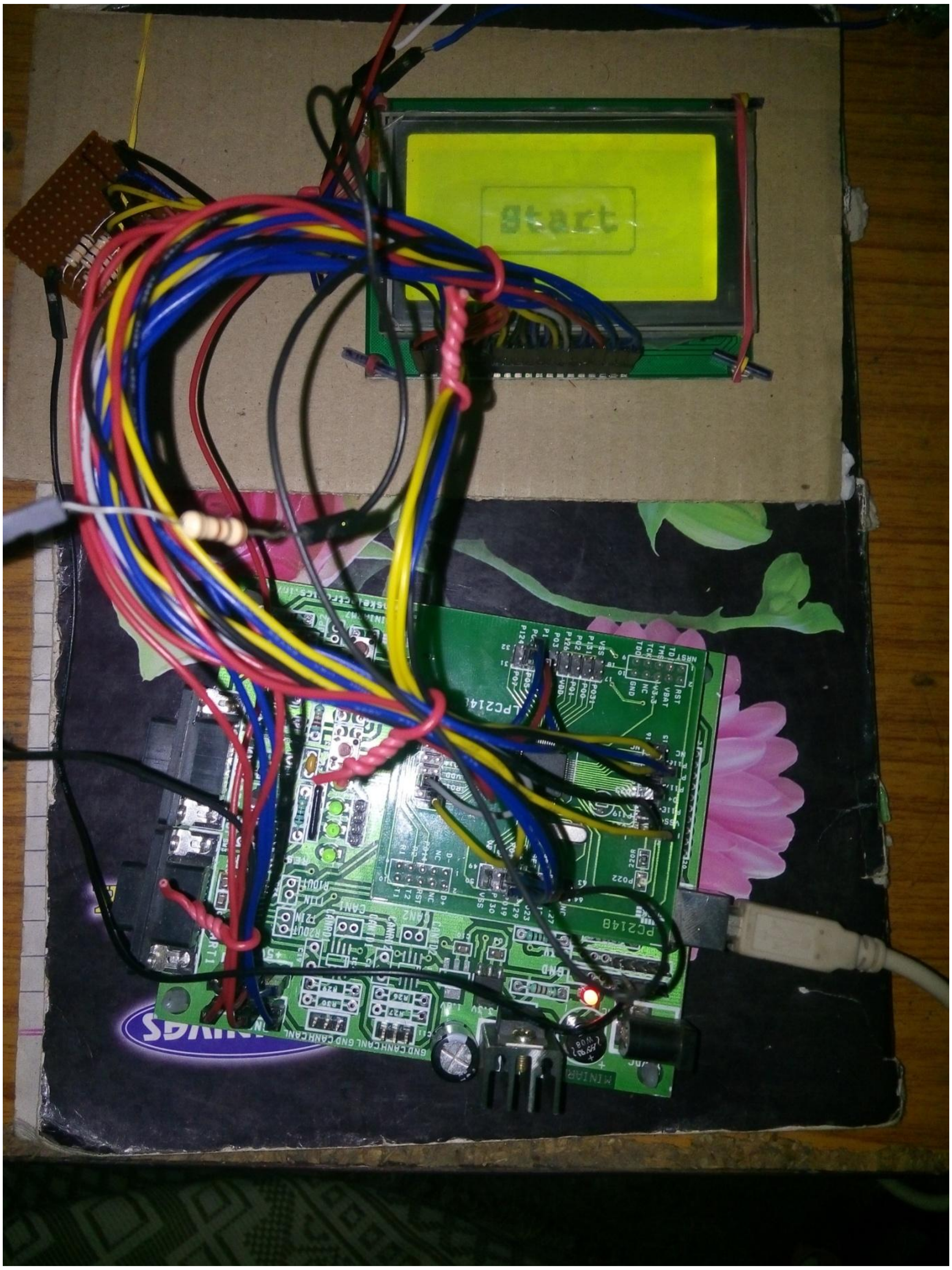

Start
LPC2148
PC2148

Blood Pressure Measurement &Monitoring System Using Interactive Graphic User Interface

Submitted in Partial Fulfilment of the Requirements for the Degree of

Bachelor of Technology

in

Electronics and Telecommunication Engineering

by

Siddharth Vijay Funde

(2011BEC035)

Sushant Ranjeet Mhaisekar

(2011BEC058)

Sumit Ashok Agre

(2011BEC079)

Under the guidance of

Dr. Y.V. JOSHI

Department of Electronics & Telecommunication Engineering

SGGS Institute of Engineering & Technology,

Vishnupuri, Nanded – 431606 (MS).

CERTIFICATE

This is to certify that this project report entitled "**Blood Pressure Measurement &Monitoring System Using Interactive Graphic User Interface**" submitted to **SGGS INSTITUTE OF ENGINEERING & TECHNOLOGY, NANDED** is a bonafide record of work done by

Siddharth Vijay Funde

Sushant Ranjeet Mhaisekar

Sumit Ashok Agre

under my supervision in the year "**2014-15**" .

Dr. S.S.GAJARE	Prof. A.N.KAMTHANE	Dr. Y.V. JOSHI
Head of Department	Project Coordinator	Project Guide

Place: Nanded

Date: 15th May 2015

Acknowledgement

We express our sincere gratitude to Dr. Y.V. Joshi and Dr. S. S. Gajare(Head Of Dept. Extc) for giving us an opportunity to carry out the project work under their guidance. We are extremely grateful for their invaluable and timely guidance given during the course of our project work. We are greatly indebted to them for their critical review of our project work at each and every level. They have been always a source of inspiration for us. We feel extremely fortunate to work under their guidance. We would like to thank them for their meticulous guidance and for making this study an interesting learning experience; for their motivation, teaching and skills have added the necessary favours to this study.

We are thankful to Department of Electronics & Telecommunication Engineering, SGGSIE&T, Nanded, for providing us hardware required for the project.

We wish to take this platform to extend our sincere thanks to all our teachers for moulding us in their special way. Last but not least, we express our gratefulness to Prof. A. N. Kamthane, Project Coordinator for his encouragement, staunch support and belief in our project. We express our heartfelt thanks and sincere gratitude to him for his valuable suggestions, encouragement and moral support given during our project work. We are also thankful to him for providing the laboratory facility whenever needed.

Date: 15th May, 2015

Place: Nanded

Mr. SUMIT A. AGRE

MR.SUSHANT R. MHAISEKAR

MR. SIDDHARTH V. FUNDE

ABSTRACT

Title-Blood Pressure Measurement and Monitoring device with GUI

Idea and Inspiration:

In today's world everyone is working like a machine in order to make this world a better place to live in and they are trying to approach up to expectation. Thus so much stress and tense environment is around a cause of Blood Pressure abnormality.

The person having Blood pressure abnormality has to regularly check his blood pressure and monitor it which is generally done by a medical Practioner. But in this modern world time got the very much importance, so Doctors are asking the patients itself to buy a BP machine. In market there are BP machines which are already available; we will try to come up with a efficient and improved one. But first we have to get into it in order to further improvements that what else can be there. So first we will make simple Blood pressure measuring device with Graphical User Interface in order to make it more userFrienshly!

Rough Idea of Project:

Over all project will be consisting of mainly two parts

Embedded part (GUI and Processing Part)

Measuring part (Blood Pressure Measuring part)

Basic Requirements:

A microcontroller (8051 or Higher Controller), LCD, touch detecting screen(touch screen), blood pressure detection block, ADC.

Block Diagram:

TABLE OF CONTENTS

List of Figure, Abbreviations and Nomenclature

List of figures:

Abbreviation:

MPX2050- Pressure sensor

JHD21864E- GLCD

LF356- FET- based Op-amp

ARM- Advance RISC Machine

LP2148- ARM7 Microcontroller

Nomenclature:

“my_adc.h”- Header file to interface ADC

“glcd.h”- Header file to interface GLCD

“my_uart0.h” - Header file to interface UART0 “data.h” – To store display data

Chapter 0

Introduction

In today’s world everyone is working like a machine in order to make this world a better place to live in and they are trying to approach up to expectation. Thus so much stress and tense environment is around a cause of Blood Pressure abnormality.

The person having Blood pressure abnormality has to regularly check his blood pressure and monitor it which is generally done by a medical Practioner. But in this modern world time got the very much importance, so Doctors are asking the patients itself to buy a BP machine. In market there are BP machines which are already available; we will try to come up with a efficient and improved one. But first we have to get into it in order to further improvements that what else can be there. So first we will make simple Blood pressure measuring device with Graphical User Interface in order to make it more user Friendly!

Rough Idea of Project:

Over all project will be consisting of mainly two parts

Embedded part (GUI and Processing Part)

Measuring part (Blood Pressure Measuring part)

Chapter 1

Blood Pressure Measurement Techniques

1.0 Introduction

Blood pressure (BP) is the pressure exerted by circulating blood upon the walls of blood vessels and is one of the principal vital signs. When used without further specification, "blood pressure" usually refers to the arterial pressure of the systemic circulation, usually measured at a person's upper arm. A person's blood pressure is usually expressed in terms of the systolic (maximum) pressure over diastolic (minimum) pressure and is measured in millimetres of mercury (mm Hg). Normal resting blood pressure for an adult is approximately 120/80 mm Hg.

Blood pressure varies depending on situation, activity, and disease states, and is regulated by the nervous and endocrine systems. Blood pressure that is pathologically low is called hypotension, and pressure that is pathologically high is hypertension. Both have many causes and can range from mild to severe, with both acute and chronic forms. Chronic hypertension is a risk factor for many diseases, including kidney failure, heart attack, and stroke. Chronic hypertension is more common than chronic hypotension in Western countries. Chronic hypertension often goes undetected because of infrequent monitoring and the absence of obvious symptoms.

Arterial pressure is most commonly measured via a sphygmomanometer, which historically used the height of a column of mercury to reflect the circulating pressure. Blood pressure values are generally reported in millimetres of mercury (mm Hg), though aneroid and electronic devices do not contain mercury.

For each heartbeat, blood pressure varies between systolic and diastolic pressures. Systolic pressure is peak pressure in the arteries, which occurs near the end of the cardiac cycle when the ventricles are contracting. Diastolic pressure is minimum pressure in the arteries, which occurs near the beginning of the cardiac cycle when the ventricles are filled with blood. An example of normal measured values for a resting, healthy adult human is 120 mm Hg systolic and 80 mm Hg diastolic.

1.1 Methods for Measurement

1.1.1 Auscultatory

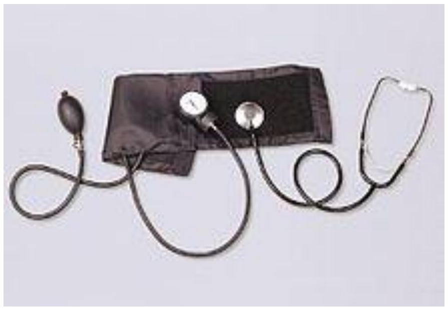


Figure 1.0 Auscultatory method aneroid sphygmomanometer with stethoscope

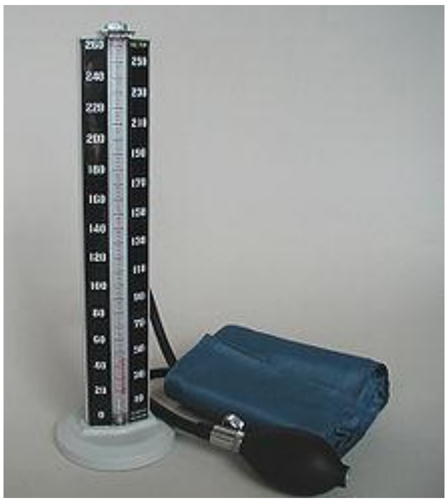


Figure1.1 Mercury manometer

The auscultatory method (from the Latin word for "listening") uses a stethoscope and a sphygmomanometer. This comprises an inflatable cuff placed around the upper arm at roughly the same vertical height as the heart, attached to a mercury or aneroid manometer. The mercury manometer, considered the gold standard, measures the height of a column of mercury, giving an absolute result without need for calibration and, consequently, not subject to the errors and drift of calibration which affect other methods. The use of mercury manometers is often required in clinical trials and for the clinical measurement of hypertension in high-risk patients, such as pregnant women.

A cuff of appropriate size is fitted smoothly and also snugly, and then inflated manually by repeatedly squeezing a rubber bulb until the artery is completely occluded. Listening with the stethoscope to the brachial artery at the antecubital area of the elbow, the examiner slowly releases the pressure in the cuff. When blood just starts to flow in the artery, the turbulent flow creates a "whooshing" or pounding (first Korotkoff sound). The pressure at which this sound is first heard is the systolic blood pressure. The cuff pressure is further released until no sound can be heard (fifth Korotkoff sound), at the diastolic arterial pressure.

The auscultatory method is the predominant method of clinical measurement

1.1.2 Oscillometric

The oscillometric method was first demonstrated in 1876 and involves the observation of oscillations in the sphygmomanometer cuff pressure which are caused by the oscillations of blood flow, i.e., the pulse. The electronic version of this method is sometimes used in long-term measurements and general practice. It uses a sphygmomanometer cuff, like the auscultatory method, but with an electronic pressure sensor to observe cuff pressure oscillations, electronics to automatically interpret them, and automatic inflation and deflation of the cuff. The pressure sensor should be calibrated periodically to maintain accuracy.

Oscillometric measurement requires less skill than the auscultatory technique and may be suitable for use by untrained staff and for automated patient home monitoring.

The cuff is inflated to a pressure initially in excess of the systolic arterial pressure and then reduced to below diastolic pressure over a period of about 30 seconds. When blood flow is nil(cuff pressure exceeding systolic pressure) or unimpeded (cuff pressure below diastolic pressure), cuff pressure will be essentially constant. It is essential that the cuff size is correct: undersized cuffs may yield too high a pressure; oversized cuffs yield too low a pressure. When blood flow is present, but restricted, the cuff pressure, which is monitored by the pressure sensor, will vary periodically in synchrony with the cyclic expansion and contraction of the brachial artery, i.e., it will oscillate.

Over the deflation period, the recorded pressure waveform forms a signal known as the cuff deflation curve. A band pass filter is utilized to extract the oscillometric pulses from the cuff deflation curve. Over the deflation period, the extracted oscillometric pulses form a signal known as the oscillometric waveform. The amplitude of the oscillometric pulses increases to a maximum and then decreases with further deflation. A variety of analysis algorithms can be employed in order to estimate the systolic, diastolic, and mean arterial pressure.

In practice the different methods do not give identical results; an algorithm and experimentally obtained coefficients are used to adjust the oscillometric results to give readings which match the auscultatory results as well as possible. Some equipment uses computer-aided analysis of the instantaneous arterial pressure waveform to determine the systolic, mean, and diastolic points. Since many oscillometric devices have not been validated, caution must be given as most are not suitable in clinical and acute care settings.

