Open Source, Real-Time Temperature Monitoring & Control using Scilab & Arduino

A PROJECT REPORT

Submitted by

Mr. AVINASH SHETTY
Miss. FARHEEN ANSARI
Mr. MOIZ FADNIS
Mr. SHOEB QURAISHI

In partial fulfillment for the award of the degree B.E EXTC

Under The Guidance Of

Mrs. CHAYA S

Anjuman-I-Islam's
Kalsekar Technical Campus

At

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING

ANJUMAN-I-ISLAM‘S KALSEKAR TECHNICAL CAMPUS

PANVEL

MAY2016

DECLARATION

We hereby declare that the project entitled **“Open Source Real-Time Temperature Monitoring & Control using Scilab & Arduino”** submitted for the B.E Degree is our original work and the project has not formed the basis for the award of any degree, associate ship, fellowship or any other similar titles.

Signature of the Students:

Mr. Avinash Shetty ______________________

Miss. FarheenAnsari ______________________

Mr. Moiz Fadnis ______________________

Mr. Shoeb Quraishi ______________________

Place:

Date:

CERTIFICATE

This is to certify that the project entitled **"Open Source, Real-Time Temperature Monitoring & Control using Scilab & Arduino"** is the bonafide work carried out by the above mentioned students of B.E, KALSEKAR Technical Campus, Panvel, during the year 2015-2016, in partial fulfillment of the requirements for the award of the Degree of B.E EXTC and that the project has not formed the basis for the award previously of any degree, diploma, associate ship, fellowship or any other similar title.

(Prof.Mujib Tamboli)	(Prof. Chaya S)
H.O.D	Asst.Prof.

(External)

ACKNOWLEDGEMENT

The completion of any project brings with it a sense of satisfaction, but it is never complete without thanking those people who made it possible and whose constant support has crowned our efforts with success. One cannot even imagine the power of the force that guides us all and neither can we succeed without acknowledging it. Our deepest gratitude to Almighty God for holding our hands and guiding us throughout our lives.

We owe a deep sense of gratitude to our project guide **Mrs. Chaya S**, Dept. Electronics Engineering for her expert guidance, encouragement, valuable suggestions at every step and for stimulating our interest in the subject. Her involvement right from conceptualization of the study through its execution and meticulous perusal of the manuscript ensured its timely completion.

We would also like to express our gratitude and thank **Mr. Mujib Tamboli (**Head of the Department), Electronics Engineering for encouraging and inspiring us to carry out the project in the department lab and for his thought provoking comments, valuable suggestions constant motivation, encouragement and support.

We are extremely happy to acknowledge and express our sincere gratitude and heartfelt thanks to our parents and family members for their constant support and encouragement and last but not the least, to our friends and well-wishers for their help and cooperation and solutions to problems during the course of the project.

Also our friends at Engineers Academy who provided solutions at times when we were against the wall in need of help.

ABSTRACT

Our project is a working model which incorporates temperature sensors to measure temperature. "Temperature Monitoring System", is a system that can be used in an industrial like factory or inside the main distribution frame room for industries. Here we are using two open source platforms Arduino and Scilab for temperature monitoring. The Arduino and Scilab communicated with the help of serial port connection "USB-UART". The main issues in a temperature monitoring system are how the system communicates with the user, if the temperature is beyond the stable temperature defined by the user. It is a system that is applied to detect temperature and display the value of temperature on the monitor, as well as it will be displayed on the LDC screen. The graph of the temperature variation with respect to time will be displayed on the monitor, which will give the value of temperature every second. While the monitoring system uses a computer system to monitoring the temperature data.

Table of Contents

Title
Page……………………………….……………1

Declaration……………………….…….….2

Certificate…………….................................3

Acknowledgement...……………………...4

Abstract………………………………….....6

1. INTRODUCTION --8
   i. BLOCK DIAGRAM --9
   ii. SOFTWARE ENVIRONMENT --10
   i. ARDUINO --12
   ii. SCILAB ARDUINO TOOLBOX ---14

2. GRAPHICAL USER INTERFACE (GUI) ---16
   i. SCILAB AS A SUBSTITUTE OF MATLAB --18
   ii. ARDUINO & SCILAB FIRMWARE ---20

3. DEVELOPING FRONT-END USING GUI --21

4. HARDWARE ---23

   i. LM35---24

   ii. LCD (LIQUID CRYSTAL DISPLAY)---27

   iii. POTENTIOMETER---31

5. WORKING --32

6. APPLICATION, CONCLUSION --36

- APPENDIX

  i. FUTURE SCOPE---38

  ii. REFERENCE--39

List of Figures

Fig 1.1: Proposed block diagram --9
Fig 1.2: Arduino UNO board---12
Fig 1.3: GUI sample--16
Fig 1.4: Radio buttons in MATLAB--18
Fig 1.5: Check box in scilab--19
Fig 1.6: Front-End for monitoring & control purpose---------------------------------------21
Fig 1.7: LM35---24
Fig 1.8: LCD display---27
Fig 1.9: LCD module interface---30
Fig 2.0: potentiometer--31
Fig 2.1: GUI front-end---32
Fig 2.2: Graph on monitor --33
Fig 2.3: Pin connection for LCD with arduino--34
Fig 2.4: Message displayed on LCD---35

LIST OF TABLES

Table 2.1: Arduino Uno hardware specifications------------------------------------13

