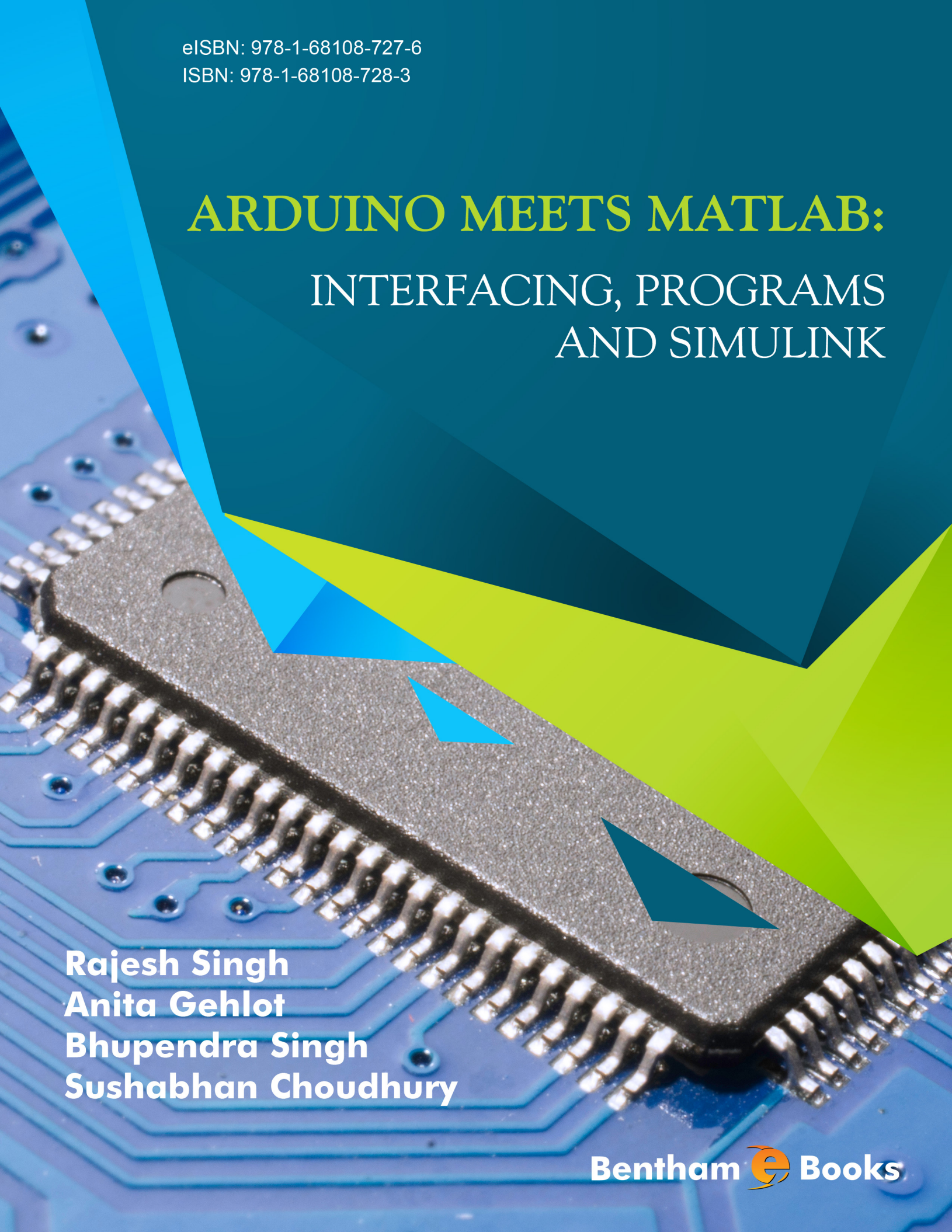
eISBN: 978-1-68108-727-6
ISBN: 978-1-68108-728-3

ARDUINO MEETS MATLAB:

INTERFACING, PROGRAMS AND SIMULINK



Rajesh Singh
Anita Gehlot
Bhupendra Singh
Sushabhan Choudhury

Bentham Books

Arduino Meets MATLAB Interfacing, Programs and Simulink

Authored by:

Rajesh Singh
School of Electronics & Electrical Engineering, Lovely Professional University Jalandhar, Punjab, India

Anita Gehlot
School of Electronics & Electrical Engineering, Lovely Professional University Jalandhar, Punjab, India

Bhupendra Singh
Schematics Microelectronics, Dehradun, Uttrakhand, India

&

Sushabhan Choudhury
Department of Electronics & Electrical Engineering, University of Petroleum and Energy Studies, Dehradun, Uttrakhand, India

Arduino Meets MATLAB: Interfacing, Programs and Simulink

Authors: Rajesh Singh, Anita Gehlot, Bhupendra Singh, Sushabhan Choudhury

ISBN (Online): 978-1-68108-727-6

ISBN (Print): 978-1-68108-728-3

General:

1. Any dispute or claim arising out of or in connection with this License Agreement or the Work (including non-contractual disputes or claims) will be governed by and construed in accordance with the laws of the U.A.E. as applied in the Emirate of Dubai. Each party agrees that the courts of the Emirate of Dubai shall have exclusive jurisdiction to settle any dispute or claim arising out of or in connection with this License Agreement or the Work (including non-contractual disputes or claims).
2. Your rights under this License Agreement will automatically terminate without notice and without the need for a court order if at any point you breach any terms of this License Agreement. In no event will any delay or failure by Bentham Science Publishers in enforcing your compliance with this License Agreement constitute a waiver of any of its rights.
3. You acknowledge that you have read this License Agreement, and agree to be bound by its terms and conditions. To the extent that any other terms and conditions presented on any website of Bentham Science Publishers conflict with, or are inconsistent with, the terms and conditions set out in this License Agreement, you acknowledge that the terms and conditions set out in this License Agreement shall prevail.

Bentham Science Publishers Ltd.
Executive Suite Y - 2
PO Box 7917, Saif Zone
Sharjah, U.A.E.
Email: subscriptions@benthamscience.org

CONTENTS

FOREWORD

This book titled Arduino meets MATLAB. .. Interfacing, Programs and Simulink will provide a platform for the beginners to get started with Arduino and its interfacing with the MATLAB. The book provides the basic knowledge of the programming and interfacing of the devices with Arduino and MATLAB. This book also explains in a lucid manner the basic steps to understand the interfacing and programming with Arduino and MATLAB. This book goes from basic to advanced level of Arduino and interfacing with various input/output devices through various communication modules in well defined sequence which will be easily understandable to the undergraduate and post graduate students. This book will be specially beneficial to those researchers looking for hardware based implementation platforms.

The USP of this book lies in the fact that a new concept has been introduced for the researchers and students for prototype development of the real time projects using Arduino, MATLAB and I/O devices in the single platform. Another important point to mention will be that this book is entirely based on the practical experience of the authors while undergoing projects with the students and industries.

Dr. Babu Sena Paul
Department of Electrical and Electronic Engineering Technology
University of Johannesburg
South Africa

PREFACE

The primary objective of writing this book is to provide a platform for the beginners to get started with Arduino and its interfacing with the MATLAB. The book provides the basic knowledge of the programming and interfacing of the devices with Arduino and its interfacing with MATLAB.

The aim is to explain the basic steps to understand the interfacing and programming to interface Arduino with MATLAB.

This book provides basics to advanced knowledge of Arduino and its interfacing with input/output devices (display devices, actuators, sensors), communication modules (RF modem, Zigbee) and MATLAB. This would be beneficial for the people who want to get started with hardware based project prototypes. Embedded system based on Arduino with simulation, programming and interfacing with MATLAB all at a single platform. Arduino interfacing with MATLAB with and without I/O packages is included. Basics of the Arduino are covered in section-A, how to interface Arduino with basic input/output devices. Section-B covers Arduino interfacing with MATLAB with I/O package and section-C covers Arduino interfacing with Arduino without I/O package.

The concept which makes this book unique is a book programming and simulation of Arduino and MATLAB based real time project prototypes at a single platform.

This book is entirely based on the practical experience of the authors while undergoing projects with the students and industries. Although the circuits and programs mentioned in the text are tested on real hardware but in case of any mistake we extend our sincere apologies. Any suggestions to improve in the contents of book are always welcome and will be appreciated and acknowledged.

CONSENT FOR PUBLICATION

Not applicable.

CONFLICT OF INTEREST

The author(s) declared no conflict of interest regarding the contents of each of the chapters of this book.

ACKNOWLEDGEMENTS

We acknowledge the support from Nutty Engineer to use its products to demonstrate and explain the working of the systems. We would like to thank BENTHAM SCIENCE for encouraging our idea about this book and the support to manage the project efficiently.

We are grateful to the honorable Chancellor (Lovely Professional University) Ashok Mittal, Mrs. Rashmi Mittal (Pro Chancellor, LPU), Dr. Ramesh Kanwar (Vice Chancellor, LPU), Dr. Loviraj Gupta (Executive Dean, LPU) for their support. We are also thankful to the

chancellor (UPES) Dr. S.J Chopra, Dr. Dependra Jha (Vice Chancellor, UPES), Dr. Kamal Bansal (Dean, SoE, UPES), Dr. Piyush Kuchhal (Associate Dean, UPES) and Dr. Suresh Kumar (Director, UPES) for their support and constant encouragement. In addition we are thankful to our family, friends, relatives, colleagues and students for their moral support and blessings.

Rajesh Singh
School of Electronics &
Electrical Engineering,
Lovely Professional University,
Jalandhar, Punjab,
India

Anita Gehlot
School of Electronics &
Electrical Engineering,
Lovely Professional University,
Jalandhar, Punjab,
India

Sushabhan Choudhry
Department of Electronics &
Electrical Engineering,
University of Petroleum and Energy Studies,
Dehradun, Uttrakhand,
India

Bhupendra Singh
Schematics Microelectronics,
Dehradun, Uttrakhand,
India

SECTION A: Arduino and its Interfacing

CHAPTER 1

Introduction to Arduino, Arduino IDE and Proteus Software

Abstract: Arduino is an open source platform and easy to use software. The chapter is about to discuss the advantages of Arduino with brief description to each Arduino board including UNO, MEGA and NANO. Arduino Integrated Development Environment is used to write the program for Arduino, this chapter elaborates step to step description of writing and compiling the program. Proteus simulator is also introduced, which is used for checking the feasibility of program and working of the designed system without actual implementation on hardware. Design steps are described for the beginners.

Keywords: Arduino, Arduino IDE, Open Source Platform.

Arduino is a user friendly open-source platform. Arduino has on board microcontroller and IDE is used to program it. As compared to similar platforms it is easy to program and has many advantages over them.

ADVANTAGES

Low Cost - Arduino boards are of relatively low-cost as compared to other microcontroller platforms.

Cross-platform - The Arduino Software (IDE) is compatible with Windows, Macintosh OSX, and Linux operating systems, which most of microcontroller systems are not.

User Friendly - The Arduino Software (IDE) is user friendly and easy-to-use for beginners and much flexibility for skilled programmers.

Open Source - The Arduino is an open source software and can be programmed with C, C++ or AVR-C languages. So a variety of modules can be designed by users.

Arduino platform comprises of a microcontroller. It can be connected to PC *via* a USB cable. It is freely accessible and can be easily downloaded from

http://www.arduino.org/downloads. It can also be modified by the programmer. In the market different versions of Arduino boards are available and depending on the requirement of user.

1.1. ARDUINO UNO

The Arduino/Genuino Uno has on board ATmega328 microcontroller. It has on board six analog input ports (A0-A5). Each pin can operate on 0-5V of voltage. It has 14 digital I/O pins out of which 6 are PWM output, 6 analog inputs, 2 KB SRAM, 1 KB EEPROM and operates at 16 MHz of frequency. Table **1.1** shows the pin description of Arduino UNO. Fig. (**1.1**) shows the Arduino Uno board.

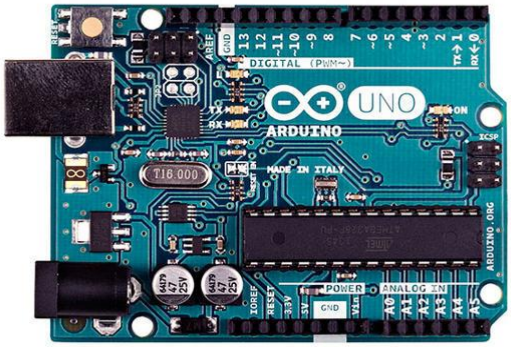


Fig. (1.1). Arduino Uno Board.

Table 1.1. Pin Description of Arduino UNO.

Pin	Description
Vin	The external voltage to the Arduino board.
+5V	Output a regulated 5V
3.3 V	On board 3.3 volt supply
GND	Ground
IOREF	provides the voltage reference and select appropriate power source
Serial	Transmits and receives serial data, Pins: 0(Rx) 1(Tx)
External Interrupts	trigger an interrupt on low value (Pins 2 & 3)
PWM	Provides 8 bit PWM output (pins: 3,5,6,9,10,11)
SPI	Supports SPI communication (Pins: 10(SS), 11(MOSI), 12 (MISO) and 13 (SCK))
LED	LED driven by pin 13
TWI	Supports TWI communication (Pins: A4 (SDA), A5(SCL))

(Table 1.1) cont.....

Pin	Description
AREF	Reference voltage for the analog inputs
Reset	It is used to reset the onboard microcontroller

1.2. ARDUINO MEGA

The Arduino Uno has on board ATmega2560 microcontroller. It has on board 16 analog inputs, 54 digital I/O, USB connection, 4 UART, power jack and reset button. It operates on 16 MHz frequency. The board can be operated with 5-12 volts of external power, if supplied more than this it can damage the board. It has on board 256 KB flash memory, 8 KB SRAM, 4 KB EEPROM. Table **1.2** shows the pin description of Arduino Mega. Fig. (**1.2**) shows the Arduino Mega board.

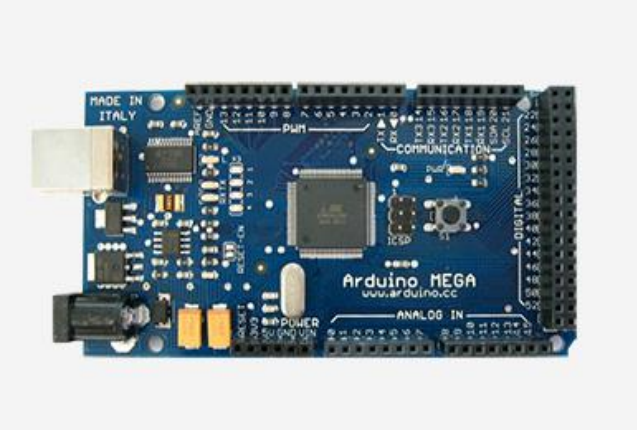


Fig. (1.2). Arduino Mega Board.

Table 1.2. Pin Description of Arduino Mega.

