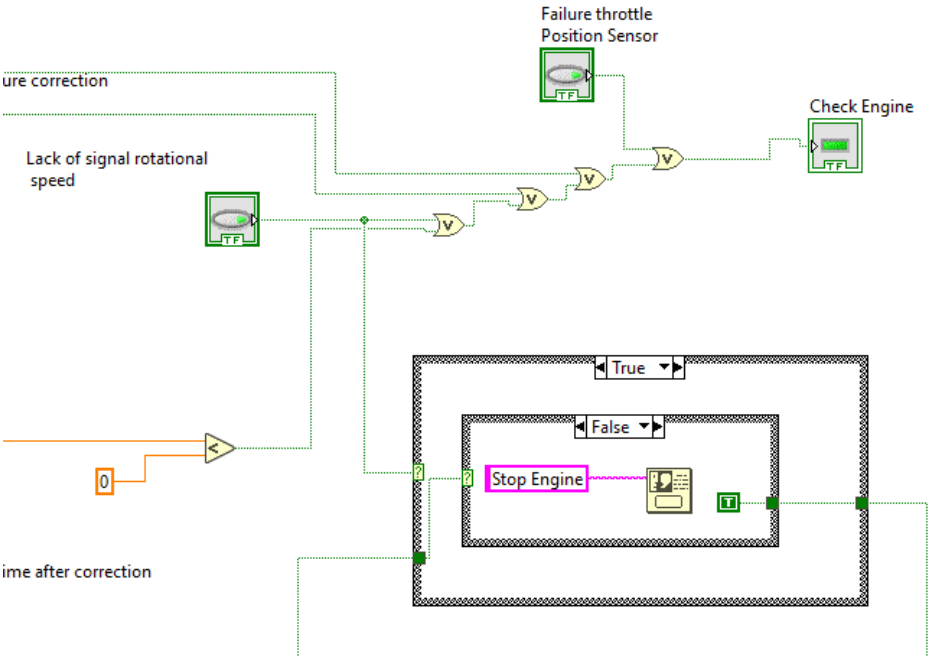

Figure 3.7: Concept of realizing the circuit of Check Engine control lighting

3.3.6 Integrated functional block of the ECU system

As in the Fig 3.8 shown below by combining all the block diagrams an integrated functional block diagram is designed. This block diagram leads to the output of the ECU system for sensor simulation. And when the system is ON the parameters will be controlled by the user which will circulate the signals through the entire mechanism. Starting from the Engine Rotational speed, as the speed increases the parameters will showcase the data on the monitoring and controlling panel of the output. As well as the data to these sensors are employed from the ECU memory. The ECU memory consisting of the data stored of the temperature,pressure in the suction manifold and speed of the vehicle.