Table 2.2: Pin description of LM35---25

CHAPTER – 1

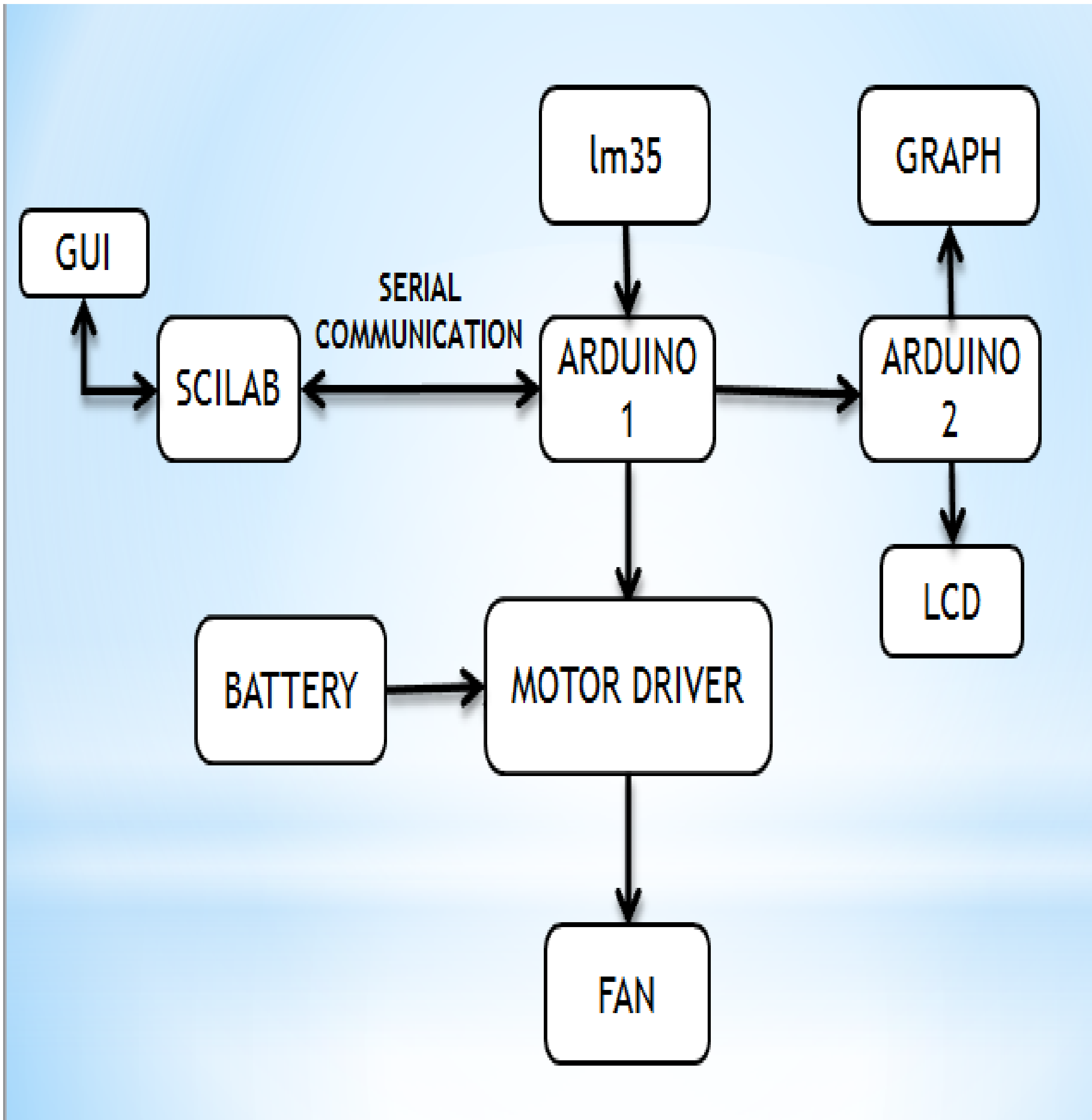
INTRODUCTION

Monitoring & Controlling temperature has been a prime objective in various applications including refrigerators, air conditioners, air coolers, heaters, industrial temperature conditioning and so on. Temperature controllers vary in their complexities and algorithms. Some of these use simple monitoring and control techniques like simple on-off control while others use complex Proportional Integral Derivative (PID) or fuzzy logic algorithms.

Our project is a working model which incorporates temperature sensors to measure temperature. “Temperature Monitoring & Control System”, is a system that can be used in an industrial like factory or inside the main distribution frame room for industries. Here we are using two open source platforms Arduino and Scilab for temperature monitoring. The Arduino and Scilab communicated with the help of serial port connection “USB-UART”

The main issues in a temperature monitoring system are how the system communicates with the user, if the temperature is beyond the stable temperature defined by the user. It is a system that is applied to detect temperature and display the value of temperature on the monitor, as well as it will be displayed on the LDC screen. The graph of the temperature variation with respect to time will be displayed on the monitor, which will give the value of temperature every second. While the monitoring system uses a computer system to monitoring the temperature data.

i- BLOCK DIAGRAM



Scilab:

Scilab is an open source cross platform numerical computational package and a high level, numerically oriented programming language. It can be used for signal processing, statistical analysis, image enhancement, fluid dynamic simulations, numerical optimization, modeling, simulation of explicit and implicit dynamical systems and symbolic manipulations (if the corresponding toolbox is installed).

Scilab is one of the two major open-source alternatives to MATLAB. Scilab is similar enough to MATLAB that some book authors argue that it is easy to transfer skills between the two systems. Scilab how ever puts less emphasis on syntactic compatibility with MATLAB than octave does.

Scilab is a high-level, numerically oriented programming language. The language provides an interpreted environment, with matrices as the main data type. By using matrix based computation, dynamic typing, and automatic memory management, many numerical problems may be expressed in a reduced number of code lines, as compared to similar solutions using traditional languages, such as Fortan, C, C++.This allows users to rapidly construct models for a range of mathematical problems. While the language provides a library of high-level operations such as correlation and complex multidimensional arithmetic.

The software can be used for signal processing, statistical analysis, image enhancement, fluid dynamics simulations, and numerical optimization.

As the syntax of scilab is similar to MATLAB, scilab includes a source code translator for assisting the conversion of code from MATLAB to scilab. Scilab is available free of cost under an open source license. Due to the open source nature of the software, some user contributions have been integrated into the main program.

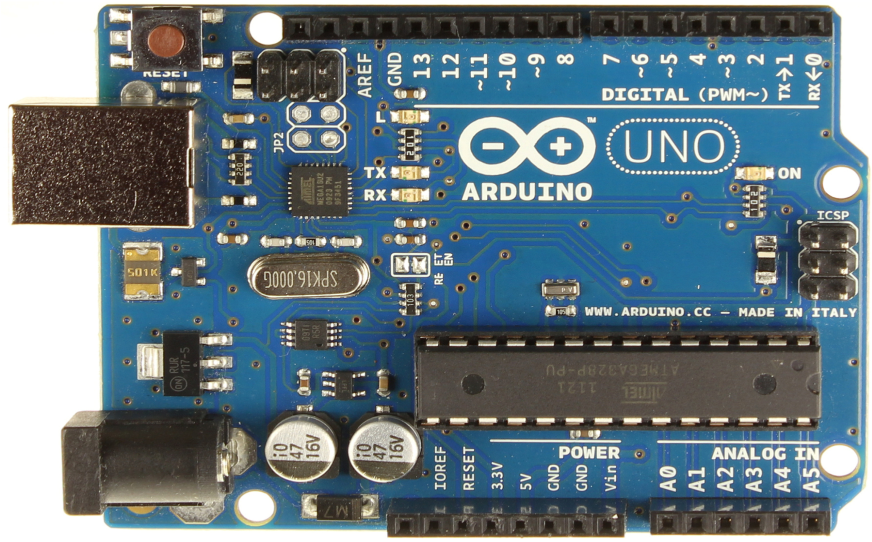


Fig. 1.2: Arduino UNO board

Arduino is an open source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs – light on a sensor and turn it into an output – activating a motor, turning on a LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board.

The Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

ARDUINO UNO SPECIFICATIONS:

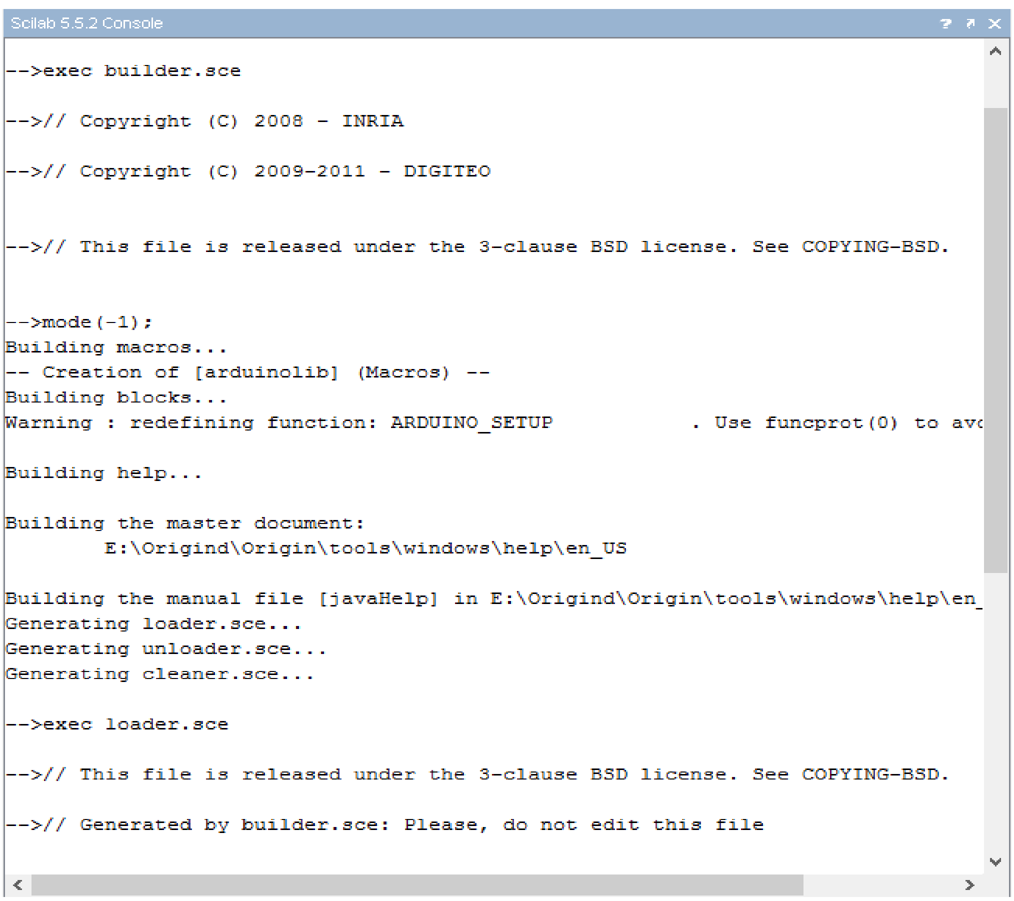
parameter	value
Microcontroller	2
Operating voltage	5V
Input voltage(recommended)	7-12V
Digital I/O pins	14(of which 6 provide PWM o/p)
Analog I/O pins	6
DC current per I/O pin	40mA
DC current for 3.3V pin	50mA
Flash memory	32 KB (ATmega328)
SRAM	2KB (ATmega328)
EEPROM	1KB (ATmega328)
Clock speed	16 Mhz
Length	68.6mm
Width	53.4mm
Weight	25g

Table 2.1: Arduino Uno hardware specifications

iv- SCILAB ARDUINO TOOLBOX

Scilab and arduino cannot be connected with each other by default. This connection is possible only with the help of scilab arduino toolbox. We have different toolboxes for different operating systems. We are using scilab arduino toolbox according to our need. This tool box helps in serial communication of scilab and arduino. We can make scilab arduino toolbox by normal scilab coding also, but scilab coding for toolbox is really bulky and time taking process. For the same purpose we have GUI buider.