Pin	Description
Vin	The external voltage to the Arduino board.
+5V	Output a regulated 5V
3.3 V	On board 3.3 volt supply
GND	Ground
IOREF	provides the voltage reference and select appropriate power source
Serial0	Transmits and receives serial data, Pins: 0(Rx) 1(Tx)
Serial1	Transmits and receives serial data, Pins: 19(Rx) 18(Tx)
Serial2	Transmits and receives serial data, Pins: 17(Rx) 16(Tx)

CHAPTER 2

Arduino Interfacing with Display Devices

Abstract: Display devices are important part of any prototype developed by the learners. It is required to display the sensory values and other important information and check its validity at transmitter and receiver end. This chapter explores the interfacing of display devices with Arduino with the help of circuit diagrams and programs.

Keywords: Arduino, LED, Liquid Crystal Display, Seven Segment Display.

2.1. LIGHT EMITTING DIODE (LED)

Light emitting diode is most basic display device, which is mostly used to indicate the status of the system. For this purpose any color of LED can be used as per designer's choice. Fig. (**2.1**) shows the block diagram of LED interfacing with Arduino, comprises of Arduino board, power supply, resistors and LED.

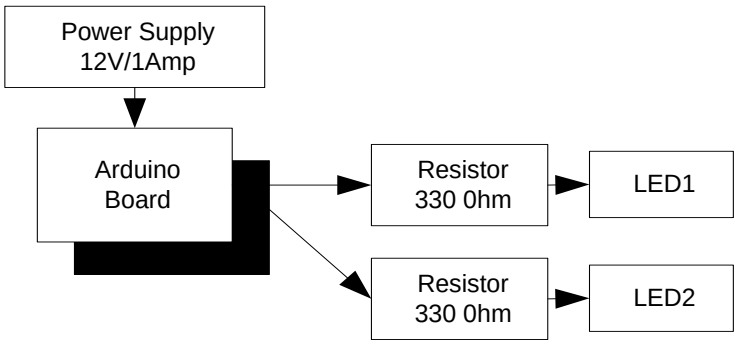


Fig. (2.1). Block diagram for LED interfacing with Arduino.

2.1.1. Circuit Diagram

Connect all the components to Arduino as per the connections as described-

1. LED1 indicator connected to pin13 of Arduino Uno through 330 ohm of resistor.
2. LED2 indicator connected to pin12 of Arduino Uno through 330 ohm of resistor.

3. DC jack of +12 V power supply is connected to power supply DC jack of Arduino. Fig. (**2.2**) shows the circuit diagram of the system.

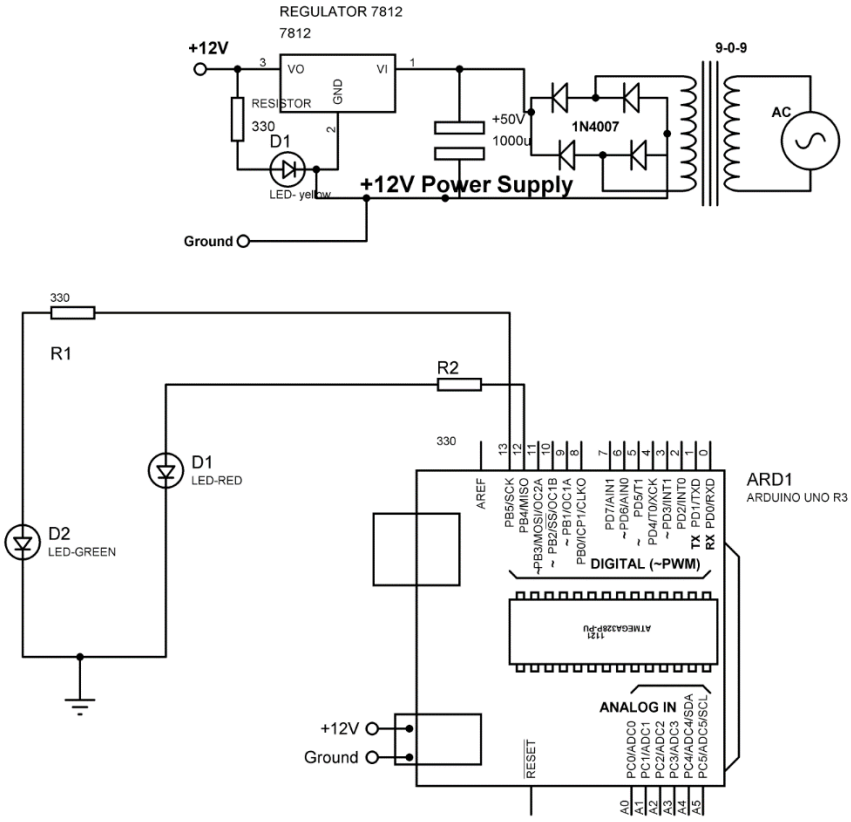


Fig. (2.2). Circuit diagram to interface LED with Arduino.

2.1.2. Program

Case1: (LED1 ‘ON’)

```
int led_FIRST = 13;

int led_SECOND = 12;

void setup()

{

// initialize the digital pin as an output.

pinMode(led_FIRST, OUTPUT);
```

```
pinMode(led_SECOND, OUTPUT);
}
void loop()
{
digitalWrite(led_FIRST, HIGH); // turn the LED on by making the voltage HIGH
digitalWrite(led_SECOND, HIGH);
delay(1000); // wait for 1000 millisecond
digitalWrite(led_FIRST, LOW); // turn the LED off by making the voltage LOW
digitalWrite(led_SECOND, LOW);
delay(1000); // wait for 1000 millisecond
}
```

Case2: (LED2 ‘ON’)

```
int led_FIRST = 13;
int led_SECOND = 12;

void setup()
{
// initialize the digital pin as an output.
pinMode(led_FIRST, OUTPUT);
pinMode(led_SECOND, OUTPUT);
}
void loop()
{
digitalWrite(led_FIRST, HIGH); // turn the LED ON/OFF by making the voltage
HIGH and LOW
```

CHAPTER 3

Arduino Interfacing with Digital Sensors

Abstract: Digital sensor gives digital output and changes the status with respect to change in conditions which can be processed through a controller unit. This chapter describes the interfacing of digital sensors like flame sensor, PIR sensor and Gas sensor with Arduino with the help of circuit diagram and programs.

Keywords: Arduino, Flame sensor, Gas sensor, PIR sensor.

3.1. FLAME SENSOR

Flame sensor or fire sensor acts as digital sensor which changes its output status with respect to the environmental changes. Fig. (**3.1**) shows the block diagram of the system, comprises of Arduino, power supply, flame sensor, LED. It is designed to sense the fire in the surrounding and corresponding change in the LED takes place.

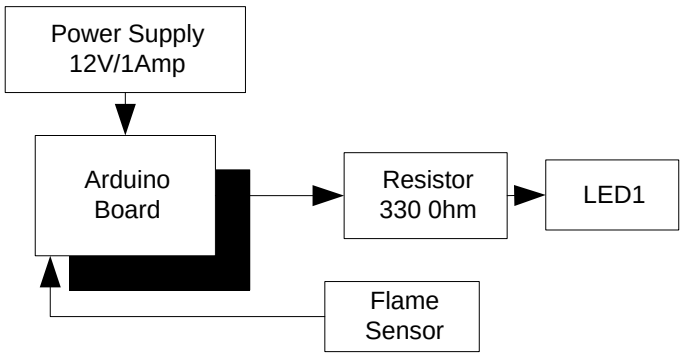


Fig. (3.1). Block diagram to interface flame sensor.

3.1.1. Circuit Diagram

Connect all the components to Arduino as per the connections as described-

1. Flame sensor ‘OUT’ pin is connected to pin3 of Arduino Uno.

2. +Vcc and GND pins of flame sensor are connected to +5V and GND pin of power supply patch/explorer respectively.
3. Anode of LED1 is connected to pin2 of Arduino Uno through 330 ohm of resistor.
4. Cathode of LED1 is connected to 'GND'.
5. +12V power supply jack is connected to DC jack of Arduino Uno.

Fig. (**3.2**) shows circuit diagram to interface flame sensor.

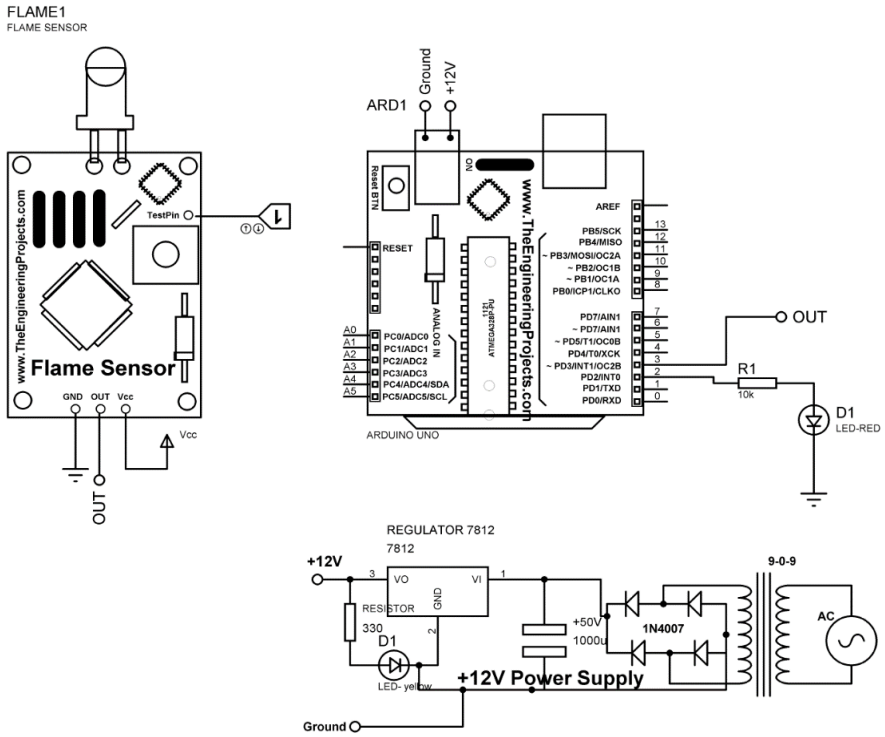


Fig. (3.2). Circuit diagram to interface flame sensor.

3.1.2. Program

```
int FlameOUT = 3;

int LED=2;

void setup()

{
```

```
pinMode(FlameOUT, INPUT_PULLUP);
pinMode(LED, OUTPUT);
}
void loop()
{
if(digitalRead(FlameOUT) == HIGH)
{
digitalWrite(LED,HIGH);
delay(20);
}
if(digitalRead(FlameOUT) == LOW)
{
digitalWrite(LED,LOW);
delay(20);
}
}
```

3.1.3. Proteus Simulation Model

Connect the components with Arduino as described in section 3.1.2 in the virtual environment of Proteus simulator. Power supply need not to be connected in the virtual environment of Proteus. As Proteus is virtual environment to make the change in the status of sensor a 'logic' is connected to logic state pin, which can be changed to '0' or '1' to check the working of sensor. Load the program as described in section 3.1.3 and check the feasibility and working of the circuit Fig. (**3.3**) shows the Proteus model for the system.

CHAPTER 4

Arduino Interfacing with Analog Sensors

Abstract: Analog sensor gives analog output and changes the status with respect to change in conditions which can be processed through a controller unit. This chapter describes the interfacing of analog sensors like ultrasonic sensor and temperature sensor with Arduino with the help of circuit diagram and programs.

Keywords: Arduino, Analog sensor, Temperature sensor, Ultrasonic sensor.

4.1. ULTRASONIC SENSOR

Ultrasonic sensor is an analog sensor which changes its output status with respect to the change in the distance from an object. Fig. (**4.1**) shows the block diagram of the system, comprises of Arduino, power supply, ultrasonic sensor, LCD. It is designed to measure the distance from an object and corresponding change is displayed on LCD.

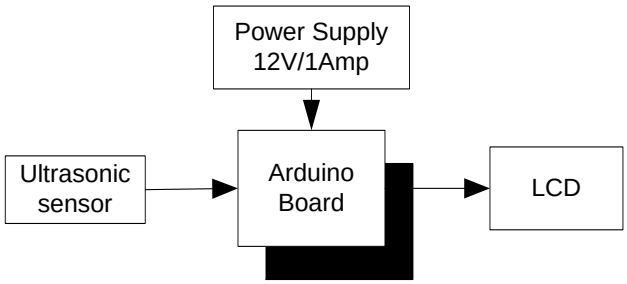


Fig. (4.1). Block diagram of the interfacing of ultrasonic sensor.

4.1.1. Circuit Diagram

Connect all the components to Arduino as per the connections as described-

1. Trigger pin is connected to pin3 of Arduino Uno.
2. Echo pin is connected to pin2 of Arduino Uno.
3. +Vcc and GND pins of ultrasonic sensor are connected to +5V and GND pins of power supply patch respectively.

4. RS pin of LCD is connected to pin12 of Arduino Uno.
5. RW pin of LCD is connected to GND pin of Arduino Uno.
6. RS pin of LCD is connected to pin11 of Arduino Uno.
7. D4 pin of LCD is connected to pin10 of Arduino Uno.
8. D5 pin of LCD is connected to pin9 of Arduino Uno.
9. D6 pin of LCD is connected to pin8 of Arduino Uno.
10. D7 pin of LCD is connected to pin7 of Arduino Uno.
11. Pins 1and 16 are connected to GND pin of power supply patch.
12. Pins 2 and 15 are connected to +5V pin of power supply patch.
13. +12V power supply jack is connected to DC jack of Arduino Uno.

Note- To vary the intensity of LCD Potentiometer can be used. Variable terminal of 10K POT is connected to pin 3 of LCD. Two-fixed terminals are connected to +5V and GND pin of patch respectively. Vary the value to change the background light intensity.

Fig. (**4.2**) shows circuit diagram for the interfacing of ultrasonic sensor.

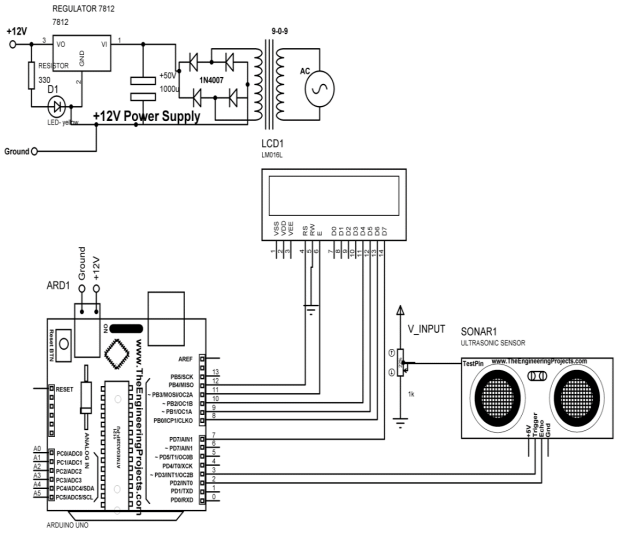


Fig. (4.2). Circuit diagram for the interfacing of ultrasonic sensor.