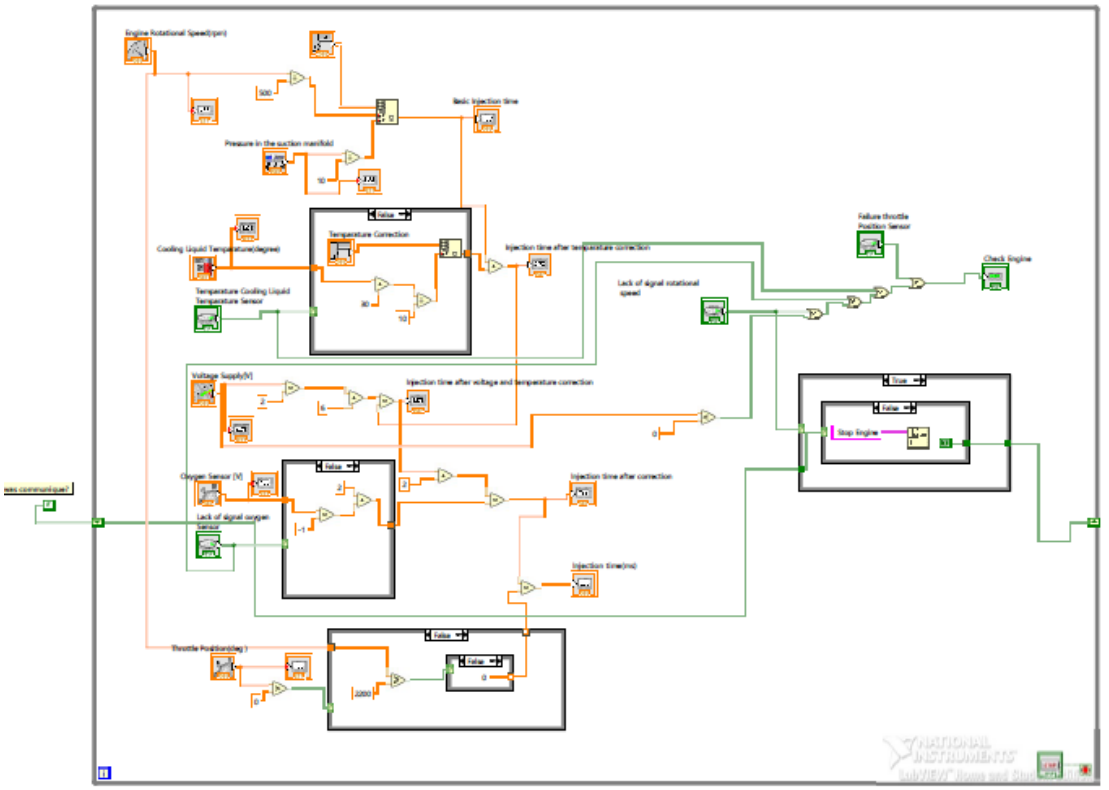

Figure 3.8: Integrated functional block

3.4 General Model of small scale SCADA Using LabVIEW and Microcontroller

The design of small-scale SCADA measurement system has been proposed.The use of LabVIEW and the Atmega16 microcontroller is used for the data acquisition (DAQ) system and is presented as a real time monitoring of and controlling of important mechanical and light parameters such as like motors rortating at certain speed and can be controlled at what speed the motor should run.The LED switching up when there is any high speed of the rotatory functions. The simulation is shown using the proteus8 software where the proteus is linked to the LabVIEW software. For the implementation of such design, the mechanical and electrical parameters are acquired in LabVIEW from the circuit containing software.

3.4.1 System Description

For development of the system, the LabVIEW software is used which will be monitoring the entire model also which is to be controlled by the user. And also we are using a 8-bit ATmega 16 micro-controller for the circuit which will help us in integrating the parameters driven. For the circuit simulation we have combined the circuit with the virtual ports which intern is set by the virtual serial port driver. This helps in transferring data from the proteus8 software to the labVIEW software for monitoring and controlling purpose. Thus,the controlled parameters are the motors as well as the LED's which is used for alarming condition. The virtual port driver will serially transmit the data to the VISA resource for data communication. So that as a small scale the user can get to know at what stage the targeted parameters are build for the load ends.And the controlling can happen vice-versa from LabVIEW as well as the proteus softwares so that it is represented as a DAC system.

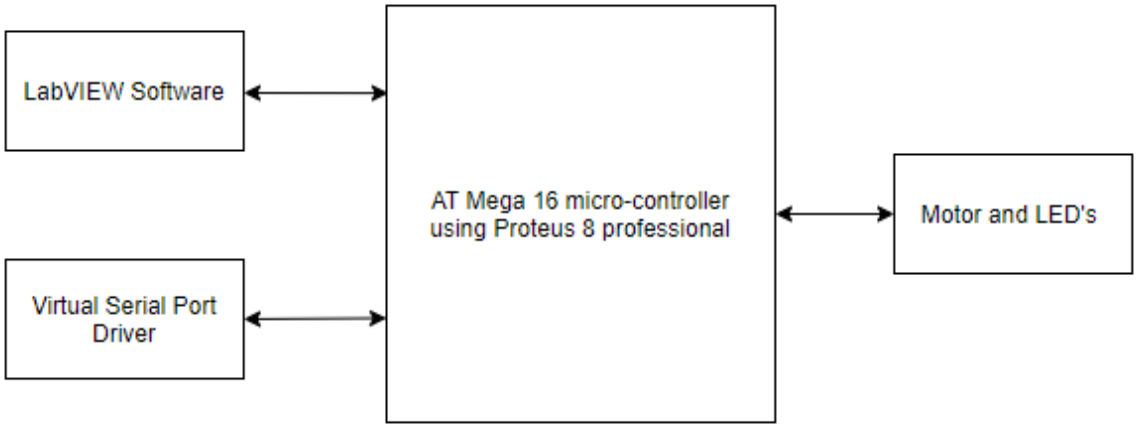

Figure 3.9: Block diagram of the system

The input flow of the data is transferred from the proteus8 software to the labVIEW where the data bits are in terms of bits. The data is collected at the Visa Resource name is then processed further which is shown in the Fig 3.9 .

