

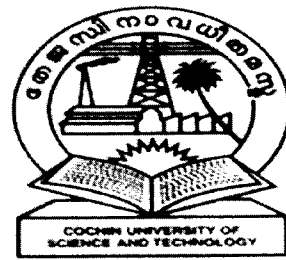
A Project Report on

**PIC+ Graphical LCD based Mini-Oscilloscope
for frequency stabilization of the
External Cavity Diode Laser**

Submitted in Partial Fulfillment
of the Requirement for the
Award of the Degree of
MASTER OF TECHNOLOGY
in Optoelectronics and Laser Technology

By

PRAVEEN KUMAR.C.K



International School of Photonics

Cochin University of Science and Technology

Kochi - 682 022

JUNE 2006

COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

INTERNATIONAL SCHOOL OF PHOTONICS

KOCHI-682 022



CERTIFICATE

This is to certify that Mr.Praveen Kumar .C.K has satisfactorily completed the dissertation work titled' *PIC+Graphical LCD based Mini-Oscilloscope for frequency stabilization of the External Cavity Diode Laser*' carried out at **Raman Research Institute**, Bangalore in partial fulfillment of the requirements for the award of the degree of **MASTER OF TECHNOLOGY in OPTOELECTRONICS AND LASER TECHNOLOGY** during the academic year 2004-2006.

Dr.P.Radhakrishnan

Director

Dr.V.P.N.Nampoori

Project Co-ordinator

Contents

| | | |
|----------|--|----------|
| 1 | Introduction..... | 1 |
| 1.1 | Diode Lasers..... | 2 |
| 1.2 | Saturation Absorption spectroscopy..... | 4 |
| 1.3 | Frequency locking of a laser line..... | 5 |
| 2 | Hardware : Mini – Oscilloscope..... | 6 |
| 2.1 | Introduction..... | 6 |
| 2.2 | Microcontroller..... | 7 |
| 2.21 | Defining the Task..... | 8 |
| 2.22 | Designing and Building the circuit..... | 8 |
| 2.23 | PIC 16F877A microcontroller..... | 10 |
| 2.24 | Analog-to-Digital Converter Module..... | 12 |
| 2.3 | Display Unit- Graphical LCD..... | 14 |
| 2.31 | The Main Features T6963C controller..... | 14 |
| 2.32 | Display Memory..... | 16 |
| 2.33 | MPU Connection..... | 17 |
| 2.4 | Setting up the T6963C..... | 19 |
| 2.41 | Powering sequence for T6963C..... | 20 |
| 2.5 | LCD screen..... | 21 |
| 2.6 | IC Programming..... | 22 |

| | | |
|----------|--|-----------|
| 3 | Software: Mini-Oscilloscope..... | 25 |
| 3.1 | Introduction..... | 25 |
| 3.2 | General Considerations of Commands and Data for..... the T6963C controller. | 27 |
| 3.3 | Writing the Control Program..... | 30 |
| 3.31 | Testing and Debugging..... | 31 |
| 3.4 | PIC Initialization..... | 32 |
| 3.5 | LCD Initialization..... | 36 |
| 3.51 | Setting the Display Size..... | 36 |
| 3.52 | Setting different Modes of Display..... | 37 |
| 3.6 | Address Pointer Considerations..... | 41 |
| 3.7 | A/D Conversion..... | 41 |
| 3.8 | Displaying the Waveforms on the LCD screen..... | 42 |
| 4 | Building up of External Cavity Diode Laser..... | 43 |
| 4.1 | External Cavity Diode Laser (ECDL)..... | 41 |
| 4.2 | Temperature controller..... | 48 |
| 4.3 | Laser Diode protection circuit..... | 49 |
| 4.4 | Saturation Absorption spectroscopy..... | 51 |
| 4.41 | Doppler broadening..... | 53 |
| 4.42 | Saturation Absorption Spectroscopy:..... Experimental set-up | 56 |
| 4.5 | Frequency Locking..... | 59 |
| 5 | Design and Results..... | 61 |
| 6 | Conclusions..... | 66 |