Recently, several coefficient-free oscillometric algorithms have developed for estimation of blood pressure. These algorithms do not rely on experimentally obtained coefficients and have been shown to provide more accurate and robust estimation of blood pressure.

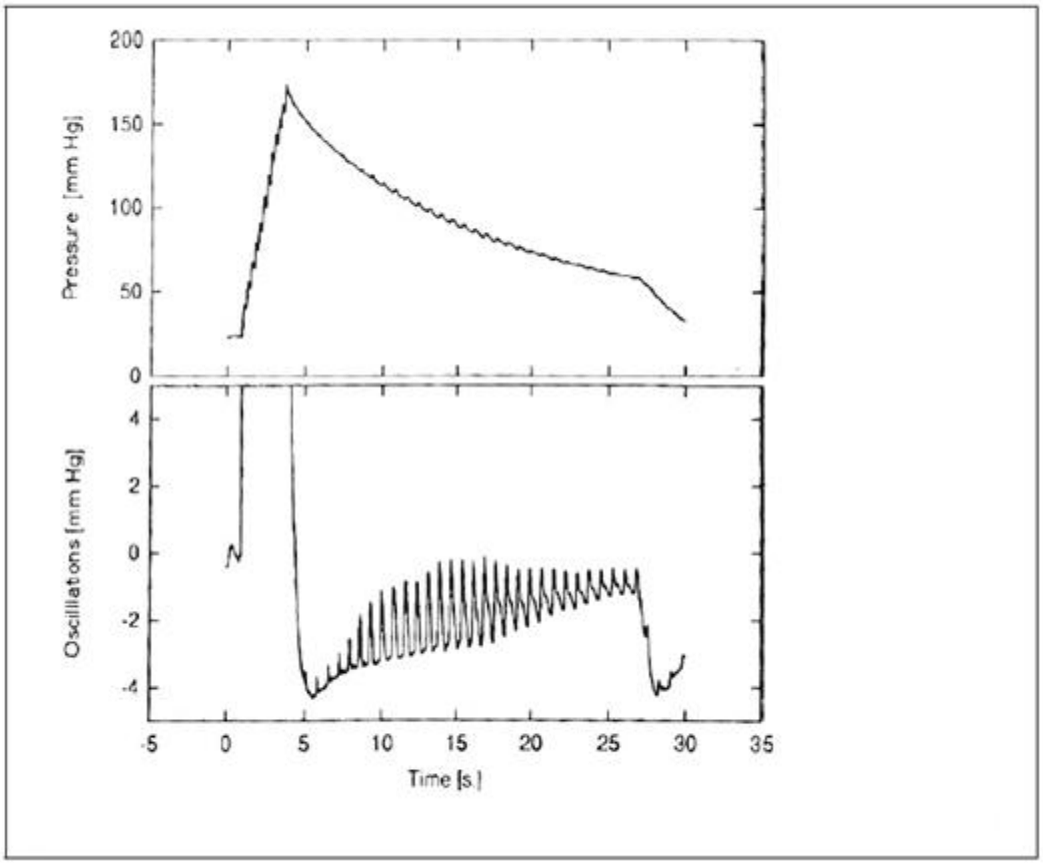


Figure 1.2 Recording of Cuff Pressure

Chapter 2

Touch Screen

2.0 Introduction

Touch screens are two dimensional input devices. Nowadays most of the electronic gadgets use them. Laptops, smart phones, tablets and even some home appliances like washing machines & microwave ovens also use a touch screen nowadays.

2.1 Types

- Resistive Touch
- Surface Capacitive
- Projected Capacitive
- Surface Acoustic Touch
- Infrared Touch

2.2.1 4-Wire Resistive Touch

4-Wire Resistive Touch is the most widely used Touch Technology. A resistive touch screen monitor is composed of a glass panel and a film screen,

Each covered with a thin metallic layer, separated by a narrow gap. When a user touches the screen, the two metallic layers make contact, resulting in electrical flow. The point of contact is detected by this change in voltage.

2.3 Why?

Touch screens are preferred over keypads because they need very little or no pressure to operate whereas the Keypads/ buttons need a minimum pressure to operate and our hands start aching after some time of continuous usage.

- Lowest cost touch technology
- Low power consumption

And one more great advantage in using a touch screens is that it enables us to make more room for the screen itself instead of wasting the space on the permanent keypad. And that's the reason for our smart phone's screens to become big enough to browse web pages also and still fit in our pockets.

2.4 Pin Description

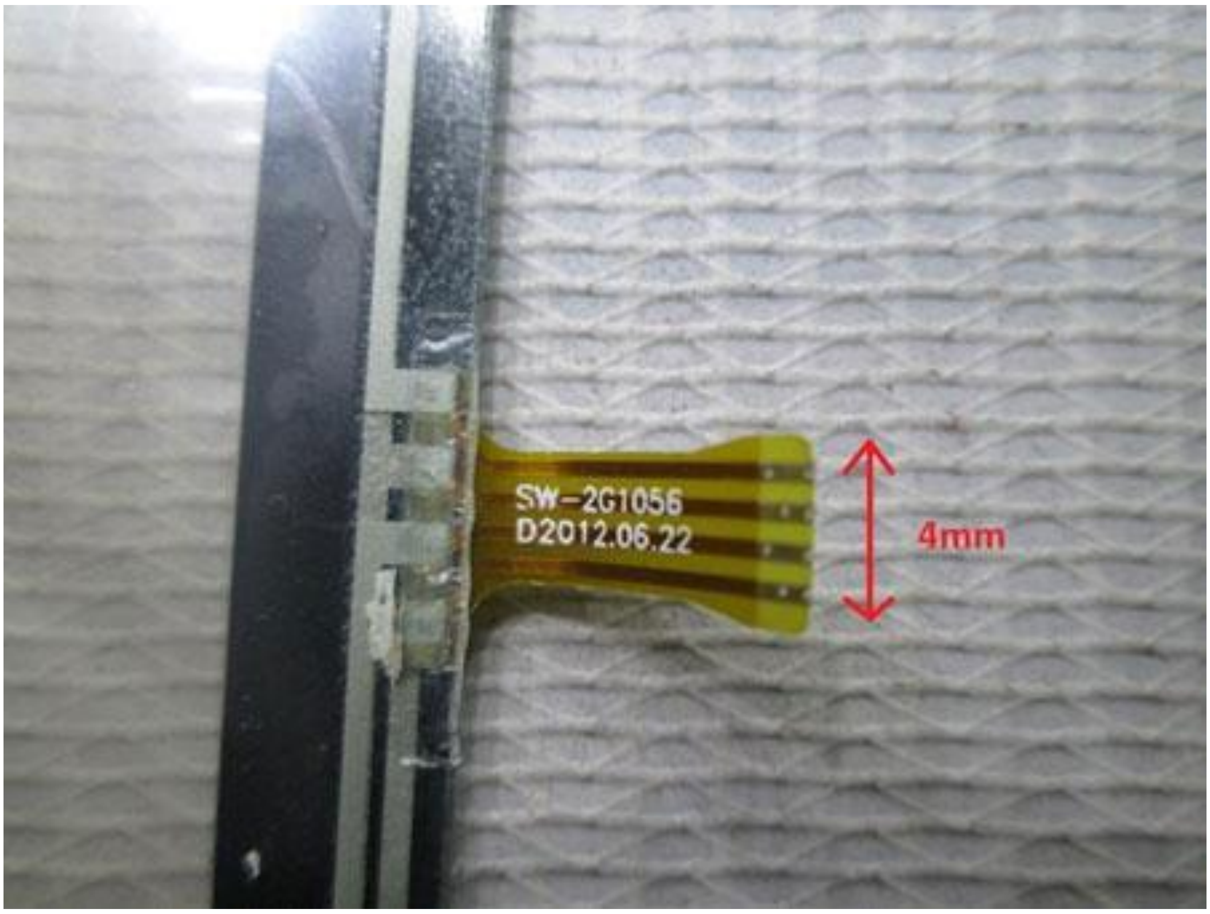


Figure 2.0 pin description of 4 wire resistive touch

2.5 Working?

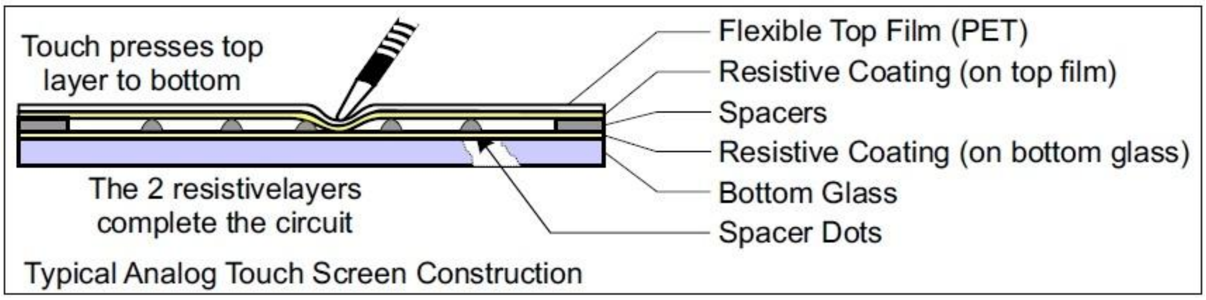

Touch presses top layer to bottom
Flexible Top Film (PET)
Resistive Coating (on top film)
Spacers
Resistive Coating (on bottom glass)
Bottom Glass
Spacer Dots
The 2 resistivelayers complete the circuit
Typical Analog Touch Screen Construction

Figure 2.1 Typical analog Touch Screen Construction

2.6 Position Sensing?

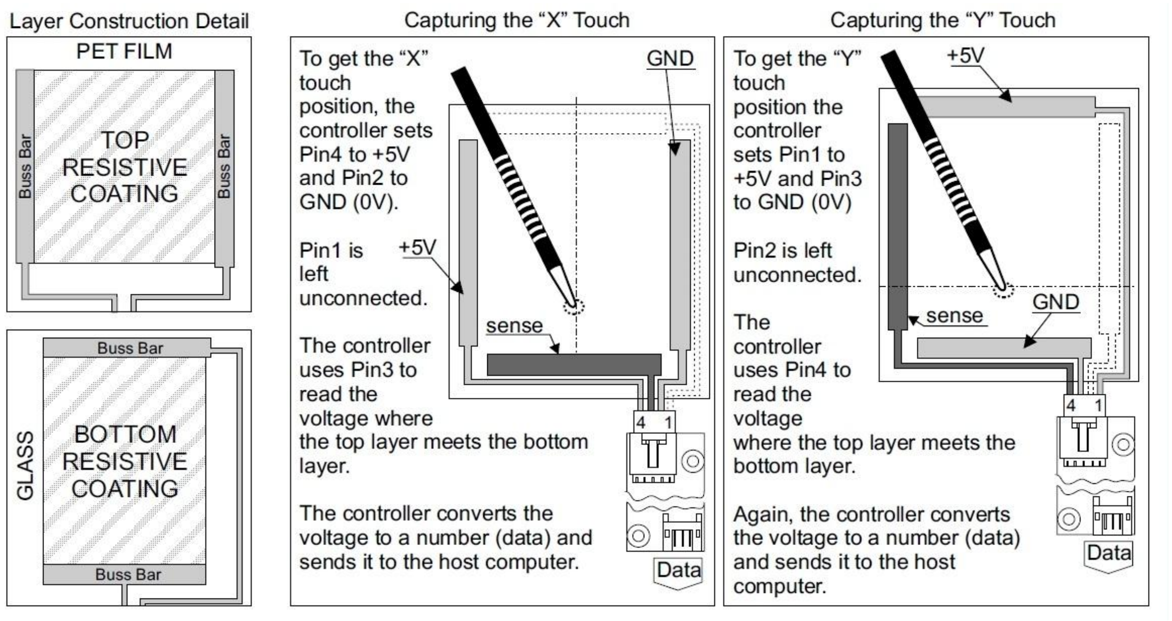

Layer Construction Detail
PET FILM
Buss Bar
TOP RESISTIVE COATING
Buss Bar
GLASS
Buss Bar
BOTTOM RESISTIVE COATING
Buss Bar
Capturing the "X" Touch
To get the "X" touch position, the controller sets Pin4 to +5V and Pin2 to GND (0V).
Pin1 is left unconnected.
+5V
GND
sense
The controller uses Pin3 to read the voltage where the top layer meets the bottom layer.
The controller converts the voltage to a number (data) and sends it to the host computer.
4 1
Data
Capturing the "Y" Touch
To get the "Y" touch position the controller sets Pin1 to +5V and Pin3 to GND (0V)
Pin2 is left unconnected.
+5V
GND
sense
The controller uses Pin4 to read the voltage where the top layer meets the bottom layer.
Again, the controller converts the voltage to a number (data) and sends it to the host computer.
4 1
Data

Figure 2.2 Position Sensing of Touch

Chapter 3

Graphics LCD

3.0 Introduction

User friendly visual displays are used nowadays to keep track of working of any device. Such a visual display can be anything ranging from old analog meters to new and smart Digital meters.