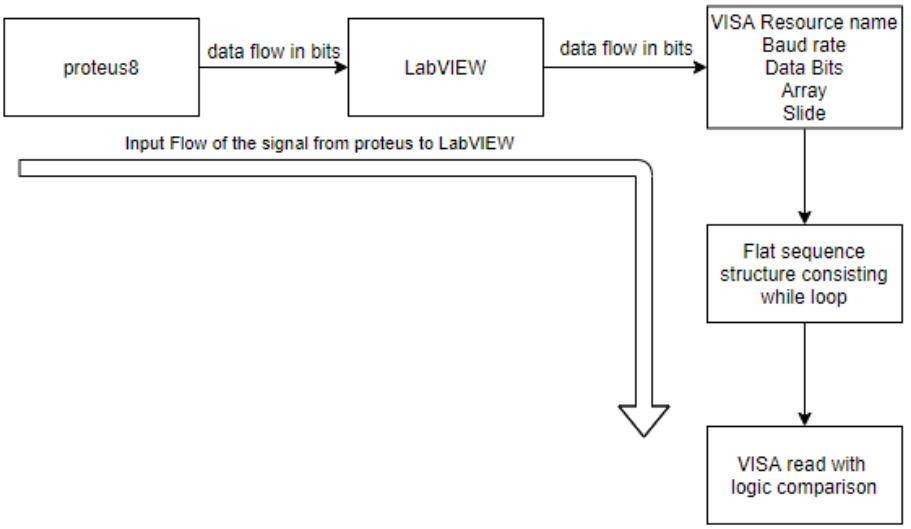

Figure 3.10: Input flow of the data

The output is shown on the front panel of the LabVIEW software as the data is analysed by the virtual configuration port and the logic is represented for the output to be classified. The block diagram of the output is shown in the Fig 3.10 .

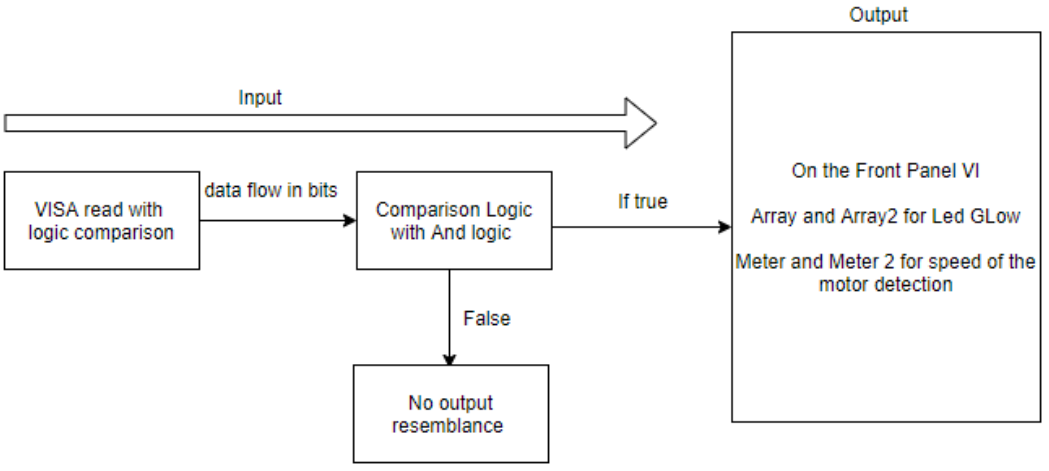

Figure 3.11: Output flow of the data

3.4.2 Algorithm to design the system

1. Design the block diagram in the LabVIEW with the certain loops generated.
2. Check for the errors by simulating the block diagrams.
3. Code the ATmega16 micro-controller in the BASCOM AVR-IDE software.
4. Baud rate of the labVIEW as well as AVR-IDE must be equal.
5. Dump the HEX file intothe microcontroller in the proteus8 professional software.
6. Design the circuit using various components and check for any error in the simulation.
7. Connect the LabVIEW and the proteus with virtual configuration serial port driver.
8. Check for the dual simulation and data transfer between both the softwares.

3.4.3 Functional block of the small scale SCADA system

The three tyre integration details that the the block diagram is represented using the mentioned algorithms and techniques. Having VISA source name for input/output interface,baud rate of 9600,data bits of 8 bits, array properties for the LEDS when alarm is generated and side for the meter speed of the rotatory motion.All these data are collected from the proteus or the microcontroller connected to the LabVIEW and then analyzed.The VISA write gets the data collected at certain point of communication and is designed with the help of string for the certain LED's.Then the data is flowed towards the Flat Sequence Structure which consists of a while loop. While loop in the Flat sequence Structure node is used which contains the bits propagation and finally to the output. Later the data is send through the data read and then the data check system is closed. The string subset consists of length comparing with each other and is released further to the and gate and later to the case structure where it consists of two meters and two tanks which shows the readings of the motor and the LEDs. The code is written as per the connection made and is dumped into the proteus8 simulation software.Using virtual port configuration software of the data flow is serially communicated to LabVIEW software.

