

# Design of SCADA Systems for Medical & Industrial Applications using Arduino AtMega8 & MATLAB

Gajjala Ashok 1 M. Lakshmipathy 2 T Siva Kumar 3 K Karthik 4

*M.Tech(Embedded systems), Dept. of ECE, Kuppam Engineering College, kuppam, Chittoor(dst), A.P*<sup>1</sup>.

*Asst. Professor, Dept of ECE, Kuppam Engineering College, Kuppam, Chittoor(dst) .A.P*<sup>2</sup>.

*M.Tech(Embedded systems), Dept. of ECE, Kuppam Engineering College, Kuppam, Chittoor(dst) A.P*<sup>3,4</sup>.

## ABSTRACT

The main intention of this Research work is to design low cost supervisory control and data acquisition(SCADA) systems using Arduino AtMega8 and MATLAB.

This Research work will be implemented for two distinct areas that are Industrial and Medical. In industries this can be used for the controlling of Water tank Motor that means ON and OFF mechanism, various power section controlling, data acquisition from the sensors for example boiler temperature measurement etc. And in the Medical field which is used for pulse rate monitoring of patient and it warns if the pulse rate exceeds beyond the limit.

The design interpretation of SCADA Systems as follows,. First the sever programming is to port on to the microcontroller ( AtMega8). This program facilitates to access all pins from MATLAB. This project includes the design of GUI(Graphical User Interface) using Guide tool in MATLAB environment. The user communicates with the external hardware through MATLAB GUI. This can be achieved by interfacing MATLAB GUI with AtMega8 microcontroller. The microcontroller responds according to instructions given by the MATLAB GUI.

**Index Terms:** MATLAB, AtMega8, Sensors, Data Acquisition, Arduino.

## 1. INTRODUCTION

If you are going to be a physicist or engineer, you will have a lot to study: two hundred years of the most rapidly developing field of knowledge that there is. So much knowledge, in fact, that you might think that you cannot learn all of it in four years, and truly you cannot; you will have to go to graduate school too!

Surprisingly enough, in spite of the tremendous amount of work that has been done for all this time it is possible to condense the enormous mass of results to a large extent—that is, to find *laws* which summarize all our knowledge. Even so, the laws are so hard to grasp that it is unfair to you to start exploring this tremendous subject without some kind of map or outline of the relationship of one part of the subject of science to another. Each piece, or part, of the whole of nature is always merely an *approximation* to the complete truth, or the complete truth so far as we know it. In fact, everything we

know is only some kind of approximation, because *we know that we do not know all the laws* as yet. Therefore, things must be learned only to be unlearned again or, more likely, to be corrected.

The principle of science, the definition, almost, is the following: *The test of all knowledge is experiment.* Experiment is the *sole judge* of scientific "truth." But what is the source of knowledge? Where do the laws that are to be tested come from? Experiment, itself, helps to produce these laws, in the sense that it gives us hints. But also needed is *imagination* to create from these hints the great generalizations—to guess at the wonderful, simple, but very strange patterns beneath them all, and then to experiment to check again whether we have made the right guess. This imagining process is so difficult that there is a division of labor in physics: there are *theoretical* physicists who imagine, deduce, and guess at new laws, but do not experiment; and then there are *experimental* physicists who experiment, imagine, deduce, and guess [14]. Two fundamental questions,

1. What is a science ?.
2. What is an Engineering?.

Already science is defined above, An Engineering can be define *the purposeful use of science.* And the duty of engineer is to simplify the things. Today we come crass a lot of things are simplified, like advanced Android mobiles can be used by ordinary persons. That means the internal circuit of mobile is complex but to use it ,there is need not know about its internal structure, because those are user guided.

3. What is meant by SCADA?

In general SCADA systems are used to monitor and control equipment in industries such as telecommunications, water and waste control, energy, oil and gas refining, transportation and Medical. These systems encompass the transfer of data between a SCADA central host computer and a number of Remote Terminal Units (RTUs) and/or Programmable Logic Controllers (PLCs), and the central host and the operator terminals. A SCADA system gathers information (such as where a leak on a pipeline has occurred), transfers the information back to a central site, then alerts the home station that a leak has occurred, carrying out necessary analysis and control, such as determining if the leak is critical, and displaying the information in a logical and organized fashion.