Here is the code for GUI builder which will be helpful in building of scilab arduino toolbox.




```
-->exec builder.sce

-->// Copyright (C) 2008 - INRIA

-->// Copyright (C) 2009-2011 - DIGITEO


-->// This file is released under the 3-clause BSD license. See COPYING-BSD.


-->mode(-1);
Building macros...
-- Creation of [arduinolib] (Macros) --
Building blocks...
Warning : redefining function: ARDUINO_SETUP           . Use funcprot(0) to avo

Building help...

Building the master document:
        E:\Origind\Origin\tools\windows\help\en_US

Building the manual file [javaHelp] in E:\Origind\Origin\tools\windows\help\en_
Generating loader.sce...
Generating unloader.sce...
Generating cleaner.sce...

-->exec loader.sce

-->// This file is released under the 3-clause BSD license. See COPYING-BSD.

-->// Generated by builder.sce: Please, do not edit this file
```

```
-->exec(get_absolute_file_path("loader.sce")+"etc\"+"arduino.start");
Start Arduino
        Load macros
        Load serial dll
Shared archive loaded.
Link done.
        Load palette
        Load help
        Load demos
        Arduino Version: 1.2

-->exec('E:\Origind\Origin\tools\windows\guibuilder\builder.sce', -1)
Building macros...
-- Creation of [guibuilderlib] (Macros) --
Generating loader.sce...
Generating unloader.sce...
Generating cleaner.sce...

-->exec('E:\Origind\Origin\tools\windows\guibuilder\loader.sce', -1)
START GUI Builder 3.0 for Scilab 5.5
        Load macros
        Load help
        Type "guibuilder" to start the GUI
```

CHAPTER-2

GRAPHICAL USER INTERFACE (GUI)

In computer science, a graphical user interface or GUI is a type of interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, as opposed to text-based interfaces, typed command labels or text navigation.

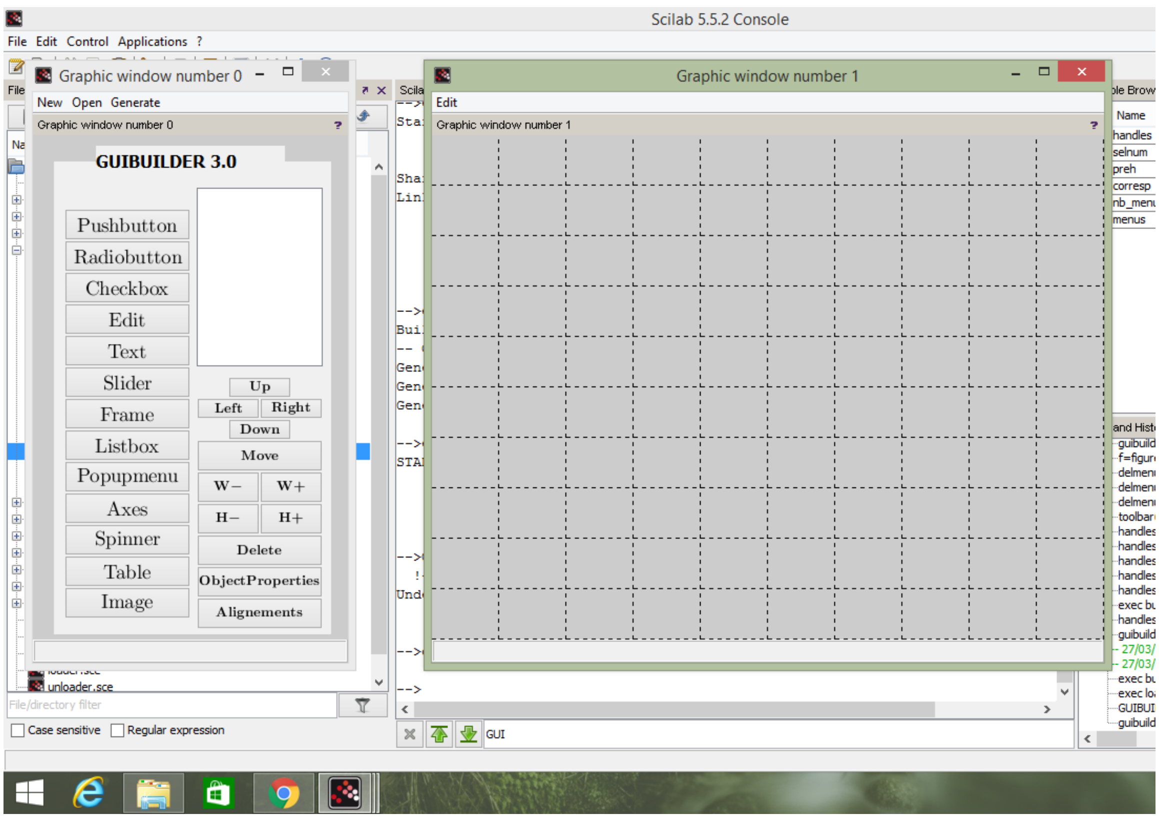


Fig. 1.3: GUI sample

Typically, the user interacts with information by manipulating visual widgets that allow for interactions appropriate to the kind of data they hold. The widgets of a well-designed interface are selected to support the actions necessary to achieve the goals of the user.

A model-view-controller allows for a flexible structure in which the interface is independent from and indirectly linked to application functionality, so the GUI can be easily customized. This allows the user to select or design a different skin at will, and eases the designer's work to change the interface as the user needs evolve. Good user interface design relates to the user, not the system architecture.

A GUI may be designed for the requirements of a vertical market as application-specific graphical user interfaces. Examples of application-specific GUIs include automated teller machines (ATM), point-of-sale touchscreens at restaurants, self-service checkouts used in a retail store, airline self-ticketing and check-in, information kiosks in a public space, like a train station or a museum, and monitors or control screens in an embedded industrial application which employ a real time operating system (RTOS).

In our project we used GUI Builder which is a Graphic User Interface Builder under Scilab. The program allowed us to build our GUI quickly, and the code for the GUI was generated automatically. We are using scilab 5.5, which includes the features like:

1. Ask confirmation before create a new GUI or open an existing one
2. Possibility to move by 4 buttons a group of buttons.

i. SCILAB AS A SUBSTITUTE OF MATLAB

In MATLAB, the user can click the run button, or a slider control multiple times. This calls multiple instances of the function, with bizarre results. In a couple cases enough functions are called to crash MATLAB.

Also, if user presses any one button or selects any option, then user is restricted to that option only. User cannot press other radio button simultaneously.