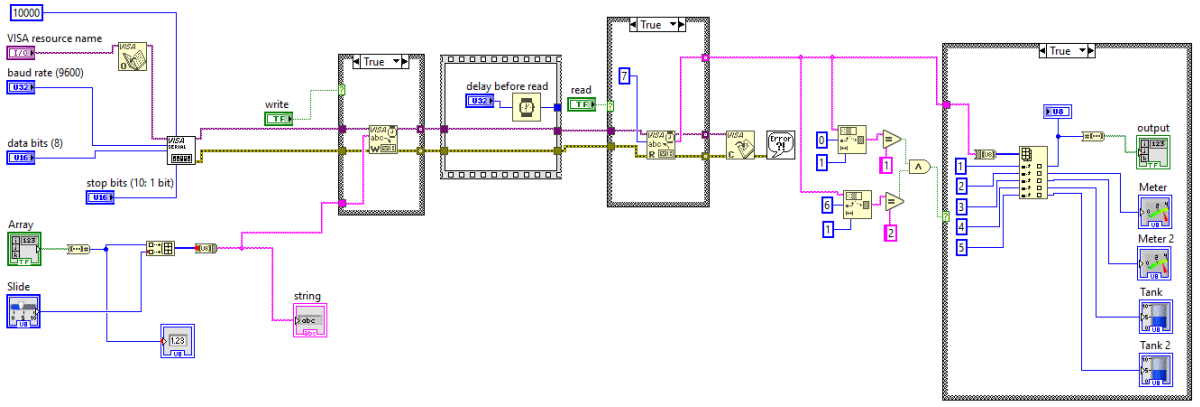

Figure 3.12: Integrated block diagram

3.4.4 Circuit Diagram in Proteus for Simulation

The circuit contains the motor connection with a diode which is regulated with respect to the connection made with the LabVIEW.The potentiometer controls the position of the LED's and at what excat time the LED should glow. This is controlled by the Atmega 16 microcontroller where the code is generated in the Hex file and dumped into it. The code determines the scalable structure as well as the setting of the parameters.The connection to the LabVIEW is made by the COMPIM configuration with the same connection properties with that of the LabVIEW. So with this circuit we can control the LabVIEW equipment directly with this circuit shown in the Fig.3.13.