List of Figures

| | | |
|-----|--|----|
| 1.1 | Spectral transitions in Rubidium..... | 2 |
| 1.2 | Laser modes..... | 3 |
| 2.1 | Block diagram of the hardware units used for Mini-Oscilloscope.... | 6 |
| 2.2 | Pin diagram of PIC16F877 micro-controller..... | 10 |
| 2.3 | Block diagram of Graphic LCD module..... | 15 |
| 2.4 | Programming the PIC micro-controller using IC Programmer..... | 23 |
| 3.1 | Flow chart for Status checking of T6963C controller..... | 28 |
| 3.2 | Flow chart for sending a command to T6963C controller..... | 29 |
| 3.3 | Flow chart of a single channel oscilloscope..... | 30 |
| 3.4 | Examples for graphic and text pattern..... | 37 |
| 3.5 | Different display modes..... | 37 |
| 4.1 | Schematic of ECDL..... | 44 |
| 4.2 | Components used in an ECDL..... | 45 |
| 4.3 | Littrow configuration of Grating..... | 46 |
| 4.4 | Wheatstone Bridge used in Temperature Controller..... | 48 |
| 4.5 | Laser Diode Protection Circuit..... | 50 |
| 4.6 | Experimental set up of saturation absorption spectroscopy..... | 57 |
| 4.7 | (A) Doppler broadened signal (B) Doppler along with Rb85 cooling.. | 58 |

| | | |
|--------|---|----|
| 4.8 | Cooling spectrum of Rb ⁸⁵ | 58 |
| 4.9 | Lock box connected to ECDL..... | 59 |
| 5.1 | Block diagram of the Mini-oscilloscope connected to the ECDL... | 61 |
| 5.2 | Circuit built for Mini-Oscilloscope..... | 62 |
| 5.3 | Mini-oscilloscope as two channel display..... | 63 |
| 5.4(a) | ⁸⁵ Rb cooling signal on Mini-oscilloscope..... | 64 |
| 5.4(b) | ⁸⁵ Rb cooling signal on Agilent oscilloscope..... | 64 |

List of Tables

| | | |
|-----|--|----|
| 2.1 | The key features of PIC16F87x family Microcontrollers..... | 10 |
| 2.2 | Pins for VRAM Communication..... | 16 |
| 2.3 | Pins for MPU Communication..... | 18 |
| 3.1 | PIC 16F877 Bank selection..... | 26 |
| 3.2 | Status bits read from the T6963C..... | 28 |
| 3.3 | The five bit combination corresponds to a memory block..... addresses | 39 |

Preface

In the Laser Cooling and Trapping related experiments, the frequency stabilized lasers are extensively used. There are many lasers that work simultaneously in such experiments. The simplest procedure to stabilize the lasers are by using the Saturation Absorption Spectroscopy and adjusting the parameters by monitoring the error signal that is fed back to the laser. So there will be many signals that are to be monitored simultaneously. Using the commercial digital oscilloscope to display all the signals will be very expensive.

The main aim of the Project is to design and develop a system, which works as a Mini-oscilloscope, for displaying signals of interest, and make the unit stand-alone and portable. The Mini-Oscilloscope is built using the microcontroller and a Graphical LCD. The Mini-Oscilloscope is integrated to the Laser, which will minimize the cost as well as reduce the maintenance.

The Chapter one gives a brief introduction to the Diode lasers, its characteristics and the need for saturation absorption spectroscopy. Chapter two deals with the hardware, the advantages of the components used to build the mini-oscilloscope. Chapter Three gives the software approach for running the entire program Chapter four in its first section gives the procedure to build the External Cavity Diode Laser including the tuning and temperature controlling of the diode laser and second section gives the theory and experimental set up of the saturation absorption spectroscopy.