These systems can be relatively simple, such as one that monitors environmental conditions of a small office building, or very complex, such as a system that monitors all the activity in a nuclear power plant or the activity of a municipal water system. Traditionally, SCADA systems have made use of the Public Switched Network (PSN) for monitoring purposes.

## 2. RELATED WORK

### 2.1 ARDUINO AND ITS PROGRAMMING METHODS

**Arduino** is a single-board microcontroller designed to make the process of using electronics in multidisciplinary projects more accessible. The hardware consists of a simple open source hardware board designed around an 8-bit Atmel AVR microcontroller, though a new model has been designed around a 32-bit Atmel ARM. The software consists of a standard programming language compiler and a boot loader that executes on the microcontroller.

Arduino boards can be purchased pre-assembled or do-it-yourself kits. Hardware design information is available for those who would like to assemble an Arduino by hand. There are sixteen official Arduinos that have been commercially produced to date. Numerous hardware variations of the Arduino are being sold by third parties [15].

#### 2.1.1 History

In 2005, in Ivrea, Italy, a project was initiated to make a device for controlling student-built interaction design projects with less expense than with other prototyping systems available at the time. Founders Massimo Banzi and David Cuartielles named the project after Arduin of Ivrea, the main historical character of the town, and began producing boards in a small factory located in the same region as the computer company Olivetti. The Arduino project is a fork of the open source Wiring platform and is programmed using a Wiring-based language (syntax and libraries), similar to C++ with some slight simplifications and modifications, and a Processing-based integrated development environment (IDE). Arduino was built around the Wiring project of Hernando Barragan. Wiring was Hernando's thesis project at the Interaction Design Institute Ivrea. It was intended to be an electronic version of Processing that used our programming environment and was patterned after the Processing syntax. It was supervised by myself and Massimo Banzi, an Arduino founder. I don't think Arduino would exist without Wiring and I don't think Wiring would exist without Processing. And I know Processing would certainly not exist without Design By Numbers and John Maeda.

#### 2.1.2 Software

The Arduino integrated development environment (IDE) is a cross-platform application written in Java, and is derived from the IDE for the Processing programming language and the Wiring projects. It is designed to introduce programming to artists and other newcomers unfamiliar with software

development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. There is typically no need to edit make files or run programs on a command-line interface.

Arduino programs are written in C or C++. The Arduino IDE comes with a software library called "Wiring" from the original Wiring project, which makes many common input/output operations much easier. Users only need define two functions to make a runnable cyclic executive program:

- `setup()`: a function run once at the start of a program that can initialize settings
- `loop()`: a function called repeatedly until the board powers off

A typical first program for a microcontroller simply blinks an LED on and off. In the Arduino environment, the user might write a program like this:

The integrated pin 13 LED

```
#define LED_PIN 13

void setup () {
  pinMode (LED_PIN, OUTPUT); // enable pin 13 for digital
  output
}

void loop () {
  digitalWrite (LED_PIN, HIGH); // turn on the LED
  delay (1000); // wait one second (1000 milliseconds)
  digitalWrite (LED_PIN, LOW); // turn off the LED
  delay (1000); // wait one second
}
```

#### 2.1.3 Pin IO Methods

- pinMode
- digitalRead
- digitalWrite
- analogRead
- analogWrite

i. **a.pinMode(pin, str);** specifies the pin mode of a digital pins.

The first argument before the function name, a, is the arduino object. The first argument, pin, is the number of the digital pin (2 to 19). The second argument, str, is a string that can be 'input' or 'output', Called with one argument, as a.pin(pin) it returns the mode of the digital pin, called without arguments, prints the mode of all the digital pins. Note that the digital pins from 0 to 13 are located on the upper right part of the board, while the digital pins from 14 to 19 are better known as "analog input" pins and are located in the lower right corner of the board.

