PRESSURE SENSING SYSTEM

An undergraduate mini project synopsis submitted to Department of Biomedical Engineering in partial fulfillment of the requirement for the

MICROCONTROLLERS LAB (BME3111)

In

V semester Biomedical Engineering

Submitted by

Dwija Thakar - 170902018

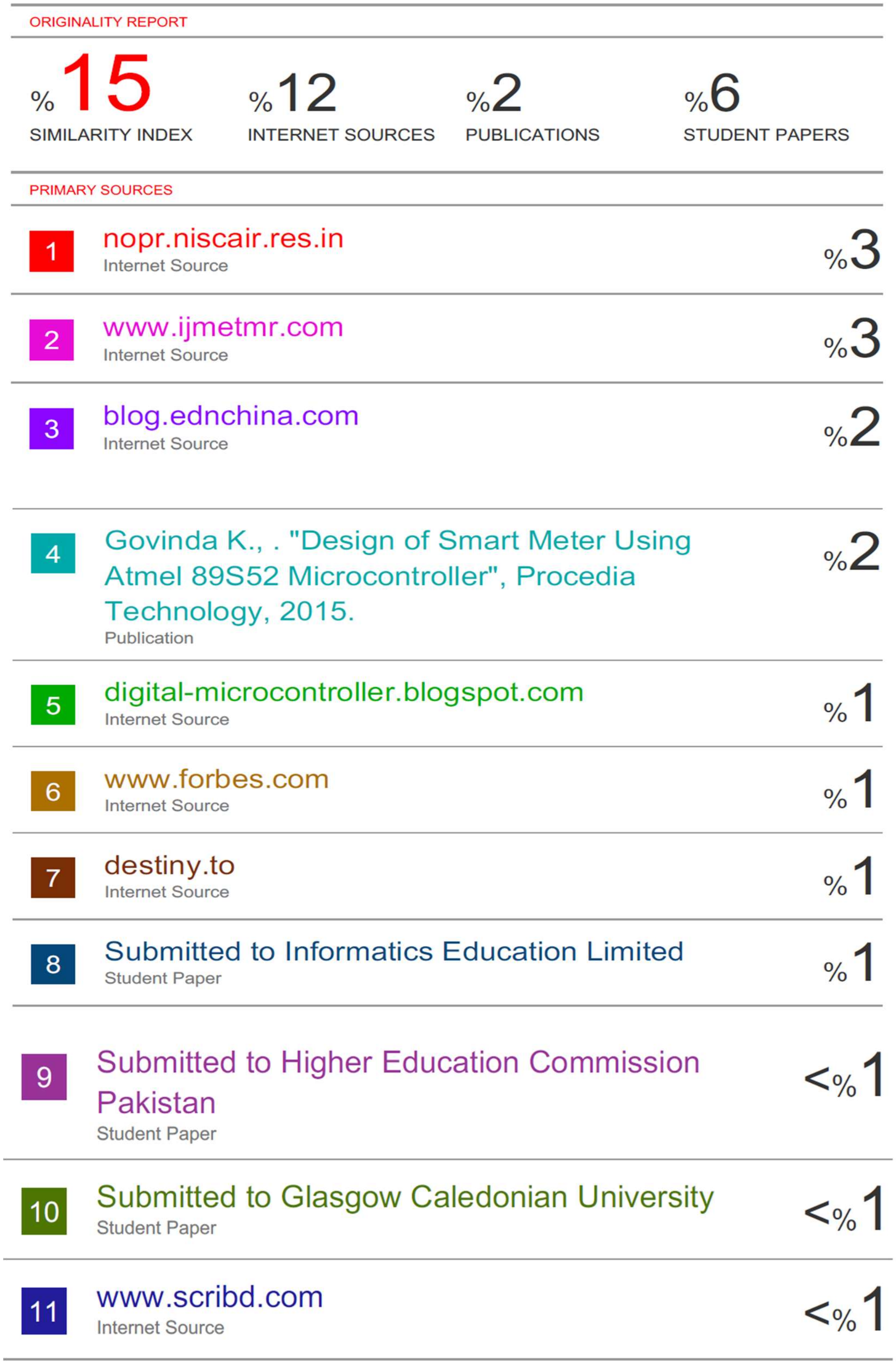
Mohammed Sinad - 170902020

Gopika Satish - 170902005

DEPARTMENT OF BIOMEDICAL ENGINEERING
MANIPAL INSTITUTE OF TECHNOLOGY
(A Constituent College of Manipal Academy of Higher Education)
MANIPAL – 576104, KARNATAKA, INDIA

November 2019

TURNITIN SIMILARITY REPORT



ORIGINALITY REPORT

%15	%12	%2	%6
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	nopr.niscair.res.in Internet Source	%3
2	www.ijmetmr.com Internet Source	%3
3	blog.ednchina.com Internet Source	%2
4	Govinda K., . "Design of Smart Meter Using Atmel 89S52 Microcontroller", Procedia Technology, 2015. Publication	%2
5	digital-microcontroller.blogspot.com Internet Source	%1
6	www.forbes.com Internet Source	%1
7	destiny.to Internet Source	%1
8	Submitted to Informatics Education Limited Student Paper	%1
9	Submitted to Higher Education Commission Pakistan Student Paper	<%1
10	Submitted to Glasgow Caledonian University Student Paper	<%1
11	www.scribd.com Internet Source	<%1

ACKNOWLEDGEMENTS

Sincere thanks and gratitude to Mr. Devadas Bhat for constantly supporting us throughout the project. His valuable guidelines and inputs was a constant source of motivation and improvement for us. We also extend our sincere respect and thanks to the lab assistants microcntrollers lab for letting us use some of their equipments and working space. Without their help we could not have completed our project.

ABSTRACT

A sensor is a device which converts a certain input stimulus to it's corresponsing electrical output. It is like a transducer. The piezoelectric sensor used in this project works on the simple principle of piezoresistivity. On application of force the dimensions of the material changes causing the electrons to move in a certain direction generating electricity and volatge. Using 8051 microcontroller a pressure sensing system has been deviced which finds its use in various biomedical and non biomedical applications like-

1. to measure the body weight (on either sides/ full body weight) leading to gait analysis
2. to measure the amount of pressure a person suffering from Parkinson's with trembling fingers/hand can exert
3. for heart rate measurement
4. to measure water levels in borewells using drop testing
5. manomerters
6. automotive and aerospace industry
7. safety circuit in a spa system
8. medical ventilators as pressure flow sensor

The curent circuit designed measure pressure in the range of 9 to 250Kpa.

The piezoelectric sensor is connected to ADC which converts the analog output by force exerted by the finger on the sensor to digital output. Hence the ADC is interfaced to the microcontroller as microcontroller can only read digital data. The LCD screen is interfaced with the microcontroller which acts as a display unit for th output.