Chapter 1

Introduction

The frequency stabilized laser diodes are used in various application in atomic physics. The Lasers with combination of high power, small linewidth and fast tuneability are essential for many fields in high resolution spectroscopy. One example is the quickly developing field of laser atom cooling and trapping. Requirements for a laser system used in this field of applications are extensive: a mode hop free tuning range of a few GHz, with a linewidth in the regime of 1-MHz with an output power of few hundreds of mW.

In the Laser cooling and trapping experiments the laser frequency has to be stabilized to either to the side of or at the peak of any sufficiently well resolved transition of the Rubidium (the element being used in the experiment) hyperfine absorption spectrum. Rubidium is a convenient atom to study with absorption spectroscopy. Because it has a single electron outside of closed shells, it is relatively simple hydrogen like structure of energy levels, figure 1.1 shows the Spectral transitions in Rubidium atom . With output frequency stabilized against atomic transition frequency through saturated absorption, diode lasers can be a very effective laser source for many high precision atomic physics experiments.

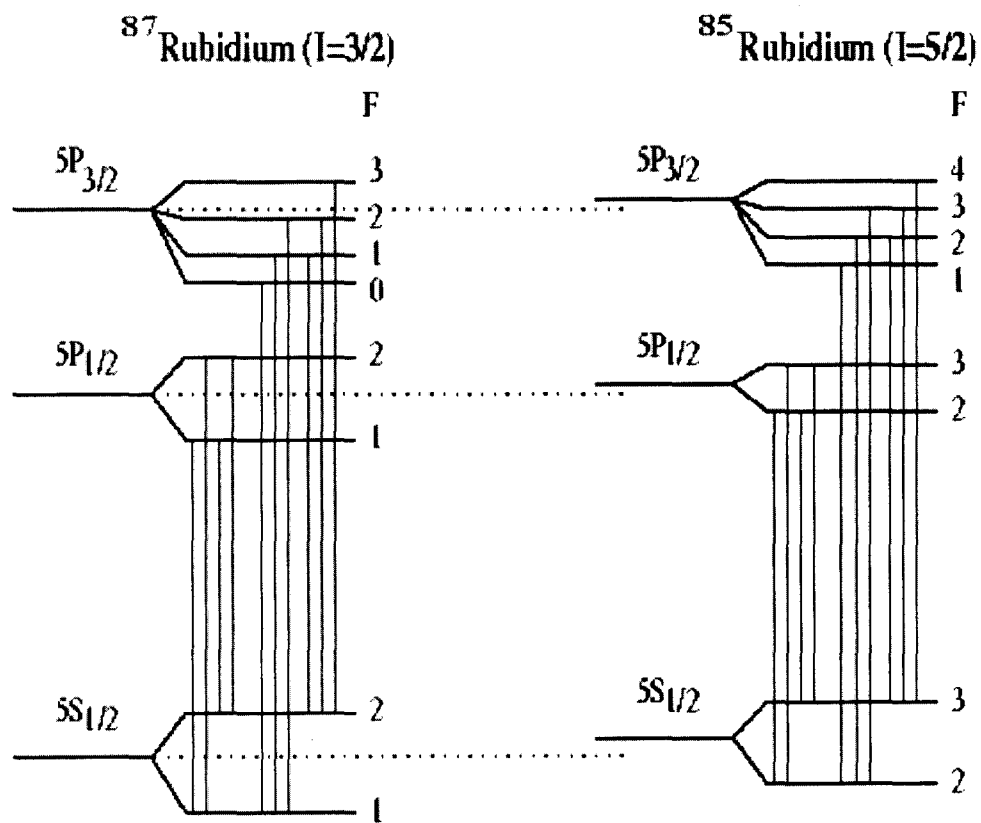


Figure 1.1 : Spectral transitions in Rubidium

1.1 Diode Lasers

Diode lasers are small devices with dimensions of less than a millimeter. They operate on a wavelength ranging from 0.6 to 1.55 micrometer, depending on the materials of the laser medium. The diode laser light is generated in small rectangular active region of typical thickness of 0.1 micron, width 3 micron and length 300 micron. materials of the laser medium. (The diode laser light is generated in small rectangular active region of typical thickness of 0.1 micron, width 3 micron and length 300 micron) Diode lasers produce linearly polarized