Examples:

```

a.pinMode(11,'output') % sets digital pin #11 as output
a.pinMode(10,'input') % sets digital pin #10 as input
val=a.digitalRead(10); % returns the status of digital pin #10
a.pinMode(5);          % prints the status of digital pin #5
a.pinMode;            % prints the status of all pins
ii. val=a.digitalRead(4);          % reads pin #4
iii. a.digitalWrite(13,1);        % sets pin #13 high
     a.digitalWrite(13,0);        % sets pin #13 low
iv. val=a.analogRead(0);          % reads analog input pin # 0
v. a.analogWrite(11,90);          % sets pin #11 to 90/255
     a.analogWrite(3,10);         % sets pin #3 to 10/255

```

## 2.2 SENSOR NETWORKS(WN)

A sensor is often defined as a *device that receives and responds to a signal or stimulus*. This definition is broad. In fact, it is so broad that it covers almost everything from a human eye to a trigger in a pistol. The purpose of a sensor is to respond to some kind of an input physical property (stimulus) and to convert it into an electrical signal which is compatible with electronic circuits. We may say that a sensor is a translator of a generally nonelectrical value into an electrical value. When we say “electrical,” we mean a signal which can be channeled, amplified, and modified by electronic devices. The sensor’s output signal may be in the form of voltage, current, or charge. These may be further described in terms of amplitude, frequency, phase, or digital code. This set of characteristics is called the *output signal format*. Therefore, a sensor has input properties (of any kind) and electrical output properties[4][5].

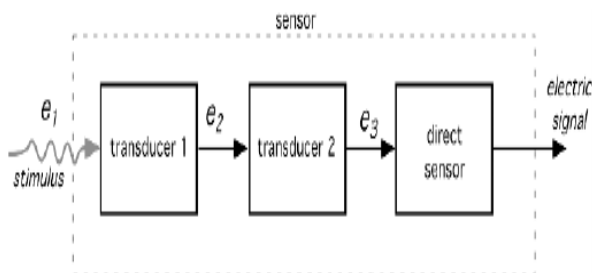


Figure 1: A sensor may incorporate several transducers.  $e_1$ ,  $e_2$ , and so on are various types of energy. Note that the last part is a direct sensor

### 2.2.1. Direct Digitization and Processing

Most sensors produce low-level signals. To bring these signals to levels compatible with data processing devices, amplifiers are generally required. Unfortunately, amplifiers and connecting cables and wires may introduce additional errors, add cost to the instrument, and increase complexity. Some emerging

trends in the sensor-based systems are causing use of the signal conditioning amplifiers to be reevaluated (at least for some transducers). In particular, many industrial sensor-fed systems are employing digital transmission and processing equipment. These trends point toward direct digitization of sensor outputs—a difficult task. It is especially true when a sensor-circuit integration on a single chip is considered.

Classical A/D conversion techniques emphasize high-level input ranges. This allows the LSB step size to be as large as possible, minimizing offset and noise error. For this reason, a minimum LSB signal is always selected to be at least 100–200  $\mu\text{V}$ . Therefore, a direct connection of many sensors (e.g., RTD temperature transducers or piezoresistive strain gauges) is unrealistic. Such transducers’ full-scale output may be limited by several millivolts, meaning that a 10-bit A/D converter must have about 1  $\mu\text{V}$  LSB.

In this work, all necessary components for to process the digitalized signal are integrated in a single silicon chip, called microcontroller(AVR Microcontroller ie.,AtMega8).This Advanced Virtual Risk Machine can handle complex tasks.

### 2.2.2 Semiconductor P-N Junction Sensors

Since prehistoric times people were aware of heat and tried to assess its intensity by measuring temperature. Perhaps the simplest and certainly the most widely used phenomenon for temperature sensing is thermal expansion. This forms the basis of the liquid-in-glass thermometers. For electrical transduction, different methods of sensing are employed. Among them are the resistive, thermoelectric, semi conductive, optical, acoustic, and piezoelectric detectors.