CONTENTS

CHAPTER 1 – INTRODUCTION

Introduction

Our project aims to device an easy to use, cost effective primitive model of pressure sensing system that could highlight the pressure exerted by a certain body part.

Introduction to the area of work

A sensor is a device which converts a certain input stimulus to it's corresponsing electrical output. It is like a transducer. The piezoelectric sensor used in this project works on the simple principle of piezoresistivity. On application of force the dimensions of the material changes causing the electrons to move in a certain direction generating electricity and volatge. Using 8051 microcontroller a pressure sensing system has been deviced which finds its use in various biomedical and non biomedical applications.microcontroller is used as it is a system on chip device which is easily programmable, has programmable input output peripherals, a processing core, 128 bytes of RAM and 4Kbyte of ROM. It can easily provide us with a digital output.

Brief present-day scenario regarding the work area

1. measures the body weight (on either sides/ full body weight) leading to gait analysis
2. measures the amount of pressure a person suffering from Parkinson's with trembling fingers/hand can exert
3. used in heart rate measurement
4. measures water levels in borewells using drop testing
5. manomerters
6. used automotive and aerospace industry
7. used in safety circuit in a spa system
8. measure pressure flow in medical ventilaor
9. drug infusion
10. providing pressure masks to treat sleep apnea

The curent circuit designed measure pressure in the range of 9 to 250Kpa.

Motivation to do the project work

To come up with an improvised version of the already existing circuit which can later be implemented to –

1. measure the body weight (on either sides/ full body weight) leading to gait analysis
2. measurs the amount of pressure a person suffering from Parkinson's with trembling fingers/hand can exert
3. measure heart rate
4. measure amount of pressure exerted by the soled of foot in a diabetic patient (foot plantar pressure measurement
5. measure pressure flow in medical ventilaor
6. automatic drug infusionproviding pressure masks to treat sleep apnea.

Objective of the work

- Primary Objective
  The aim of the project is to device a simple, safe, compact, user friendly, cost effective pressure sensing system using 8051 microcontroller
- Secondary Objective
  With this as a primitive model, to design higher end models that can be used in gait analysis

Target Specifications

<u>Importance of the result:</u>

The pressure exerted by the finger or hand on the sensor can be easily measured and determined which can be used for gait analysis.

Project work schedule

MONTH	**ACTIVITY**	**STATUS**
August 2019	Collecting of components	completed
September 2019	Model on developer’s board	completed
October 2019	Code generation	completed
November 2019	Final circuit designing and assembling	completed

Table 1

CHAPTER 2 - LITERATURE REVIEW

Introduction

Our project aims to device an easy to use, cost effective primitive model of pressure sensing system that could highlight the pressure exerted by a certain body part.

Introduction to the project title

A sensor is a device which converts a certain input stimulus to it's corresponsing electrical output. It is like a transducer. The piezoelectric sensor used in this project works on the simple principle of piezoresistivity. On application of force the dimensions of the material changes causing the electrons to move in a certain direction generating electricity and volatge. Using 8051 microcontroller a pressure sensing system has been deviced which finds its use in various biomedical and non biomedical applications. A microcontroller is used as it is a system on chip device which is easily programmable, has programmable input output peripherals, a processing core, 128 bytes of RAM and 4Kbyte of ROM. It can easily provide us with a digital output.

Present state/ recent developments in the work area

Nowadays, research is oriented in such a manner where all the sensors are being manufactured as solid-state silicon sensors and optical-fiber devices. Miniaturization of the sensors is being done using semiconductor fabrication techniques of microelectronics. The unique advantage of silicon derived sensors is that by combining the sensor and the signal electronics on the same silicon chip, "integrated smart or intelligent sensors" have been fabricated. Due to the combination and participation of material scientists, physicists, chemists, biologists and engineering experts, the field of "physical sensors" is advancing rapidly through the discovery of better and newer sensing materials, replacement of conventional electronic devices by optical-fiber devices, and the implementing novel designs.

Brief background theory

A sensor is a device which converts a certain physical input stimulus to it's corresponsing electrical output. It is like a transducer. The piezoelectric sensor used in this project works on the simple principle of piezoresistivity. On application of force the dimensions of the material changes causing the electrons to move in a certain direction generating electricity and volatge.

Theoretical discussions and general analysis

8051 microcontroller is a system on chip device which is easily programmable, has programmable input output peripherals, and a processing core.

1. It has 128 bytes of RAM and 4Kbyte of ROM.
2. It can easily provide us with a digital output.
3. 2 timers
4. 8 bit data bus and 16 bit address bus
5. 3 internal and 2 external interrupts
6. Four 8 bit ports
7. 1 microsecond instruction cycle with 12MHz crystal

8. 16 bit program counter and data pointer

The piezoelectric sensor is connected to ADC which converts the analog output of force exerted by the finger on the sensor to digital output. Hence the ADC is interfaced to the microcontroller as microcontroller can only read digital data. The LCD screen is interfaced with the microcontroller which acts as a display unit for the output.

CHAPTER 3 METHODOLOGY

Our project aims at developing an easy to use, cost effective model of pressure sensing system that could highlight the pressure exerted by the certain body part. In this chapter the detailed methodology and working of the circuit, circuit layout, block diagrams, component lists, detailed component specification and conclusions that we obtained from our preliminary result will be discussed.

Working of the circuit

1. The circuit consists of a 5 V regulated power supply that is obtained using LM7805 voltage regulator, powering up all the components of the circuit. The piezoelectric disc is a sensor that works on piezoelectric effect, to measure changes in voltages corresponding to the mechanical strain applied on it. This voltage (Vin) ranging from 0 to 5V is applied as the Analog input to ADC0804, which converts it to an 8-bit digital value ranging from 0 to 255.
2. For the ADC to start converting, a start conversion signal as a pulse is sent into the ADC, sampling the analog input value to give a corresponding binary value. When the conversion is completed, the ADC activates its data ready output signal. This signal is basically used to inform the Microcontroller/ processor to start reading the binary data. The start conversion pulse is applied to pin 3(WR), and the data ready signal is from pin 5(INTR) of ADC.
3. The output of the ADC O804 (binary) is given as input to AT89S52 microcontroller whose Port1 was configured as the input port and Port 2 was configured as the output port. Inside the controller the pressure and voltage are related using the formula Vo = Vs* (0.004 x P-0.04), where Vs is the supply voltage and P is pressure in KPa (kilo Pascal). 101325 Pascal is one atmosphere, and the piezoelectric sensor measures pressure between 0 and 250 KPa, so it can measure up to 2.5 times the normal atmospheric pressure.
4. The output of the microcontroller from port P2 is given to 16x2 LCD display so the changes in the voltage from piezoelectric sensor is converted into pressure using the formula in the controller and could display the pressure on the LCD display. The contrast of the LCD display could be adjusted using a potentiometer connected to Vo pin of LCD.