light with electric vector parallel to the junction ^{Plane}. Diode lasers oscillate in TE and TM modes. The TE modes are characterized by an electric field that oscillates in a direction parallel to the junction plane, while TM modes have an electric field that oscillates perpendicular to the junction plane. Since the reflectivity of the TE mode is always greater than that of the TM modes, cavity end losses are smaller for TE modes. This leads to a lower TE mode threshold for lasing. Thus the lasing transverse mode in a diode laser will be a TE mode.

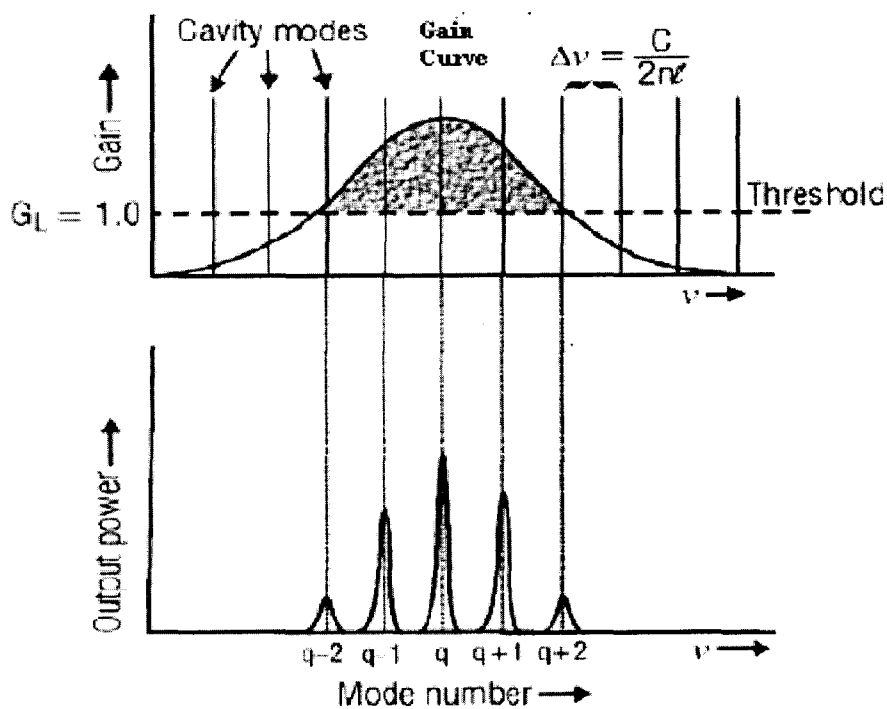


Figure 1.2: Laser modes

The typical output spectrum of a diode laser consists of a series of wavelength peaks corresponding to different longitudinal modes or cavity modes within the structure as shown in Fig1. 2. The spacing of these modes is dependent on the optical cavity length as each one corresponds to an integral number of ^{cavity} lengths. The spacing of these modes also depends on the refractive index which in turn is controlled by temperature and carrier density i.e., injection

current. The injection current tuning arises because of the heating produced by the current. The injection current also changes the carrier density which changes the index of refraction. Thus there are two tuning parameters-temperature and current, which has to be controlled for stabilizing the Diode laser.

In spite of the passive stability of the laser diode, injection current and the active stabilization of the temperature, drifts in frequency of few tens of MHz can occur over a period of few minutes. Mechanical vibrations too can cause changes in the laser wavelength. Thus some form of active stabilization of the frequency of the laser is required. This requires a frequency reference, which is obtained by using Doppler free saturated absorption spectroscopy signal of Rb atoms.