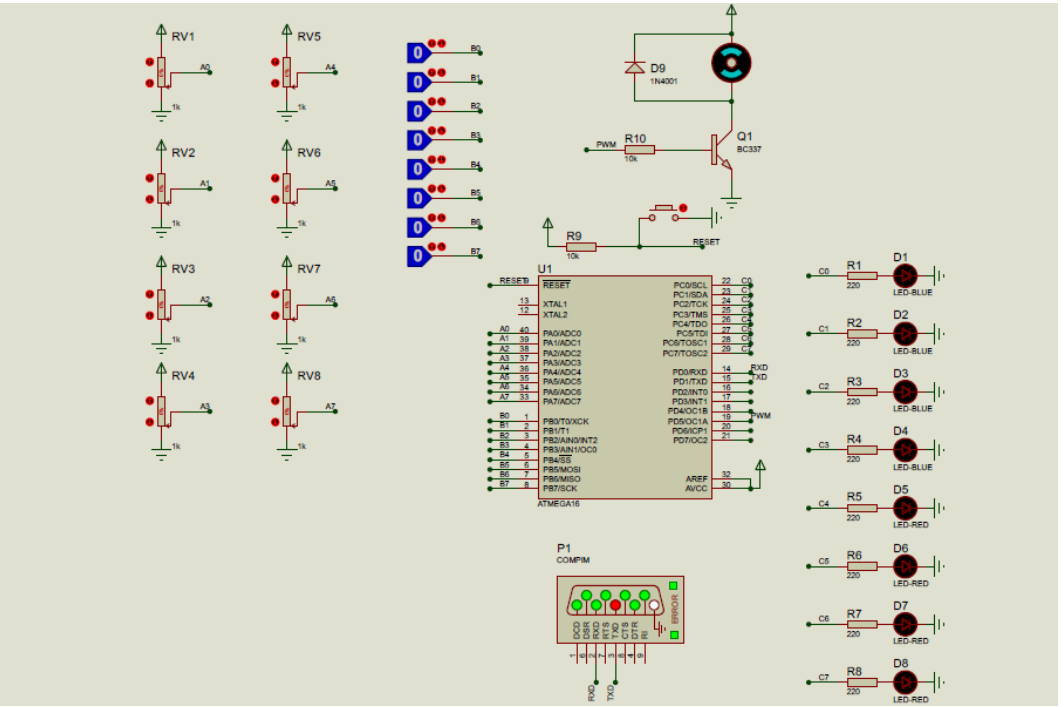

Figure 3.13: Circuit diagram of the system

Chapter 4

Optimization and Debugging Details

4.1 Optimization

In computing,the word optimization is the process of altering a system to make some applications of it work more efficiently or use less resources. For such instance, a computer program may be optimized so that it runs faster, or to run with less memory requirements or other resources. The optimization can have sense at different levels, from the lowest up to the highest levels of making of implementation, use or design of algorithms.The optimization technique is generalized to leave until the end of the process of development, since the premature optimization can introduce new errors. The Optimization techniques used for LabVIEW interface were differentiated in three categories and is as follows:
•No Change of Indicators, Array and constants without loops
•Model Without Loops
•Model With Loops

Thus we used the Model with loops and case structures optimization for our project.

No change of Indicators, Array and constants without loop	Model without loops and case structures	Model with loops and case structures
Can't be differentiated and can't be integrated	Can be differentiated and can be integrated	Can be differentiated and can be integrated
Flaws in output	Less flaws in output	Minute flaws or NIL flaws in output
Debugging is at high chance	Debugging is at medium chance	Debugging is at low chance
Logic and G programming errors	Can have logic and G programming errors	No logical and G programming errors

Figure 4.1: Optimization with categories

The optimization technique is also implemented for the assembly code of the Atmega-16 microcontroller and is as follows:
• Using Data types and sizes
• Global variables and local values
• Loop index
• Constants in program space
• Low level assembly instructions

4.2 Debugging Details

The major debugging stage in the G programming for the LabVIEW are:
•Logic generation and the logic about certain sensors which is demonstrated.
•The Boolean and the mathematical formulas are set as the algorithm.
•Determine the numeric constants.
•Symbols generalizing in the Graphical phase.
•Loop State for certain check position.

4.2.1 Logic Generation for ECU System

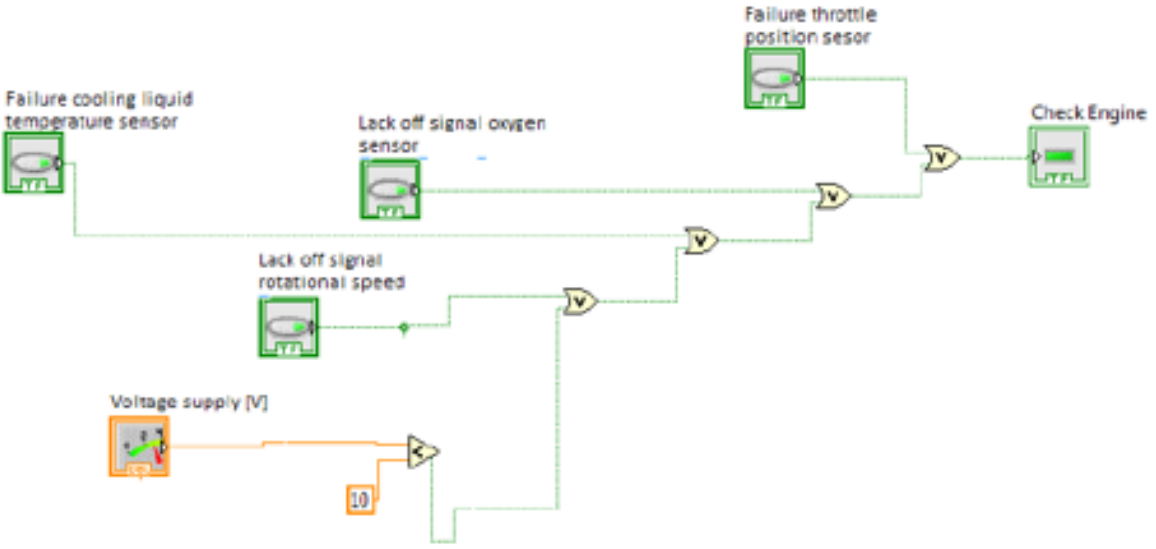

Figure 4.2: Logic Generation

A simple logic here is the use of OR gates which will compare the logic of the parameters and will display the alarm through which we will get to know when to drop the Engine for a hault.