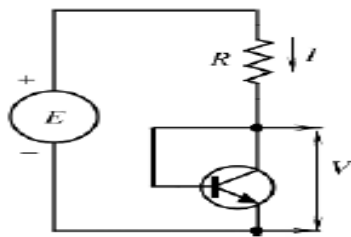
A semiconductor p-n junction in a diode and a bipolar transistor exhibits quite a strong thermal dependence. If the forward-biased junction is connected to a constant-current generator, the resulting voltage becomes a measure of the junction temperature. A very attractive feature of such a sensor is its high degree of linearity. This allows a simple method of calibration using just two points to define a slope (sensitivity) and an intercept. The current-to-voltage equation of a p-n junction diode can be expressed as

$$I = I_0 + \exp\left(\frac{qv}{2kT}\right)$$

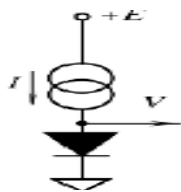
where  $I_0$  is the saturation current, which is a strong function of temperature. It can be shown that the

temperature-dependent voltage across the junction can be expressed as

$$V = \frac{E_g}{q} - \frac{2kT}{q} (\ln k - \ln I)$$



(a)



(b)

Figure 2: Voltage-to-temperature dependence of a forward-biased semiconductor junction under constant-current conditions.

### 3. DATA ACQUISITION

There are two types of sensors: *direct* and *complex*. A direct sensor converts a stimulus into an electrical signal or modifies an electrical signal by using an appropriate physical effect, whereas a complex sensor in addition needs one or more transducers of energy before a direct sensor can be employed to generate an Electrical output[6].

A sensor does not function by itself; it is always a part of a larger system that may incorporate many other detectors, signal conditioners, signal processors, memory devices, data recorders, and actuators. The sensor's place in a device is either intrinsic or extrinsic. It may be positioned at the input of a device to perceive the outside effects and to signal the system about variations in the outside stimuli. Also, it may be an internal part of a device that monitors the devices' own state to cause the appropriate performance. A sensor is always a part of some kind of a data acquisition system. Often, such a system may be a part of a larger control system that includes various feedback mechanisms[8] [9].

To illustrate the place of sensors in a larger system, Figure shows a block diagram of a data acquisition and

control device. An object can be anything: a car, space ship, animal or human, liquid, or gas. Any material object may become a subject of some kind of a measurement. Data are collected from an object by a number of sensors. Some of them (2, 3, and 4) are positioned directly on or inside the object. Sensor 1 perceives the object without a physical contact and, therefore, is called a *noncontact* sensor. Examples of such a sensor is a radiation detector and a TV camera. we say "noncontact", we remember that energy transfer always occurs between any sensor and an object.

Sensor 5 serves a different purpose. It monitors internal conditions of a data acquisition system itself. Some sensors (1 and 3) cannot be directly connected to standard electronic circuits because of inappropriate output signal formats. They require the use of interface devices (signal conditioners). Sensors 1, 2, 3, and 5 are passive. They generate electric signals without energy consumption from the electronic circuits. Sensor 4 is active. It requires an operating signal, which is provided by an excitation circuit. This signal is modified by the sensor in accordance with the converted information. An example of an active sensor is a thermistor, which is a temperature-sensitive resistor. It may operate with a constant-current source, which is an excitation circuit. Depending on the complexity of the system, the total number of sensors may vary from as little as one (a home thermostat) to many thousands (a space shuttle).

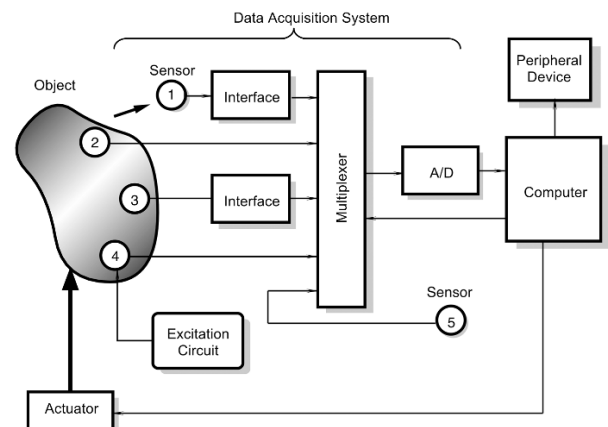


Figure 3: Positions of sensors in a data acquisition system. Sensor 1 is noncontact, sensors 2 and 3 are passive, sensor 4 is active, and sensor 5 is internal to a data acquisition system.