1.2 Saturation Absorption spectroscopy

Doppler free Saturated absorption eliminates Doppler broadening to resolve hyperfine spectral lines that are spaced a few tens of MHz apart. Doppler widths of about 500MHz caused by the random thermal motion of atoms limited the resolution of conventional optical spectroscopy. Doppler free saturated absorption spectroscopy is insensitive to this effect and thus allows the splitting between energy levels of a sample to be measured with much greater precision. The Saturation Absorption spectroscopy theory and experimental setup is discussed in detail in chapter 4.

1.3 Frequency locking of a laser line

In laser cooling experiments , a Doppler free saturation absorption setup is used to monitor in real time the frequency of the laser. The laser frequency is actively stabilized by detecting deviation from set frequency as changes in absorption and correcting for it by a servo mechanism . This is also described in chapter 4. For this purpose the saturation absorption signals are displayed on an oscilloscope. As a typical experiment has at least four such frequency stabilized lasers, one requires four oscilloscopes for working as a stand alone system. In this project we have eliminated the need for commercial oscilloscope by using a graphical LCD to display the signals. The analog detector output is acquired by a PIC microcontroller and after processing passed onto the graphical LCD, which may then be used to monitor in real time, the laser frequencies.

Chapter 2

Hardware : Mini – Oscilloscope

2.1 Introduction

The hardware for building Mini – Oscilloscope consists of a number of individual components which are linked together to produce the overall device.

The main hardware units are :

- Processing unit : Microcontroller - PIC 16F877A
- Display unit : Graphical LCD module with in-built T6963C controller
- PIC IC-programmer

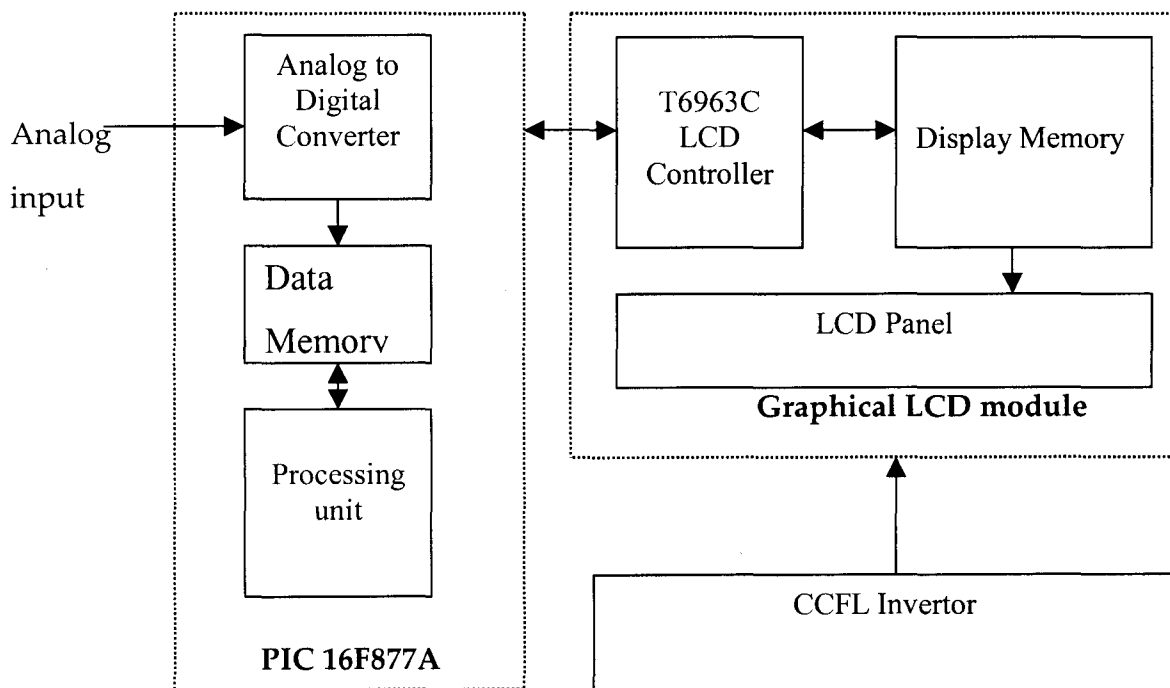


Fig. 2.1 Block diagram of the hardware units used for “Mini-Oscilloscope”