4.1.2. Program

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 10, 9, 8, 7);

const int pingPin = 3; // Trigger Pin of Ultrasonic Sensor
```

```
const int echoPin = 2; // Echo Pin of Ultrasonic Sensor
void setup()
{
lcd.begin(16,2); // initialize the 16*2 LCD
}
void loop()
{
lcd.clear();
long duration, inches, cm;
pinMode(pingPin, OUTPUT);
digitalWrite(pingPin, LOW);
delayMicroseconds(2);
digitalWrite(pingPin, HIGH);
delayMicroseconds(10);
digitalWrite(pingPin, LOW);
pinMode(echoPin, INPUT);
duration = pulseIn(echoPin, HIGH);
inches = microsecondsToInches(duration);
cm = microsecondsToCentimeters(duration);
lcd.setCursor(0, 0);
lcd.print(inches);
lcd.print("inches");
lcd.setCursor(0, 1);
lcd.print(cm);
```

CHAPTER 5

Arduino Interfacing with Actuators

Abstract: An actuator is a component which is responsible for moving or controlling a mechanism. An actuator requires a control signal and a source of energy. This chapter explains the working of actuator with the help of different methods.

Keywords: AC motor, Arduino, DC motor, L293D, Stepper motor, Servo motor.

5.1. DC MOTOR CONTROL WITH TRANSISTOR 'H' BRIDGE

A DC motor is a device that converts electrical energy into mechanical energy. It has vital importance for the industry. Fig. (**5.1**) shows the block diagram of the system, comprises of Arduino, power supply, DC motor, LED. It is designed to control the DC motor with 'H' bridge (2N2222), LEDs are connected to check the change in the status of inputs to motor in order to make it move in forward and reverse direction. To make H bridge four 2N2222 transistors are used- Q1, Q2, Q3, Q4.

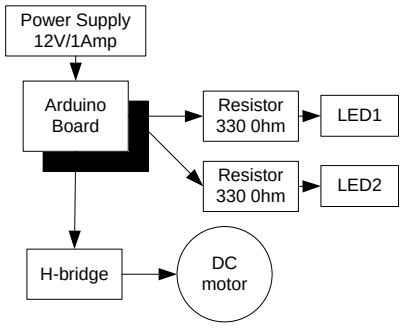


Fig. (5.1). Block diagram for the interfacing of DC motor.

5.1.1. Circuit Diagram

Connect all the components to Arduino as per the connections as described-

1. Make collector of Q1 & Q2 common and connect to positive terminal of +12V DC.
2. Make emitter of Q3 & Q4 common and connect to negative terminal of +12V DC and 'GND'.

3. Make base of Q1 & Q4 common and connect to pin9 of Arduino.
4. Make base of Q2 & Q3 common and connect to pin10 of Arduino.
5. LEDs are also connected parallel to inputs of H -bridge.
6. +12V DC jack of power supply is connected to DC jack of Arduino Uno.

Fig. (**5.2**) shows circuit diagram for the interfacing of DC motor.

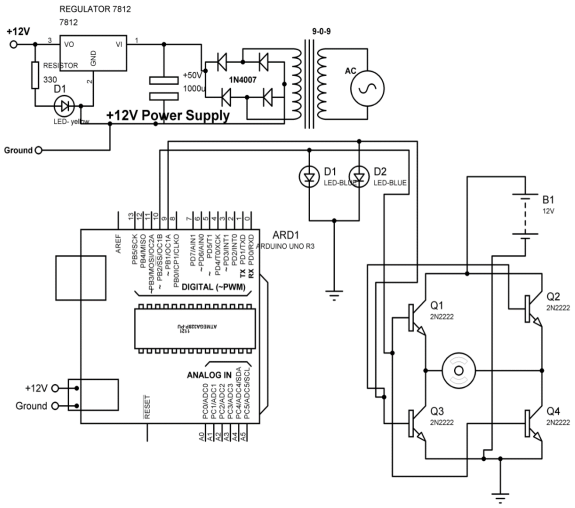


Fig. (5.2). Circuit diagram for the interfacing of DC motor.

5.1.2. Program

```
int MPIN1 = 10;

int MPIN2 = 9;

void setup()

{

// initialize pin10 and 9 as output

pinMode(MPIN1, OUTPUT);

pinMode(MPIN2, OUTPUT);

}

void loop()