Electrical signals from the sensors are fed into a multiplexer (MUX), which is a switch or a gate. Its function is to connect sensors one at a time to an analog-to-digital (A/D) converter if a sensor produces an analog signal, or directly to a computer if a sensor produces signals in a digital format. The computer



controls a multiplexer and an A/D converter for the appropriate timing. Also, it may send control signals to the actuator, which acts on the object. Examples of actuators are an electric motor, a solenoid, a relay, and a pneumatic valve. The system contains some peripheral devices (for instance, a data recorder, a display, an alarm, etc.) and a number of components, which are not shown in the block diagram. These may be filters, sample-and-hold circuits, amplifiers, and so forth[10] [11].

#### 4. MATLAB GUI (Graphical User Interface) DESIGN ASPECTS

*The Language of Technical Computing:*

MATLAB® is a high-level language and interactive environment for numerical computation, visualization, and programming. Using MATLAB, you can analyze data, develop algorithms, and create models and applications. The language, tools, and built-in math functions enable you to explore multiple approaches and reach a solution faster than with spreadsheets or traditional programming languages, such as C/C++ or Java®. You can use MATLAB for a range of applications, including signal processing and communications, image and video processing, control systems, test and measurement, computational finance, and computational biology. More than a million engineers and scientists in industry and academia use MATLAB, the language of technical computing[2].

*Key Features:*

- ✓ High-level language for numerical computation, visualization, and application development.
- ✓ Interactive environment for iterative exploration, design, and problem solving.
- ✓ Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration, and solving ordinary differential equations.
- ✓ Built-in graphics for visualizing data and tools for creating custom plots.
- ✓ Development tools for improving code quality and maintain ability and maximizing performance.

- ✓ Tools for building applications with custom graphical interfaces.
- ✓ Functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET, and Microsoft® Excel®.

#### 4.1 MATLAB GUI

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components, that enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the task sare performed.

GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders—just to name a few. GUIs created using MATLAB® tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots.

#### 5. INSIGHTS INTO PRACTICAL DESIGN ASPECTS, WORKING STRATEGIES AND RESULTS

##### 5.1 GUI (Graphical User Interfacing) Design

The developed GUI as shown below, it has the controlling and data acquisition.

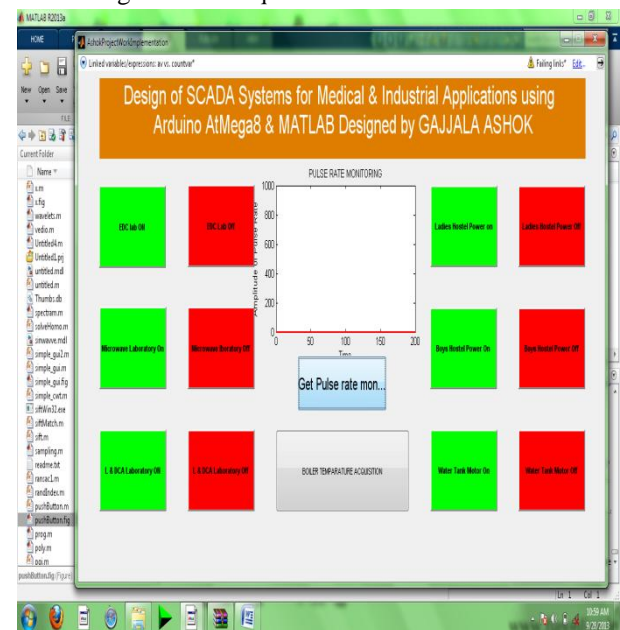


Figure 4: Developed Graphical User Interfacing

### 5.2 Arduino I/O Program Porting

Upload srv.pde (or adiosrv.pde) to the arduino board (to be done only once).The srv.pde (or adiosrv.pde) is the "server" program that will continuously run on the microcontroller. It listens for MATLAB commands arriving from the serial port, executes the commands, and, if needed, returns a result [15].