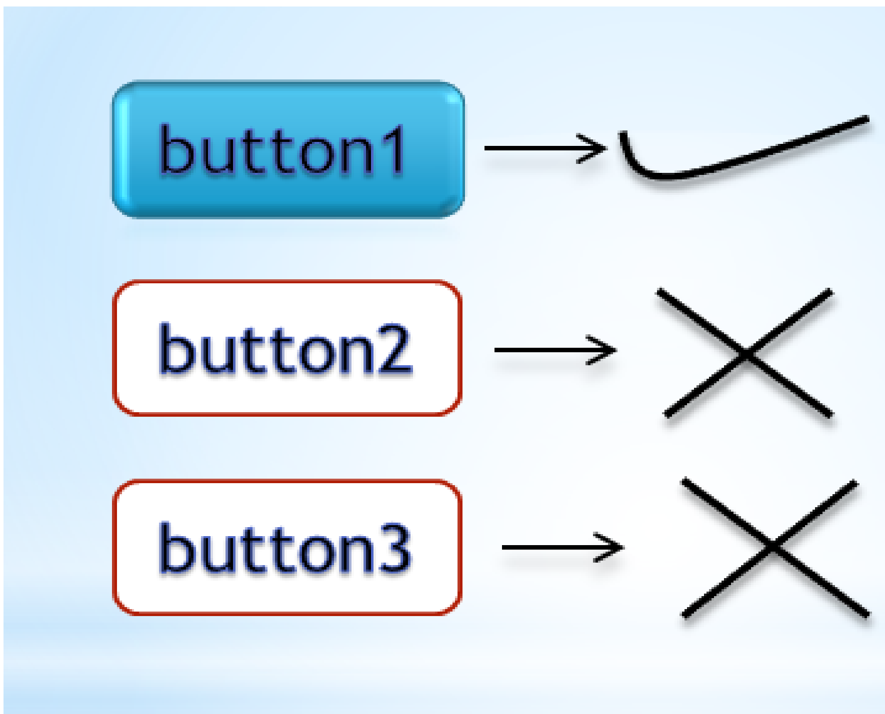


Fig. 1.4 : radio buttons in MATLAB

MATLAB is paid software which would make our project pretty much expensive.

On the other hand if we use scilab, then there will not be any issues regarding radio buttons. Because in scilab there exists CHECK BOX instead of radio buttons. If user wants to select more than one option then he is free for that.

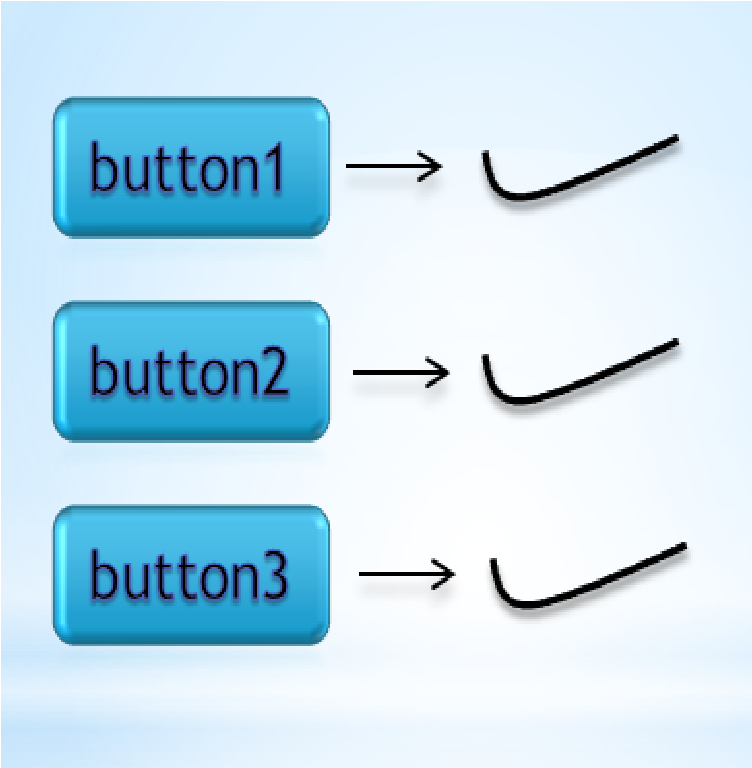


Fig. 1.5 : check box in scilab

In scilab, matrices and vectors can be created easily – no typing, or storage allocation is needed. I Matrix-vector product, scalar-vector/matrix products are written without any fuss - like the mathematicians do. Scilab is extensively used for linear algebra and simulation, control system design. Scilab has good graphics capability as well. Because of all these features we have chosen scilab as the programming language for our project.

ii. ARDUINO & SCILAB FIRMWARE

ARDUINO FIRMWARE:

Arduino firmware is important parameter when we have to deal with the interfacing of Scilab and Arduino. It is always required whenever scilab code is being run.

SCILAB FIRMWARE:

Scilab firmware is for user test purpose. User can confirm by using this firmware that the program has been successfully executed and it is ready for further use. Basically, scilab and arduino firmwares are used as language converter which are very much essential while serial communication of arduino and scilab and also while interfacing these functionalities with hardware. This is used only for checking whether the connection between arduino and scilab is established or not

SCILAB FIRMWARE CODES:

```
mode(0)
h=open_serial(1,2,115200);
for i=1:3
write_serial(1,"v",1);
read_serial(1,3)
endclose_serial(1)
```

CHAPTER-3

DEVELOPING FRONT-END USING GUI

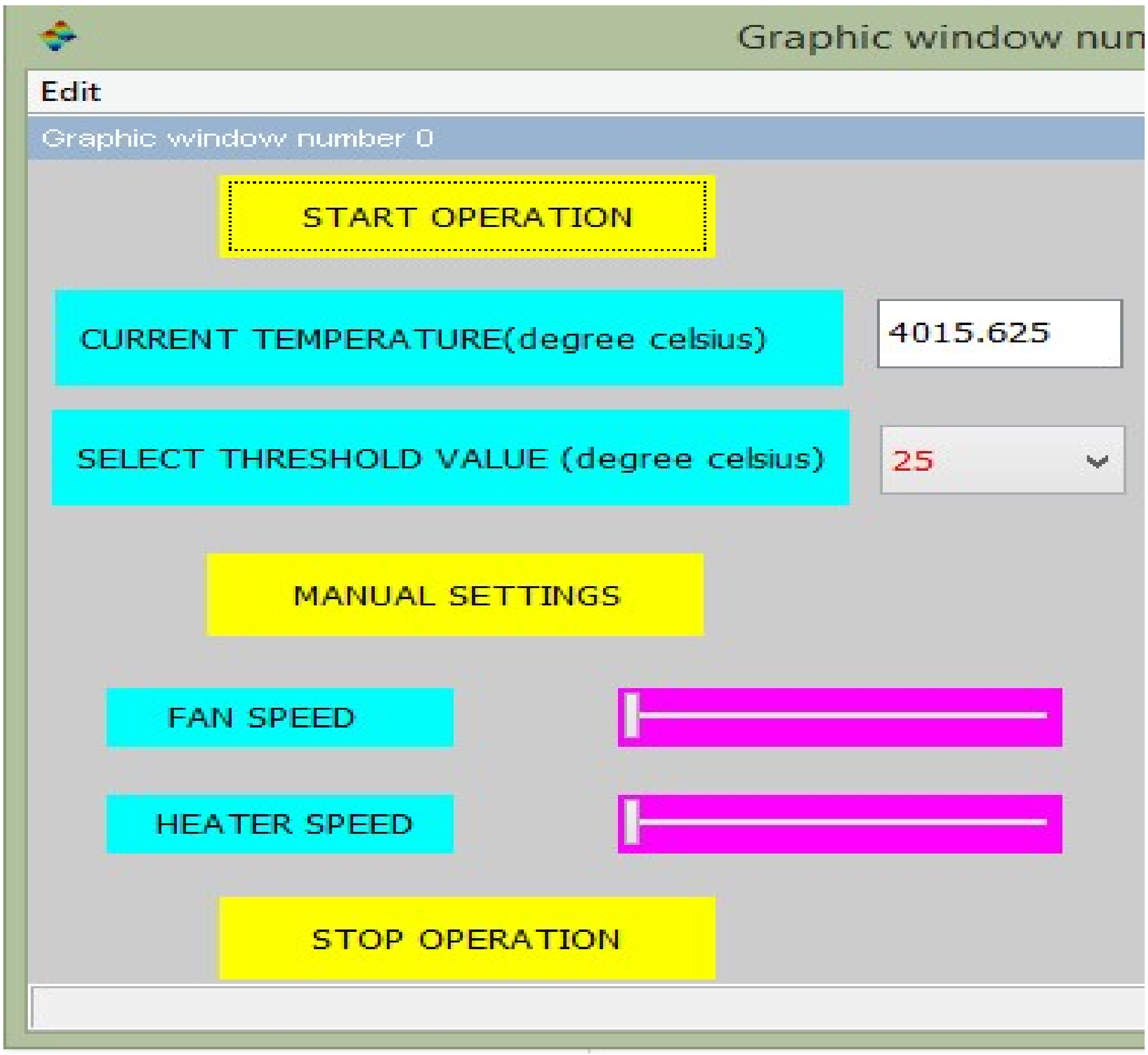


Fig. 1.6 : Front-End for monitoring & control purpose

The front-end we have developed, consists of push buttons for start and stop operations. There is a block which will give current temperature value in degree Celsius. Threshold values can be selected whose values are prior set inside the main program. For the manual setting there are two additional blocks to control the speed of the fan as well as the temperature. The front-end is formed using drag and drop operation with the help of GUI toolbox. There is option where we can change the color of each block for simplification purpose or to make front-end more attractive.