{
```

```
digitalWrite(MPIN1, HIGH); // make 10 and 9 pin HIGH and LOW respectively
digitalWrite(MPIN2, LOW);
delay(1000); // wait for a 1000 millisecond
digitalWrite(MPIN1, LOW); // make 9 and 10 pin HIGH and LOW respectively
digitalWrite(MPIN2, HIGH);
delay(1000); // wait for a 1000 millisecond
}
```

5.1.3. Proteus Simulation Model

Connect the components with Arduino as described in section 5.1.1 in the virtual environment of Proteus simulator. Power supply need not to be connected in the virtual environment of Proteus. Load the program as described in section 5.1.2 and check the feasibility and working of the circuit. Fig. (**5.3**) shows the Proteus model for the system.

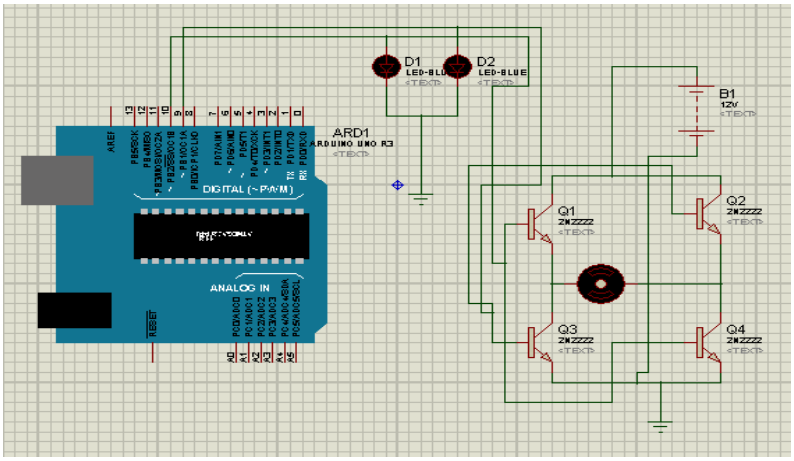


Fig. (5.3). Proteus simulation model for the Arduino interfacing with DC motor.

5.2. DC MOTOR CONTROL WITH L293D

L293D is 14 pin motor driver IC. DC motor upto 12V/1A ca be controlled with this.Fig. (**5.4**) shows the block diagram of the system, comprises of Arduino, power supply, DC motor, L293D, LED. It is designed to control the DC motor with IC L293D. LEDs are connected to check the change in the status of inputs to motor in order to make it move in forward and reverse direction.

CHAPTER 6

Arduino Interfacing with Wireless Modems

Abstract: The wireless communication is technology in which information is communicated through air without cables from one device to other by using different modems like IR, RF, satellite *etc*. This chapter describes the wireless modem (XBee) used to communicate with programs and connecting examples.

Keywords: Arduino, XBee.

6.1. XBEE MODEM

XBee module communicates with protocol 802.15.4, which is a point-to-point communication protocol. It can be used to design WPAN with free band. It has range of 1600 meters in line of sight and 90 meters in indoors or urban area. It is used for embedded solutions providing addressable wireless end – point connectivity to devices. This XBee wireless device can be directly connected to the serial port (at 3.3 V) of the microcontroller.

Before it can be used in the system, it needs to be configured first. To configure XBee module please follow following steps-

6.1.1. Configuration of XBee

Step 1: Download X-CTU Software from link- https://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu#productsupport-utilities and install [XCTU v.6.3.10].

Fig. (**6.1**) shows DIGI XCTU software.

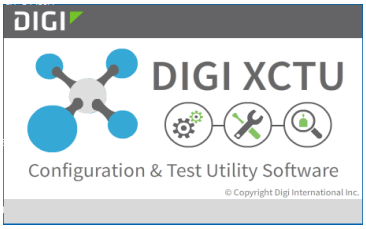


Fig. (6.1). DIGI XCTU.

Fig. (**6.2**) shows XCTU starting Window.

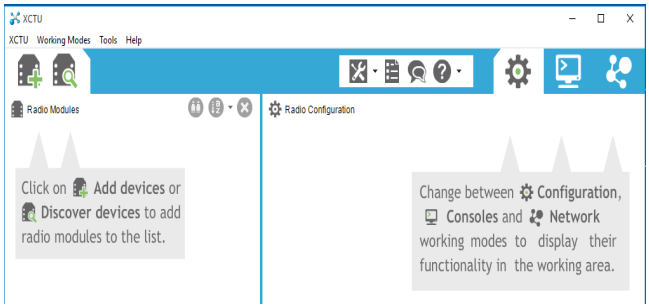


Fig. (6.2). XCTU starting Window.

Step 2: Connect two XBee board at same PC with two serial ports (COMPORT can be different for every PC), here XBee are connected at COM16 & COM8. Fig. (**6.3**) shows window after adding first XBee module at COM16.

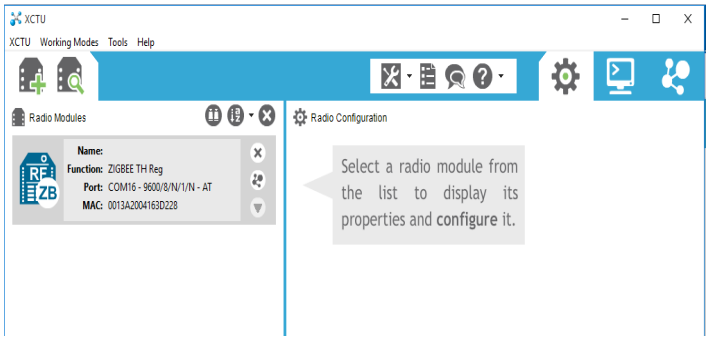


Fig. (6.3). Window after adding first XBee module at COM16.

Fig. (**6.4**) shows Window after adding second XBee module at COM8

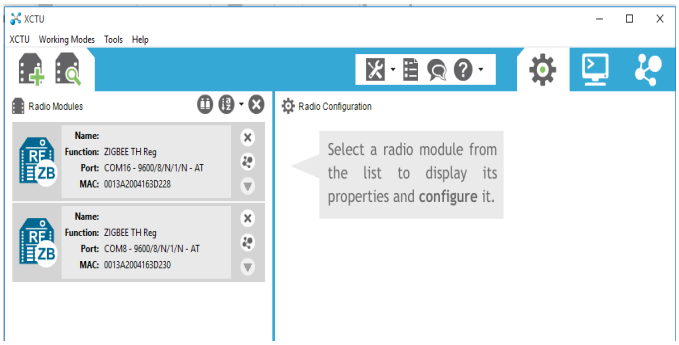


Fig. (6.4). Window after adding second XBee module at COM8.

Step 3: Configure 1st XBee as a coordinator.

Click XBee at COM16, the following settings will open.

Fig. (**6.5**) shows settings window.

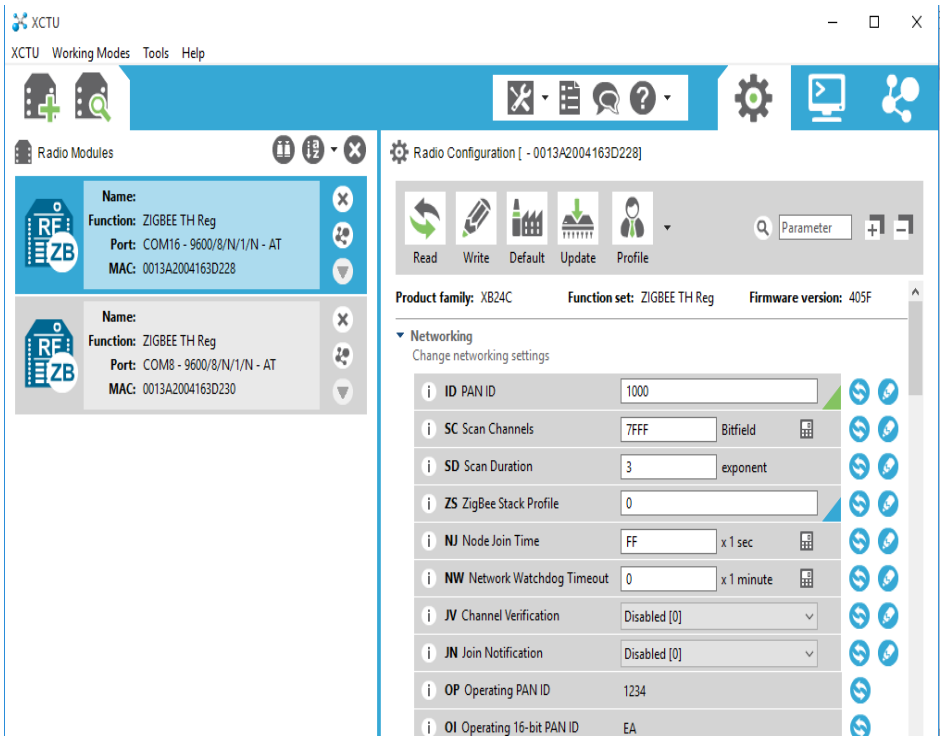


Fig. (6.5). Settings window.

To configure XBee as coordinator, settings are as follows-

PAN ID-1000

CE coordinator Enable=enabled [1]

DL destination address low=FFFF

Fig. (**6.6a**) shows configuring XBee as coordinator and Fig. (**6.6b**) shows snapshot of the setting for coordinator.

CHAPTER 7

MATLAB GUI

Abstract: This chapter describes the steps to design MATLAB GUI. MATLAB is a programming language developed by MathWorks. It is used for, matrix manipulations, plotting of functions and data, implementation of algorithms and user interfaces. MATLAB applications include, signal processing and communications, Image and video processing, control systems, test and measurement, computational finance, computational biology *etc*.

Keywords: GUI, MATLAB.

7.1. GRAPHICAL USER INTERFACE (GUI)

It is a MATLAB tool that enables a user to perform interactive tasks.

GUI manipulates the commands that is given by the end user and responds accordingly. Each control and the GUI have one or more *callbacks* as command.

7.1.1. Steps to Create GUI in MATLAB

Step 1: Open GUI in MATLAB by clicking on icon shown in Fig. (**7.1**).

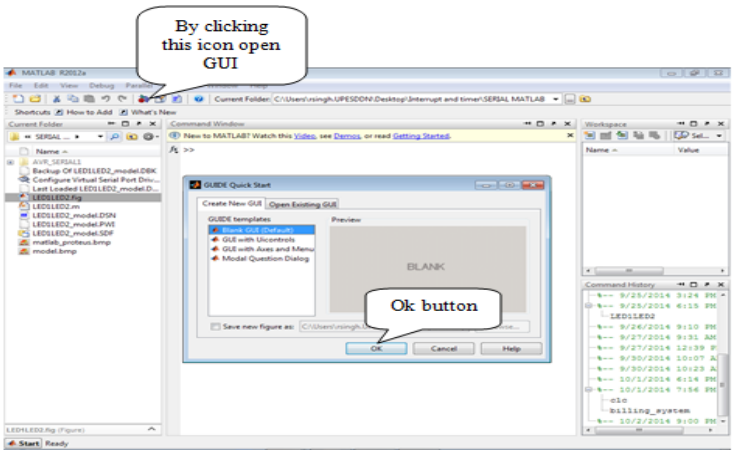


Fig. (7.1). Blank GUI MATLAB window.

Step 2: Click on OK button then the window as shown in Fig. (**7.2**) will be opened.

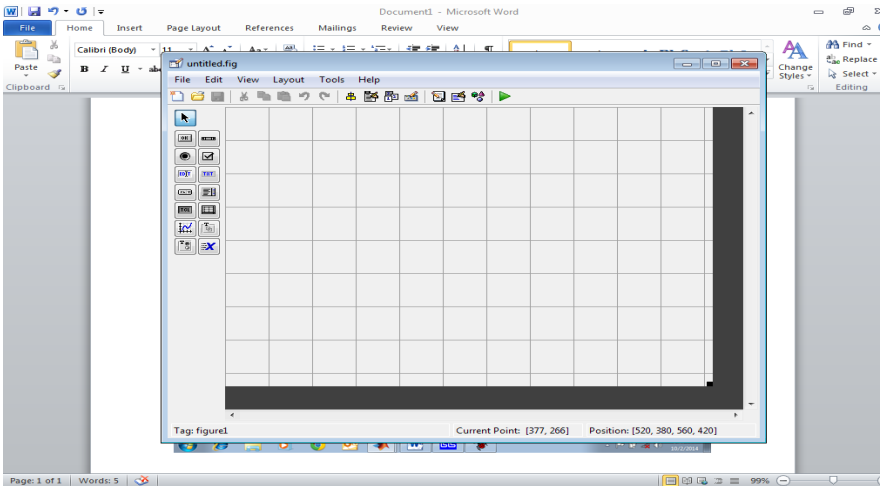


Fig. (7.2). GUI window.

Step 3: Click on the push button to select two push buttons and draw on the GUI window as shown in Fig. (**7.3**).

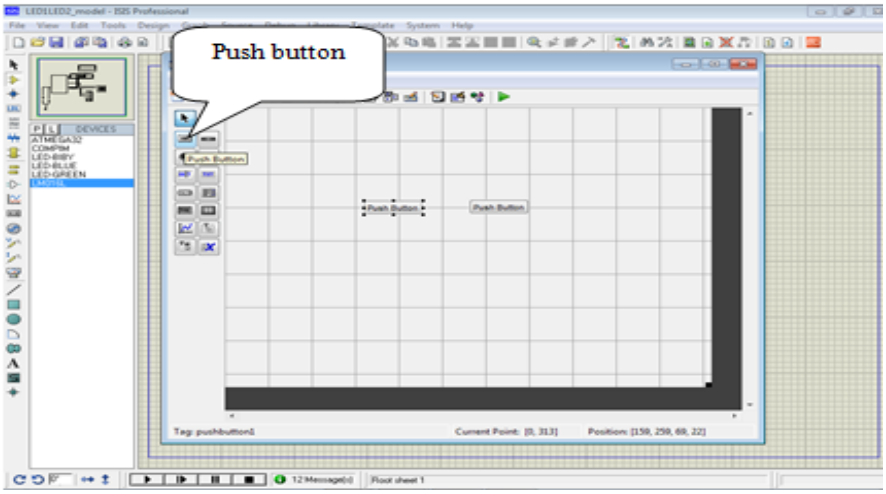


Fig. (7.3). Push buttons on GUI.

Step 4: Double click on push button will open the window as shown in Fig. (**7.4**), where some options like- Font size, color and name can be assigned, then save it and repeat it for another push button. Let's say button name as 'A' and font size-10.

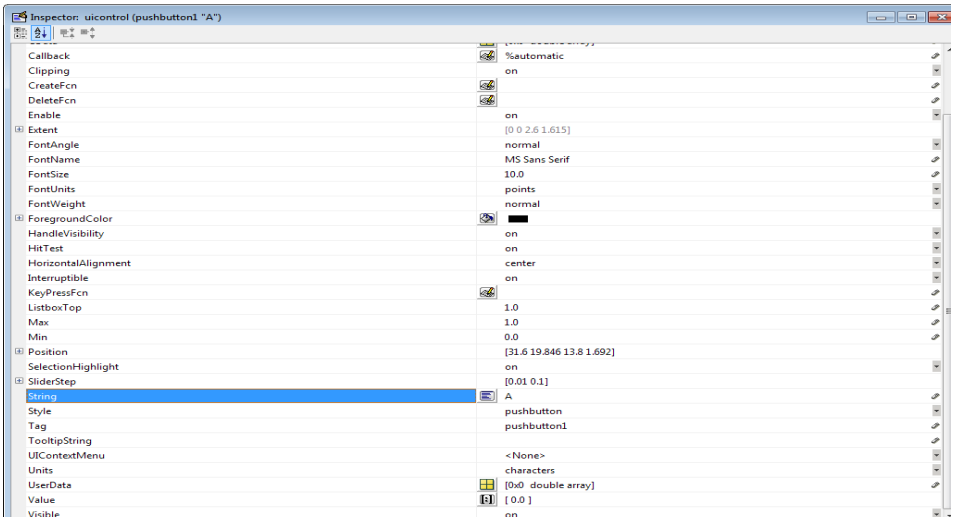


Fig. (7.4). Window to configure the push button.

Step 5: Right click on 'A' button then go to view callbacks to callback as shown in Fig. (**7.5**).

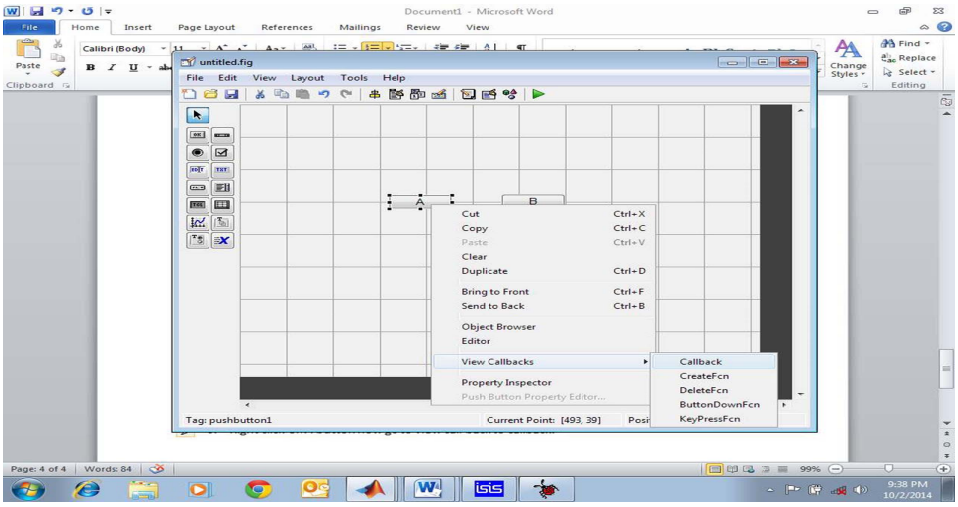


Fig. (7.5). Window to go for callback.

SECTION B: Arduino Interfacing with MATLAB using I/O Package

CHAPTER 8

Simulink and Arduino I/O Package

Abstract: Arduino I/O package from Mathworks provides an interface between Simulink and hardware system. It allows real time communication between Arduino and MATLAB. It needs a simple program to load in Arduino and create Simulink by simple steps. The program acts a server for communicating the information from input devices, Arduino and Simulink model with special blocks and commands.

In this case Arduino will not run separate program to control the devices rather run the executable program from Simulink.

Keywords: Arduino I/O package, Simulink.

8.1. GET STARTED WITH SIMULINK ARDUINO I/O PACKAGE

To get started with Simulink and Arduino I/O package, following components are required.

Component Requirement-

1. Arduino (Mega2560 or UNO).
2. MATLAB 2013 (or above).
3. Installed Arduino package in Simulink.
4. Arduino USB cable (for communication).

8.1.1. Steps to Install Arduino I/O Package in MATLAB