3.1 Pin Diagram

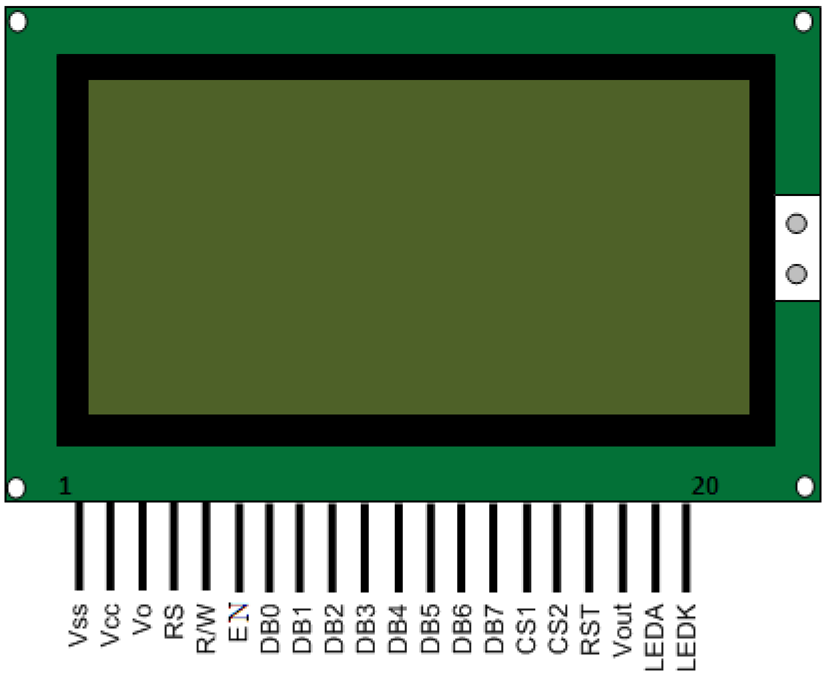
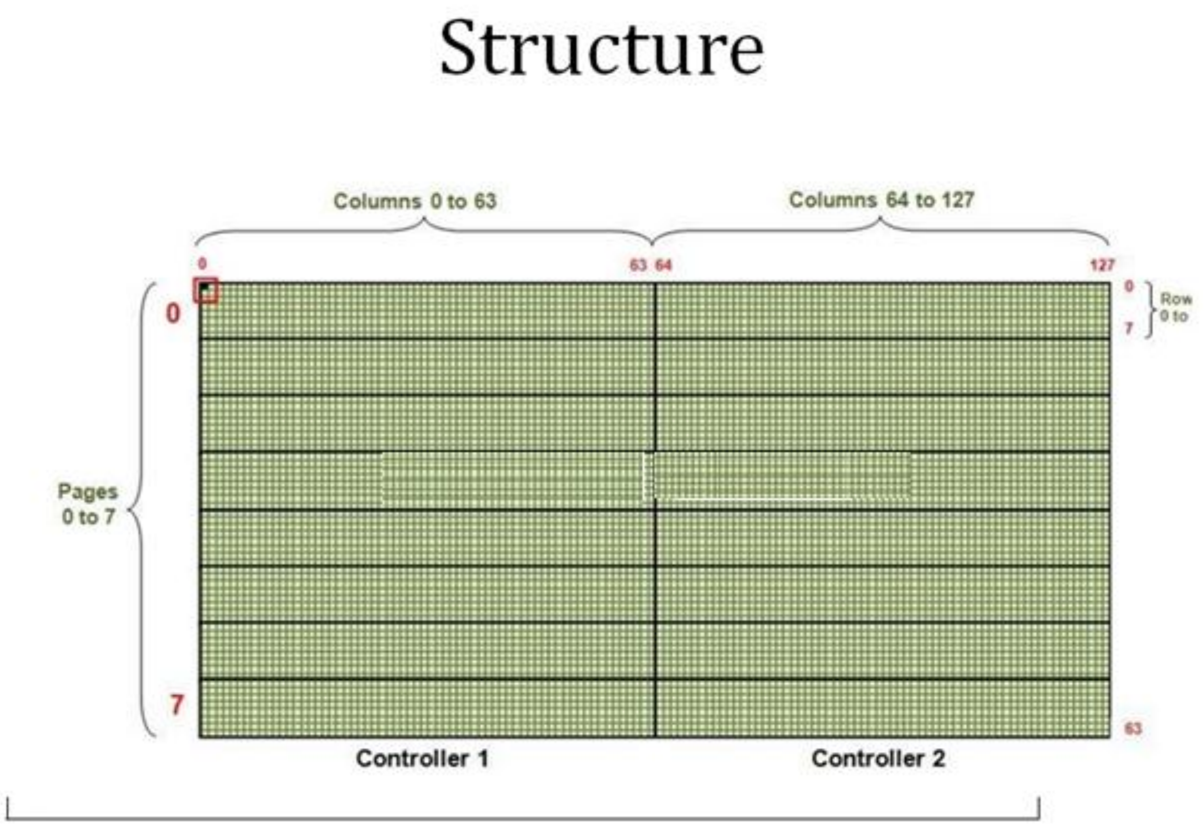


Figure 3.0 Pin diagram of GLCD

3.2 Description

Pin no.	Symbol	Level	Description
1	$\overline{CS1}$	L	Select segments 1-64
2	$\overline{CS2}$	L	Select segments 65-128
3	V_{SS}	0 V	Ground
4	V_{DD}	5.0 V	Supply voltage for logic
5	Vo	Variable	Contrast adjustment
6	D/I or RS	H/L	H : Data, L: Instruction
7	R/W	H/L	H: Read data, L: Write data
8	E	H	Enable signal
9-16	D0-D7	H/L	Data bus
17	RST	L	Reset the LCD module
18	V_{EE}		Negative voltage output
19	A		Anode (+) of B/L LED
20	K		Cathode (-) of B/L LED

3.3 Structure



3.4 Registers

Input Register:

Used while giving instructions and writing data to LCD.

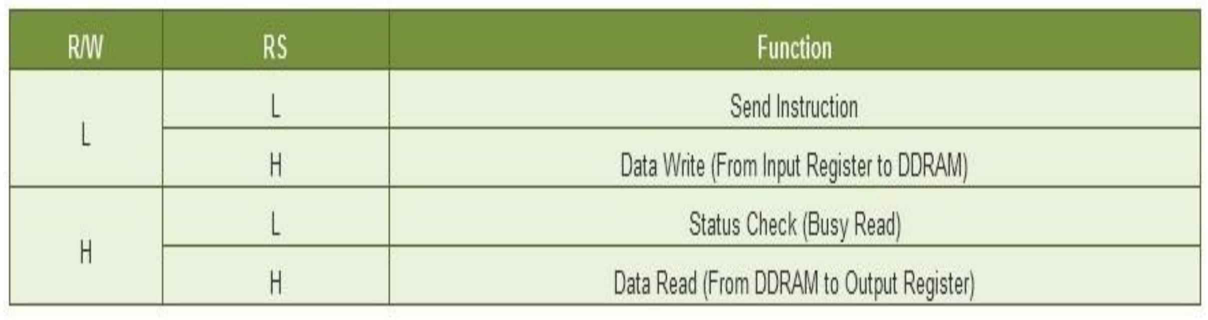
Holds the data/instruction temporarily before writing to DDRAM(Data Display RAM).

Output Register:

Used to read data from DDRAM and to check status data(busy check).

3.5 Working

3.5.0 How?



R/W	RS	Function
L	L	Send Instruction
	H	Data Write (From Input Register to DDRAM)
H	L	Status Check (Busy Read)
	H	Data Read (From DDRAM to Output Register)

3.5.1 Algorithm

The basic operation with graphical LCD requires following steps:

- LCD Initialization
- Page Selection
- Column Selection
- Data Display

LCD Initialization

Before displaying anything on graphics LCD, display must be puton and column/page selection be made.

1.Put these values in Data Register

Data appears when D=1 and disappears when D=0. When the display is off, there is no effect on the data which is stored in DDRAM.

2. CS1=1, CS2=1(to activate display of both halves of LCD)

3. RS=0, R/W=0 (to select the instruction mode)

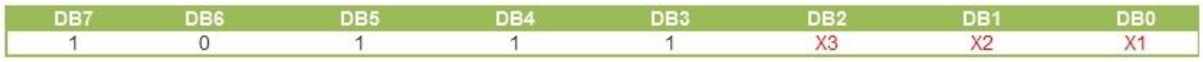
4. EN=1

5. Delay

6. EN=0 (to latch data into the input register)

Page selection

Put these values in Data Register



DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	1	1	1	X3	X2	X1

Since there are a total of 8 pages (0–7), a particular page is selected by setting three bits (X1-X3).

CS1=1, CS2=1(to activate display of both halves of LCD)

RS=0, R/W=0 (to select the instruction mode)

EN=1

Delay

EN=0 (to latch data into the input register)

For example, if X3=0, X2=1 and X1=0, then the second page is selected. Reading or writing operation is done on this page until next page is set. Depending on the column selection, the page is selected from either left or right half of the graphics LCD.

Column selection

1. There are 128 [64 (=26) columns per half] in graphics LCD and they are automatically incremented.

2. This means that after selecting a column, it increases on its own by one, after each write cycle.

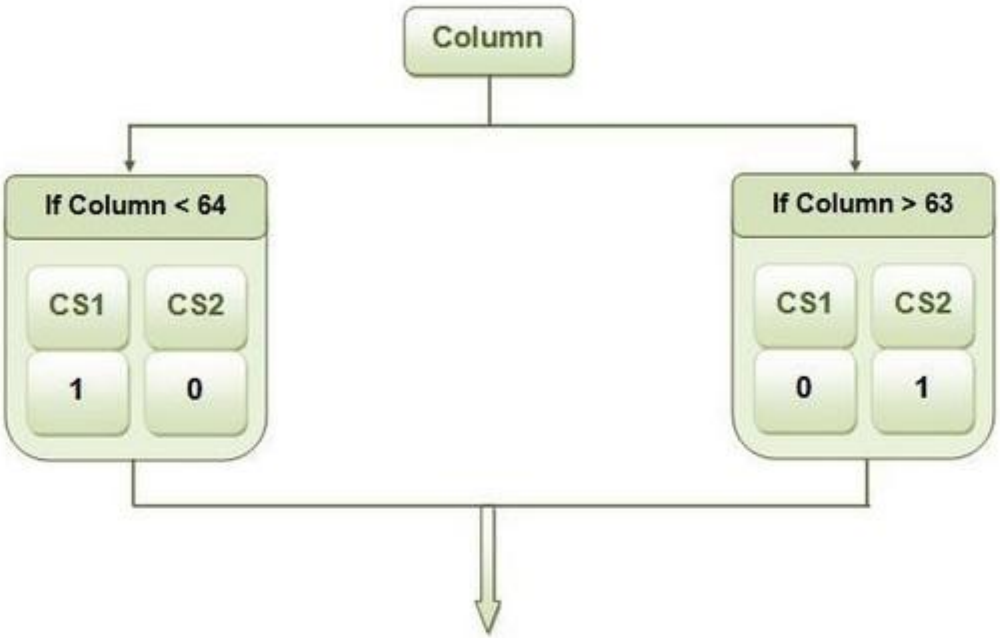
3. So it is easier to write data column by column.

4. A column can be chosen through following instructions:

DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	Y5	Y4	Y3	Y2	Y1	Y0

5. a) Put these values in Data Register.

The corresponding controller (CS1 or CS2) is selected depending on the Column number as shown below.



b) RS=0, R/W=0 (to select the instruction mode)

c) EN=1

d) Delay

e) EN=0 (to latch data into the input register)

Display data

After page and column selection, data can be sent to LCD for display. The programming steps for display data are as given below:

Put the data values in Data Register.

With every write cycle, data is written in one column and the column then gets auto incremented.

A high data bit (DBx = 1) corresponds to activated (dark) pixel and low data bit (DBx = 0) corresponds to deactivated (light) pixel.

Here MSB corresponds to 8th row in column and LSB to 1st row of column.

b) If column<63 then (CS1=1 & CS2=0) else (CS1=0 & CS2=1)

c) RS=1 and R/W=0 (to select write mode of LCD)

d) EN=1

e) Delay

f) EN=0 (to latch data into the input register)

Chapter 4

Sensors

4.0 Pressure Sensor

We need to measure pressure from above the highest systolic pressure (including any fluctuations that may be present) to below the lowest possible diastolic pressure (again, including any fluctuations that may be present).

Systolic pressures above 260 mmHg are rarely seen and so a range of 0mmHg to 300mmHg should achieve this. A range of 0 to 5.8 psi is required (1 mmHg is 0.01933 psi).

It must also have sufficient resolution to accurately describe the fluctuations in pressure which are of the order of a few mmHg, i.e., about 1% of the pressure range.

Using the smaller range will restrict the range of Blood Pressure’s which can be measured whereas using the larger range may result in less accurate measurement because the accuracy of these components is typically a fraction (of the full scale measurement).

Typically, the accuracy of the reading is 0.2 to 0.5% of the full scale reading. 0.5% of 15 psi is 0.075 psi or 3.9bar which is too course to capture the fluctuations.

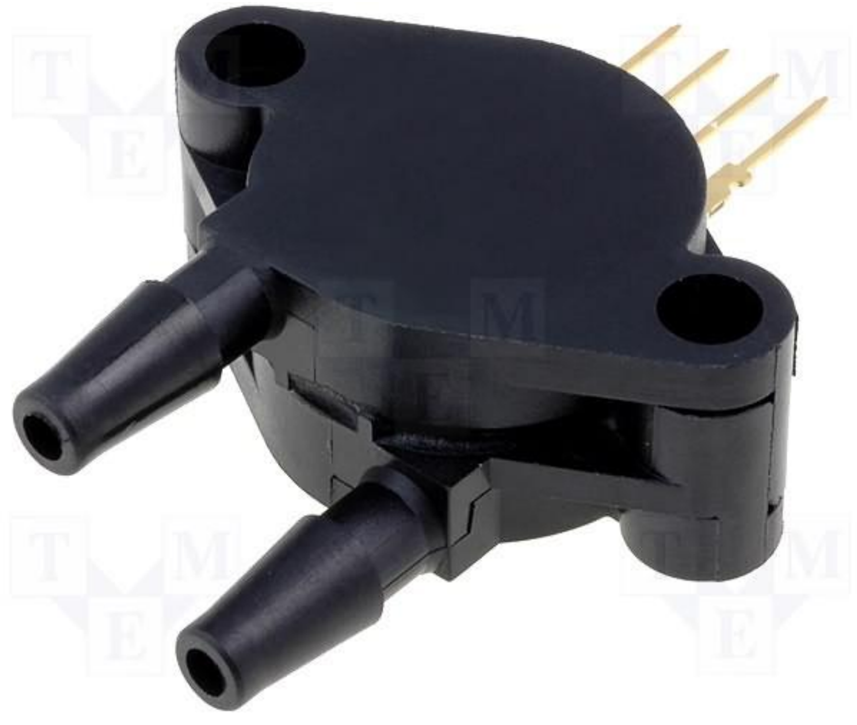


Figure 4.0 Physical Structure of MPX-2050

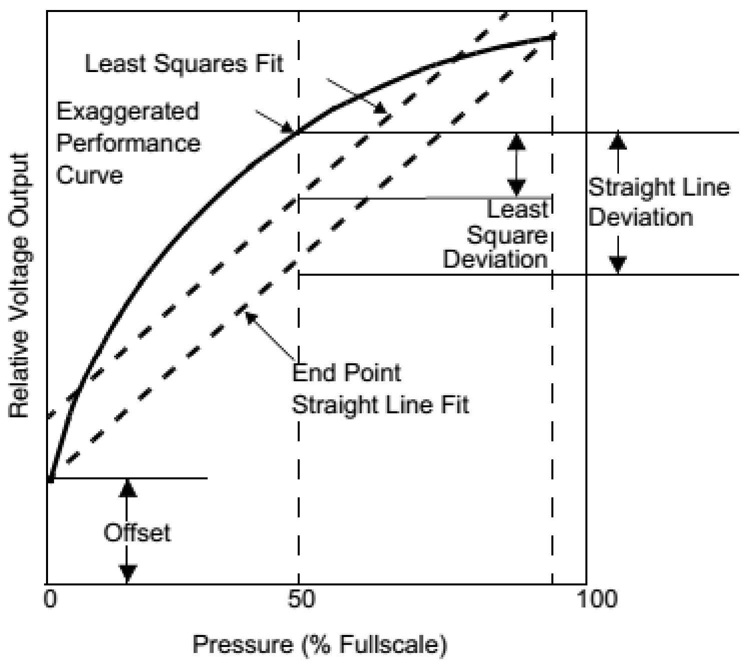


Figure 4.1 Linearity Specification Comparisons

4.1 Features of MPX2050

• Temperature Compensated Over 0°C to +85°C

• Unique Silicon Shear Stress Strain Gauge

• Easy to Use Chip Carrier Package Options

• Ratio metric to Supply Voltage

• Differential and Gauge Options

• ±0.25% Linearity

Chapter 5

Microcontroller (LPC2148 –ARM7 TDMI Based)

5.0 Introduction

ARM is a family of instruction set architectures for computer processors based on are reduced instruction set computing (RISC) architecture developed by British company ARM Holdings.