4.2.2 Logic Generation for Small Scale SCADA application

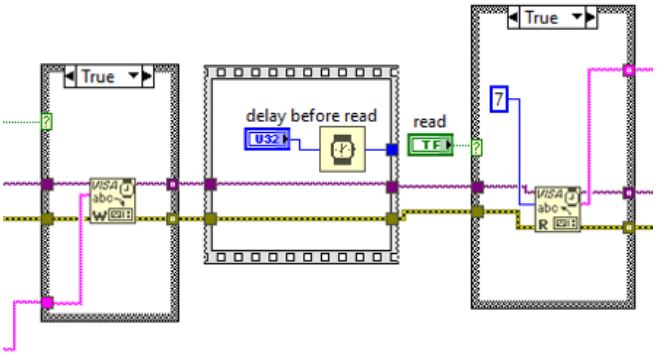

Figure 4.3: Logic Generation

The logic implemented here contain the set of case structures for the true and false functions that will generate the data flow sequence and will read the values from the COM port.

Chapter 5

Results

5.1 Control Panel of ECU

The output on the control panel of the LabVIEW designed shows controlling scheme of the Electronic Control Unit (ECU). This makes possible to present each and particular knob values which represent the sensors and makes an easy user control mechanism. Especially the ECU memory which is used for data formation and a proper GUI.

From this front panel we can control various sensors which get the certain information from the ECU memory and is shown in the Fig 5.1. The final output is shown in the output block of the system as:
•Basic Injection time(ms)
•Injection time after temparature correction(ms)
•Injection time after voltage and temperature correction(us)
•Injection time after correction (us)

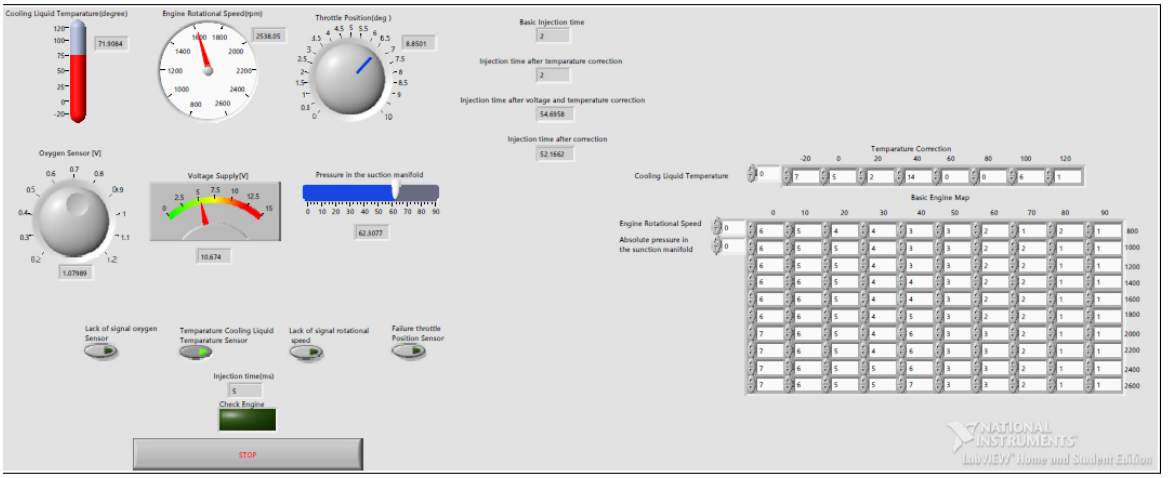

Figure 5.1: Output demonstration of an ECU control system using LabVIEW program

5.1.1 Surface Diagram of different parameters

Injection time

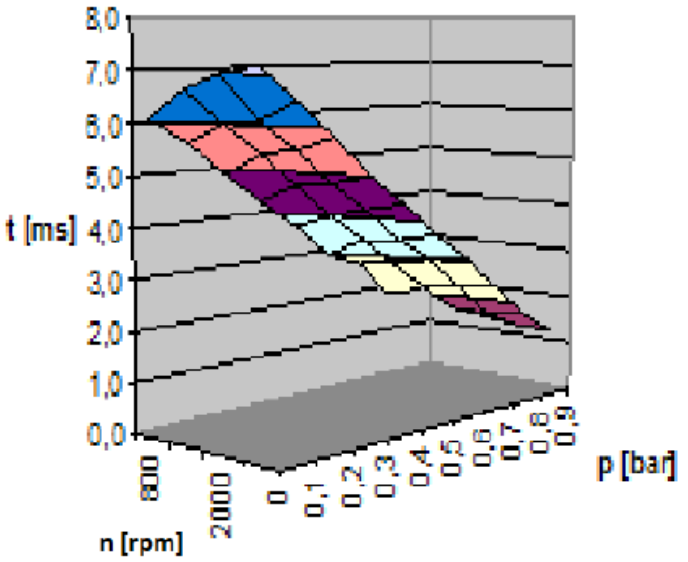

Figure 5.2: 3D graph representing the injection time(t),engine speed(rpm) and pressure(p) generated by an Engine