CHAPTER –4

HARDWARE

This part explains about the hardware design and construction involves in this system.

- Arduino UNO board
- Arduino ARK
- Fan (to reduce temperature)
- LCD screen (16*2)
- Motor driver (to run motor)
- Battery source (for power supply)
- LM35 (temperature sensor IC)
- Potentiometer (10k)
- Jumper wires

- LM35

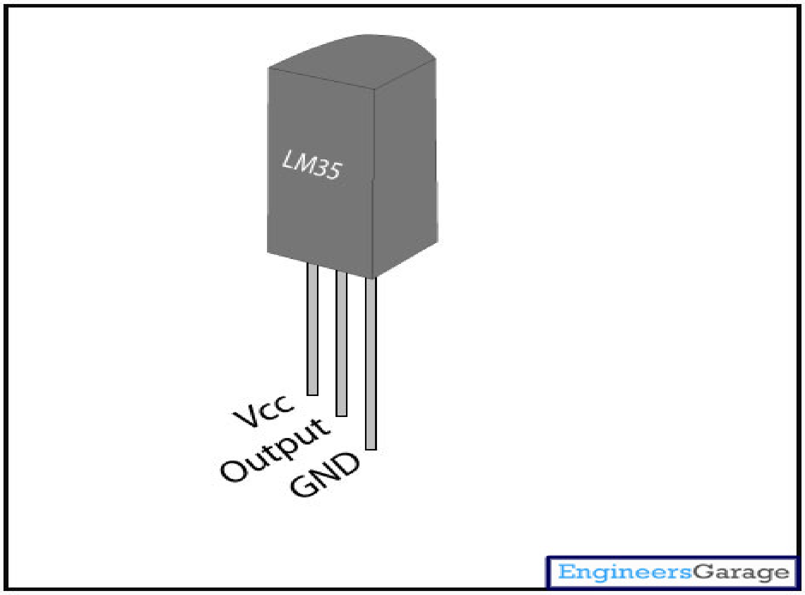


Fig. 1.7: LM35

The LM35 is precision integrated-circuit temperature devices with an output voltage linearly-proportional to the Centigrade temperature. The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°Cover a full −55°C to 150°C temperature range The low-output impedance, linear output, and precise inherent calibration of the LM35 device makes interfacing to readout or control circuitry especially easy. It has very low self-heating of less than 0.1°C in still air. The LM35 device is rated to operate over a −55°C to 150°C temperature range.

- ✓ PIN DISCRIPTION:

Pin No	Function	Name
1	Supply voltage; (5V)	V_{cc}
2	Output voltage (+6V to -1V)	Output
3	Ground (0V)	Ground

Table 2.2: Pin description of LM35

- ✓ FEATURES:

- Calibrated directly in ˚ Celsius (Centigrade)
- Linear + 10.0 mV/˚C scale factor n 0.5˚C accuracy (at +25˚C)
- Rated for full −55˚ to +150˚C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts n Less than 60 µA current drain
- Low self-heating, 0.08˚C in still air
- Nonlinearity only ±1⁄4˚C typical
- Low impedance output, 0.1 Ω for 1 mA load

- ✓ APPLICATIONS:

The LM35 can be applied easily in the same way as other integrated-circuit temperature sensors. It can be glued or cemented to a surface and its temperature will be within about 0.01˚C of the surface temperature. This presumes that the ambient air temperature is almost the same as the surface temperature; if the air temperature were much higher or lower than the surface temperature, the actual temperature of the LM35 die would be at an intermediate temperature between the surface temperature and the air temperature. This is especially true for the TO-92 plastic package, where the copper leads are the principal thermal path to carry heat into the device, so its temperature might be closer to the air temperature than to the surface temperature. To minimize this problem, be sure that the wiring to the LM35, as it leaves the device, is held at the same temperature as the surface of interest. The easiest way to do this is to cover up these wires with a bead of epoxy which will insure that the leads and wires are all at the same temperature as the surface, and that the LM35 die’s temperature will not be affected by the air temperature.

The TO-46 metal package can also be soldered to a metal surface or pipe without damage. Of course, in that case the V− terminal of the circuit will be grounded to that metal. Alternatively, the LM35 can be mounted inside a sealed-end metal tube, and can then be dipped into a bath or screwed into a threaded hole in a tank. As with any IC, the LM35 and accompanying wiring and circuits must be kept insulated and dry, to avoid leakage and corrosion. This is especially true if the circuit may operate at cold temperatures where condensation can occur. Printed-circuit coatings and varnishes such as Humiseal and epoxy paints or dips are often used to insure that moisture cannot corrode the LM35 or its connections. These devices are sometimes soldered to a small light-weight heat fin, to decrease the thermal time constant and speed up the response in slowly-moving air. On the other hand, a small thermal mass may be added to the sensor, to give the steadiest reading despite small deviations in the air temperature.

- LCD (LIQUID CRYSTAL DISPLAY)

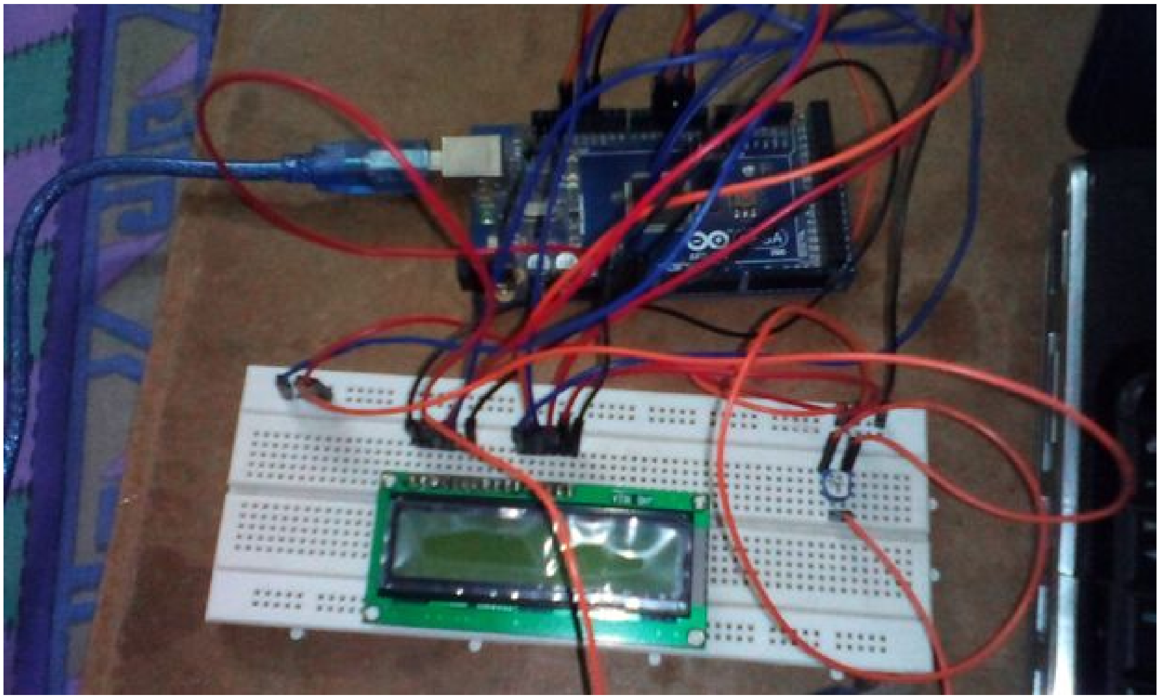


Fig. 1.8: LCD display

✓ Part to be Required:

1. LCD
2. Arduino board
3. Potentiometer
4. Jumper wires

✓ What is LCD?