MATLAB Simulink don't come with pre-installed packages for Arduino.

1. Download it from the official website of Mathworks. For MATLAB 2013 and above versions.
2. Run the MATLAB program
3. Go to the *HOME* menu, in menu bar
4. Go to option of ad-Ons in the block of *RESOURCES*
5. 'Click' on 'get hardware support packages'
6. On clicking 'support Package', Installer window will open

This window will have following options-

a. Install from internet.
b. Download from internet.
c. Install from folder.
d. Uninstall.

Install package from where you want to.

7. Support package can be installed by writing target installer in command window.

1. 'Click' *Next* on that Support Package installer window in bottom right.
2. All the available hardware support packages will be shown in Next window.
3. 'Click' on '*Arduino'* and 'click' *Next.*
4. *Sign in to* Mathworks account, if don't have create it with working email address.
5. Login and click next and next and it will take a little time to download and install. Then 'click' finish.
6. Now install this package.

8.1.2. Making Arduino Compatible with MATLAB

Follow the steps as below for making Arduino compatible with MATLAB.

1. Arduino IO package will download by folder name 'ArduinoIO'
2. Connect Arduino board with the computer using its data cable.
3. Open this folder, one folder by name of 'pde' will be there.
4. Open this folder and then open the Arduino program in folder 'ADIOES'.

Fig. (**8.1**) shows the ADIOES program to load to Arduino to work with IO package.

1. Burn this program in the Arduino.
2. Now Arduino is compatible with Simulink and model created in it.

8.1.3. Creating Model in Simulink

To create a Simulink model, go to Simulink menu, 'Click' on the Simulink Library and then open ArduinoIO library. There are blocks available to use according to the requirement of the project. Mainly two types of blocks- read blocks and write blocks.

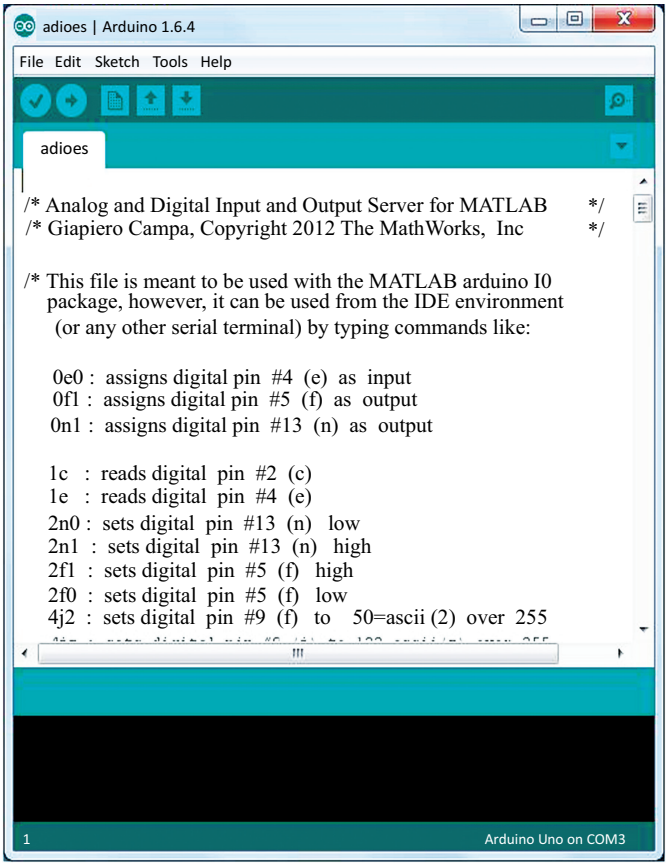


Fig. (8.1). ADIOES program to load to Arduino to work with IO package.

8.2. READ BLOCKS

These blocks takes the information from input pin or communicate the real time data to the Simulink.

8.3. WRITE BLOCKS

These blocks are to communicate Simulink data to the outer world.

Simulink model can be designed by connecting the blocks from library to the blank space provided to create model. Build the Simulink model & run on hardware to communicate with outer world.

CHAPTER 9

Digital Read/Write and Analog Read/Write with Arduino I/O Package

Abstract: This chapter describes about the examples of digital read/write and analog read/write with Arduino I/O package, its designing process and working.

Keywords: Analog read/write, Arduino I/O Package, Arduino, Digital read/write.

9.1. DIGITAL READ/WRITE

To understand the digital read/write with Arduino I/O package, let's take example of making LED 'ON/OFF' with a button. Fig. (**9.1**) shows the block diagram for the system, which comprises of Arduino board, a switch or button, LED (connected to Arduino through resistor).

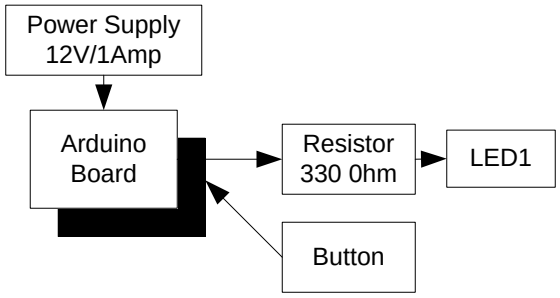


Fig. (9.1). Block diagram for digital read/write.

9.1.1. Circuit Diagram

Connect all the components to Arduino as per the connections as described-

1. Push Button or switch is connected at pin8 of Arduino Uno.
2. LED indicator at connected to pin7 of Arduino Uno through 330 ohm of resistor.
3. DC jack of +12 V power supply is connected to power supply DC jack of Arduino mega.

Fig. (**9.2**) shows circuit diagram for the system.

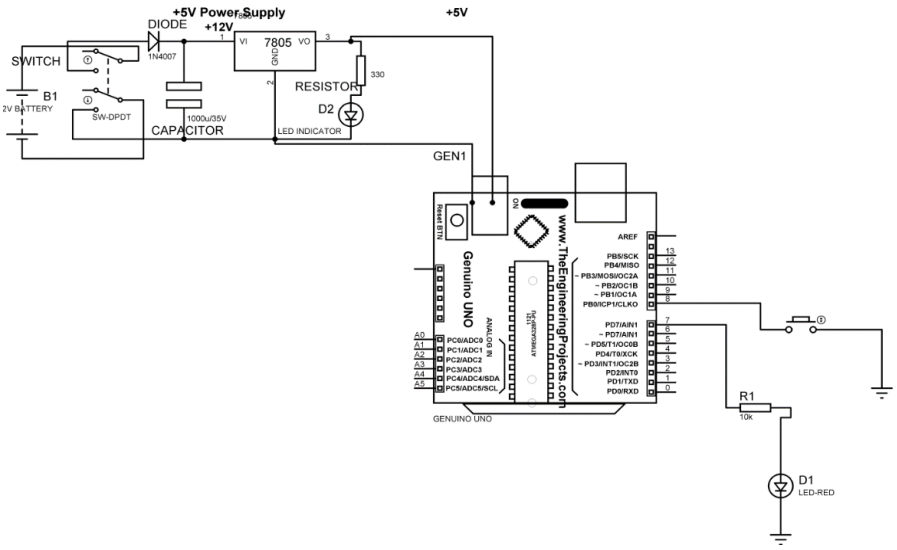


Fig. (9.2). Circuit diagram for digital read/write.

9.1.2. Simulink Model

Connect the required blocks for the system in Simulink space and burn the specified program for ADIOES to Arduino and check the working of the system. Fig. (**9.3**) shows simulink model for digital read/write.

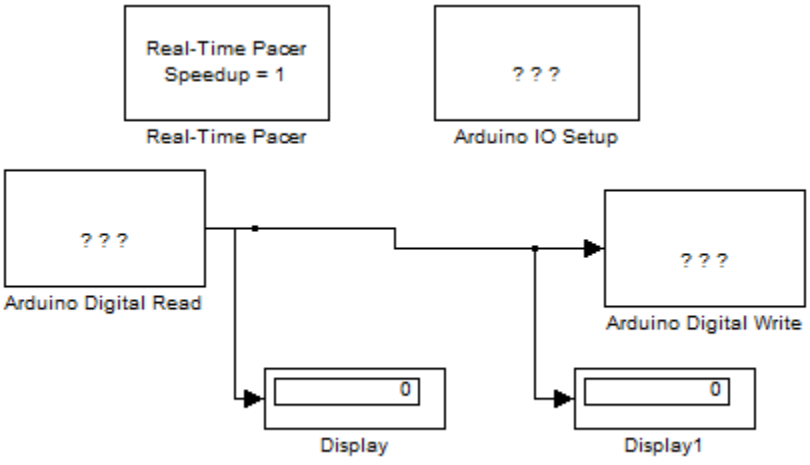


Fig. (9.3). Simulink model for digital read/write.

9.2. ANALOG READ/WRITE

To understand the analog read/write with Arduino I/O package, let's take example of making LED 'ON/OFF' with a POT. Fig. (**9.4**) shows the block diagram for the system, which comprises of Arduino board, a POT, LED (connected to Arduino through resistor).

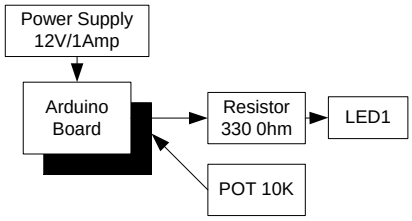


Fig. (9.4). Block diagram for analog read/write.

9.2.1. Circuit Diagram

Connect all the components to Arduino as per the connections as described:

1. Variable terminal of POT is attached at A0 analog pin of Arduino Uno.
2. Fixed two terminals of POT is connected to +5V and GND of power supply extension/explorer.
3. LED indicator at connected to pin 3 of Arduino Uno through 330 ohm of resistor.
4. DC jack of +12 V power supply is connected to power supply DC jack of Arduino mega.

Fig. (**9.5**) shows circuit diagram for analog read/write.

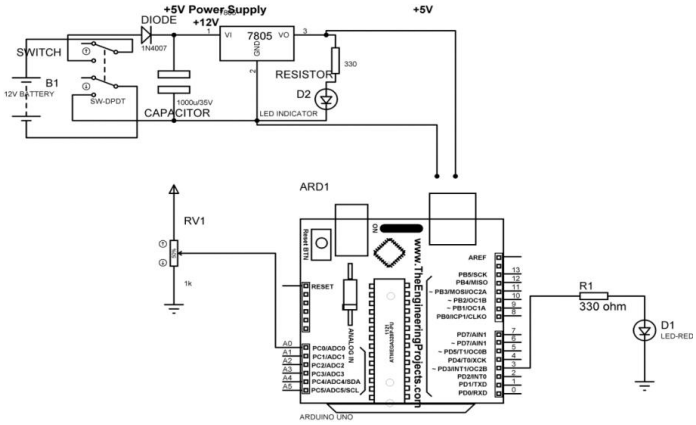


Fig. (9.5). Circuit diagram for analog read/write.

CHAPTER 10

Digital Read with Proximity and Touch Sensor and Digital Write on LED with Arduino I/O Package

Abstract: This chapter describes the process for digital read and digital write with Arduino I/O package, with the help of proximity a & touch sensor and LED.

Keywords: Arduino, Digital read, Digital write.

The system comprises of Arduino board, two LED, proximity sensor, touch sensor and power supply. Fig. (**10.1**) shows the block diagram for the system

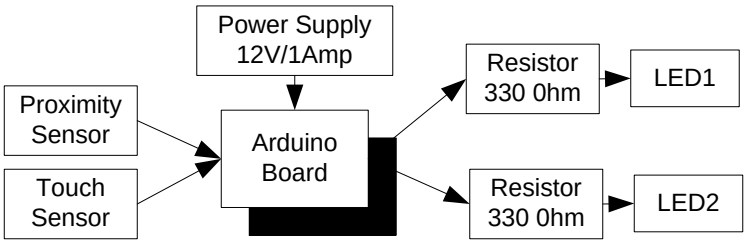


Fig.(10.1). Block diagram of system.

10.1. CIRCUIT DIAGRAM

Connect all the components to Arduino as per the connections as described-

1. Touch sensor OUT pin is connected to 7 pin of Arduino mega.
2. Proximity sensor OUT pin is connected to 6pin of Arduino mega.
3. LED D1 is connected with 8 pin of Arduino mega which is indicator to touch sensor through 330 ohm resistor.
4. LED D2 is connected with 8 pin of Arduino mega which is indicator to proximity sensor through 330 ohm resistor.
5. +Vcc and Ground pin of individual sensors is connected to +5V and GND pin of power Patch/explorer.
6. DC jack of +12 V power supply is connected to power supply DC jack of Arduino mega.

Fig. (**10.2**) shows circuit diagram for the system.

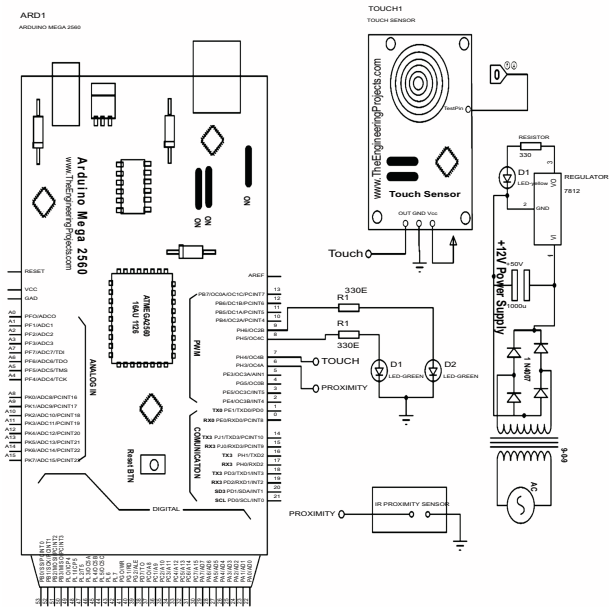


Fig. (10.2). Circuit diagram for the system.

10.2. SIMULINK MODEL

Follow the steps defined in chapter 8 and design the Simulink model. Burn the predefined I/O program in Arduino and see the results on oscilloscope. Figs. (**10.3**) & (**10.4**) shows the Simulink model for the system.