### 5.3 Hardware Design Architecture

The following two figures shows the Block diagrammatic representation and Technological representation respectively.

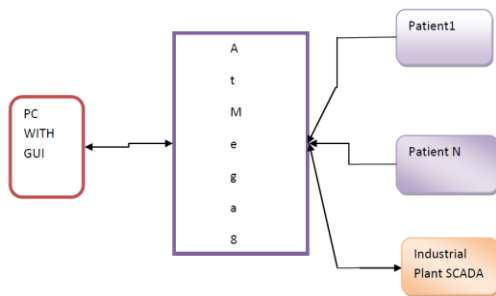


Figure 5: Block Diagram of the designed system.

all pins from MATLAB. This project includes the design of GUI(Graphical User Interface) using Guide tool in MATLAB environment. The user communicates with the external hardware through MATLAB GUI. This can be achieved by interfacing MATLAB GUI with AtMega8 microcontroller. The microcontroller responds according to instructions given by the MATLAB GUI. In industries this can be used for the controlling of Water tank Motor that means ON and OFF mechanism, various power section controlling, data acquisition from the sensors for example boiler temperature measurement etc. And in the Medical field which is used for pulse rate monitoring of patient and it warns if the pulse rate exceeds beyond the limit.

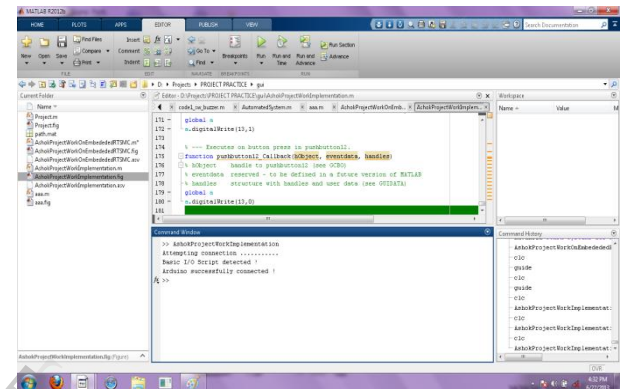


Figure 7: Arduino Connection to Matlab

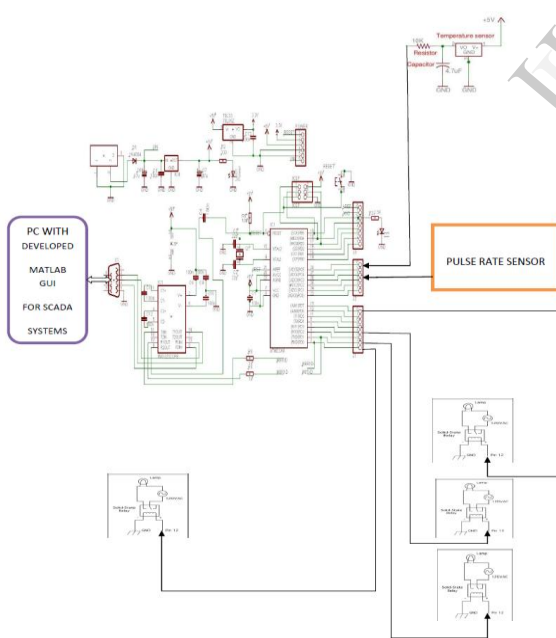


Figure 6: Technological Schematic

### 5.4 Working mechanism and Result

The design interpretation of SCADA Systems as follows., First the sever programming is to port on to the microcontroller ( AtMega8).This program facilitates to access