LCD (liquid crystal display) is the technology used for displays in notebook and other smaller computers. Like light-emitting diode (LED) and gas-plasma technologies, LCDs allow displays to be much thinner than cathode ray tube (CRT) technology. LCDs consume much less power than LED and gas-display displays because they work on the principle of blocking light rather than emitting it.

An LCD is made with either a passive matrix or an active matrix display grid. The active matrix LCD is also known as a thin film transistor (TFT) display. The passive matrix LCD has a grid of conductors with pixels located at each intersection in the grid. A current is sent across two conductors on the grid to control the light for any pixel. An active matrix has a transistor located at each pixel intersection, requiring less current to control the luminance of a pixel. For this reason, the current in an active matrix display can be switched on and off more frequently, improving the screen refresh time.

✓ The name and functions of each pin of the LCD module:

Pin1 (Vss): Ground pin of the LCD module.

Pin2 (Vcc): +5V supply is given to this pin

Pin3 (VEE): Contrast adjustment pin. This is done by connecting the ends of a 10K potentiometer to +5V and ground and then connecting the slider pin to the VEE pin. The voltage at the VEE pin defines the contrast. The normal setting is between 0.4 and 0.9V.

Pin4 (RS): Register select pin. The JHD162A has two registers namely command register and data register. Logic HIGH at RS pin selects data register and logic LOW at RS pin will select command register. If we make the RS pin HIGH and put a data on the data lines (DB0 to DB7) it will be recognized as a data. If we make the RS pin LOW and put a data on the data lines, then it will be taken as a command.

Pin5 (R/W): Read/Write modes. This pin is used for selecting between read and write modes. Logic HIGH at this pin activates read mode and logic LOW at this pin activates write mode.

Pin6 (E): This pin is meant for enabling the LCD module. A HIGH to LOW signal at this pin will enable the module.

Pin7 (DB0) to Pin14 (DB7): These are data pins. The commands and data are put on these pins.

Pin15 (LED+): Anode of the back light LED. When operated on 5V, a 560 ohm resistor should be connected in series to this pin. In arduino based projects the back light LED can be powered from the 3.3V source on the arduino board.

Pin16 (LED-): Cathode of the back light LED.

✓ Steps to be followed:

Step 1: Make a Breadboard circuit and Set contrast with potentiometer:

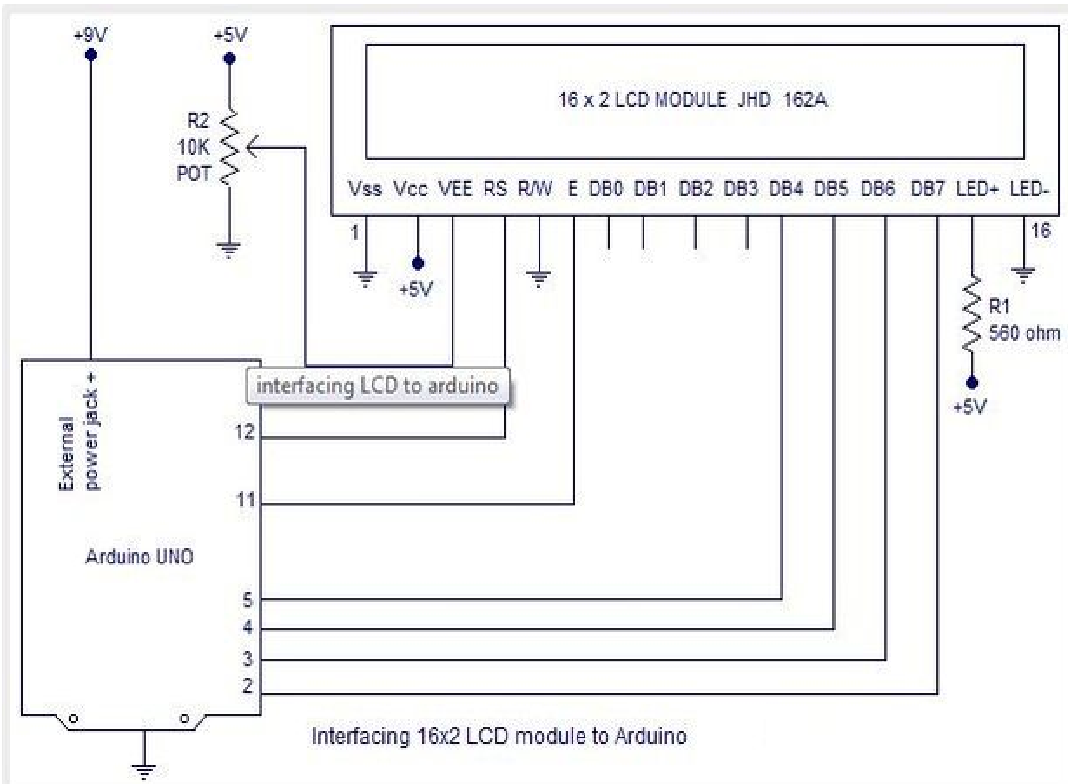


Fig. 1.9: LCD module interface

- Potentiometer (10k):

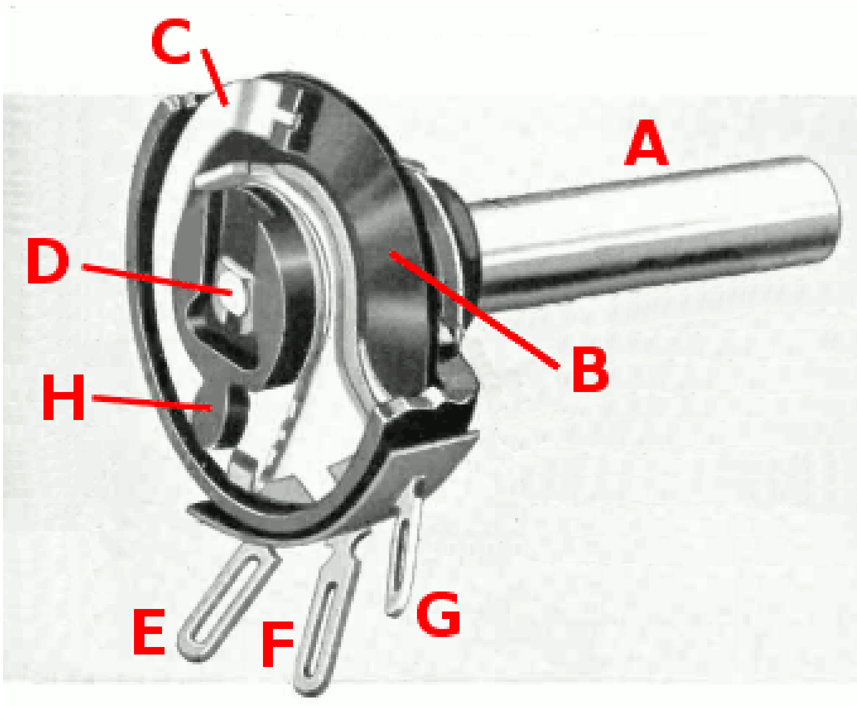


Fig. 2.0: potentiometer

A potentiometer, informally a pot, is a three-terminal resistor with a sliding or rotating contact that forms an adjustable voltage divider. If only two terminals are used, one end and the wiper, it acts as a variable resistor or rheostat.

The measuring instrument called a potentiometer is essentially a voltage divider used for measuring electric potential (voltage); the component is an implementation of the same principle, hence its name.

Potentiometers are commonly used to control electrical devices such as volume controls on audio equipment. Potentiometers operated by a mechanism can be used as position transducers, for example, in a joystick. Potentiometers are rarely used to directly control significant power (more than a watt), since the power dissipated in the potentiometer would be comparable to the power in the controlled load.