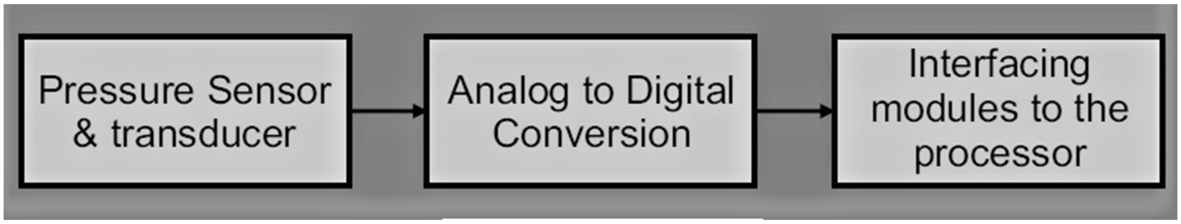

Pressure Sensor & transducer
Analog to Digital Conversion
Interfacing modules to the processor

Figure 1: Block Diagram

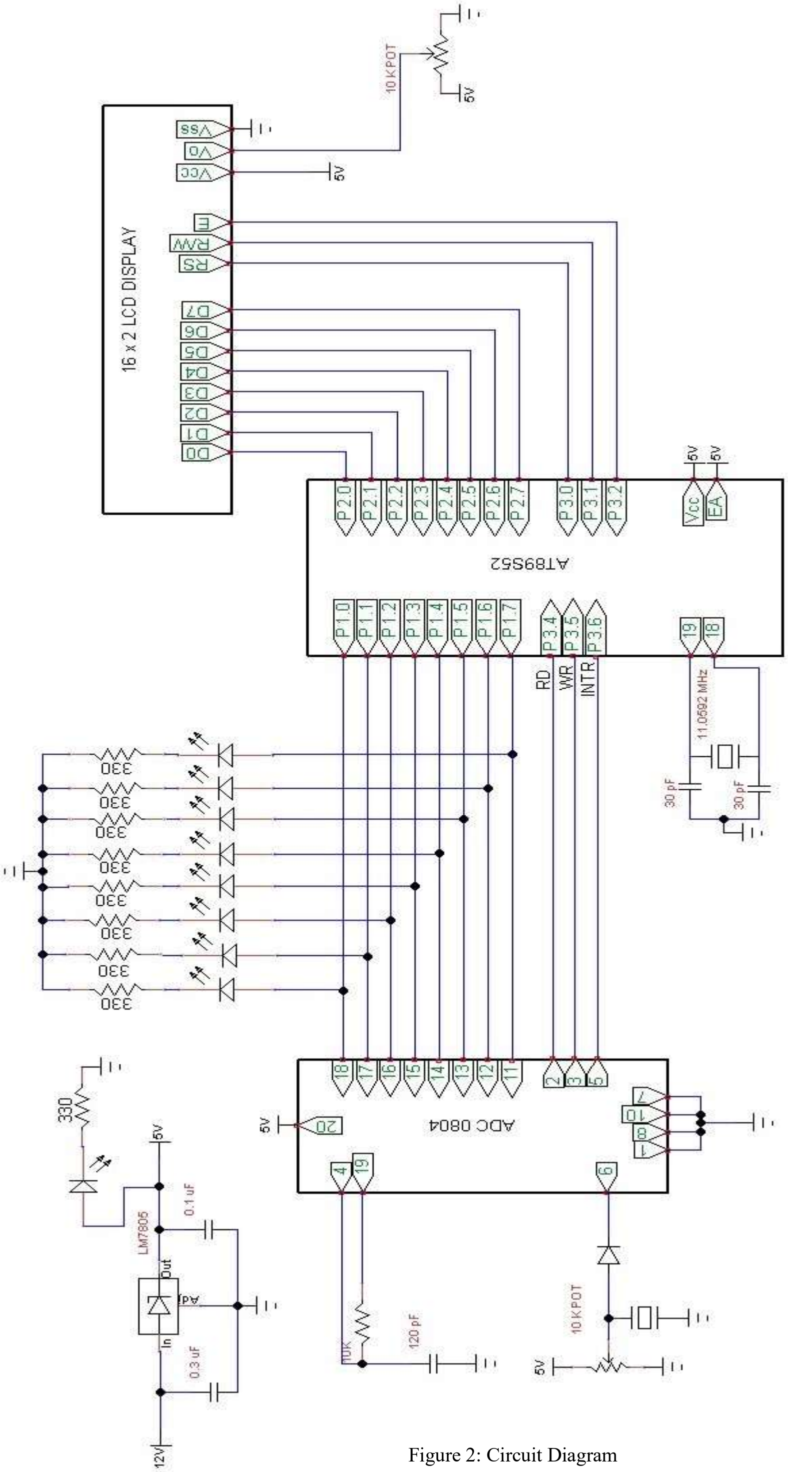


Figure 2: Circuit Diagram

COMPONENT LIST

Table 2

COMPONENT	VALUE	QUANTITY
AT89S52	-	1
ADC 0804	-	1
16X2 LCD Display	-	1
LM7805	-	1
LED	-	9
Piezoelectric disc	-	1
DC Jack for power supply	-	1
Resistor	330Ω	9
Potentiometer	10KΩ	2
Crystal oscillator	11.0592MHZ	1
Capacitor	30pF	2
Capacitor	0.3μF	1
Capacitor	0.1μF	1
Capacitor	120pF	1
Resistor	10KΩ	1

- AT89S52 - It is identical to the 8051, with the addition of 8KB on chip program ROM, a 3rd 16-bit timer/counter with 16-bit auto-reload, and a capture module that works with timer 2. The 8052 also has an additional 128B of RAM, making for a total of 256 bytes of on chip RAM space. The upper 128 bytes of RAM can only be indirectly addressed. The AT89S52 is Atmel's derivative of the 8052. It is compatible with Atmel's In-System Programming (ISP) via an on-chip SPI interface, which allows it to be programmed via Atmel's AT89ISP programmer while the MCU is in the target circuit.

- ADC 0804 IC (since microcontroller can read only digital codes)
  This is a 20 pin Analog to Digital converter. It converts analog voltage produced from differential pressure transducer to 8-bit digital value that could lie anywhere between 0 to 255.It can take only one Analog signal as input.
  - ➔ To activate the ADC the Chip, select pin (pin no:1 of the ADC) should be grounded.
  - ➔ The Read pin of ADC should be set to ground (logic 0) to be able to read the Analog value and the Write pin must be pulsed high to begin the data conversion.
  - ➔ It has an interrupt pin which goes high on completion of the conversion from Analog value to 8-bit digital value which generates an interrupt request, informing the 8051 microcontroller that the 8-bit digital value data is ready.
- LCD display for displaying value of pressure in kPa.
- LM 7805 for maintaining a 5v supply line.