A RISC-based computer design approach means ARM processors require significantly fewer transistors than typical processors in average computers. This approach reduces costs, heat and power use. These are desirable traits for light, portable, battery-powered devices—including smart phones, laptops, tablet and notepad computers), and other embedded systems. A simpler design facilitates more efficient multi-core CPUs and higher core counts at lower cost, providing higher processing power and improved energy efficiency for servers and supercomputers. It is manufactured by Philips and it is pre-loaded with many inbuilt peripherals making it more efficient and a reliable option for the beginners as well as high end application developer.

Features OF LPC 2148

- 8 to 40kB of on-chip static RAM and 32 to 512kB of on-chip flash program memory.128 bit wide interface/accelerator enables high speed 60 MHz operation.
- In-System/In-Application Programming (ISP/IAP) via on-chip boot-loader software. Single flash sector or full chip erase in 400 ms and programming of 256 bytes in 1ms.
- Embedded ICE RT and Embedded Trace interfaces offer real-time debugging with the on-chip Real Monitor software and high speed tracing of instruction execution.
- USB 2.0 Full Speed compliant Device Controller with 2kB of endpoint RAM. In addition, the LPC2146/8 provides 8kB of on-chip RAM accessible to USB by DMA.
- One or two (LPC2141/2 vs. LPC2144/6/8) 10-bit A/D converters provide a total of 6/14analog inputs, with conversion times as low as 2.44 us per channel.
- Single 10-bit D/A converter provide variable analog output.
- Two 32-bit timers/external event counters (with four capture and four compare channels each), PWM unit (six outputs) and watchdog.
- Low power real-time clock with independent power and dedicated 32 kHz clock input.
- Multiple serial interfaces including two UARTs (16C550), two Fast I2C-bus (400 Kbit/s), SPI and SSP with buffering and variable data length capabilities.
- Vectored interrupt controller with configurable priorities and vector addresses.
- Up to 4.5 of 5 V tolerant fast general purpose I/O pins in a tiny LQFP64 package.

- Up to nine edge or level sensitive external interrupt pins available.
- On-chip integrated oscillator operates with an external crystal in range from 1 MHz to30 MHz and with an external oscillator up to 50MHz.
- Power saving modes include idle and Power-down.
- Individual enable/disable of peripheral functions as well as peripheral clock scaling for additional power optimization.
- Processor wake-up from Power-down mode via external interrupt, USB, Brown-OutDetect (BOD) or Real-Time Clock (RTC).
- Single power supply chip with Power-On Reset (POR) and BOD circuits:
- CPU operating voltage range of 3.0 V to 3.6 V (3.3 V ± 10 %) with 5 V tolerant I/O pins.

Pin Diagram

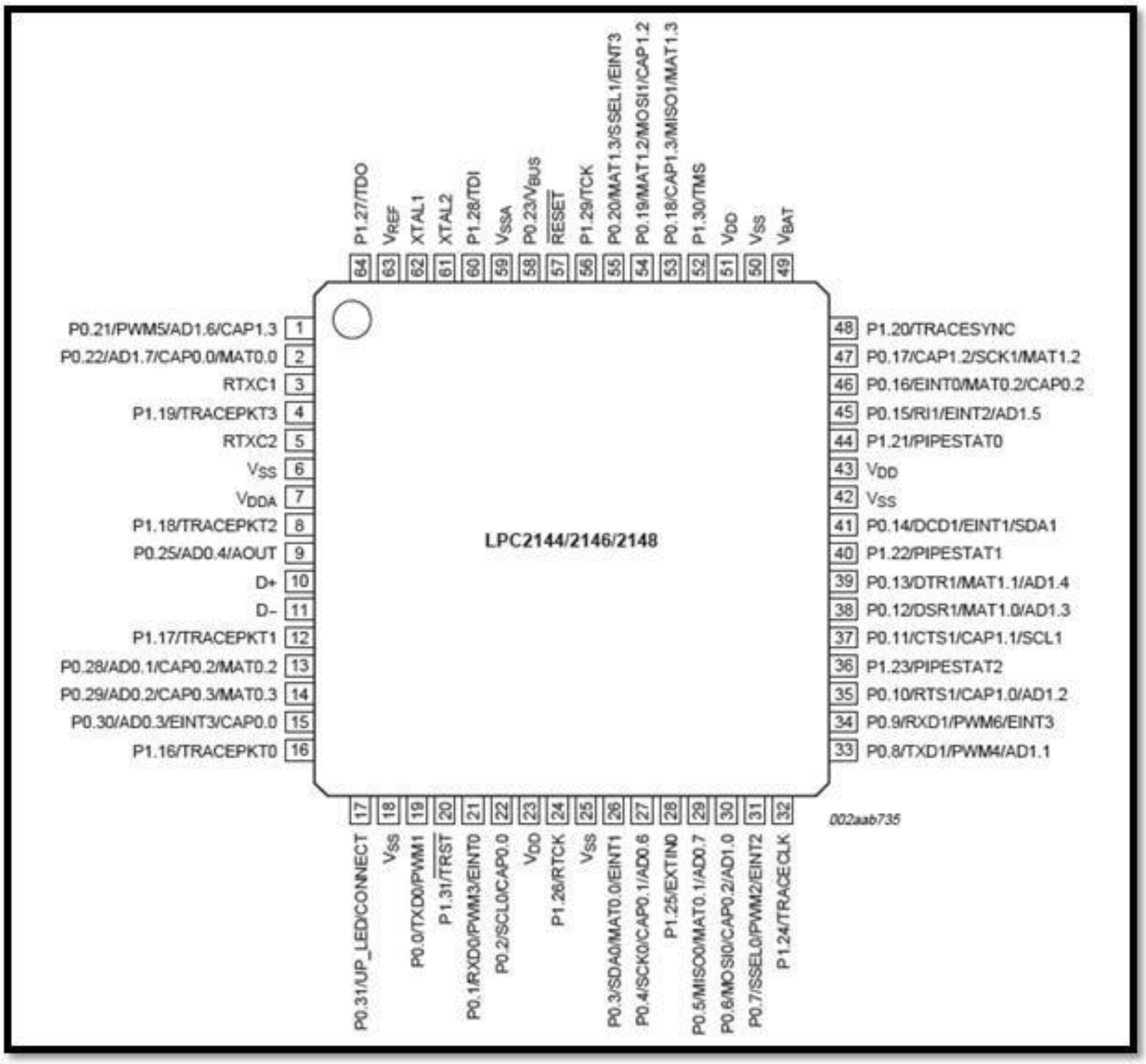


Figure 5.0 Pin Description Diagram of LPC-2148

5.3 GPIO

GPIO, or General Purpose Input/output, is the easiest way for you to interact with basic peripherals like buttons, LEDs, switches, and other components. It can also be used for more complex components like text and graphic LCDs, but for now we'll start with a few basic components that are relatively ease to get working.

Most of the function oriented pins on lpc214x Microcontrollers are grouped into ports. Lpc2148 has 2 ports viz. Port 0 and Port 1.

Port 0 is a 32 bit wide I/O port (i.e. it can be used for max 32 pins where each pin refers to a corresponding bit) and has dedicated direction bits for each of the pins present in the port. 28 out of the 32 pins can be used as bi-directional I/O (digital) pins. Pins P0.24, P0.26 & P0.27 are unavailable for use and Pin P0.30 can be used as output pin only.

Port 1 is also a 32 bit wide I/0 port but Pins 0 to 15 i.e. P1.0 – P1.15 are unavailable for use and this port too has a dedicated direction bit for each of the usable pins.

1. IOxPIN : This register can be used to read or write values directly to pins. Regardless of direction set for the particular pins it gives the current start of of GPIO pin when read.

2. IOxDIR : This is the GPIO direction control register. Setting a bit to 0 in this register will configure the corresponding pin to be used as an Input while setting it to 1 will configure it as Output.

3. IOxSET : This register can be used to drive an 'output' configured pin to Logic 1 i.e HIGH. Writing Zero does NOT have any effect and hence it cannot be used to drive a pin to Logic 0 i.e LOW. For driving pins LOW IOxCLR is used which is explained below.

4. IOxCLR : This register can be used to drive an 'output' configured pin to Logic 0 i.e LOW. Writing Zero does NOT have any effect and hence it cannot be used to drive a pin to Logic 1.

5.4 ADC

While dealing with controllers we are dealing with digital signals but real world is analog. In real world all natural quantities like temperature, pressure, light intensity etc. are analog. If we have to deal with such quantities using embedded systems, we have to convert these analog quantities which can be understood and proceed by digital signal. Devices used for this purpose is known as ADC. Special ICs such as ADC0804, ADC0808, ADC0809, Serial ADC MAX1112 etc. are handy for this. Nowadays, many controllers are having inbuilt ADC. LPC2148 of ARM7 family is one of the widely used controller and it also has multichannel ADC inbuilt. In this article, we will understand how to use inbuilt ADC of LPC2148.

To configure the A/D Converter, we need to pass a specific 32-bit value to the appropriate ADCR, or "Analog/Digital Control Register". This 'control register' (defined in lpc214x.h as 'AD0_CR' and 'AD1_CR') manages the configuration of our A/D converter, and determines a variety of things, including:

SEL - Which channel should be used (0..7)

CLKDIV - A value to divide PCLK by to determine which speed the A/D Converter should operate at (up to a maximum of 4.5MHz)

CLKS - How precise the conversion results should be (between 3 and 10 bits)

PDN - Whether the A/D Converter is currently active (1) or sleeping (0)

The 32-bit Analog/Digital Control Register has the following format:

Function	-	EDGE	START	-	PDN	-	CLKS	BURST	CLKDIV	SEL
ADCR Bit(s)	31-28	27	26-24	23.22	21	20	19-17	16	15-8	7-0

Unfortunately, that may not make very much sense to you, but without going into every little detail of every possible configuration option (see p.270-272 for the User's Manual for full details), we'll try to explain the register values that are being used in the example at the beginning of this tutorial: SEL, CLKDIV, CLKS and PDN.

5.5 UART

Features

• 16 byte Receive and Transmit FIFOs
• Register locations conform to ‘550 industry standard.
• Receiver FIFO trigger points at 1, 4, 8, and 14 bytes.
• Built-in fractional baud rate generator with autobauding capabilities.
• Mechanism that enables software and hardware flow control implementation.

The UART0 is operating at a Baud Rate of 9600bps; the Baud Rate is determined by the Crystal Frequency, U0DLM, U0DLL, DivAddVal, MulVal Values.
The Formula for Calculation is as follow:-

$$UART0_{baudrate} = \frac{PCLK}{16 \times (256 \times U0DLM + U0DLL) \times \left(1 + \frac{DivAddVal}{MulVal}\right)}$$

U0THR(Transmit holding reg.) (Bit 0-7):-

The U0THR is the top byte of the UART0 TX FIFO. The top byte is the newest character in the TX FIFO and can be written via the bus interface. The LSB represents the first bit to transmit. The Divisor Latch Access Bit (DLAB) in U0LCR must be zero in order to access the U0THR. The U0THR is always Write Only.

U0DLL (Divisor Latch LSB Registers):-

The UART0 Divisor Latch is part of the UART0 Fractional Baud Rate Generator and holds the value used to divide the clock supplied by the fractional prescaler in order to produce the baud rate clock, which must be 16x the desired baud rate. The U0DLL and U0DLM registers together form a 16 bit divisor where U0DLL contains the lower 8 bits of the divisor and U0DLM contains the higher 8 bits of the divisor. A 0x0000 value is treated like a 0x0001 value as division by zero is not allowed. The Divisor Latch Access Bit (DLAB) in U0LCR must be one in order to access the UART0 Divisor Latches.

The UART0 Divisor Latch LSB Register, along with the U0DLM register, determines the baud rate of the UART0.

U0LSR (Line Status Register):

1:0 Word Length Select: -

00 - 5 bit character length 0
01 - 6 bit character length
10 - 7 bit character length
11 - 8 bit character length

2 Stop Bit Select: -

0 - 1 stop bit.
1 - 2 stop bits

3 Parity Enable: -

0- Disable parity generation and checking.
1- Enable parity generation and checking.

5:4 Parity Select: -

00-Odd parity: - Number of 1s in the transmitted character and the attached parity bit will be odd.

01-Even Parity: - Number of 1s in the transmitted character and the attached parity bit will be even.

Chapter 6

Filters and Amplifiers

6.0 Pre-amplifiers

A preamplifier (preamp) is an electronic amplifier that prepares a small electrical signal for further amplification or processing. A preamplifier is often placed close to the sensor to reduce the effects of noise and interference. It is used to boost the signal strength to drive the cable to the main instrument without significantly degrading the signal-to-noise ratio (SNR). The noise performance of a preamplifier is critical; according to Friis's formula, when the gain of the preamplifier is high, the SNR of the final signal is determined by the SNR of the input signal and the noise figure of the preamplifier.

In a home audio system, the term 'preamplifier' may sometimes be used to describe equipment which merely switches between different line level sources and applies a volume control, so that no actual amplification may be involved. In an audio system, the second amplifier is typically a power amplifier (power amp). The preamplifier provides voltage gain (e.g. from 10millivolts to 1 volt) but no significant current gain. The power amplifier provides the higher current necessary to drive loudspeakers.

Preamplifiers may be incorporated into the housing or chassis of the amplifier they feed in a separate housing mounted within or near the signal source, such as a turntable, microphone or musical instrument.