Change of fuel injection time

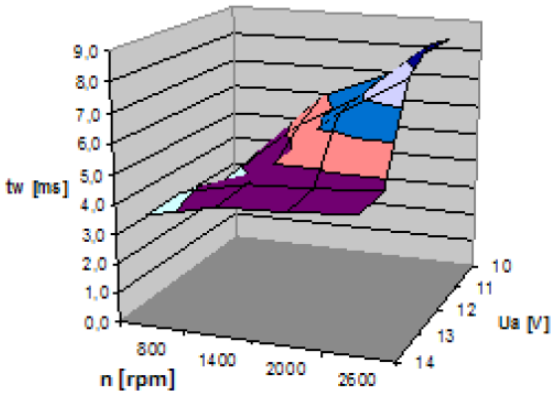

Figure 5.3: 3D graph representing the injection time(t),engine speed(rpm) and voltage(V) generated by an Engine

Fuel Injector time

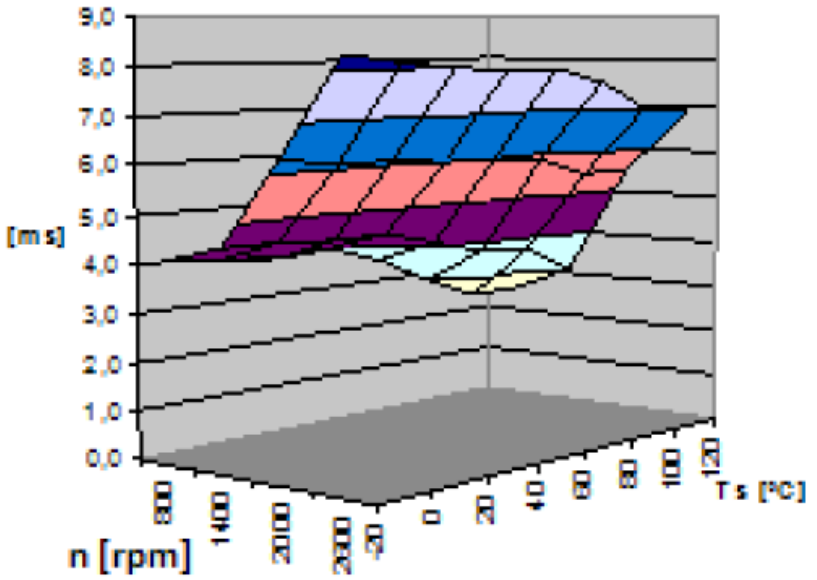

Figure 5.4: 3D graph representing the injection time(t),engine speed(rpm) and temperature(T) generated by an Engine

5.2 Small Scale SCADA application

The Fig 5.5 shows the final output of the interface with the proteus8 simulation. The array created in the LabVIEW can control the actuators like motors and LEDs. As when the LED switch is turned "on" in the LabVIEW it also shows the controlled parameters in the proteus8 too. Moreover the speed of the motor is controlled via the slide shown. This makes the remote access successful. The connection is maintained with data transfer from proteus8 software to the LabVIEW with respect to the serial port tool(virtual serial port). This generates the virtual COM port connection and maintains the link between two software so that the controlling mechanism can be done from both the software. And hence the speed of the motor is shown in the meter 1 which determines also with the tank parameters.This helps to expand the surface to control the system vice-versa.

As shown in Fig 5.6 the circuit is framed with respect to the parameters that are to be controlled with respect to the LabVIEW simulation. The code is dumped in Atmega-16 microcontroller using the HEX file which generates the register values and maintains the core of the platform so that the control gets executed.