CHAPTER 4 RESULTS AND DISCUSSION

The piezoelectric sensor is connected to ADC which converts the analog output of force exerted by the finger on the sensor to digital output. Hence the ADC is interfaced to the microcontroller as microcontroller can only read digital data. The LCD screen is interfaced with the microcontroller which acts as a display unit for the output.

When the circuit is run, we get the basic output as 9 kPa. This is due to the atmospheric interaction with the piezoelectric sensor. Then the pressure varies according to the force given on the piezoelectric sensor.

CHAPTER 5 CONCLUSION AND FUTURE SCOPE

Through this project, our chief objective is to build an effective and a basic pressure sensing system using an 8051 microcontroller. Our aim was to create a good foundation on which we can further add on while developing the future projects by incorporating modification that can improve its application in the healthcare field.

So, this system can be used to measure a given pressure.

Also, the future scopes are as given below

1. Measure the pressure exerted on each leg thus being able to correct the gait of a affected person
2. Measures the amount of pressure a person suffering from Parkinson’s Disease with trembling fingers/hand can exert.
3. Measure amount of pressure exerted by the soled of foot in a diabetic patient (foot plantar pressure measurement.

REFERENCES

Reference / Handbooks

[1] The 8051 Microcontroller and Embedded Systems, Pearson Publications, 2nd Edition, ISBN 978-81-317-10026-5

Web

[1] www.electronicsmaker.com
[2] www.allsensors.com
[3] www.rickeysworldofelectronics.com

ANNEXURES

Code:

```
#include<reg51.h>
#define dataport P2
#define adc_input P1
sbit rs = P3^0;
sbit rw = P3^1;
sbit e = P3^2;
sbit rd= P3^4;
sbit wr= P3^5;
sbit intr= P3^6;
void delay (unsigned int);
void lcd_cmd (unsigned char);
void lcd_data (unsigned char);
void lcd_data_string (unsigned char*);
void lcd_init ();
void adc_conv ();
void adc_read ();
void lcd_data_adc (unsigned int);
int num [10];
float v_out, pressure;


void main ()
{
        dataport=0x00;
        adc_input=0xff;
        P3=0x00;
        lcd_init ();
        lcd_cmd(0x84);
        lcd_data_string("PRESSURE");
        while (1)
        {
                lcd_cmd(0xc5);
                adc_conv ();
                adc_read ();
                lcd_data_string (" KPA");
                delay (50);
        }
}


void lcd_init ()
{
        lcd_cmd(0x01);
        delay (1);
        lcd_cmd(0x06);
        delay (1);
        lcd_cmd(0x0e);
        delay (1);
        lcd_cmd(0x38);
        delay (1);
        lcd_cmd(0x80);
        delay (1);
}

void delay (unsigned int sec)
{
        int i, j;
        for (i=0; i<sec; i++)
        for (j=0; j<1275; j++);
}

void lcd_cmd (unsigned char item)
{
        dataport=item;
        rs= 0;
        rw=0;
        e=1;
        delay (1);
        e=0;
        return;
}

void lcd_data (unsigned char item)
{
        dataport = item;
        rs= 1;
        rw=0;
        e=1;
        delay (1);
        e=0;
}

void adc_conv ()
{
        wr = 0;
        delay (2);
        wr = 1;
        while(intr);
        delay (2);
        intr=1;
}

void adc_read ()
{
        unsigned int value;
        rd = 0;
        delay (2);
        value=adc_input;
        v_out= (5.0/256) *value;                        //divide by 256 as it's an 8-bit converter
        pressure=((v_out/5.0) +0.09)/0.009-1;       // convert output voltage(volt) to KPA
        delay (1);
        rd=1;
        lcd_data_adc(pressure);
}

void lcd_data_adc (unsigned int i)
{
        int p;
        int k=0;
        while(i>0)
        {
                num[k]=i%10;
                i=i/10;
                k++;
        }
        k--;
        for (p=k; p>=0; p--)
        {
                dataport=num[p]+48;
                rw = 0;
                rs = 1;
                e = 1;
                delay (1);
                e = 0;
        }
}

void lcd_data_string (char *x)
{
        while (*x! ='\0')
        {
                lcd_data(*x);
                delay (2);
                x++;
        }
        }
```

Data Sheet for AT89S52

Program Memory

If the EA pin is connected to GND, all program fetches are directed to external memory. On the AT89S52, if EA is connected to VCC, program fetches to addresses 0000H through 1FFFH are directed to internal memory and fetches to addresses 2000H through FFFFH are to external memory.

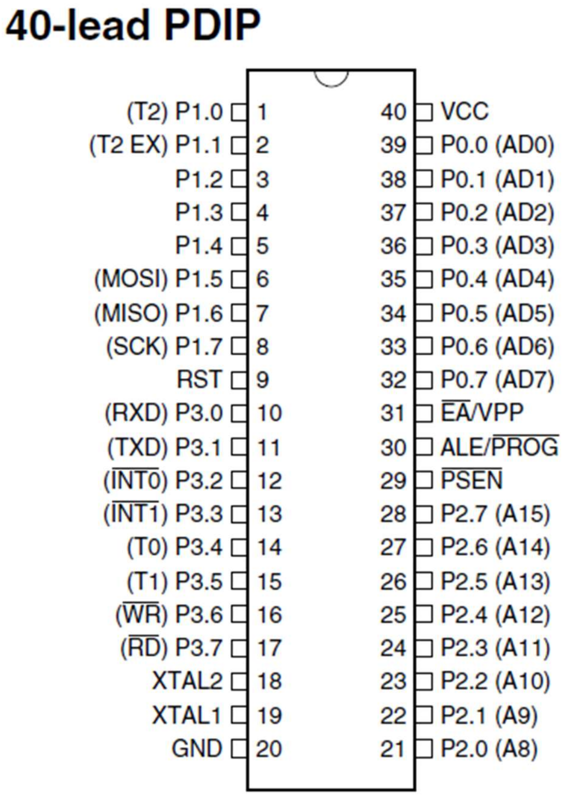
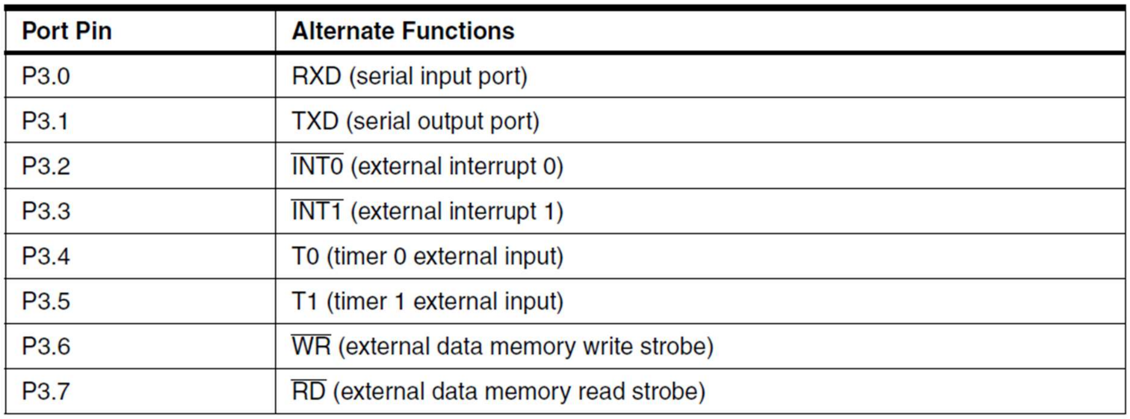