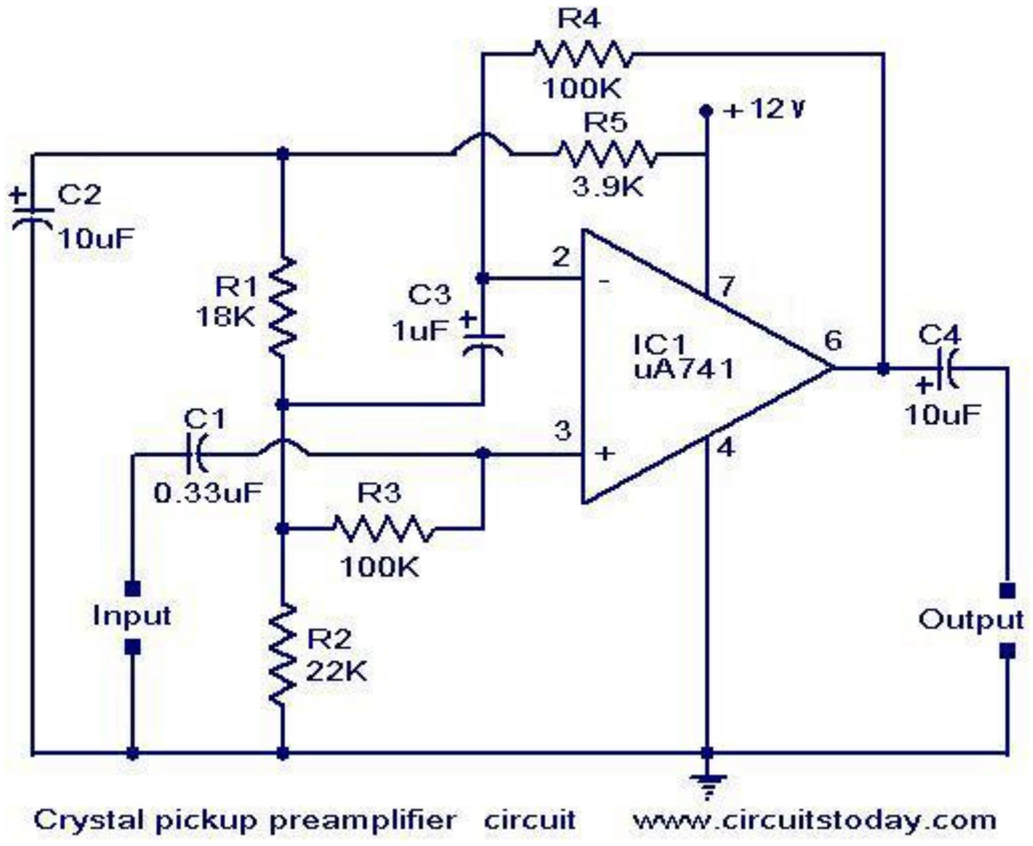


Figure 6.0 Preamplifier

6.1 Instrumentation Amplifier

An instrumentation amplifier is a type of differential amplifier that has been outfitted with input buffer amplifiers, which eliminate the need for input impedance matching and thus make the amplifier particularly suitable for use in measurement and test equipment. Additional characteristics include very low DC offset, low drift, low noise, very high open-loop gain, very high common-mode rejection ratio, and very high input impedances. Instrumentation amplifiers are used where great accuracy and stability of the circuit both short and long-term are required.

Although the instrumentation amplifier is usually shown schematically identical to a standard operational (op-amp), the electronic instrumentation amp is almost always internally composed of 3 op-amps. These are arranged so that there is one op-amp to buffer each input (+,−), and one to produce the desired output with adequate impedance matching for the function.

The most commonly used instrumentation amplifier circuit is shown in the figure. The gain of the circuit is

$$\frac{V_{\text{out}}}{V_2 - V_1} = \left(1 + \frac{2R_1}{R_{\text{gain}}}\right)\frac{R_3}{R_2}$$

The rightmost amplifier, along with the resistors labelled R_2 and R_3 is just the standard differential amplifier circuit, with gain = R_3/R_2 and differential input resistance = 2· R_2. The two amplifiers on the left are the buffers. With R_{gain} removed (open circuited), they are simple unity gain buffers; the circuit will work in that state, with gain simply equal to R_3/R_2 and high input impedance because of the buffers. The buffer gain could be increased by putting resistors between the buffer inverting inputs and ground to shunt away some of the negative feedback; however, the single resistor R_{gain} between the two inverting inputs is a much more elegant method: it increases the differential-mode gain of the buffer pair while leaving the common-mode gain equal to 1. This increases the common-mode rejection ratio (CMRR) of the circuit and also enables the buffers to handle much larger common-mode signals without clipping than would be the case if they were separate and had the same gain.

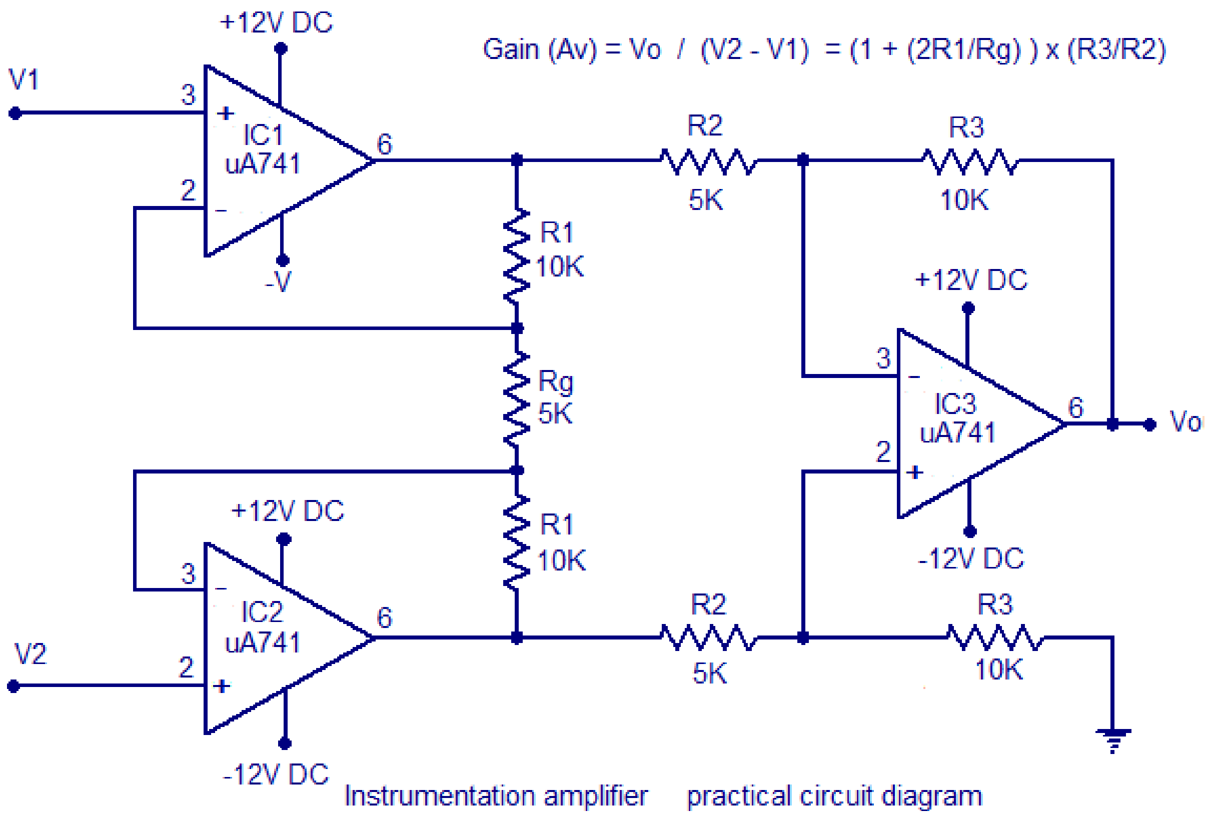


Figure 6.1 Instrumentation Amplifier

6.2 Filters

6.2.0 Low Pass Filter

A low-pass filter is a filter that passes signals with a frequency lower than a certain cut-off frequency and attenuates signals with frequencies higher than the cutoff frequency. The amount of attenuation for each frequency depends on the filter design. The filter is sometimes called a high-cut filter, or treble cut filter in audio applications. A low-pass filter is the opposite of a high-pass filter. A band-pass filter is a combination of a low-pass and a high-pass filter.

Low-pass filters exist in many different forms, including electronic circuits (such as a hiss filter used in audio), anti-aliasing filters for conditioning signals prior to analog-to-digital conversion, digital filters for smoothing sets of data, acoustic

barriers, blurring of images, and so on. The moving average operation used in fields such as finance is a particular kind of low-pass filter, and can be analyzed with the same signal processing techniques as are used for other low-pass filters. Low-pass filters provide a smoother form of a signal, removing the short-term fluctuations, and leaving the longer-term trend.

In the operational amplifier circuit shown in the figure, the cutoff frequency (in hertz) is defined as:

$$f_c = \frac{1}{2\pi R_2 C}$$

or equivalently (in radians per second):

$$\omega_c = \frac{1}{R_2 C}$$

The gain in the passband is −R2/R1, and the stopband drops off at −6 dB per octave (that is −20 dB per decade) as it is a first-order filter.

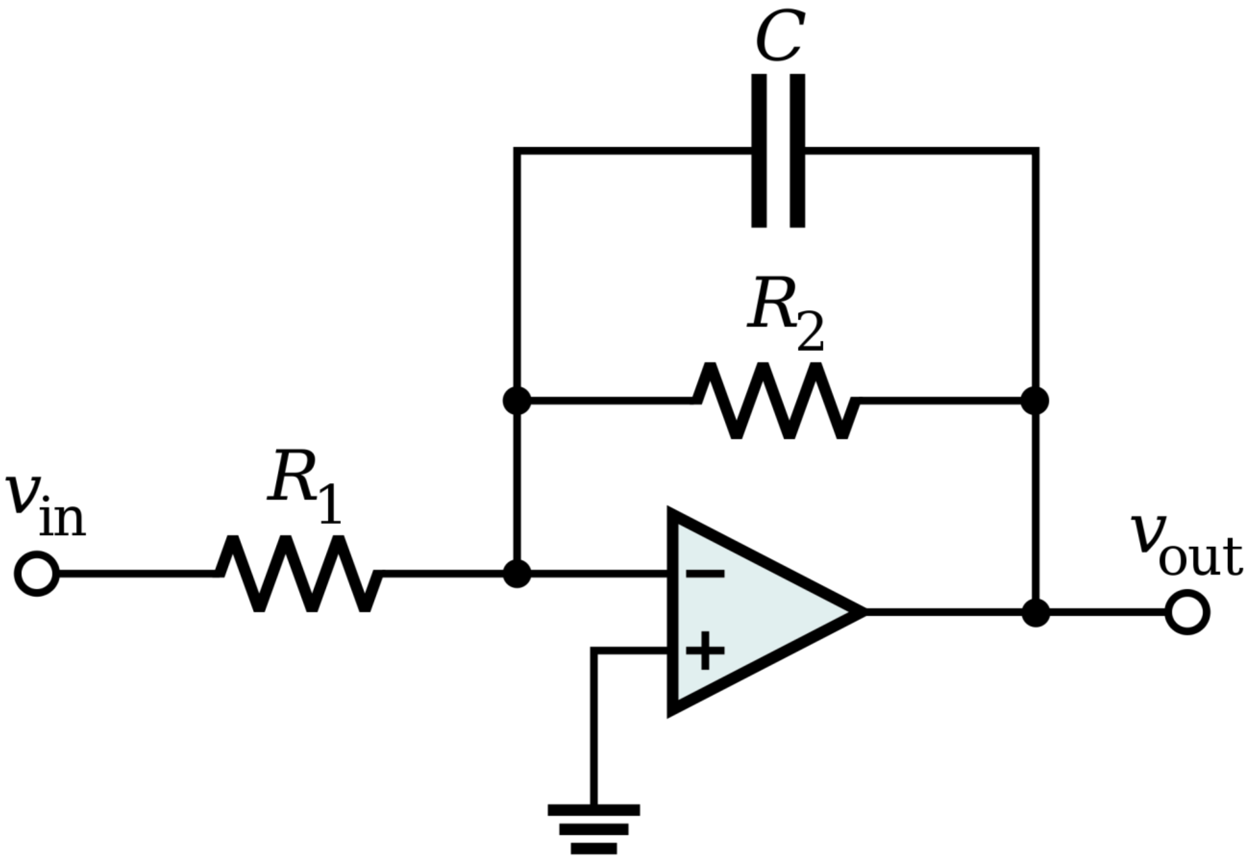


Figure 6.2 Low Pass Filter

6.2.1 High Pass Filter

A high-pass filter is an electronic filter that passes signals with a frequency higher than a certain cut off frequency and attenuates signals with frequencies lower than the cut off frequency. The amount of attenuation for each frequency depends on the filter design. A high-pass filter is usually modelled as a linear time-invariant system. It is sometimes called a low-cut filter or bass-cut filter. High-pass filters have many uses, such as blocking DC from circuitry sensitive to non-zero average voltages or radio frequency devices. They can also be used in conjunction with a low-pass filter to produce a band pass.

Figure shows an active electronic implementation of a first-order high-pass filter using an operational amplifier. In this case, the filter has a passband gain of -R2/R1 and has a cutoff frequency of

$$f_c = \frac{1}{2\pi\tau} = \frac{1}{2\pi R_1 C},$$

Because this filter is active, it may have non-unity passband gain. That is, high-frequency signals are inverted and amplified by R2/R1.