CHAPTER – 5

WORKING

LM35 is connected to arduino uno. Fan and motor drivers are also connected to arduino uno. Arduino uno will serially communicate with GUI, which is programmed in scilab. First arduino is connected to second arduino ARK. LCD screen is connected via Arduino ARK. Also the program for graph is done in arduino ARK itself. As the temperature will increase and reach to its threshold value, the fan will automatically switched on. We can see the current temperature value in front-end.

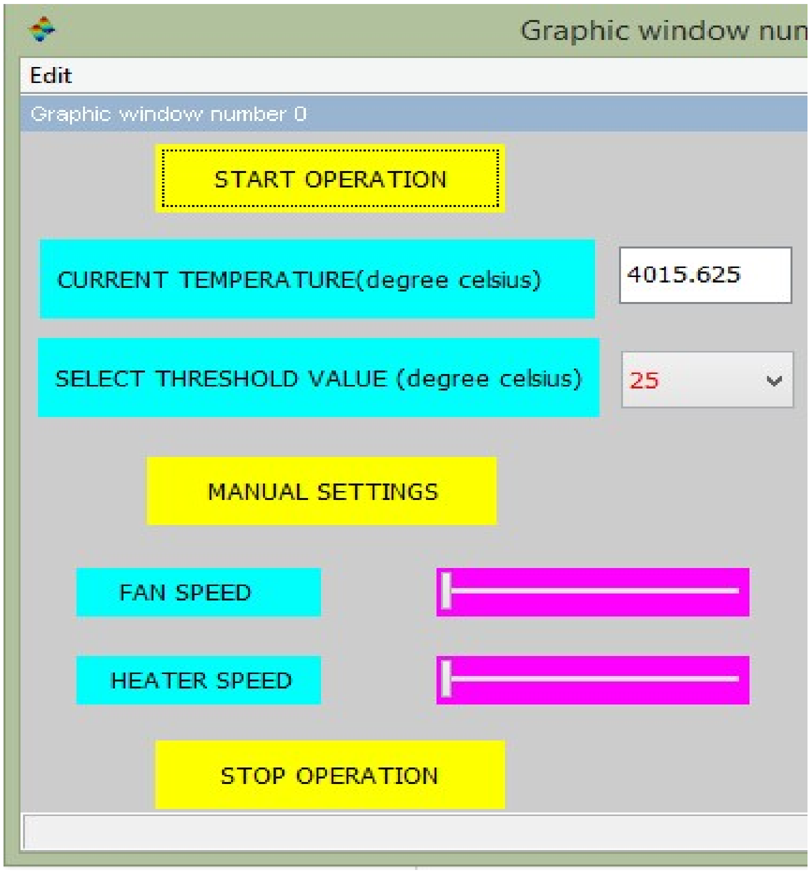


Fig. 2.1 : GUI front-end

Arduino UNO is connected to arduino ARK. There will be graph of temperature v/s time which will give the accurate variation in temperature every second. Temperature can be monitor more precisely and accurately with the help of the graph present on the monitor of computer.

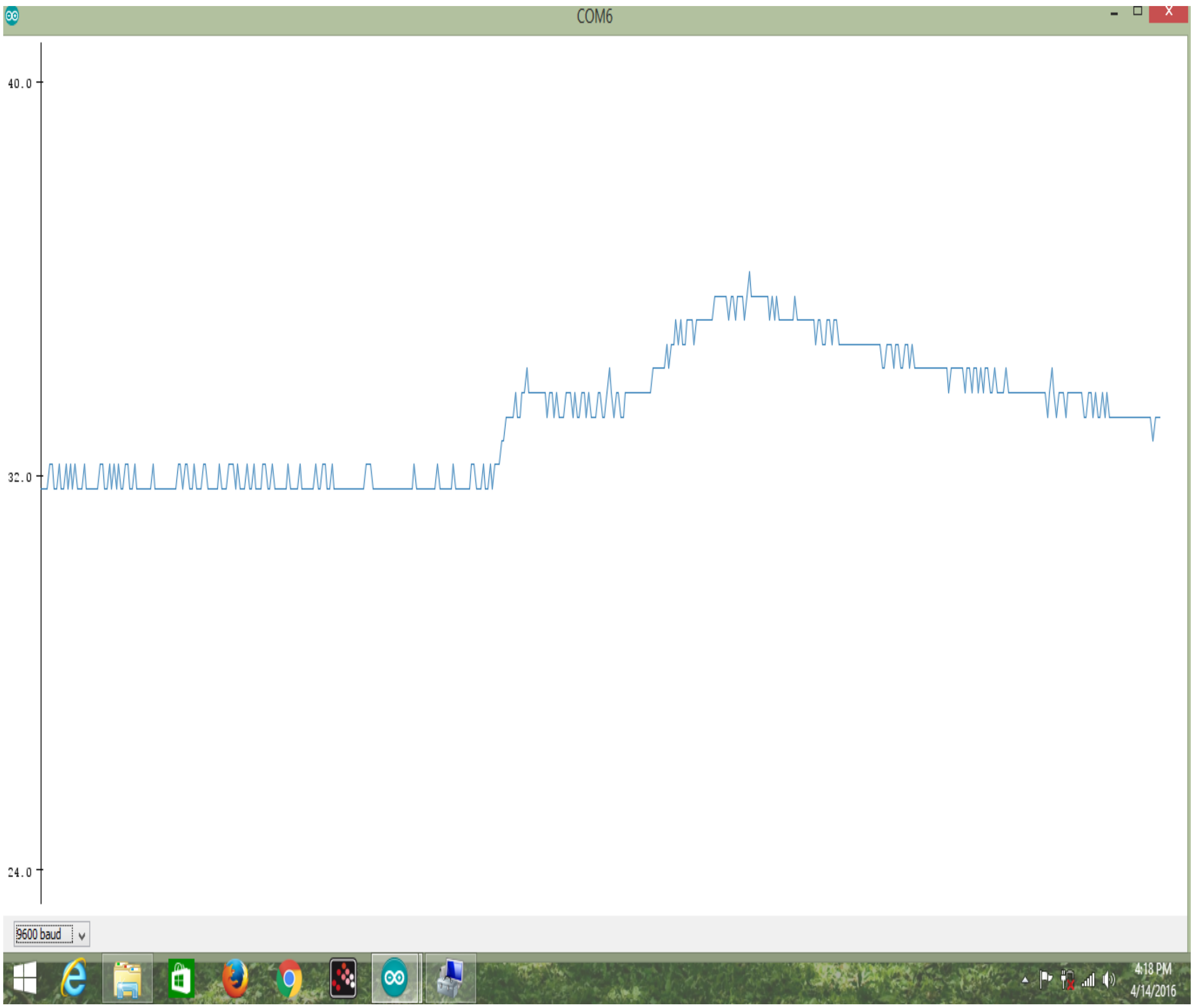


Fig. 2.2 : Graph on monitor

LCD is connected via arduino 2. The temperature value will be displayed on the LCD screen. The pin connection for LCD with arduino is:

Pin 1 to GND
Pin 2 to 5V
Pin 3 to wiper
Pin 4 to Arduino pin 12
Pin 5 to GND
Pin 6 to Arduino pin 11
Pin 11 to Arduino pin 5
Pin 12 to pin 4
Pin 13 to pin 3
Pin 14 to pin 2