Figure 3: Pinout of AT89S52

- **VCC -** Supply voltage
- **GND -** Ground.
- **Port 0** is an 8-bit open drain bidirectional I/O port. Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pull-ups. Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pull-ups are required during program verification.
- **Port 1** is an 8-bit bidirectional I/O port with internal pullups. Port 1 also receives the low-order address bytes during Flash programming and verification.
- **Port 2** is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register. Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

- **Port 3** Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 receives some control signals for Flash programming and verification. Port 3 also serves the functions of various special features of the AT89S52, as shown in the following table.



Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

Table 3

- **RST** Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives high for 98 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.
- **ALE/PROG** Address Latch Enable (ALE) is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory. With the bit set, ALE is active only during a MOVX or MOVC instruction. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.
- **PSEN** Program Store Enable (PSEN) is the read strobe to external program memory. When the AT89S52 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.
- **EA/VPP** External Access Enable. EA must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, EA will be internally latched on reset. EA should be strapped to VCC for internal program executions. This pin also receives the 12-volt programming enable voltage (VPP) during Flash programming.
- **XTAL1** Input to the inverting oscillator amplifier and input to the internal clock operating circuit.
- **XTAL2** Output from the inverting oscillator amplifier.

Data Sheet for ADC0804

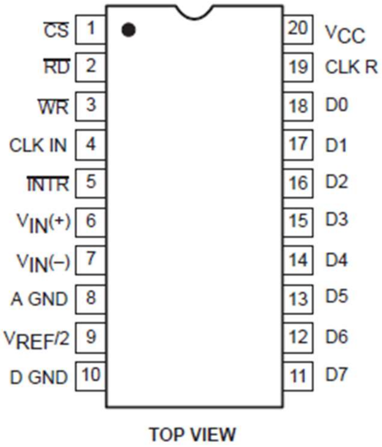


Figure 4: Pinout of ADC0804

The ADC0803 family is a series of three CMOS 8-bit successive approximation A/D converters using a resistive ladder and capacitive array together with an auto-zero comparator. These converters are designed to operate with microprocessor-controlled buses using a minimum of external circuitry. The 3-State output data lines can be connected directly to the data bus. The differential analog voltage input allows for increased common-mode rejection and provides a means to adjust the zero-scale offset. Additionally, the voltage reference input provides a means of encoding small analog voltages to the full 8 bits of resolution.

These devices operate on the Successive Approximation principle. Analog switches are closed sequentially by successive approximation logic until the input to the auto-zero comparator
[VIN (+)–VIN (–)] matches the voltage from the decoder. After all bits are tested and determined, the 8-bit binary code corresponding to the input voltage is transferred to an output latch. Conversion begins with the arrival of a pulse at the WR input if the CS input is low. On
the High-to-Low transition of the signal at the WR or the CS input, the SAR is initialized, the shift register is reset, and the INTR output is set high. The A/D will remain in the reset state if the CS and WR inputs remain low. Conversion will start from one to eight clock periods after one or both inputs makes a Low-to-High transition. After the conversion is complete, the INTR pin will make a High-to-Low transition. This can be used to interrupt a processor, or otherwise signal the availability of a new conversion result. A read (RD) operation (with CS low) will clear the INTR line and enable the output latches. The device may be run in the free-running mode as described later. A conversion in progress can be interrupted by issuing another start command.

Data Sheet for 16 x 2 LCD Display

Figure 5: Pinout of LCD Display

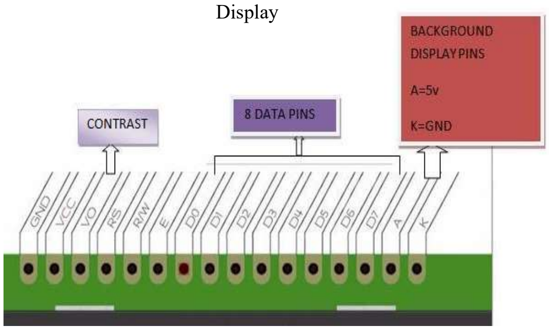
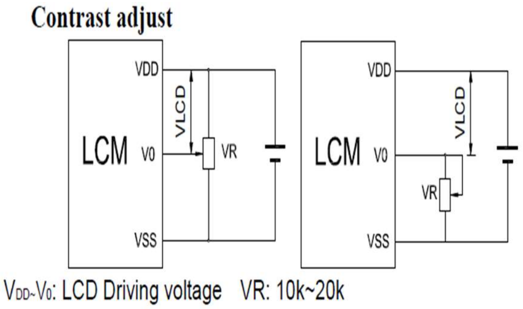
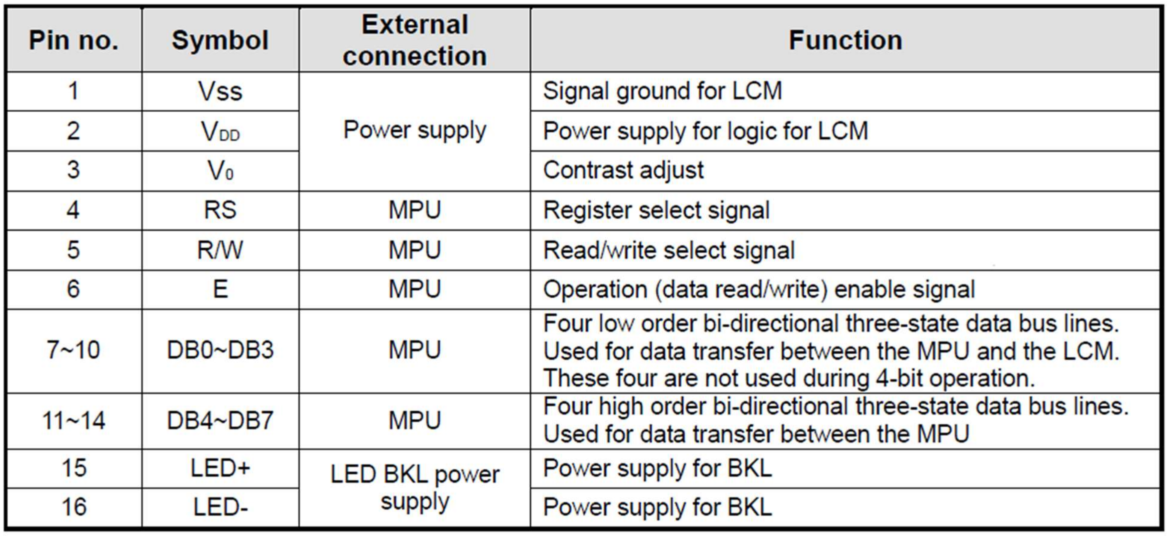