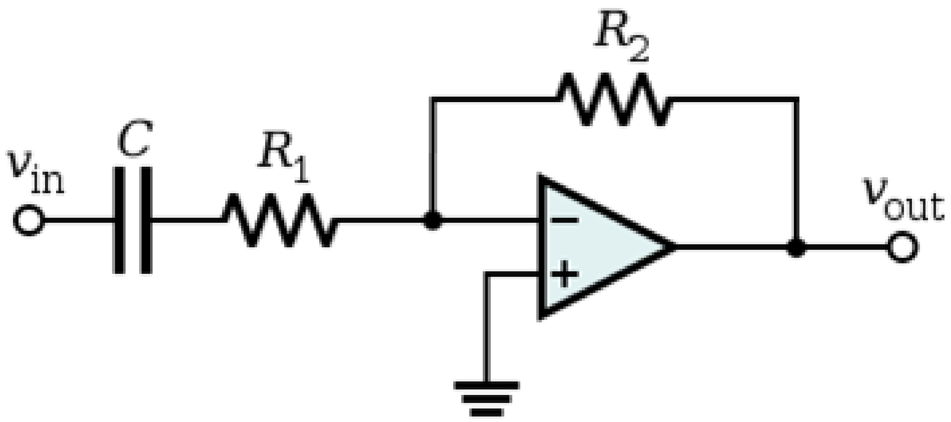


Figure 6.3 High Pass Filter

Chapter 7

Other Hardware & Software Part

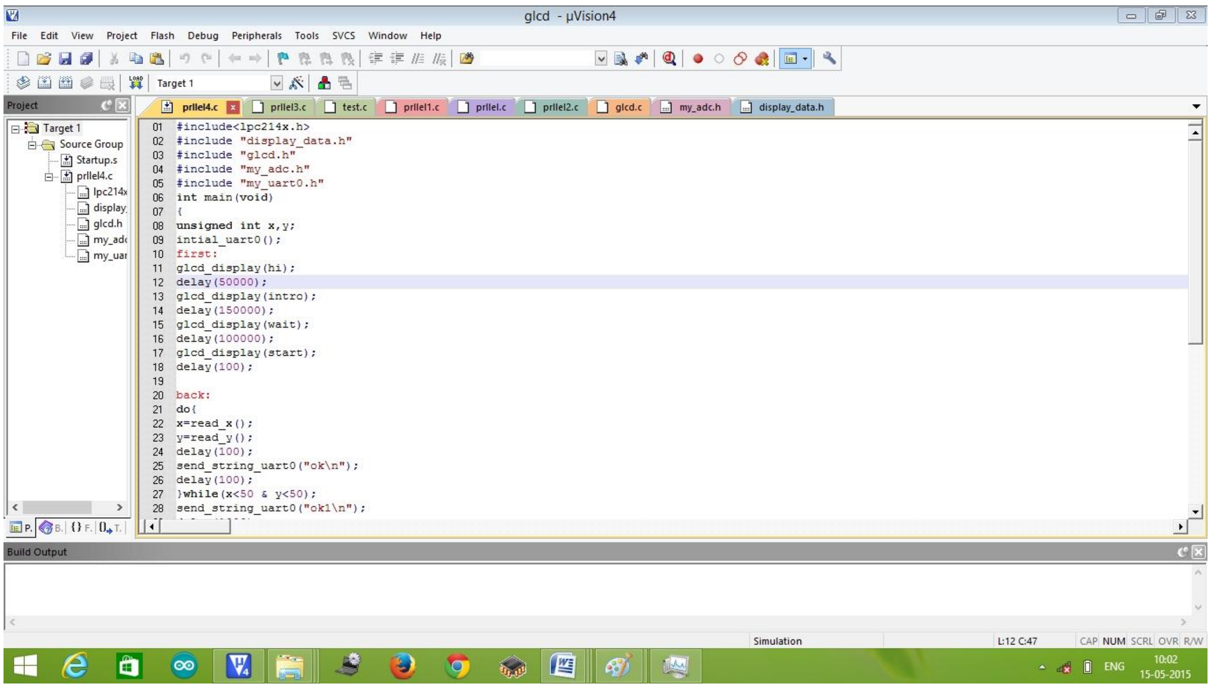
7.0 Introduction

7.1 Cuff

7.2 Micro-air pump

Compact and portable, Series AP Micro Pressure Pumps are a great solution for battery powered applications. They are easily mounted and provide maximum durability. This pump design features a rotary diaphragm and has been tested to 30,000 operating cycles.
Ideal for any industry, these pumps are unique because of their micro size and lower maximum pressure abilities. Please note these pumps only create pressure for air flow.

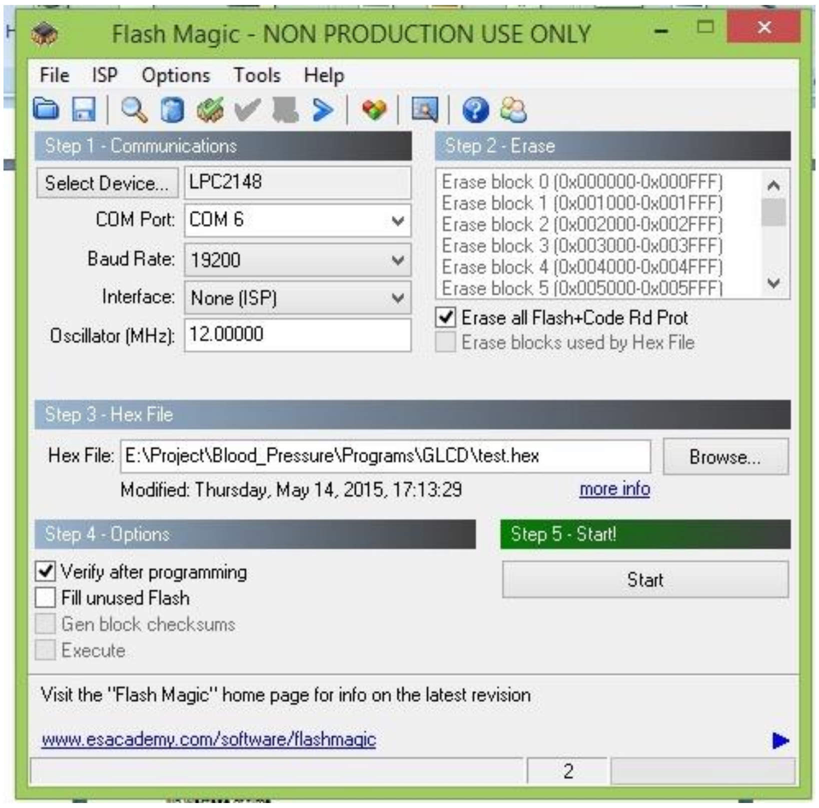
7.3 uKeil


glcd - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
prllel4.c prllel3.c test.c prllel1.c prllel.c prllel2.c glcd.c my_adc.h display_data.h
Target 1
Source Group
Startup.s
prllel4.c
Build Output
Simulation
L:12 C:47
CAP NUM SCRL OVR R/W

```
#include<lpc214x.h>
#include "display_data.h"
#include "glcd.h"
#include "my_adc.h"
#include "my_uart0.h"
int main(void)
{
unsigned int x,y;
intial_uart0();
first:
glcd_display(hi);
delay(50000);
glcd_display(intro);
delay(150000);
glcd_display(wait);
delay(100000);
glcd_display(start);
delay(100);

back:
do{
x=read_x();
y=read_y();
delay(100);
send_string_uart0("ok\n");
delay(100);
}while(x<50 & y<50);
send_string_uart0("ok1\n");
```

7.4 Flash Magic


Flash Magic - NON PRODUCTION USE ONLY

File ISP Options Tools Help

Step 1 - Communications

Select Device... LPC2148

COM Port: COM 6

Baud Rate: 19200

Interface: None (ISP)

Oscillator (MHz): 12.00000

Step 2 - Erase

Erase block 0 (0x000000-0x000FFF)
Erase block 1 (0x001000-0x001FFF)
Erase block 2 (0x002000-0x002FFF)
Erase block 3 (0x003000-0x003FFF)
Erase block 4 (0x004000-0x004FFF)
Erase block 5 (0x005000-0x005FFF)

[✓] Erase all Flash+Code Rd Prot

[ ] Erase blocks used by Hex File

Step 3 - Hex File

Hex File: E:\Project\Blood_Pressure\Programs\GLCD\test.hex Browse...

Modified: Thursday, May 14, 2015, 17:13:29 more info

Step 4 - Options

[✓] Verify after programming

[ ] Fill unused Flash

[ ] Gen block checksums

[ ] Execute

Step 5 - Start!

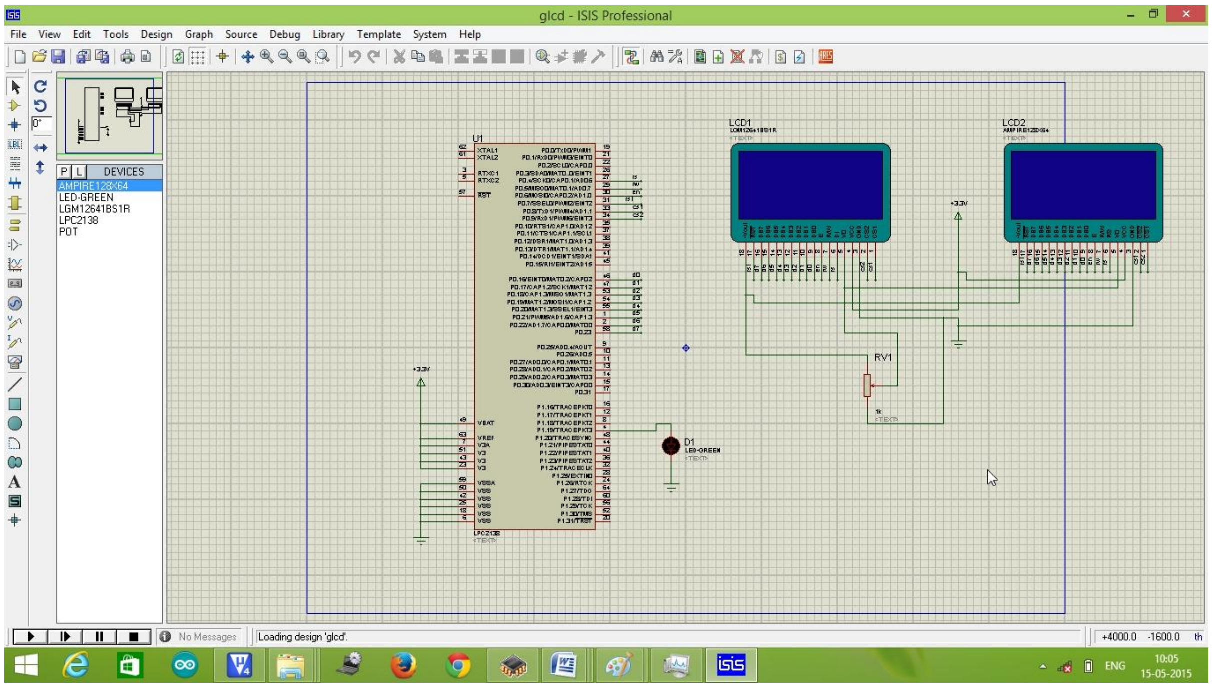
Start

Visit the "Flash Magic" home page for info on the latest revision

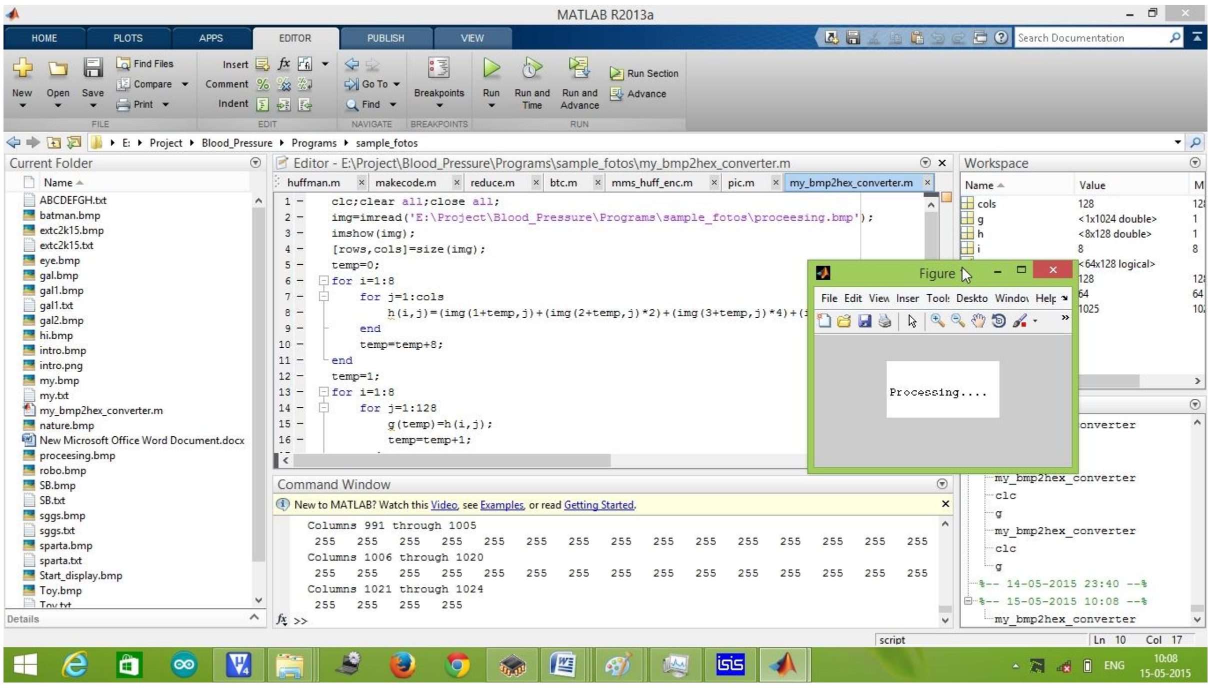
www.esacademy.com/software/flashmagic

2

7.5 Proteus



7.6 Matlab



Editor - E:\Project\Blood_Pressure\Programs\sample_fotos\my_bmp2hex_converter.m

```
clc;clear all;close all;
img=imread('E:\Project\Blood_Pressure\Programs\sample_fotos\proceesing.bmp');
imshow(img);
[rows,cols]=size(img);
temp=0;
for i=1:8
    for j=1:cols
        h(i,j)=(img(1+temp,j)+(img(2+temp,j)*2)+(img(3+temp,j)*4)+(i
    end
    temp=temp+8;
end
temp=1;
for i=1:8
    for j=1:128
        g(temp)=h(i,j);
        temp=temp+1;
```

Command Window

```
  Columns 991 through 1005
   255   255   255   255   255   255   255   255   255   255   255   255   255   255   255
  Columns 1006 through 1020
   255   255   255   255   255   255   255   255   255   255   255   255   255   255   255
  Columns 1021 through 1024
   255   255   255   255
>>
```

```
Processing....
```

Chapter 8

Conclusions & Future Scope

8.0 Advantages:

- Ease of operation
  the monitor should be easier to use than current devices relying on the blood pressure cuff while providing measurements of equal accuracy. The patient should be able to ignore it, but it is desirable for them to be able to determine that it is functioning normally and observe a blood pressure and heart rate measurement if they wish. It should be automatic, so that measurements will be taken regardless of patient behaviour. Also, it should allow manually initiated measurements.
- Low maintenance cost
- Inexpensive
  the blood pressure monitor should be inexpensive on a global scale so that patients in impoverished or developing nations have access to it. The end-product should cost no more than 2000/- and ideally The Project would like it to be 1500/- or less.
  With this in mind, our prototype, which is essentially a testing device, should cost no more than for all materials and components.
- Fit and forget system
- No wastage of time
- Durability
- Accuracy

8.1 Applications:

- Hospitals
- Remote heart rate monitoring applications
- Local monitoring applications
- Designed for Home and Clinical Applications

8.2 Conclusion

We have successfully implemented the interactive user interface part.But failed to implement preamp and instrumentation amplifier (acquiring circuitry for blood pressure data) and thus learned many things.