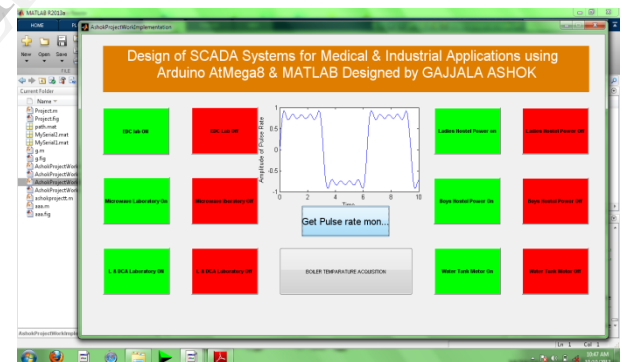


Figure 8: Data acquisition and controlling window

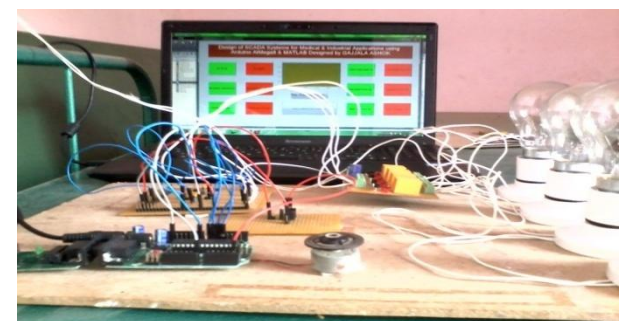


Figure 9: Snapshot of designed hardware work.

## 6. LITERATURE SURVEY

In 1999 Robert DeMoyer and E. Eugene proposed New with MATLAB 5 is the Graphical User Interface Development Environment, or GUIDE. This tool enables the engineering educator to produce pedagogical software which combines the computing and graphics capability of MATLAB with convenient Graphical User Interfaces. A brief overview of handle graphics is provided, followed by the introduction of GUIDE. GUIDE is used alone to produce an elementary example, and is used in combination with other files in order to produce a more interesting example[2].

In the year 2010 R. Arulmozhiyal, K. Baskaran, N. Devarajan and J. Kanagaraj proposed the design and implementation of Real time MATLAB Interface for speed control of induction motor drive using dsPIC 30F4011. In recent years, the field oriented control of induction motor drive is widely used in high performance drive system. It is due to its unique characteristics like high efficiency, good power factor and extremely rugged. This scheme leads to be able to adjust the speed of the motor by control the frequency and amplitude of the stator voltage, the ratio of stator voltage to frequency should be kept constant[3].

In the year 2012 Vlad Bande, Septimiu Pop, Ioan Ciascai, Dan Pitica proposed one of the most important procedures which must be done in the dam's monitoring process, from the dam behavior point of view, is the analysis of the data acquired from the sensors mounted inside or in the proximity of the dam. The analysis can be done in two ways: *real-time* - using a direct dedicated interfacing algorithm with one or more sensors, respectively a so called *historical analysis* - using large data packets acquired during long time periods. The first acquisition procedure shows the behavior of a sensor at an exact moment, while the second reveals - in the end - the dam behavior and suggests the proper actions that must be taken to secure the dam infrastructure [1].

## 7. CONCLUSIONS

This paper is a mathematical and practical approach in the medical and industrial domain. This research is employed with AVR microcontroller works with efficiency and low power consumption. It replaces high cost equipments in medical and industrial fields to

controlling and acquire of data. This Research work is applicable for any field of science and technology.

## ACKNOWLEDGMENT

I would like to acknowledge the following for help, encouragement, and support during the preparation of this research work. First, I thank God for giving me the endurance and perseverance to complete this work. I could not have completed this work without the continuous support of my academic guide Mr.M.Lakshmipathy and my Friend T Sivakumar, K Karthik . The support and encouragement of all my professors and friends at Kuppam Engineering College are appreciated. Special thanks to Prof. Kodhanda Ramaiah, HOD of ECE ,one who provided excellent research facilities at college laboratories to complete this work.