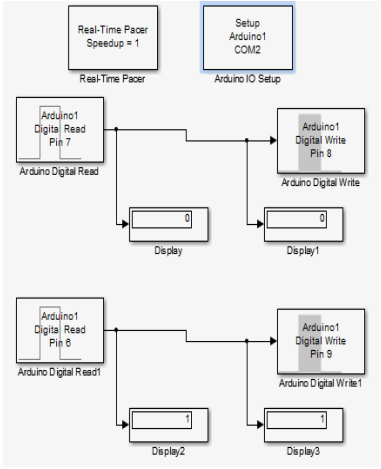


Fig. (10.3). Simulink Model 1.

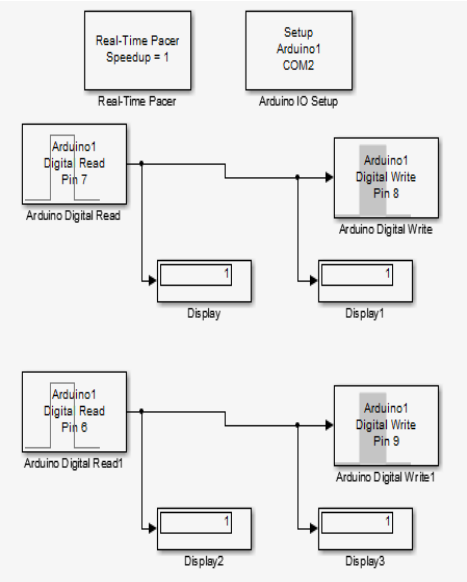


Fig. (10.4). Simulink Model 2.

CHAPTER 11

Key Touch Sensor Based Home Automation with Arduino I/O Package

Abstract: This chapter describes the home automation system with Arduino I/O package and MATLAB Simulink for monitoring the status of the appliances. The system comprises of Arduino board (connected to MATLAB through serial), four home appliances (Bulb1, Fan1, Bulb2, Fan2) which are connected to Arduino through transistor and relay, and power supply. The system is designed to control appliances with touch sensor.

Keywords: Arduino, Bulb, Fan, Home automation, Key touch sensor.

Fig. (**11.1**) shows the block diagram for the system which comprises of Arduino board (connected to MATLAB through serial), four home appliances (Bulb1, Fan1, Bulb2, Fan2) which are connected to Arduino through transistor and relay, and power supply.

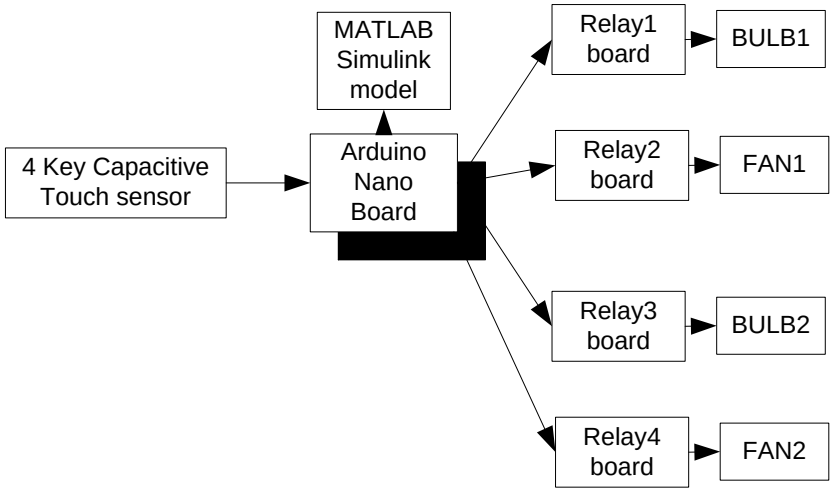


Fig. (11.1). Block diagram for home automation system.

11.1. CIRCUIT DIAGRAM

Connect all the components to Arduino as per the connections as described-

1. 4 key Touch sensor OUT1 pin is connected to 7 pin of Arduino nano.
2. 4 key Touch sensor OUT2 pin is connected to 6 pin of Arduino nano.
3. 4 key Touch sensor OUT3 pin is connected to 5 pin of Arduino nano.
4. 4 key Touch sensor OUT4 pin is connected to 4 pin of Arduino nano.
5. First appliance is connected with 12 pin of Arduino nano which is indicator to touch sensor through 330 ohm resistor.
6. Second appliance is connected with 11 pin of Arduino nano.
7. third appliance is connected with 10 pin of Arduino nano.
8. forth appliance is connected with 9 pin of Arduino nano.
9. +Vcc and Ground pin of sensor is connected to +5V and GND pin of power Patch/explorer.
10. DC jack of +12 V power supply is connected to power supply DC jack of Arduino nano.

Fig. (**11.2**) shows circuit diagram for home automation system.

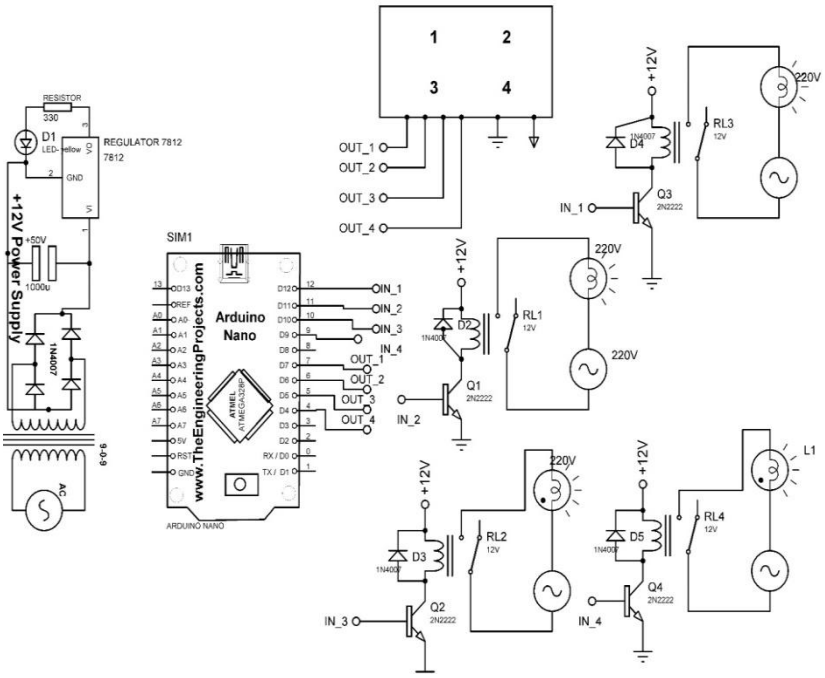


Fig. (11.2). Circuit diagram for home automation system.

11.2. SIMULINK MODEL

Connect the required blocks as discussed in chapter 7 & 8 for the system in Simulink space and burn the specified program for ADIOES to Arduino and check the working of the system. Fig. (**11.3**) shows the simulink model for the system.

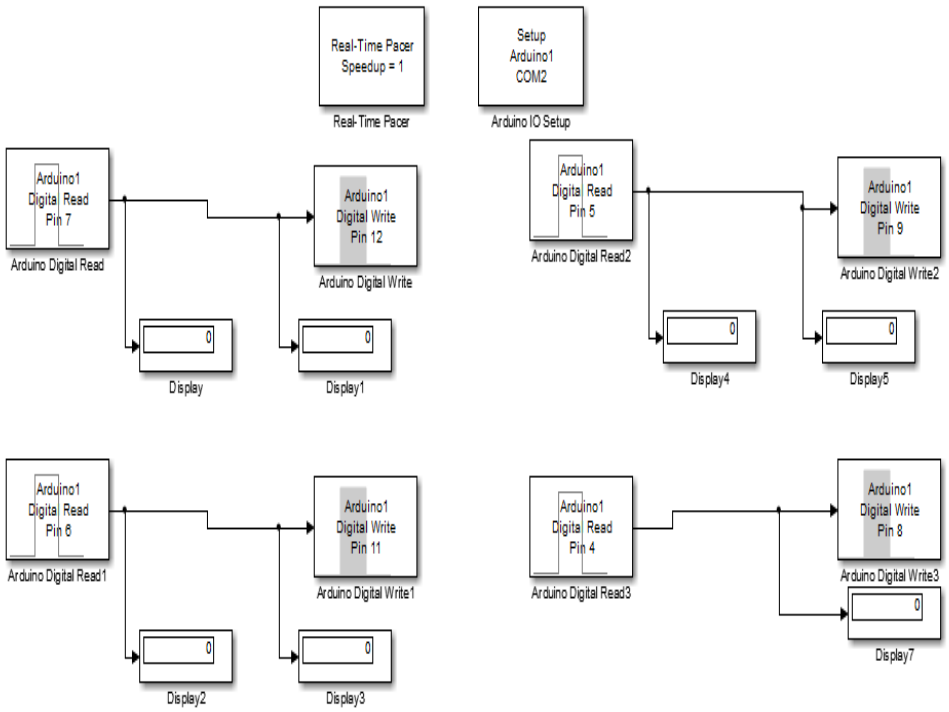


Fig. (11.3). Simulink Model.

CHAPTER 12

Sun Tracker System Using LDR with Arduino I/O Package

Abstract: A sun tracker is a device that orients PV panel towards the sun, to maximize the energy efficiency of the system. Sun trackers are designed to enhance the power generation capacity of solar panels. In this chapter Arduino based sun tracker system is designed. Sun intensity is measured with the help of light dependent resistor (LDR), placed at PV panels. The system comprises of the Arduino board, three LDR and servo motor. Servo motor is connected to orient the PV panels in such a way that PV panel is always perpendicular to the sun rays to generate maximum energy.

Keywords: Arduino, LDR, PV panel, Sun tracker system.

Fig. (**12.1**) shows the block diagram of the system. It comprises of the Arduino board, three LDR and servo motor.

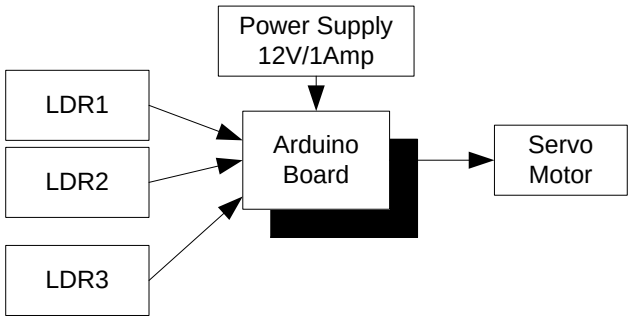


Fig. (12.1). Block diagram for the sun tracking system.

12.1. CIRCUIT DIAGRAM

Connect all the components to Arduino as per the connections as described-

1. LDR sensor1 analog out pin is connected to A0 pin of Arduino Uno.
2. LDR sensor2 analog out pin is connected to A1 pin of Arduino Uno.
3. LDR sensor3 analog out pin is connected to A2 pin of Arduino Uno.

4. +Vcc and Ground pin of individual sensors is connected to +5V and GND pin of power Patch/explorer.
5. DC jack of +12 V power supply is connected to power supply DC jack of Arduino mega.
6. Servo motor PWM in pin is connected to pin 3 of Arduino Uno.

Fig. (**12.2**) shows circuit diagram for the sun tracking system.

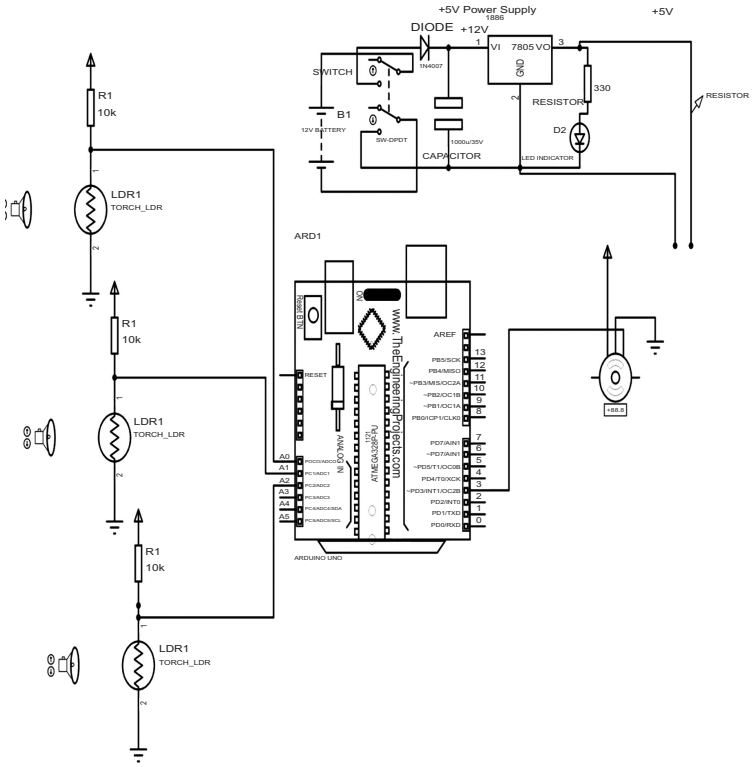


Fig. (12.2). Circuit diagram for the sun tracking system.

12.2. SIMULINK MODEL

Connect the required blocks for the system in Simulink space and burn the specified program for ADIOES to Arduino and check the working of the system. Fig. (**12.3**) shows simulink model for the system.

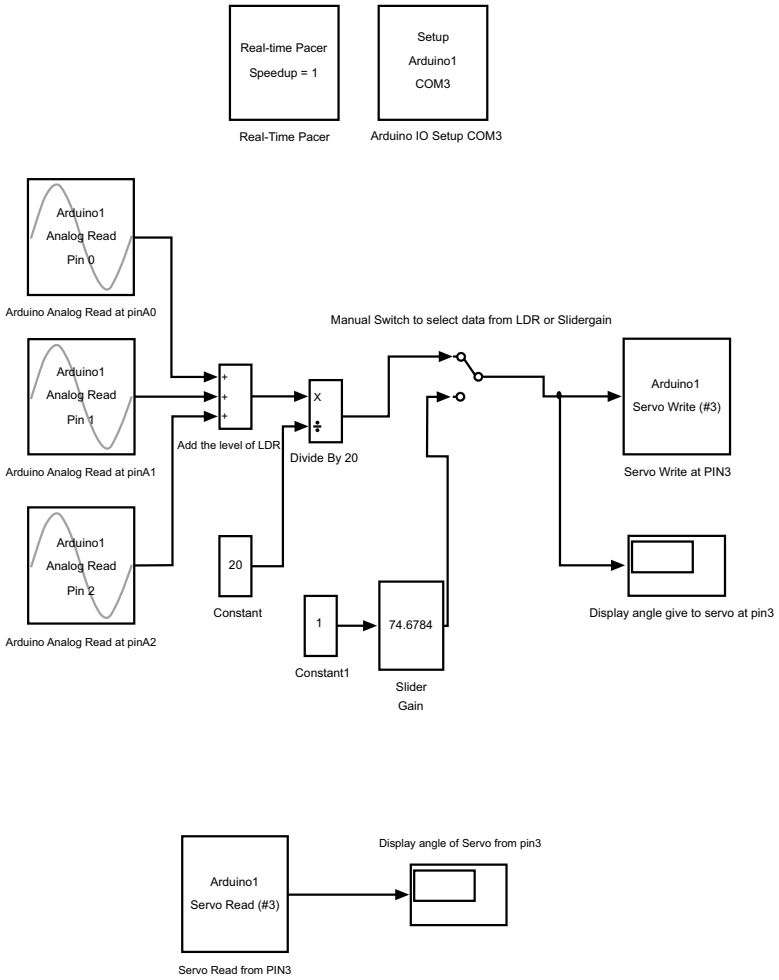


Fig. (12.3). Simulink Model for the system.

CHAPTER 13

Robot Control and Sensor Data Acquisition System with Arduino I/O Package

Abstract: This chapter describes the control of robot and sensory data acquisition system with Arduino I/O package. The system comprises of Arduino board, LM35, POT, L293D, DC motors, power supply. DC motors are used to move the robot in 'forward' 'reverse' 'left' and 'right'.

Keywords: Arduino, Arduino I/O Package, Data Acquisition, Robot Control.

Fig. (**13.1**) shows the block diagram of the system, comprises of Arduino board, LM35, POT, L293D, DC motors, power supply.

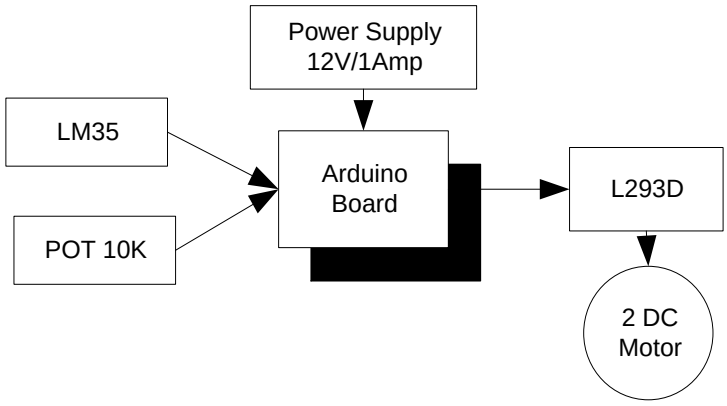


Fig. (13.1). Block diagram for the system.

13.1. CIRCUIT DIAGRAM

Connect all the components to Arduino as per the connections as described-

1. Temperature sensor analog out pin is connected to A0 pin of Arduino Uno.
2. 10K POT variable out pin is connected to A1 pin of Arduino Uno.
3. +Vcc and Ground pin of individual sensors is connected to +5V and GND pin of power Patch/explorer.

4. DC jack of +12 V power supply is connected to power supply DC jack of Arduino mega.
5. Input pins 2 and 7 of L293D IC are connected to 10 and 9 pins of Arduino Uno to supply the input.
6. Output pins 3 and 8 of L293D IC are connected to +ve and –ve terminals of DC motor.
7. +12V and ground of power supply is connected to H bridge Collector and Emitter.
8. LEDs are also connected parallel to inputs of H -bridge.
9. +12V DC jack of power supply is connected to DC jack of Arduino Uno.
10. Pins 1,9 and 16 of L293D are connected to +5V.
11. Pins 4,5 and 12 ,13 are connected to ground.

Fig. (**13.2**) shows circuit diagram for the system.

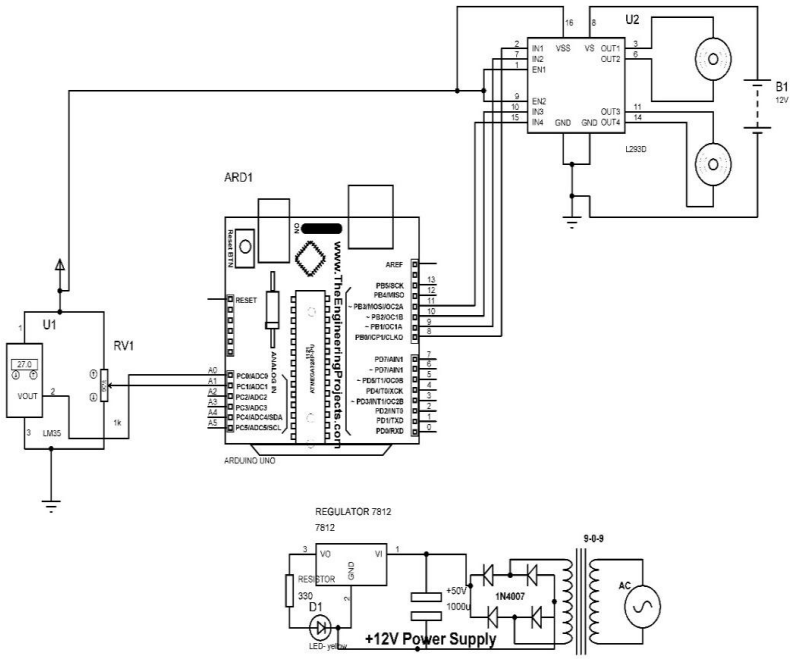


Fig. (13.2). Circuit diagram for the system.

13.2. MATLAB GUI

Create MATLAB GUI as described in chapter 7 and write program and functions to access data on created GUI. Burn the code for ADIOES to Arduino.

Program