APPENDIX A

Programs

A.0 ADC

"my_adc.h"

```
#include<lpc214x.h>
/* CONNECTIONS X+ P0.25(AD0.4), Y+ P0.29(AD0.2), X- P0.28(AD0.1), Y- P0.30(AD0.3)*/

#define adclkdiv (14<<8)
#define adpdn (1<<21)
#define adstart (1<<24)
#define adstop ~(1<<24)
int x,y;
unsigned int read_x(void);
unsigned int read_y(void);

unsigned int read_x(void)                    //CONFIGURE Y+ & Y- AS GPIO AND X+ AS ADC
{
unsigned int temp,temp1,i;
//temp=PINSEL1;
//temp1=IO0DIR;

AD0CR=(((adpdn)|(adclkdiv))|(1<<4));
PINSEL1|=(1<<18);             //pin P0.25 AS ANALOG INPUT AD0.4
PINSEL1&=(~(1<<19));
PINSEL1&=(~(((3<<24)|(3<<26))|(3<<28)));     //PIN P0.28, PIN P0.29 AND P0.30 AS GPIO
IO0DIR|=((1<<29)|(1<<30));
IO0DIR&=(~(1<<28));
IO0SET|=(1<<29);
IO0CLR|=(1<<30);
for(i=0;i<1000;i++);
AD0CR|=(adstart);
while(((AD0DR4)&0x80000000)==0x00000000);
x=((AD0DR4>>6) & 0x3FF);
AD0CR&=(adstop);

//PINSEL1=temp;
//IO0DIR=temp1;
return x;
}

unsigned int read_y(void)                    //CONFIGURE X+ & X- AS GPIO AND Y+ AS ADC
{
unsigned int temp,temp1,i;
//temp=PINSEL1;
//temp1=IO0DIR;

AD0CR=((adpdn)|(adclkdiv)|(1<<2));
PINSEL1&=(~((3<<18)|(3<<24)|(3<<28)));
PINSEL1|=(1<<26);
PINSEL1&=(~(1<<27));                         //pin P0.29 AS ANALOG INPUT
IO0DIR|=((1<<25)|(1<<28));
IO0DIR&=(~(1<<30));
IO0SET|=(1<<25);
IO0CLR|=(1<<28);
for(i=0;i<1000;i++);
AD0CR|=(adstart);
while(((AD0DR2)&(0x80000000))==0x00000000);
y=((AD0DR2>>6) & 0x3FF);
AD0CR&=(adstop);
IO0CLR|=((1<<25)|(1<<28));

//PINSEL1=temp;
//IO0DIR=temp1;
return y;}
```

A.1 GLCD

“glcd.h”

```
#include<lpc214x.h>
#define rs 4
#define rw 5
#define en 6
#define rst 7
#define cs1 8
#define cs2 9
#define d 16
void delay(unsigned int n)
{
unsigned int i,j;
for(j=n+1;j>0;j--)
for(i=0;i<100;i++);
}

void glcd_display(char *data)
{
unsigned int i,c=0,p=0;
PINSEL0&=(~0xFFF00);
PINSEL1&=(~0xFFFF);
IO0DIR|=(0xFF03F0);
IO0CLR|=(0xFF03F0);
delay(100);

//RESET
IO0CLR|=(1<<rst);
delay(10);
IO0SET|=(1<<rst);
delay(100);
        //lcd intialization Page @0th Column @0th

        IO0CLR|=(0xFF<<16);
        IO0SET|=(0x3E<<16);
        IO0SET|=((1<<cs1)|(1<<cs2));
        IO0CLR|=((1<<rs)|(1<<rw));
        IO0SET|=(1<<en);
        delay(10);
        IO0CLR|=(1<<en);
        delay(100);

        IO0CLR|=(0xFF<<16);
        IO0SET|=(((0xB8)|(p))<<16);
        IO0SET|=((1<<cs1)|(1<<cs2));
        IO0CLR|=((1<<rs)|(1<<rw));
        delay(1);
        IO0SET|=(1<<en);
        delay(1);
        IO0CLR|=(1<<en);
        delay(1);


        IO0CLR|=(0xFF<<16);
        IO0SET|=(((0x40)|(0))<<16);
        IO0SET|=((1<<cs1)|(1<<cs2));
        IO0CLR|=((1<<rs)|(1<<rw));
        delay(1);
        IO0SET|=(1<<en);
        delay(1);
        IO0CLR|=(1<<en);
        delay(1);


//display data
for(i=0;i<1024;i++)
{
        if(c<64)
        {
```

```
        IO0CLR|=(0xFF<<16);
        IO0SET|=(~data[i]<<16);
        IO0SET|=((1<<cs1)|(1<<rs));
        IO0CLR|=((1<<rw)|(1<<cs2));
        }
        if(c>63 & c<128)
        {
        IO0CLR|=(0xFF<<16);
        IO0SET|=(~data[i]<<16);
        IO0SET|=((1<<cs2)|(1<<rs));
        IO0CLR|=((1<<rw)|(1<<cs1));
        }
        IO0SET|=(1<<en);
        delay(1);
        IO0CLR|=(1<<en);
        delay(1);
        c++;
        if(c==128)
        {
        c=0;
        p++;
        IO0CLR|=(0xFF<<16);
        IO0SET|=(((0xB8)|(p))<<16);
        IO0SET|=((1<<cs1)|(1<<cs2));
        IO0CLR|=((1<<rs)|(1<<rw));
        delay(1);
        IO0SET|=(1<<en);
        delay(1);
        IO0CLR|=(1<<en);
        delay(1);

        }
}

        IO0CLR|=(0xFF<<16);
        IO0SET|=(0x3F<<16);
        IO0SET|=((1<<cs1)|(1<<cs2));
        IO0CLR|=((1<<rs)|(1<<rw));
        IO0SET|=(1<<en);
        delay(1);
        IO0CLR|=(1<<en);
        delay(100);

}
```

A.2 UART

“my_uart0.h”

```
#include<lpc214x.h>
#define Fosc 12000000
#define Fcclk (Fosc*5)
#define Fcco (Fcclk*4)
#define Fpclk (Fcclk/4)*1
#define UART_BPS 9600                                   //Set buadrate here
void intial_uart0(void);
void send_1byte_uart0(char data);
void send_string_uart0(char *str);
void serial_display_uart0(unsigned int temp);

void intial_uart1(void);
void send_1byte_uart1(char data);
void send_string_uart1(char *str);
void serial_display_uart1(unsigned int temp);

void intial_uart0(void)
{
unsigned int Baud16;
PINSEL0|=(1|(1<<2));
```

```
U0LCR=0x83;                                                 // DLAB = 1
Baud16=(Fpclk/16)/UART_BPS;
U0DLM=Baud16/256;
U0DLL=Baud16%256;
U0LCR=0x03;
}

void send_1byte_uart0(char data)         //Function to send a byte on UART0
{
U0THR=data;
while((U0LSR&0x40)==0);
}

void send_string_uart0(char *str)              //A function to send a string on UART0
{
while(1)
{
if(*str =='\0') break;
send_1byte_uart0(*str++);
}
}

void serial_display_uart0(unsigned int temp)
{
unsigned char cn,b[4],b1,b2,b3,b4;
        for(cn=0;cn<4;cn++)
        b[cn]=0;
        cn=0;
        while((temp)!=0)
        {
                b[cn]=temp%10;
                temp=temp/10;
                cn++;
        }
        b1=b[3];
        b2=b[2];
        b3=b[1];
        b4=b[0];
        send_1byte_uart0(b1+'0');
        send_1byte_uart0(b2+'0');
        send_1byte_uart0(b3+'0');
        send_1byte_uart0(b4+'0');
        send_1byte_uart0('\n');
}

void intial_uart1(void)
{
unsigned int Baud16;
PINSEL0|=((1<<16)|(1<<18));
U1LCR=0x83;                                                 // DLAB = 1
Baud16=(Fpclk/16)/UART_BPS;
U1DLM=Baud16/256;
U1DLL=Baud16%256;
U1LCR=0x03;
}

void send_1byte_uart1(char data)         //Function to send a byte on UART0
{
U1THR=data;
while((U1LSR&0x40)==0);
}

void send_string_uart1(char *str)              //A function to send a string on UART0
{
while(1)
{
if(*str =='\0') break;
send_1byte_uart1(*str++);
}
}

void serial_display_uart1(unsigned int temp)
{
unsigned char cn,b[4],b1,b2,b3,b4;
        for(cn=0;cn<4;cn++)
        b[cn]=0;
        cn=0;
        while((temp)!=0)
        {
                b[cn]=temp%10;
                temp=temp/10;
                cn++;
        }
        b1=b[3];
        b2=b[2];
        b3=b[1];
        b4=b[0];
        send_1byte_uart1(b1+'0');
        send_1byte_uart1(b2+'0');
        send_1byte_uart1(b3+'0');
        send_1byte_uart1(b4+'0');
        send_1byte_uart1('\n');
}
```