Figure 6



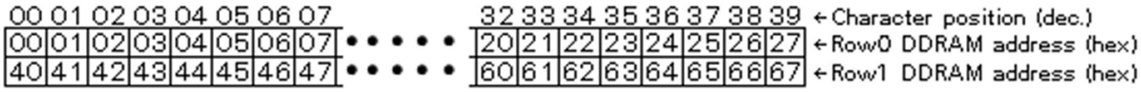


Pin no.	Symbol	External connection	Function
1	Vss	Power supply	Signal ground for LCM
2	V_{DD}		Power supply for logic for LCM
3	V_0		Contrast adjust
4	RS	MPU	Register select signal
5	R/W	MPU	Read/write select signal
6	E	MPU	Operation (data read/write) enable signal
7~10	DB0~DB3	MPU	Four low order bi-directional three-state data bus lines. Used for data transfer between the MPU and the LCM. These four are not used during 4-bit operation.
11~14	DB4~DB7	MPU	Four high order bi-directional three-state data bus lines. Used for data transfer between the MPU
15	LED+	LED BKL power supply	Power supply for BKL
16	LED-		Power supply for BKL

Table 4

DDRAM - Display Data RAM

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. So whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

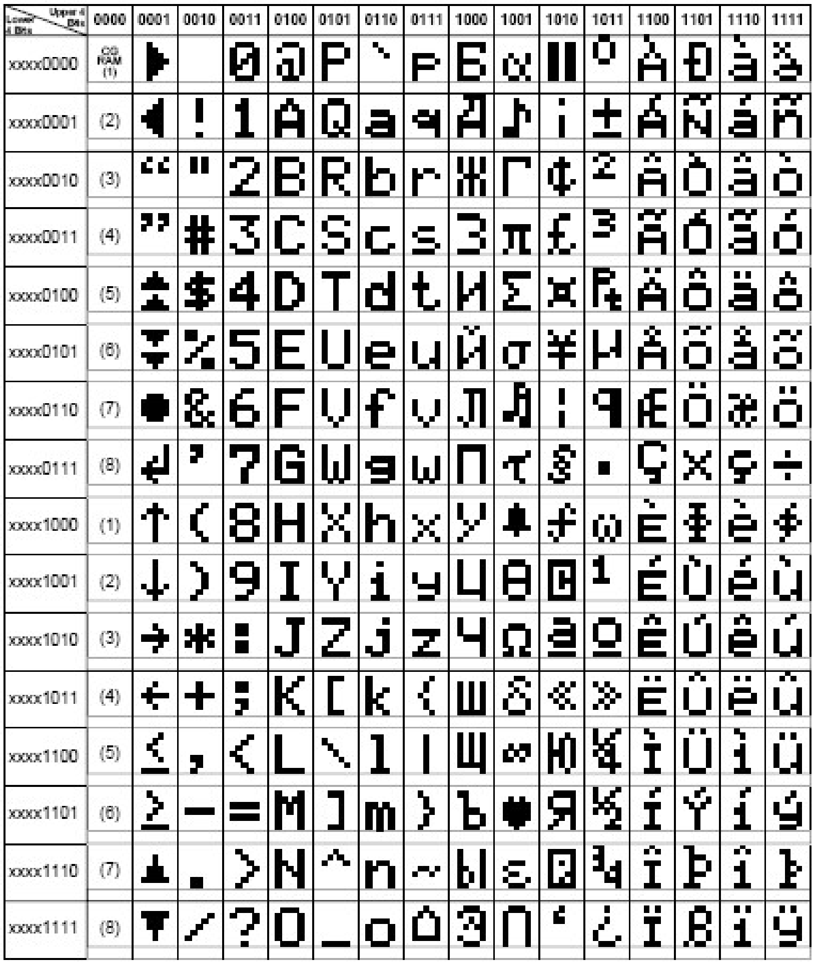


00	01	02	03	04	05	06	07		32	33	34	35	36	37	38	39	← Character position (dec.)
00	01	02	03	04	05	06	07	• • • • •	20	21	22	23	24	25	26	27	← Row0 DDRAM address (hex)
40	41	42	43	44	45	46	47	• • • • •	60	61	62	63	64	65	66	67	← Row1 DDRAM address (hex)

Table 5

CGROM - Character Generator ROM

The character generator ROM generates 5 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns. User defined character patterns are also available by mask-programmed ROM. If user want to display the fourth custom character, then the code to display it is 0x03 i.e. when user send 0x03 code to the LCD DDRAM then the fourth user created character or pattern will be displayed on the LCD.



Upper 4 Bits Lower 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)	▶		0	@	P	`	p	Б	α	‖	°	À	Đ	à	ð
xxxx0001	(2)	◀	!	1	A	Q	a	q	Д	♪	¡	±	Á	Ñ	á	ñ
xxxx0010	(3)	“	"	2	B	R	b	r	Ж	Γ	¢	²	Â	Ò	â	ò
xxxx0011	(4)	”	#	3	C	S	c	s	З	π	£	³	Ã	Ó	ã	ó
xxxx0100	(5)	⏫	$	4	D	T	d	t	И	Σ	¤	₧	Ä	Ô	ä	ô
xxxx0101	(6)	⏬	%	5	E	U	e	u	Й	σ	¥	µ	Å	Õ	å	õ
xxxx0110	(7)	●	&	6	F	V	f	v	Л	♬	¦	¶	Æ	Ö	æ	ö
xxxx0111	(8)	↲	'	7	G	W	g	w	П	τ	§	·	Ç	×	ç	÷
xxxx1000	(1)	↑	(	8	H	X	h	x	У	🔔	ƒ	ω	È	Φ	è	ø
xxxx1001	(2)	↓	)	9	I	Y	i	y	Ц	Θ	©	¹	É	Ù	é	ù
xxxx1010	(3)	→	*	:	J	Z	j	z	Ч	Ω	ª	º	Ê	Ú	ê	ú
xxxx1011	(4)	←	+	;	K	[	k	{	Ш	δ	«	»	Ë	Û	ë	û
xxxx1100	(5)	≤	,	<	L	＼	l	|	Щ	∞	Ю	¼	Ì	Ü	ì	ü
xxxx1101	(6)	≥	-	=	M	]	m	}	Ъ	♥	Я	½	Í	Ý	í	ý
xxxx1110	(7)	▲	.	>	N	^	n	~	Ы	ε	®	¾	Î	Þ	î	þ
xxxx1111	(8)	▼	/	?	O	_	o	⌂	Э	∩	‘	¿	Ï	ß	ï	ÿ

Table 6

CGRAM - Character Generator RAM

CGRAM area is used to create custom characters in LCD. In the character generator RAM, the user can rewrite character patterns by program. For 5 x 10 dots, four-character patterns can be written.