```
function varargout = READ_WRITE_ADIOES(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton', gui_Singleton, ...
                   'gui_OpeningFcn', @READ_WRITE_ADIOES_OpeningFcn, ...
                   'gui_OutputFcn', @READ_WRITE_ADIOES_OutputFcn, ...
                   'gui_LayoutFcn', [] , ...
                   'gui_Callback', []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
function READ_WRITE_ADIOES_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;

guidata(hObject, handles);

delete(instrfind({'Port'},{'COM2'}))
clear a;
```

CHAPTER 14

Two Analog Sensors [POT and LM35] Interfacing with Arduino I/O Package

Abstract: This chapter describes two analog sensors interfacing and data acquisition system with MATLAB. The system, comprises of Arduino board, LM35, POT, LED and power supply. The system is designed to receive two analog sensor data and control the LED.

Keywords : Analog sensor, Arduino, Arduino I/O package, LM35, POT.

Fig. (**14.1**) shows the block diagram of the system, comprises of Arduino board, LM35, POT, LED and power supply.

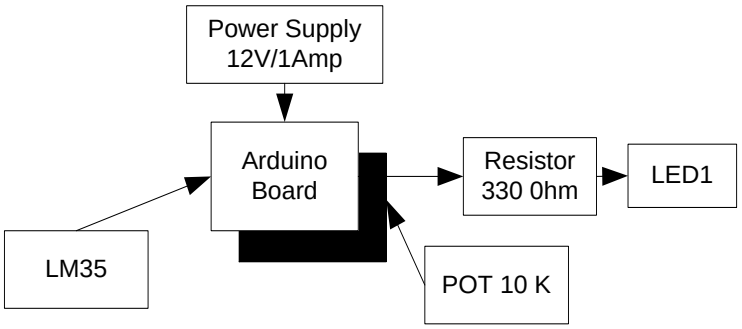


Fig. (14.1). Block diagram for interfacing of two analog sensors with Arduino with I/O package.

14.1. CIRCUIT DIAGRAM

Connect all the components to Arduino as per the connections as described-

1. Temperature sensor analog out pin is connected to A0 pin of Arduino Uno.
2. Temperature sensor analog out pin is connected to A1 pin of Arduino Uno.
3. LED D1 is connected with 8 pin of Arduino mega which is indicator through 330 ohm resistor.
4. +Vcc and Ground pin of individual sensors is connected to +5V and GND pin of power Patch/explorer.

5. DC jack of +12 V power supply is connected to power supply DC jack of Arduino mega.

Fig. (**14.2**) shows circuit diagram for interfacing two analog sensors with Arduino with I/O package.

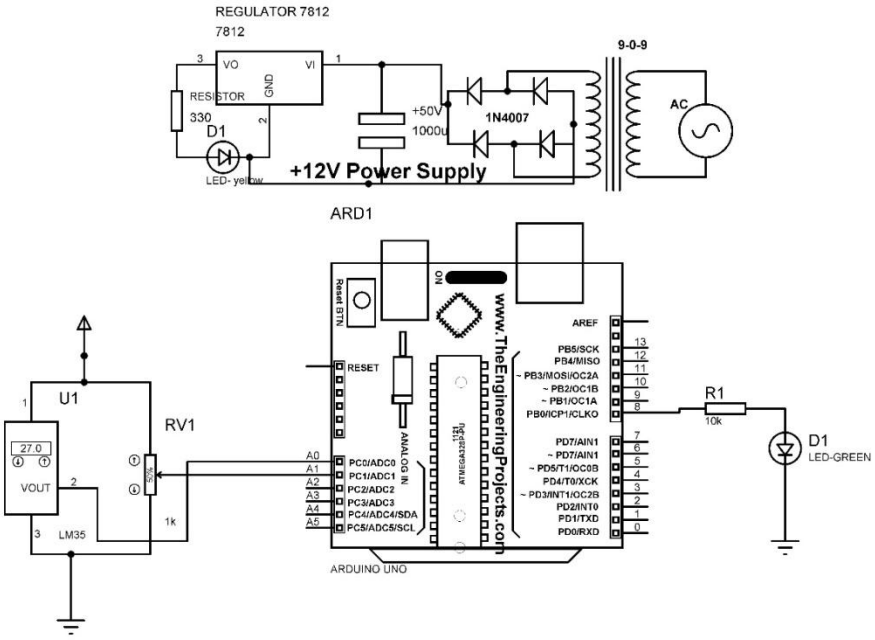


Fig. (14.2). Circuit diagram for interfacing of two analog sensors with Arduino with I/O package.

14.2. MATLAB GUI

Write program for MATLAB and develop GUI by following the steps in chapter 7.

Program

```
function varargout = Example1(varargin)

gui_Singleton = 1;

gui_State = struct('gui_Name', mfilename, ...

                   'gui_Singleton', gui_Singleton, ...

                   'gui_OpeningFcn', @Example1_OpeningFcn, ...
```

```
                'gui_OutputFcn', @Example1_OutputFcn, ...
                'gui_LayoutFcn', [] , ...
                'gui_Callback', []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
function Example1_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;

guidata(hObject, handles);
delete(instrfind({'Port'},{'COM2'}))
clear a;
global a;
global stop;
stop='e';
global entry;
entry=1;
global time;
time=0;
```

CHAPTER 15

Three Sensors Data Acquisition and Feedback System with MATLAB and Arduino I/O Package

Abstract: This chapter explores the data acquisition and feedback system for three sensors with MATLAB and Arduino I/O package. The system comprises of Arduino board (connected to MATLAB GUI through serial), three sensors (LM35 (temperature), two POT (10K)) which are connected to Arduino, LED(output device) and power supply. The system is designed to receive the sensory data and control the LED with blocks created on MATLAB GUI.

Keywords: Arduino, Arduino GUI, LED, MATLAB GUI, POT, Temperature Sensor.

Fig. (**15.1**) shows the block diagram for the system which comprises of Arduino board (connected to MATLAB GUI through serial), three sensors (LM35 (temperature), two POT (10K)) which are connected to Arduino, LED(output device) and power supply.

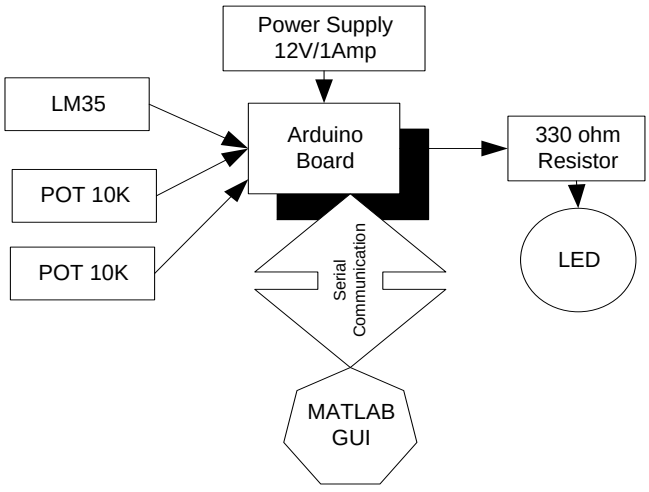


Fig. (15.1). Block diagram for interfacing of three sensors with Arduino with I/O package.

Rajesh Singh, Anita Gehlot, Bhupendra Singh & Sushabhan Choudhury

15.1. CIRCUIT DIAGRAM

Connect all the components to Arduino as per the connections as described-

1. Temperature sensor analog out pin is connected to A0 pin of Arduino Uno.
2. 10K POT variable out pin is connected to A1 pin of Arduino Uno.
3. 10K POT variable out pin is connected to A2 pin of Arduino Uno.
4. +Vcc and Ground pin of individual sensors is connected to +5V and GND pin of power Patch/explorer.
5. DC jack of +12 V power supply is connected to power supply DC jack of Arduino Uno.
6. LED is connected to pin 8 of Arduino Uno.

Fig. (**15.2**) shows Circuit diagram for interfacing of three sensors with Arduino with I/O package.

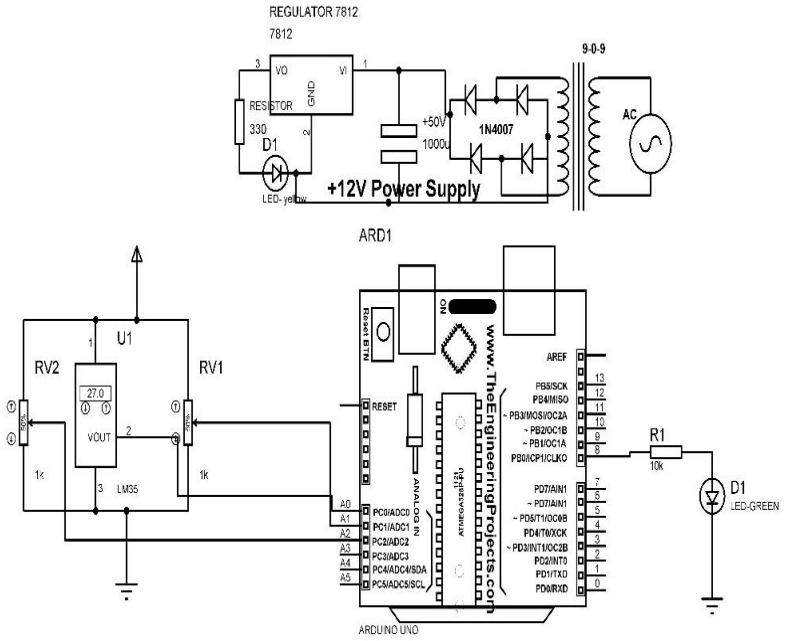


Fig. (15.2). Circuit diagram for interfacing of three sensors with Arduino with I/O package.

15.2. MATLAB GUI

Write program for MATLAB and develop GUI by following the steps in chapter 7.

Program

```
function varargout = channelread(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
                   'gui_Singleton', gui_Singleton, ...
                   'gui_OpeningFcn', @channelread_OpeningFcn, ...
                   'gui_OutputFcn', @channelread_OutputFcn, ...
                   'gui_LayoutFcn', [] , ...
                   'gui_Callback', []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
function channelread_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
delete(instrfind({'Port'},{'COM2'}))
clear a;
global a;
global stop;
```

SECTION C: Arduino Interfacing with MATLAB without I/O Package

CHAPTER 16

Building Automation System

Abstract: This chapter describes the building automaton system with MATLAB. The system comprises of Arduino board (connected to MATLAB GUI through serial), four home appliances Bulb1, Fan1, Bulb2, Fan2, relay which are connected to Arduino through transistor and relay, and power supply. The system is designed to control appliances with blocks created on MATLAB GUI.

Keywords: Arduino, Bulb, Fan, GUI, MATLAB.

Fig. (**16.1**) show the block diagram for the system which comprises of Arduino board (connected to MATLAB GUI through serial), four home appliances (Bulb1, Fan1, Bulb2, Fan2, relay which are connected to Arduino through transistor and relay, and power supply.

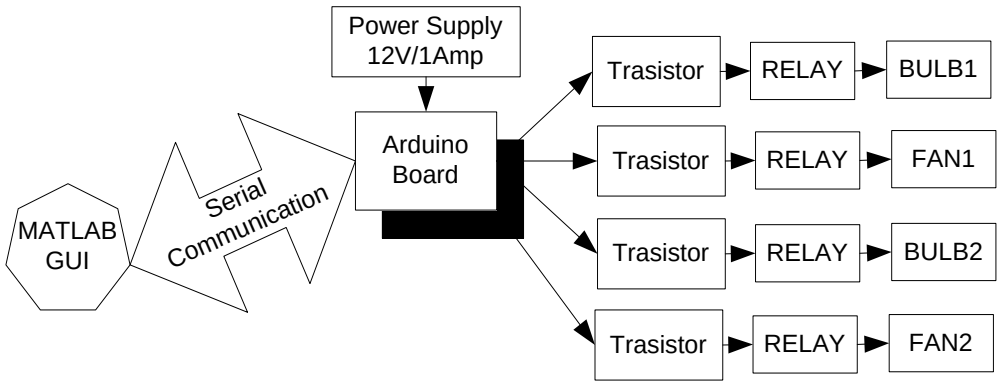


Fig. (16.1). Block diagram for building automation system.

16.1. CIRCUIT DIAGRAM

Connect all the components to Arduino as per the connections as described-

1. Pins 5,4,3 and 2 of Arduino are connected to pin IN_1, IN_2, IN_3 and IN_4 of RELAY board respectively.
2. Pin12 of Arduino is connected to pin RS of LCD.
3. Pin11 of Arduino is connected to pin E of LCD.

4. Pin10 of Arduino is connected to pin D4 of LCD.
5. Pin9 of Arduino is connected to pin D5 of LCD.
6. Pin8 of Arduino is connected to pin D6 of LCD.
7. Pin7 of Arduino is connected to pin D7 of LCD.
8. GND of Arduino is connected to pin RW of LCD
9. +12V DC jack of power supply is connected to DC jack of Arduino.

Fig. (**16.2**) shows circuit diagram for building automation system.

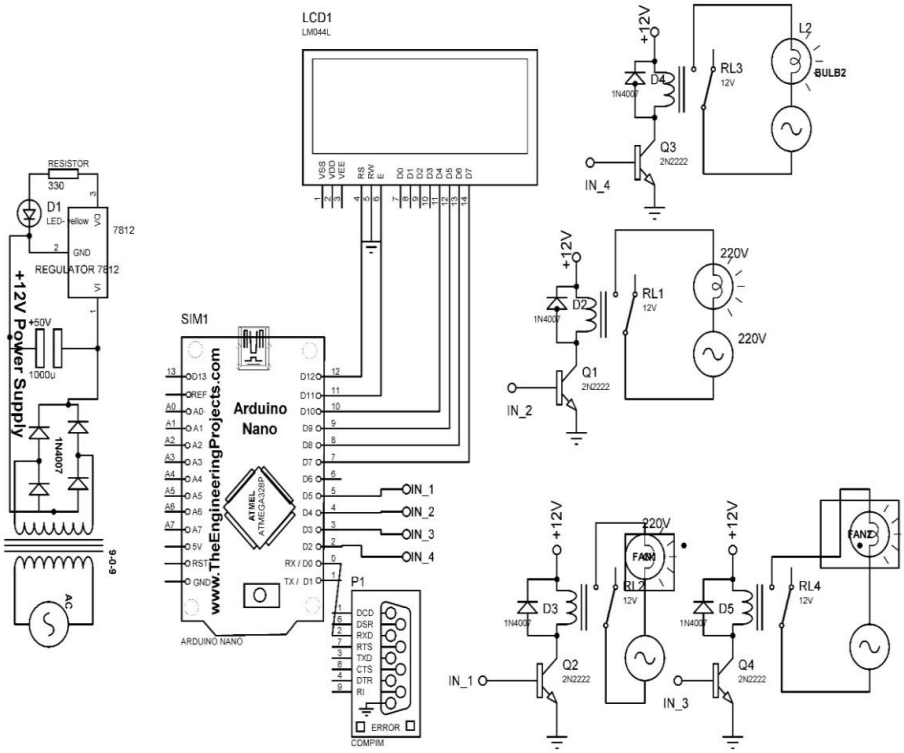


Fig. (16.2). Circuit diagram for building automation system.

16.2. PROGRAM

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 10, 9, 8, 7);


const int M1_FAN1=5;

const int M1_BULB1=4;
```

```
const int M2_FAN2=3;
const int M2_BULB2=2;
int SERIAL_VAL;

void setup()
        {
        Serial.begin(9600);
        lcd.begin(20,4);
        pinMode(M1_FAN1, OUTPUT);
        pinMode(M1_BULB1, OUTPUT);
        pinMode(M2_FAN2, OUTPUT);
        pinMode(M2_BULB2, OUTPUT);
        lcd.setCursor(0,0);
        lcd.print("ROBOT");
        lcd.setCursor(0,1);
        lcd.print("CONTROL");
        delay(2000);
        }

void loop()
        {
if(Serial.available()>0)
        {
           SERIAL_VAL=Serial.read();

        if (SERIAL_VAL == 100)
```

CHAPTER 17

Robot Control with MATLAB GUI

Abstract: This chapter describes the robot control with MATLAB. The system comprises of Arduino board (connected to MATLAB GUI through serial), four robot control (motor1, motor2) which are connected to Arduino through L293D, and power supply. The system is designed to control robot with blocks created on MATLAB GUI.

Keywords: Arduino, GUI, L293D, MATLAB, Motor, Robot.

Fig. (**17.1**) show the block diagram for the system which comprises of Arduino board (connected to MATLAB GUI through serial), four robot control (motor1, motor2) which are connected to Arduino through L293D, and power supply.