Figure 5.5: Output shown in the Front Panel of the control system

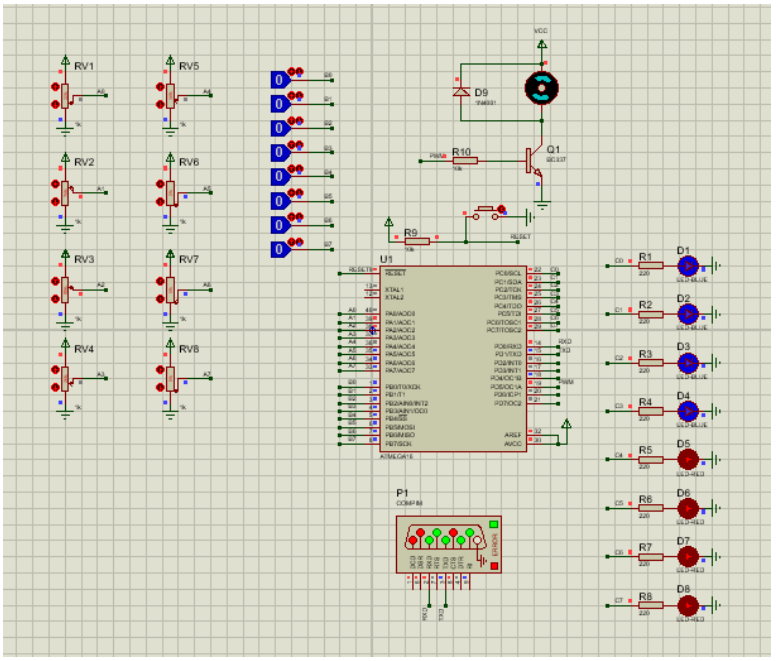

Figure 5.6: Simulation Output demonstrated in the proteus8

Chapter 6

Conclusion And Future Work

Using appropriate optimization techniques and debugging parameters,the virtual representation is demonstrated to simulate the operation for certain sensor failures of Electronic Control Unit. This makes it possible to notice signals entering control devices of modern engines with spark ignition. Moreover a small scale SCADA system was implemented uisng a ATmega16 microcontroller which controlled the system maintaining a vice-versa structure.

6.1 Advantages

1. The user can access the control panel of the remote system.
2. The algorithm generates the failure signals if there are any faults in the ECU system.
3. Continuous monitoring of sensors and actuators.

6.2 Applications

1. In most of the autonomous industries to control the substation situated in remote areas.
2. Designed ECU can generate alarms in vehicles.
3. Improving power system efficiency for the vehicles.
4. Active and reactive power control in power plants.
5. Trending and alarming to enable operators by addressing the problem spot.

6.3 Future Work

1. The SCADA or the ECU system can be interfaced with the technologies like IoT(Internet of Things),Machine Learning, Data framing(storage) and also cloud computing.
2. Data Security will play a role in securing data generated from the monitoring system.

Chapter 7

Bibliography

1. Shuangshuang Li, Baochen Jiang, Xiaoli Wang and Lubei Dong School of Mechanical, Electrical Information Engineering, Shandong University, Weihai 264209,Published: 31 March 2017
2. Igure, V.M., S.A. Laughter, and R.D. Williams, Security issues in SCADA networks. Computers and Security, 2006. 25(7): p. 498-506.
3. Linder, G. The importance of standard SCADA protocols to the reliable operation of distributed farm-scale anaerobic digesters. 2009. Seattle, WA, United states: IEEE Computer Society.
4. Mayur Avhad, Vinit Divekar, Harshad Golatkar, Sanket Joshi, "Microcontroller based automation system using industry standard SCADA", 2013 Annual IEEE India Conference (INDICON)
5. Adamiec M. and Dziubiński M. Alkaline fuel cell - aspect of efficiency. Przeglad Elektrotechniczny, 4, 2009.
6. Dziubiński M., Drozd A., Adamiec M. and Siemionek E. Electromagnetic interference in electrical systems of motor vehicles. Materials Science and Engineering. IOP Conference Series, 148, 2016.
7. Dziubiński M., Drozd A., Adamiec M. and Siemionek E. Energy balance in motor vehicles. Materials Science and Engineering. IOP Conference Series, 148, 2016.
8. Nouraei H., Ben-Mrad R. and Sinclair A. Development of a Piezoelectric Fuel Injector. IEEE Transactions on Vehicular Technology, 99, 2015, 1-8.
9. Boguta A. and Styła S. The graphic programming using simulation in the SI and CI engine management system. Autobusy. Technika, Eksploatacja, Systemy Transportowe, 2013, 3, 947-950.
10. Dziubiński M. Ecological aspect of electronic ignition and electronic injection system. Environment engineering V. CRC Press Taylor and Francis Group, 2016, 299-304.
11. Dziubiński M. Testing of exhaust emissions of vehicles combustion engines. Environment engineering V. CRC Press Taylor and Francis Group, 2016, 305-310.
12. Issermann R. Model-based fault-detection and diagnosis – status and applications. Annual Reviews in Control, 29(1), 2005, s.71-75.

13. Engineering Gandhi Institute of Engineering and Technology, India [7] P. Chou, G. Ortega, and R. Borriello, "Synthesis of the hardware/ software interface in microcontroller-based systems,"

14. G. Faraco and L. Gabriele, "Using LabVIEW for applying mathematical models in representing phenomena," Comput. Educ., vol. 49, no.3, pp. 856–872, 2007.

15. 2012 International Workshop on Information and Electronics Engineering (IWIEE) Research on the SCADA System Constructing Methodology Based on SOA Liang Wang*, Xiuting Wang School of Computer Science and Technology,Harbin institute of Technology(weihai),Weihai ,264209,China