BF - Busy Flag

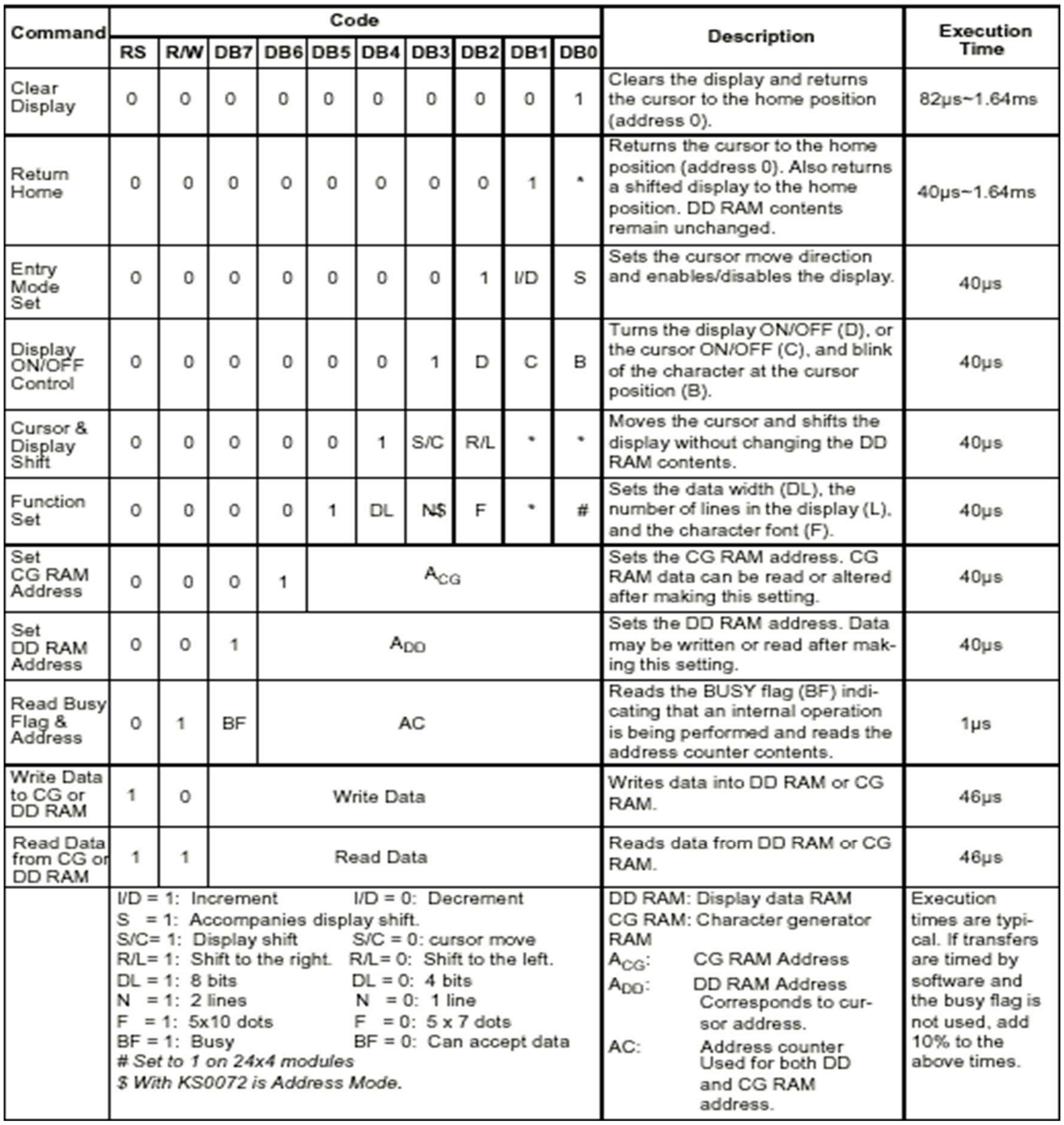
Busy Flag is a status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e. BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay. for the LCD processing. To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

Instruction Register (IR) and Data Register (DR)

There are two 8-bit registers: Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g. LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. when send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD.

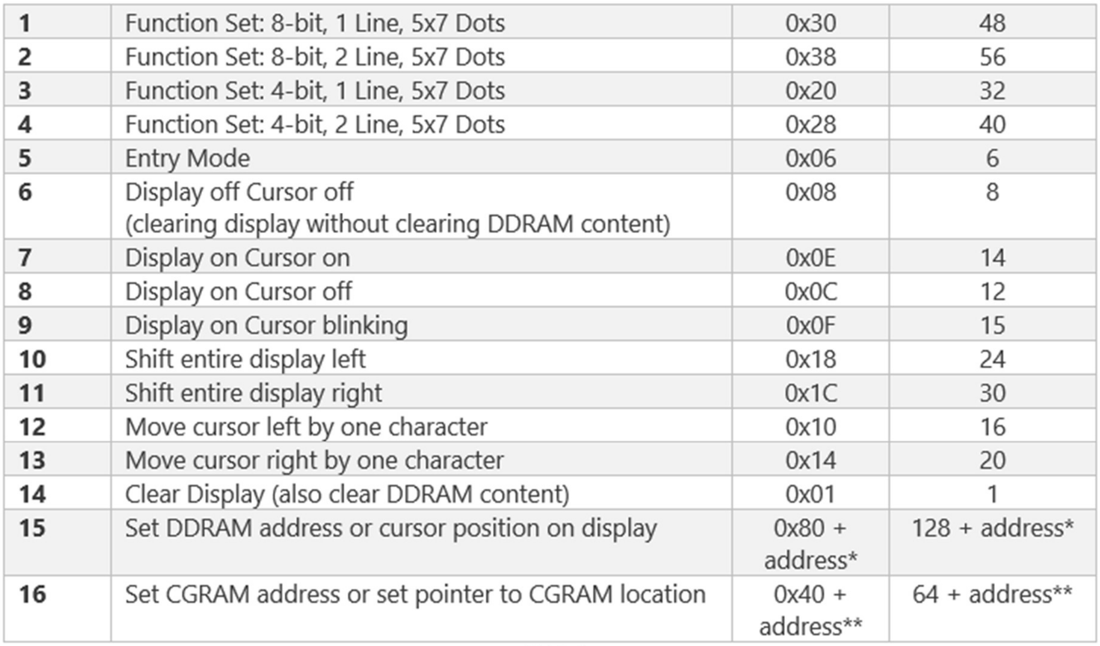
Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