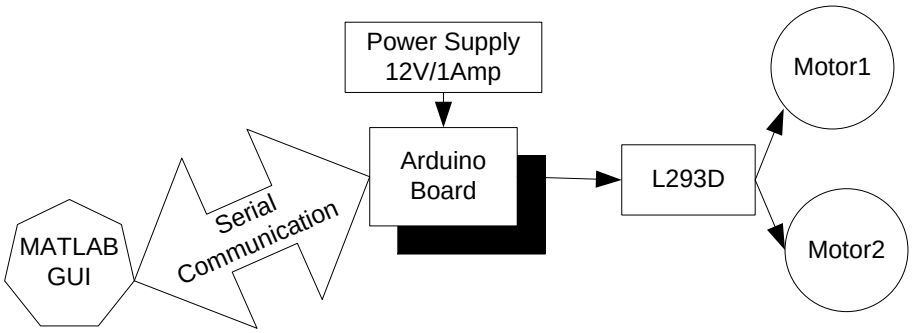


Fig. (17.1). Block diagram for robot control with MATLAB GUI.

17.1. CIRCUIT DIAGRAM

Connect all the components to Arduino as per the connections as described-

1. Pins 2,7,10 and 15 of L293Dare connected to pin 5,4,3,2 of Arduino Nano respectively.
2. Output of L293D 3,6,11 and 14 are connected to pin +ve and –ve of Motor1 and +ve and –ve of motor 2 respectively.
3. Pin12 of Arduino is connected to pin RS of LCD.
4. Pin11 of Arduino is connected to pin E of LCD.

5. Pin10 of Arduino is connected to pin D4 of LCD.
6. Pin9 of Arduino is connected to pin D5 of LCD.
7. Pin8 of Arduino is connected to pin D6 of LCD.
8. Pin7 of Arduino is connected to pin D7 of LCD.
9. GND of Arduino is connected to pin RW of LCD
10. +12V DC jack of power supply is connected to DC jack of Arduino.

Fig. (**17.2**) shows circuit diagram for robot control with MATLAB GUI.

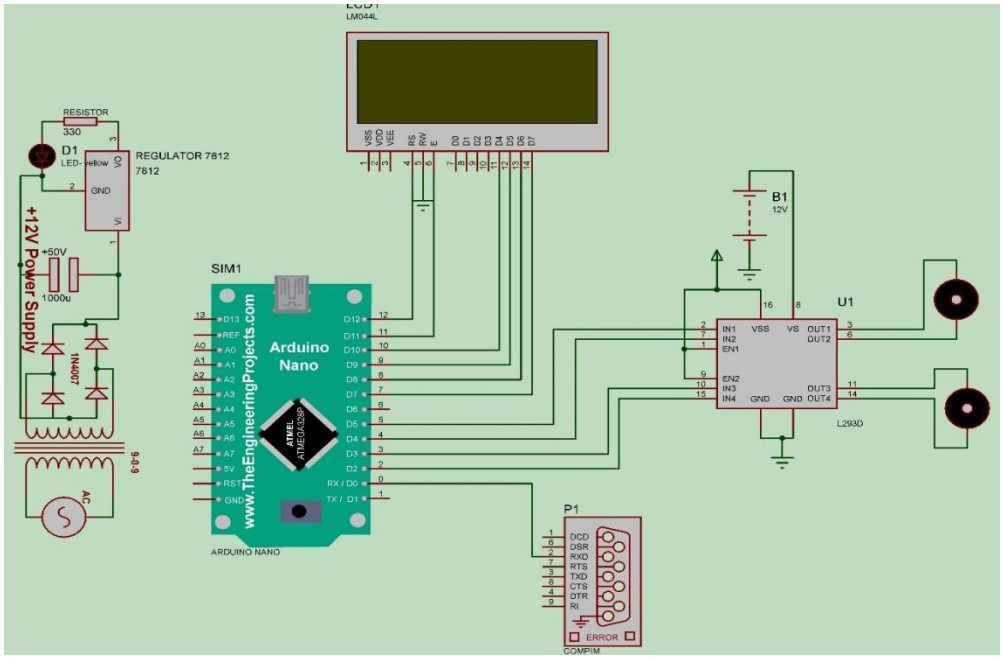


Fig. (17.2). Circuit diagram for robot control with MATLAB GUI.

17.2. PROGRAM

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 10, 9, 8, 7);


const int M1_POS=5;

const int M1_NEG=4;

const int M2_POS=3;

const int M2_NEG=2;
```

```
int SERIAL_VAL;

void setup()
        {
        Serial.begin(9600);
        lcd.begin(20,4);
        pinMode(M1_POS, OUTPUT);
        pinMode(M1_NEG, OUTPUT);
        pinMode(M2_POS, OUTPUT);
        pinMode(M2_NEG, OUTPUT);
        lcd.setCursor(0,0);
        lcd.print("ROBOT");
        lcd.setCursor(0,1);
        lcd.print("CONTROL");
        delay(2000);
        }

void loop()
        {
        if(Serial.available()>0)
        {
          SERIAL_VAL=Serial.read();

         if (SERIAL_VAL == 100)
          {
          lcd.clear();
```

CHAPTER 18

One Analog Channel and Digital Write Using MATLAB GUI

Abstract: This chapter describes how to read one analog channel with MATLAB and output as digital write for controlling the device connected go it. The system comprises of Arduino board which is connected to MATLAB GUI through serial, POT, LED and power supply. The system will sense the change in POT and processed through MATLAB and send output to digital pin.

Keywords: Analog channel, Arduino, Digital write, GUI, MATLAB.

Fig. (**18.1**) shows the block diagram for the system, comprises of Arduino board which is connected to MATLAB GUI through serial, POT, LED and power supply.

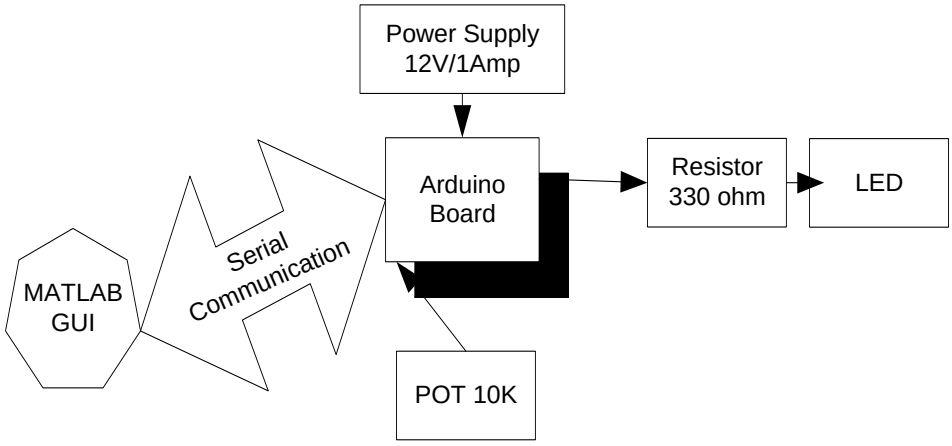


Fig. (18.1). Block diagram for one analog channel and digital write using MATLAB GUI.

18.1. CIRCUIT DIAGRAM

Connect all the components to Arduino as per the connections as described-

1. Variable terminal of POT of 10K is connected to A0 pin Arduino.

2. Fixed terminals of POT is connected to +5V and GND of Power supply patch.
3. LED is connected to pin4 of Arduino through 330-ohm resistor.
4. +12V DC jack of power supply is connected to DC jack of Arduino.

Fig. (**18.2**) shows circuit diagram for one analog channel and digital write using MATLAB GUI.

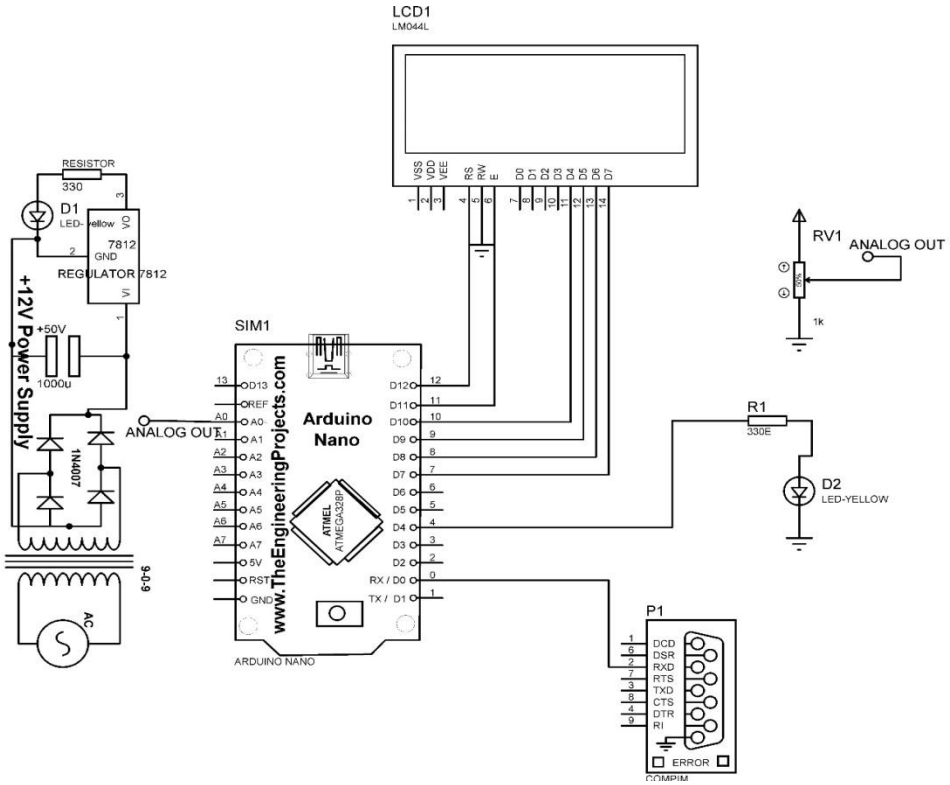


Fig. (18.2). Circuit diagram for one analog channel and digital write using MATLAB GUI.

18.2. PROGRAM

```
const int led_pin_OUTPUT=13;

int SERIAL_READ_DATA;

int adc_POT=0;

void setup()

{

Serial.begin(9600);
```

```
pinMode(led, OUTPUT);
}

void loop()
{
//// read analog sensor from arduino
adc_POT = analogRead(A0);
Serial.println(adc);
delay(200);

///// write to arduino
if(Serial.available()>0)
{
SERIAL_READ_DATA=Serial.read();

if (SERIAL_READ_DATA == 100)
{
digitalWrite(led_pin_OUTPUT, HIGH);
delay(10);
}
if(SERIAL_READ_DATA == 101)
{
digitalWrite(led_pin_OUTPUT, LOW);
delay(10);
}
```

BIBLIOGRAPHY

[1] https://in.mathworks.com/help/matlab/creating_guis/about-the-simple-guide-gui-example.html?requestedDomain=true

[2] https://www.arduino.cc/en/Main/Software

[3] https://www.arduino.cc/en/Guide/Windows

[4] http://ctms.engin.umich.edu/CTMS/index.php?aux=Activities_IOpack

[5] R. Singh, A. Gehlot, B. Singh, and S. Choudhury, *"Arduino-Based Embedded Systems: Interfacing, Simulation, and LabVIEW GUI"- CRC Press.,* Taylor & Francis, . ISBN 9781138060784

[6] R. Singh, A. Gehlot, S. Choudhury, and B. Singh, *Embedded System based on Atmega Microcontroller- Simulation, Interfacing and Projects.,* Narosa Publishing House, . ISBN: 978-8--8487-5720

[7] P.S. Pandey, R. Singh, A. Gehlot, B. Singh, and S. Choudhury, *Power Line Carrier Communication and Arduino based Automation.,* RI Publication, . ISBN: 9780387374010

SUBJECT INDEX

A

B

C

D

E

F

G

H

I

R

S

T

U

V

W

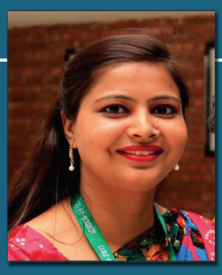
X

Z

RAJESH SINGH

Dr. Rajesh Singh is currently associated with Lovely Professional University as Professor with more than fifteen years of experience in academics. He has been awarded as gold medalist in M.Tech and honors in his B.E. His area of expertise includes embedded systems, robotics, wireless sensor networks and Internet of Things. He has organized and conducted a number of workshops, summer internships and expert lectures for students as well as faculty. He has twenty three patents in his account. He has published around hundred research papers in referred journals/conferences.

Under his mentorship students have participated in national/international competitions including Texas competition in Delhi and Laureate award of excellence in robotics engineering in Spain. Twice in last four years he has been awarded with "certificate of appreciation" and "Best Researcher award- 2017" from University of Petroleum and Energy Studies for exemplary work. He got "certificate of appreciation" for mentoring the projects submitted to Texas Instruments Innovation challenge India design contest, from Texas Instruments, in 2015.He has been honored with young investigator award at the International Conference on Science and Information in 2012. He has published ten books in the area of Embedded Systems and Internet of Things with reputed publishers like CRC/Taylor & Francis, Narosa, GBS, IRP, NIPA and RI publication. He is editor to a special issue published by AISC book series, Springer with title "Intelligent Communication, Control and Devices"-2017 & 2018.



ANITA GEHLOT

Dr. Anita Gehlot is associated with Lovely Professional University as Associate Professor with more than ten years of experience in academics. She has twenty patents in her account. She has published more than fifty research papers in referred journals and conference. She has organized a number of workshops, summer internships and expert lectures for students. She has been awarded with "certificate of appreciation" from University of Petroleum and Energy Studies for exemplary work. She has published ten books in the area of Embedded Systems and Internet of Things with reputed publishers like CRC/Taylor & Francis, Narosa, GBS, IRP, NIPA and RI publication. She is editor to a special issue published by AISC book series, Springer with title "Intelligent Communication, Control and Devices-2018".

BHUPENDRA SINGH

Bhupendra Singh is Managing director of Schematics Microelectronics and provides Product design and R&D support to industries and Universities. He has completed BCA, PGDCA, M.Sc. (CS), M.Tech and has more than eleven years of experience in the field of Computer Networking and Embedded systems. He has published ten books in the area of Embedded Systems and Internet of Things with reputed publishers like CRC/Taylor & Francis, Narosa, GBS, IRP, NIPA and RI publication.

SUSHABHAN CHOUDHURY

Dr. S.Choudhury is the Head of the Department of Electronics, Instrumentation, and Control in the University of Petroleum and Energy Studies. He has teaching experience of 26 years and he has completed his Ph.D from the University of Petroleum and Energy Studies, M.Tech (Gold Medalist) from Tezpur Central University, Tezpur, India and received his B.E. degree from NIT, Silchar University, India. He has published more than 70 papers in various national/international conferences/journals. He has filed ten patents. His area of interest is Zigbee-based networks. He has been selected as the outstanding scientist of the twenty-first century by the Cambridge Biographical Centre, UK. He has also been selected in the who's who of the world in science by Marquis Who's Who, USA. He has published three books in the area of Embedded Systems with reputed publishers like CRC/Taylor & Francis, Narosa and RI publication.