## 6. REFERENCES

- [1] Vlad Bande, Septimiu Pop, Ioan Ciascai, Dan Pitica," Real-time Sensor Acquisition Interfacing using MatLAB"2012 IEEE 18th International Symposium for Design and Technology in Electronic Packaging (SIITME) 978-1-4673-4760-0.
- [2] A. Siciliano,"MatLAB" – Data Analysis and Visualization" -World Scientific Publishing, Singapore, 2008. ISBN: 978-981-283-554-3.
- [3] S.E.Lyshevski,"Engineering and Scientific Computations Using MATLAB", John Wiley & Sons, Inc, United States of America, 2003. ISBN: 0-471-46200-4.
- [4] V.K. Ingle, J.G. Proakis, "Digital Signal Processing using MATLAB", Cengage Learning, Stamford, United States of America, 2010. ISBN: 1-111-42737-2.
- [5] Thompson, S. *Control Systems: Engineering & Design*. Longman Scientific & Technical, Essex, UK, 1989.
- [6] Norton, H. N. *Handbook of Transducers*. Prentice Hall, Englewood Cliffs, NJ, 1989.
- [7] Benedict, R. P. *Fundamentals of Temperature, Pressure, and Flow Measurements*, 3rd ed. JohnWiley & Sons, New York, 1984.
- [8] Callendar, H. L. On the practical measurement of temperature. *Phil. Trans. R. Soc. London* 178, 160, 1887.

[9] Sapoff, M. Thermistor thermometers. In: *The Measurement, Instrumentation and Sensors Handbook*. J.G. Webster, ed., CRC Press, Boca Raton, FL, 1999, pp. 32.25–32.41.

[10] Fraden, J. Two-point calibration of negative temperature coefficient thermistors. *Rev. Sci. Instrum.* 71(4), 1901–1905, 2000.

[11] Steinhart, J.S. and Hart, S.R. *Deep Sea Res.*, 15, 497, 1968.

[12] Mangum, B.W. *Rev. Sci. Instrum.* 54(12), 1687, 1983.

[13] Sapoff, M., Siwek, W.R., Johnson, H.C., Slepian, J., and Weber, S. In: *Temperature. Its Measurement and Control in Science and Industry*. J.E. Schooley, ed. American Institute of Physics, New York, 1982, Vol. 5, p. 875.

[14] Fynman" Fynman Lectures on Physics", MIT Compous 1963.

[15] <http://www.arduino.com>

#### Authors Profile:



He completed B.Tech in the stream of Electrical & Electronics Engineering from "Srikalahasteeswara institute of Technology "srikalahasti campus under the Endowment Department of A.P Govt, Affiliated to JNTUA, in the year 2011. He is persuing M.Tech in "EMBEDDED SYSTEMS" in Kuppam Engineering College from JNTUA. His Research interests in Real Time Operating Systems for handling complex operations, Experts Systems, Artificial neural networks ,Graphine based computer chip design and EHVAC,HVDC, Power System Engineering. Follow this person on the website <http://www.indianmagictchnology.blogspot.com>.



He completed B.Tech in E.C.E from Sree Vidhyanikethan College of Engineering, A Rangampet, chittoor(District), Andhrapradesh. He received M.Tech( Electronic Design & Technology) from National Institute of Technology, Calicut. He worked as a Assistant Professor in Dept. of ECE, Gokula Krishna College of Engineering, Sullurpeta ,Nellore (District) from 2007 to 2009. Currently he is working as a Assistant Professor in Dept. of ECE, Kuppam Engineering College. His areas of interests are Embedded Systems and Analog Circuit Design.



He completed B.E in ECE from "Sri Nandhanam College of Engineering and Technology", "ANNA University" from Tirupatur campus in the year 2010. He is persuing M.Tech (EMBEDDED SYSTEM) from Kuppam Engineering College, "JNTU" from Kuppam campus. His Research interests in real time system, control system, networks and digital communication.



He completed B.Tech in ECE from "Kuppam Engineering college",kuppam campus from "JNTU University" in the year 2008 and he received M.B.A (HRM) form Annamalai University in the 2011. He is persuing M.Tech (EMBEDDED SYSTEMS) from Kuppam Engineering College, "JNTU" from Kuppam campus. His Research interests in real time system, control system, networks and digital communication.