Commands and Instruction set



Command	Code										Description	Execution Time
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears the display and returns the cursor to the home position (address 0).	82μs~1.64ms
Return Home	0	0	0	0	0	0	0	0	1	*	Returns the cursor to the home position (address 0). Also returns a shifted display to the home position. DD RAM contents remain unchanged.	40μs~1.64ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets the cursor move direction and enables/disables the display.	40μs
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Turns the display ON/OFF (D), or the cursor ON/OFF (C), and blink of the character at the cursor position (B).	40μs
Cursor & Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	Moves the cursor and shifts the display without changing the DD RAM contents.	40μs
Function Set	0	0	0	0	1	DL	N$	F	*	#	Sets the data width (DL), the number of lines in the display (L), and the character font (F).	40μs
Set CG RAM Address	0	0	0	1	A_{CG}						Sets the CG RAM address. CG RAM data can be read or altered after making this setting.	40μs
Set DD RAM Address	0	0	1	A_{DD}							Sets the DD RAM address. Data may be written or read after making this setting.	40μs
Read Busy Flag & Address	0	1	BF	AC							Reads the BUSY flag (BF) indicating that an internal operation is being performed and reads the address counter contents.	1μs
Write Data to CG or DD RAM	1	0	Write Data								Writes data into DD RAM or CG RAM.	46μs
Read Data from CG or DD RAM	1	1	Read Data								Reads data from DD RAM or CG RAM.	46μs
	I/D = 1: Increment I/D = 0: Decrement S = 1: Accompanies display shift. S/C= 1: Display shift S/C = 0: cursor move R/L= 1: Shift to the right. R/L= 0: Shift to the left. DL = 1: 8 bits DL = 0: 4 bits N = 1: 2 lines N = 0: 1 line F = 1: 5x10 dots F = 0: 5 x 7 dots BF = 1: Busy BF = 0: Can accept data *# Set to 1 on 24x4 modules* *$ With KS0072 is Address Mode.*										DD RAM: Display data RAM CG RAM: Character generator RAM A_{CG}: CG RAM Address A_{DD}: DD RAM Address Corresponds to cursor address. AC: Address counter Used for both DD and CG RAM address.	Execution times are typical. If transfers are timed by software and the busy flag is not used, add 10% to the above times.

Table 7



1	Function Set: 8-bit, 1 Line, 5x7 Dots	0x30	48
2	Function Set: 8-bit, 2 Line, 5x7 Dots	0x38	56
3	Function Set: 4-bit, 1 Line, 5x7 Dots	0x20	32
4	Function Set: 4-bit, 2 Line, 5x7 Dots	0x28	40
5	Entry Mode	0x06	6
6	Display off Cursor off (clearing display without clearing DDRAM content)	0x08	8
7	Display on Cursor on	0x0E	14
8	Display on Cursor off	0x0C	12
9	Display on Cursor blinking	0x0F	15
10	Shift entire display left	0x18	24
11	Shift entire display right	0x1C	30
12	Move cursor left by one character	0x10	16
13	Move cursor right by one character	0x14	20
14	Clear Display (also clear DDRAM content)	0x01	1
15	Set DDRAM address or cursor position on display	0x80 + address*	128 + address*
16	Set CGRAM address or set pointer to CGRAM location	0x40 + address**	64 + address**

Table 8

* DDRAM address

** CGRAM address

Initialization of LCD

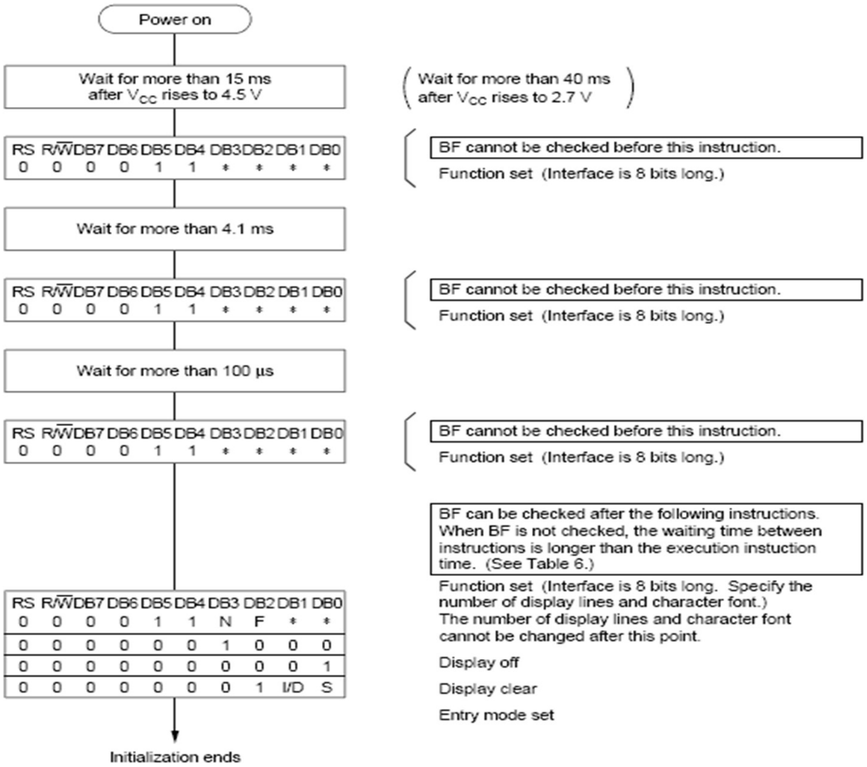

Power on
Wait for more than 15 ms after V_{CC} rises to 4.5 V
(Wait for more than 40 ms after V_{CC} rises to 2.7 V)
RS R/W̄ DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0
0 0 0 0 1 1 * * * *
BF cannot be checked before this instruction.
Function set (Interface is 8 bits long.)
Wait for more than 4.1 ms
RS R/W̄ DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0
0 0 0 0 1 1 * * * *
BF cannot be checked before this instruction.
Function set (Interface is 8 bits long.)
Wait for more than 100 µs
RS R/W̄ DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0
0 0 0 0 1 1 * * * *
BF cannot be checked before this instruction.
Function set (Interface is 8 bits long.)
BF can be checked after the following instructions. When BF is not checked, the waiting time between instructions is longer than the execution instuction time. (See Table 6.)
RS R/W̄ DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0
0 0 0 0 1 1 N F * *
0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 1 I/D S
Function set (Interface is 8 bits long. Specify the number of display lines and character font.) The number of display lines and character font cannot be changed after this point.
Display off
Display clear
Entry mode set
Initialization ends

Figure 7