A.3 Data to be Display

"data.h"

```
char
toy[]={255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,252,0,135,128,192,192,227,255,255,255,254,248,248,249,249,249,253,
253,253,252,252,252,254,
254,254,254,254,254,254,254,254,254,254,254,254,254,254,60,12,1,0,0,0,0,0,0,0,0,0,0,0,0,
240,127,63,31,15,15,7,3,1,0,0,0,0,0,0,0,0,0,0,128,192,224,248,252,252,252,248,248,
252,252,254,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,255,255,255,255,255,255,
255,255,255,255,255,255,255,127,127,63,31,31,15,15,7,7,7,3,3,3,3,3,195,199,231,199,199,199,135
,135,
131,3,3,3,3,35,123,123,3,3,3,1,1,1,1,1,1,1,1,1,0,128,128,128,192,192,224,224,224,
224,224,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,32,112,240,248,248,248,248,248,240,240,251,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,63,15,7,1,0,0,0,0,0,0,0,0,0,0,0,0,128,192,192,
239,255,135,247,243,247,247,1,3,3,251,249,250,2,252,252,252,240,252,248,248,248,240,252,244,25
2,2,250,249,249,
3,1,1,3,251,249,123,243,255,247,224,224,0,0,128,128,0,0,0,0,0,0,0,0,0,0,0,0,1,3,15,255,255,2
55,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,254,252,248,248,240,240,224,224,192,192,192,128,188,254,254,127,255,255,255,255,25
5,255,255,189,245,245,239,
254,254,254,255,127,63,158,207,207,255,63,59,59,59,59,63,63,255,207,142,190,126,254,254,254,25
4,214,254,254,254,
239,255,255,255,255,0,252,253,253,191,158,196,192,192,192,224,224,240,240,248,248,252,252,254,
255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,255,255,255,255,255,255,
127,127,127,127,127,63,63,63,63,63,63,63,31,31,31,31,31,31,62,190,190,190,190,190,190,31,31,31
,31,31,
31,31,31,31,31,31,31,31,63,63,63,63,131,254,255,255,255,255,255,255,255,255,255,255,255,255,25
5,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,255,255,255,255,255,255,
255,255,255,255,255,255,247,231,231,198,198,130,2,2,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,128,129,193,193,227,227,243,251,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,126,62
,28,12,192,32,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,28,62,126,127,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,255,255,255,255,255,255,
255,255,255,127,191,159,95,111,55,23,11,1,1,0,128,192,240,252,159,129,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,128,128,128,128,128,0,0,0,0,0,5,11,19,55,47,79,31,63,127,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,255,255,255,255,255,255,255,255,255,255
},
sparta[]={227,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,159,63,63
,31,63,63,31,31,15,255,255,63,255,255,255,255,255,255,255,255,255,0,0,255,255,253,231,127,255,
255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,255
,255,255,255,255,255,255,255,255,89,7,63,127,255,255,255,255,255,255,255,255,255,223,207,129,1
29,3,3
,3,3,3,15,31,31,15,3,3,7,31,255,255,255,255,255,255,255,255,255,255,1,1,7,15,7,255
,255,255,0,7,63,31,127,255,255,255,255,255,255,255,255,255,255,143,128,192,192,192,192,192,192
,192,143
,143,0,0,231,255,255,1,59,255,255,255,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,
255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,255
,241,120,60,158,239,247,251,191,255,255,255,255,255,255,255,255,255,255,255,255,255,254,254,25
2,252,252,252
,248,248,248,240,179,255,255,255,255,255,255,255,255,255,224,0,0,0,0,63,255,127,240,248,248,25
2,252
,252,253,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,254,253,253,251,246,2
37,223,187,112
,248,255,191,31,0,0,255,255,255,255,255,255,255,252,231,255,255,255,255,255,255,255,255,255,25
5,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,254,191,2
55,255,255,255
,223,255,239,239,247,247,255,63,115,251,121,123,27,3,7,3,3,7,15,15,223,255,191,127,255,255,255
,63,255,255,255,255,255,255,255,0,0,0,0,0,15,255,255,255,127,191,223,255,15,39,119,123,123,59
,3,3,7,1,3,3,7,255,243,255,239,239,199,255,255,255,255,255,255,255,254,249,247,206,0,0,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,247,47,79,95,159,159,63,127,2
55,255,255
,252,248,240,230,228,236,236,236,228,230,224,240,252,255,255,255,255,255,255,255,0,7,63,255,25
5,255,255
,255,0,0,0,0,12,255,255,255,255,255,255,255,255,255,252,248,244,230,228,236,236,228,230,240,24
8,252
,255,255,255,127,127,32,63,63,31,15,15,135,135,195,227,225,195,0,0,255,255,255,255,255,255,255
,127
,63,63,127,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,241,31,254,192,127,253,255,255,255,255,255,255,25
5,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,64,0,0,3,255,255,7,0,0,0,16,4,47
,63,63,63,63,63,31,31,31,143,143,143,143,199,199,199,227,227,227,225,241,241,240,248,120,0,248
,252
,254,254,255,255,255,255,255,255,255,127,0,0,255,127,63,15,7,243,121,61,29,60,124,253,63,31,15
,7,31,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255
,255,255,255,255,255,254,225,31,248,129,3,15,31,63,255,255,255,255,255,255,255,255,255,255,255
,255,255
,255,254,254,255,253,253,251,228,208,96,192,7,31,0,128,224,248,252,252,252,254,254,254,254,255
,255,255
,255,255,255,255,255,255,255,255,255,255,255,127,63,15,7,1,224,255,255,255,255,255,255,255,255
,255,127
,15,1,4,0,0,3,0,0,64,16,7,0,0,0,0,0,127,0,3,4,0,32,64,7,63,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,255
```

```
,254,227,181,240,240,228,244,244,245,245,247,241,229,211,151,175,117,113,247,245,247,251,245,2
43,229,245,241
,241,255,255,255,252,252,255,255,255,255,255,255,253,255,243,255,255,255,255,127,127,191,191,2
23,239,47,135
,99,249,252,254,230,248,224,224,252,255,255,255,255,255,255,255,127,15,1,0,0,0,0,0,0
},
sumit[]={255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,127,127,127,63,63
,63,63,63,31,31,31,31,31,31,15,15,15,15,15,15
,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15
,15,15,15,31,31,31,31,63,63,63,63,127,127,127,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,251,255,255,223,223,95,95,255,255,255,255,255,223,223
,255,255,255,255,252,254,253,252,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,127,127,63,31,31,15,7,7,7,3,3,1,1,1
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,1,1,1,3,7
,7,15,15,31,31,63,127,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,31,63,63,63,127,255,127,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,3,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,32,32,32,48,48,48,56,184,184,176,176,176,160,160
,160,32,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,1,3,7,15,31,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,254,252,252,252,252,252
,248,248,249,249,255,247,255,255,255,255,255,255,255,255,255
,252,240,128,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,2,2,2,3,3,3,7,6,7
,7,135,135,131,3,3,3,3,3,3,1,1,1,1,1
,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,56,63
,63,63,63,63,63,63,63,63,127,127,127,127,255,255,255
,255,255,255,255,255,255,255,249,3,131,251,255,255,255,255
,255,7,7,255,255,255,255,191,31,31,31,159,156,28,48
,32,32,0,128,128,0,0,0,0,192,240,248,252,124,124
,124,124,120,120,56,56,17,17,0,0,0,0,0,0,0
,16,16,16,16,16,28,14,28,28,31,31,31,31,30,30
,28,28,28,24,24,16,0,0,0,0,8,8,24,24,28
,28,60,62,62,126,126,254,252,240,192,224,240,240,248,248
,248,124,127,127,63,63,39,22,62,63,62,30,30,31,31
,31,30,31,31,31,31,30,31,31,31,3,131,243,243,23
,0,224,247,247,247,247,231,231,1,128,238,239,239,239,239
,15,0,192,207,207,223,223,0,0,128,255,255,255,255,254
,252,252,255,255,239,204,128,0,0,0,0,0,0,0,0
,0,0,80,240,224,224,32,64,4,4,4,4,4,12,4
,4,4,4,12,4,4,12,4,4,4,4,0,0,176,160
,160,0,0,0,0,0,0,0,0,0,0,128,192,224,253
,255,255,255,255,255,255,255,224,224,224,240,240,240,240,240
,240,240,240,240,240,240,224,224,224,224,224,224,0,0,0
,0,0,0,0,0,0,0,0,0,1,3,3,7,7,7
,0,0,127,127,255,255,255,31,0,240,255,255,255,255,131
,0,96,127,127,127,63,31,15,7,7,7,7,7,7,7
,7,6,6,60,24,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,1,1,1,1,1,1,1,1,1,0,64,64,96,120,60
,60,30,31,63,63,127,127,127,255,255,255,255,255,255,255
,79,15,15,31,31,31,31,31,63,63,63,27,31,31,7
,7,7,3,3,3,1,0,32,32,224,224,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,3,3,3
},
gal[]={0,0,128,128,128,128,128,128,128,128,192,192,192,0,0
```

```
,0,0,0,0,0,0,0,0,0,0,0,128,128,128,224
,224,240,248,252,127, 63, 31, 15,7,1,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,8, 16,1,0,0,0,0,0,0,0,0,0,0,0
, 96,192,192,192,129,131,3,7, 15, 63,127,255,252,252,248
,248,224,224,192,192,128,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0, 63,255,255, 15, 15, 15,7
,3,3,1,1,1,0,0,224,224,224,240,240,248,252,252
,254,254,255,251,243,243,227,227,227,227,225, 96,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,128,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,1,3,7,7, 15
, 15, 94,192,192,191,255,255,255,255,255,255,255,255,255,255
,254,252,252,248,240,192,192,128,0,0,0,0,0,0,0
,0,204,231,249,248,252,254,246,192,192,128, 48,120,176,160
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255, 15,1,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,128,192,240,248,127,127,126,120,240,192
,128,0,128,128,0,0,0, 16,224,192,128, 56,112, 96, 64
,204,240,224,128,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,192,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,254,252,248,224,192,128,0,219,239,239,247,255,251
,255,253,255,254,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,240,0,0,0,0
,0,128,240, 12,1,0,0,0,0,224,240,252,230,225,253
,254,198,135,130,160,160,255,255,255,254,255,255,255,254,252
,248,255,255,255,195,135,135,128,160,224,253,249,247,254,252
,224, 32,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,3,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,0,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,248,224,192,128,0,127,255,128,0,0,0
,0,0,241,3, 31, 63,127,127,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,127,127, 63, 30,1,0,128,0,0,0,0
,128, 16,0,0,0,128,192,192,224,248,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,0,255,255,255,255,255
,191,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,240,128, 48, 32, 96,111,254,248,224,192
,128,129,7, 31,127,255,255,255,255,255,252,252,253,253,253
,253,252,252,254,255,255,255,255,255, 63, 31,130,128,192,192
,224, 56, 28,3,0,0,128,242,241,248,252,254,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,0,127,255,255,255,255,255,255,255,239,239,255,255
,255,127,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255, 63, 15,255,255,255,255,255,255,255
,255,254,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,254,254,254,255,255
,255,255,255,255,255,255,255,255,127, 31,159,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,0,191,255,255,187
,223,127,255, 63, 63, 61,0, 25,0,192,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,128
,0, 63,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,127,127,127, 63, 63, 63, 63, 31
, 31, 63, 63, 63, 63,127,127,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,127, 15
,3,224,252,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255, 63, 31
,7,4,0,0},
sumit1[]={255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,127,127
,63,63,31,31,31,31,15,15,15,7,7,3,3,3,3
,3,1,3,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1,1,1,1,1,1,1,3,3,3,3,7
,7,7,7,15,15,15,15,31,31,31,63,63,127,127,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
```

```
,255,255,255,255,255,255,255,255,255,255,255,255,127,31,15
,15,7,3,1,1,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,128,128,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,1,1,3,3,7,15,31
,63,127,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,31,7,1,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,192,128,192,192,192,224,192,192
,252,252,255,255,226,243,224,224,255,254,255,255,255,255,255
,254,254,255,254,254,254,254,252,252,248,176,184,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,3,127,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,254,248,240,224,128
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,2,7,15,7
,15,15,7,15,15,15,15,15,15,15,15,15,15,15,7
,7,7,7,7,7,7,7,7,3,3,3,3,3,3,3
,1,1,17,48,48,49,48,48,48,48,32,96,192,224,192
,0,0,0,224,240,248,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,240,224,192,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,198,255,255,255,255,254,252,252,248,252,248,248,240
,240,224,224,224,224,224,192,192,192,192,226,226,227,224,224
,240,240,252,252,254,255,255,255,255,248,0,63,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,243,240,192,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,3,7,7,7,7
,15,15,7,7,7,7,7,103,103,255,127,121,121,113,3
,3,3,7,15,15,15,15,15,15,15,31,31,31,15,31
,15,7,7,224,240,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,254,252,248,240,224,192,128,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,16,16,0,0,16,16,16,16,16,16,16,16,16,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,128,192,224,240,248,254,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,191,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,127,124,96,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,2,2,0,2,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,30,63,127,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255},
extc2k15[]={255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
```

```
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,127,127,127,127,127,127,127,127,63,63,255,255,255,255
,255,255,255,127,127,255,255,255,255,255,255,255,255,127,127
,127,127,127,127,127,127,255,255,255,255,127,127,127,127,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,127,127,127,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,127,63,15,135,128,152,222,223,223,223,255,255
,255,255,255,0,0,15,143,199,227,241,248,252,254,255,253
,253,60,30,134,224,240,252,255,255,255,255,127,31,15,199
,227,249,188,158,142,195,240,252,255,159,159,207,239,111,15
,15,255,255,255,63,15,7,131,217,228,240,248,254,255,255
,191,159,31,15,7,199,247,255,255,63,31,7,71,103,119
,247,247,247,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,63,7,1,112,60,191,191
,191,191,191,127,63,31,143,199,227,241,248,224,0,7,255
,255,255,255,255,255,127,15,3,224,248,254,255,255,255,255
,255,255,7,1,32,252,255,127,63,159,207,231,123,63,31
,15,71,103,99,113,120,252,254,63,15,131,192,0,0,124
,62,158,207,239,255,255,127,15,131,225,248,254,255,255,239
,135,135,190,190,223,199,224,240,253,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,254,254,254,255,255,255,255,255,255,255,254,254,255,255
,255,255,255,255,255,252,252,252,255,255,255,255,255,254,254
,255,255,255,255,255,255,255,255,255,255,255,254,254,254,254
,254,255,255,255,255,254,254,254,254,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255
},
sggs[]={255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,127,31,15,199,231,99,51,19,131,199,255,63
,31,15,199,227,243,49,1,131,255,127,31,15,135,199,227
,113,17,3,255,255,63,31,143,199,227,115,51,3,131,255
,255,127,31,7,195,243,255,63,7,3,195,243,243,249,249
,249,249,251,255,31,15,199,7,15,159,255,255,255,255,255
,255,255,207,207,199,231,103,7,3,195,243,243,243,243,243
,243,243,255,255,255,255,255,63,15,7,3,247,255,63,15
,131,225,251,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,127,127,255,224,128,15,57
,248,252,254,31,7,129,240,248,254,255,115,113,248,60,14
,3,224,248,252,255,127,113,240,248,126,127,127,240,192,6
,29,120,252,252,254,127,31,3,192,248,254,127,15,3,192
,240,248,249,249,253,253,255,63,31,143,231,227,0,0,201
,12,14,207,231,231,231,239,255,255,255,255,31,7,129,240
,252,255,255,255,255,255,255,255,255,255,255,63,15,195,240
```

```
,60,0,64,31,7,192,248,126,63,15,207,7,7,255,127
,31,15,31,31,143,223,127,31,143,255,255,255,255,255,255
,224,192,206,207,207,199,226,240,248,255,255,240,224,231,35
,3,1,224,248,254,255,240,224,227,231,19,1,192,240,252
,255,240,224,198,206,207,199,227,224,240,255,255,195,192,248
,255,255,255,195,192,224,227,243,243,243,243,251,255,255,255
,224,192,199,207,207,198,224,192,136,158,143,207,255,255,255
,255,255,131,128,240,254,255,255,255,255,127,15,7,239,255
,255,255,131,192,248,255,255,255,192,192,192,240,255,255,255
,224,224,227,224,192,206,227,224,248,192,192,254,255,241,224
,226,225,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,241,224,248,255,255,255,255,255,255
,255,247,224,240,254,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,249,248,252,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255
};
```

Appendix B

B.0 Matlab Code To Generate Image Data

“my_bmp2hex_converter.m”

```
clc;clear all;close all;
img=imread('E:\Project\Blood_Pressure\Programs\sample_fotos\proceesing.bmp');
imshow(img);
[rows,cols]=size(img);
temp=0;
for i=1:8
    for j=1:cols

h(i,j)=(img(1+temp,j)+(img(2+temp,j)*2)+(img(3+temp,j)*4)+(img(4+temp,j)*8)+(img(5+temp,j)*16)+(img(6+temp,j)*32)+(img(7+temp,j)*64)+(img(8+temp,j)*12
8));
    end
    temp=temp+8;
end
temp=1;
for i=1:8
    for j=1:128
        g(temp)=h(i,j);
        temp=temp+1;
    end
end
display(g);
```

Appendix C

Main Program

```
#include<lpc214x.h>
#include "display_data.h"
#include "glcd.h"
#include "my_adc.h"
#include "my_uart0.h"
int main(void)
{
unsigned int x,y;
intial_uart0();
first:
glcd_display(hi);
delay(50000);
glcd_display(intro);
delay(150000);
glcd_display(wait);
delay(100000);
glcd_display(start);
delay(100);

back:
do{
x=read_x();
y=read_y();
delay(100);
send_string_uart0("ok\n");
delay(100);
}while(x<50 & y<50);
send_string_uart0("ok1\n");
delay(1000);

x=read_x();
delay(1000);
y=read_y();
delay(100);
send_string_uart0("ok2\n");
delay(1000);
if(810<x<840 & 250<y<580)
{
send_string_uart0("ok3\n");
delay(1000);
glcd_display(processing);
delay(500000);
}
else
{
goto back;
}
goto first;
}
```

Appendix D

D.0

http://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CCcQFjAA&url=http%3A%2F%2Fwww.nxp.com%2Fdocuments%2Fdata_sheet%2FLPC2141_42_44_46_48.pdf&ei=AHxVVbiNLIy3uATFlIDICw&usg=AFQjCNHI4ODAQoUZ2hwAH0fTnK-PZQfC6Q&sig2=GjBR5tXoHR9SZ0GMYc4dCQ&bvm=bv.93564037,d.c2E

D.1

http://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CB0QFjAA&url=http%3A%2F%2Fwww.agspecinfo.com%2Fpdfs%2FJ%2FJHD12864.PDF&ei=VHxVVaTgOsXHuATcqYCwCw&usg=AFQjCNHJlX9ZU3TjeMAGjvelKnAgTIoonQ&sig2=4dcWlr-c_F6ywJiLhHVDmw&bvm=bv.93564037,d.c2E

D.2

http://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&sqi=2&ved=0CCsQFjAA&url=http%3A%2F%2Fcache.freescale.com%2Ffiles%2Fsensors%2Fdoc%2Fdata_sheet%2FMPX2050.pdf&ei=Z3xVVf3vJcyjugSL44LACw&usg=AFQjCNGq9q24G21wndCwdgph63-jtbH-OA&sig2=kbO-tJDVspexmn01RuU3Mg&bvm=bv.93564037,d.c2E

D.3

http://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&sqi=2&ved=0CB0QFjAA&url=http%3A%2F%2Fwww.ti.com%2Flit%2Fds%2Fsymlink%2Flf357.pdf&ei=inxVVejKHIe9ugTylIC4Cw&usg=AFQjCNFkDDBym2EedMFM5nmuVY-hJyXdZw&sig2=K51jhBmENNlsyFPKdo0JiA&bvm=bv.93564037,d.c2E