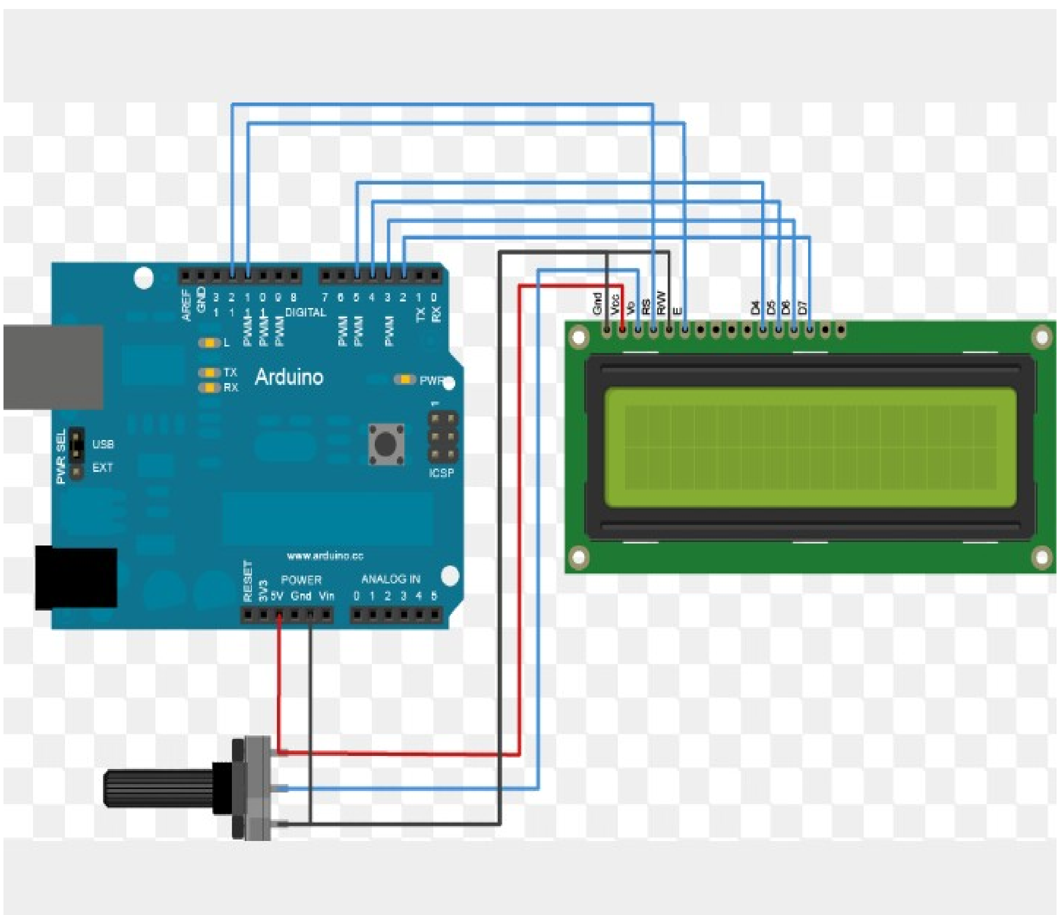


Fig. 2.3: Pin connection for LCD with arduino

If the temperature goes beyond its threshold value it will be displayed on the LCD screen.

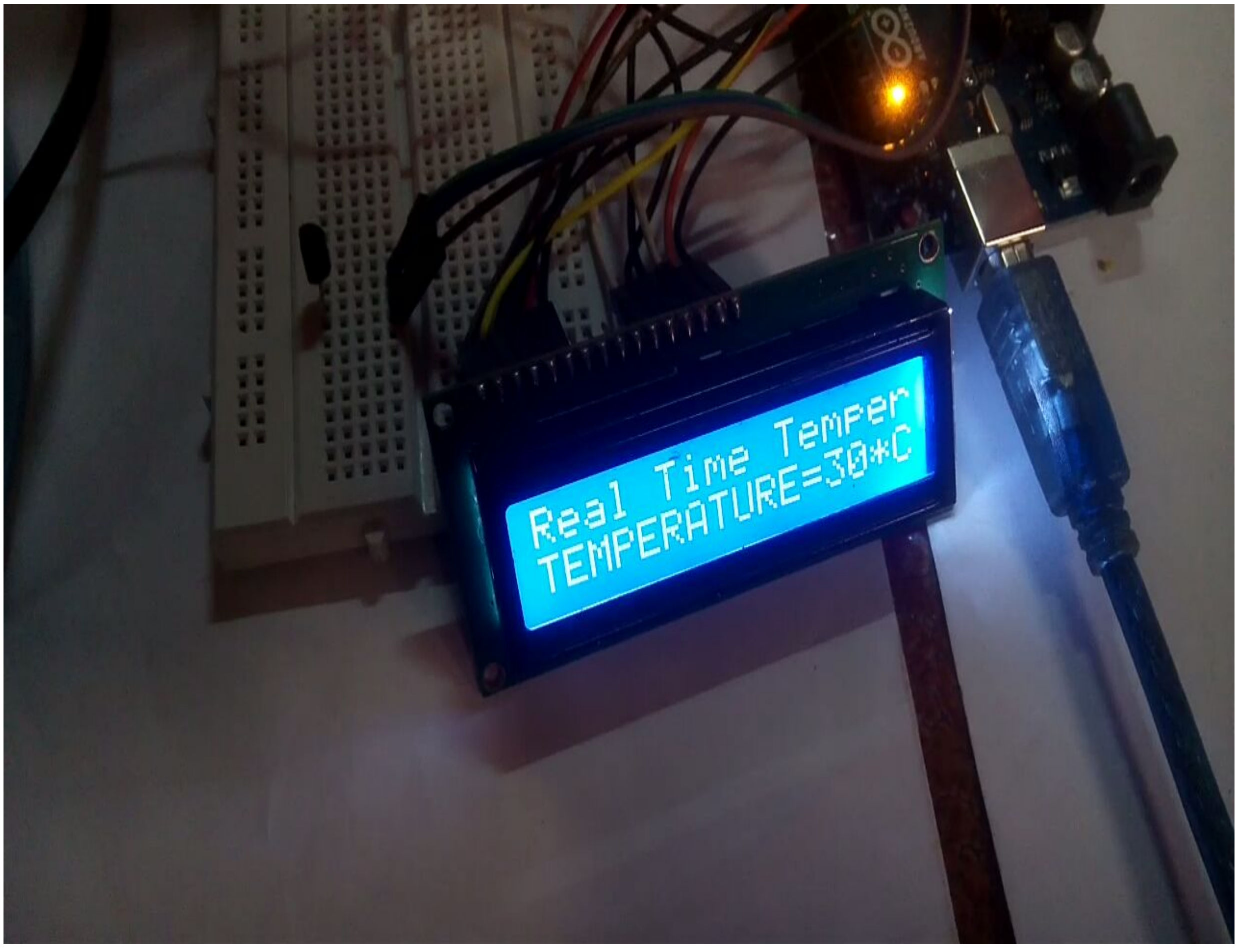


Fig. 2.4 : Message displayed on LCD

CHAPTER – 6

APPLICATION

Temperature monitoring & control system is essential in areas where the greatest variability in temperature is expected to occur within the qualified storage volume and they should be positioned so as to be minimally affected by transient events. Temperature monitoring systems should be installed in all temperature controlled rooms, cold rooms, freezer rooms, refrigerators and freezers

- The system can be used in industries like factory or main frame distribution where it is required to maintain constant temperature.
- It can be used in main frame room of telecommunication companies.
- It can be used in infant baby incubator where temperature is to be kept constant i.e. $39^{o}C$ as inside the mother's womb.
- This system can be used in Poultry management.
- The system can be used in server rooms, labs, ICUs and ATM machine room.

CONCLUSION

Thus when the temperature crosses the set temperature value, the fan will be automatically switch on. The scilab and arduino software will be used in monitoring & controlling the temperature. We can have the current value of temperature. We can change threshold value any time we wish or whenever require. We can analyze variation in temperature via graph which will be obtained in new pop-up window. If the temperature will cross its threshold, it will be displayed on the LCD screen connected via arduino ARK. It is the simple and efficient in maintaining the temperature irrespective of the outside temperature. It is the low cost solution for monitoring and controlling temperature as scilab and arduino both are open source platform and hardware part is not so complex as well.

• APPENDIX

i- FUTURE SCOPE

- Controlling parameters like:
    - Humidity
    - Moisture content
- Connecting Bluetooth
- Connecting GSM
- Speech control system
- Can be used in blood PH monitoring system

ii- REFERENCES

I. Real time temperature of oven using matlab-simulink, 11th WSEAS international conference on 23-25 July 2007

II. A resistive furnace published in Applied and Theoretical Electricity (ICATE), 2014 International conference on 23-25 October 2014

III. System using DSP technique published in multimedia, Signal Processing and Communication Technologies(IMPACT), 2011 international conference on 17-19 December.2011

IV. Real-time temperature monitoring using multiple wireless sensor networks published in sensors, 2005 IEE. Conference on 30 November 2005

V. TEMPERATURE MONITORING SYSTEM: Report by faculty of Electronics and Computer Engineering Universiti Teknikal Malaysia Melaka May 2011