2.2 Microcontroller

A microcontroller is a highly integrated chip which includes all or most of the parts needed for a controller. It typically includes: CPU (central processing unit), RAM (Random Access Memory), EPROM/PROM/ROM (Erasable Programmable Read Only Memory), I/O (input/output), serial and parallel timers, interrupt controller. The Peripheral Interface Controller(PIC) range of microcontrollers are the first "Reduced Instruction Set Computer" (RISC) microcontrollers. Although having few instructions (e.g. 35 instructions for the PIC 16F877 line versus over 90 for the Intel 8048), the PIC line has a wealth of features included as part of the chip. Separate buses for instructions and data allows simultaneous access of program and data, and overlapping of some operations for increased processing performance. The benefits of design simplicity are a very small chip, small pin count, and very low power consumption.

Putting together a microcontroller project involves several steps:

1. Defining the task
2. Designing and building the circuits
3. Writing the control program
4. Testing and debugging

Writing the control program , testing and debugging will be discussed in the Chapter 3

2.21 Defining the Task

Every project begins with an idea, or a problem that needs a solution. Once it is known what is to be accomplished, we need to determine whether or not our idea is one that requires a microcontroller at all. In general, a microcontroller is the way to go when the circuits must make complex decisions or deal with complex data. For example, a simple AND gate can easily decide whether or not two inputs are both valid logic highs, and will change its output accordingly. But it would require many small-scale chips to build a circuit that stores a series of values representing sensor outputs and the times they occurred, and displays the information in an easily understandable form. This type of application is where microcontrollers come in handy. Inside, microcontrollers are little more than a carefully designed array of logic gates and memory cells, but modern fabrication processes allow thousands of these to fit on a single chip. The basic functions of a microcontroller performing arithmetic, logic, data-moving, and program branching functions are common ones. The abilities of the microcontroller is accessed by writing a program that performs the desired functions.

2.22 Designing and Building the circuit

When it is ready to design and build the circuits for a project, there are several ways to proceed. We can design our own circuits from scratch, using manufacturers' data books as guides or we can also follow a tested design. In this

project I have approached both the methods, I have built a circuit for the PIC microcontroller and interfaced it to a tested design of Graphical LCD.

Choosing a chip.

All microcontrollers contain a CPU, and chances are that we can use any of several devices for a specific project. Within each device family, we will usually find a selection of family members, each with different combinations of options. For example, the PIC 16F877A is a member of the PIC 16x family of microcontrollers.

We should select the version that best suits the system's requirements. Microcontrollers are also characterized by how many bits of data they process at once, with a higher number of bits generally indicating a faster or more powerful chip. Eight-bit chips are popular for simpler designs, but 4-bit, 16-bit, and 32-bit architectures are also available. Power consumption is another consideration, especially for battery-powered systems. Chips manufactured with CMOS processes usually have lower power consumption than those manufactured with NMOS processes. Many CMOS devices have special standby or "sleep" modes that limit current consumption to as low as a few microamperes when the circuits are inactive. Using these modes, a data logger can reduce its power consumption.

Finally, input/output (I/O) requires design decisions. Most systems require interfaces to things like sensors, keypads, switches, relays, and displays. Microcontrollers have ports for interfacing to the world outside the chip.

2.23 PIC 16F877A microcontroller

The PIC 16F877A is a 40-pin microcontroller, it has many features which makes it well suited for the present project of building a mini-oscilloscope. In the **Table 2.1** The PIC 16F877A is compared with some of the other 16F87X family controller. **Figure 2.2** shows the pin diagram of PIC16F877A micro-controller. The aim is to use this microcontroller in a new way to run a fully multi-tasking operating system.

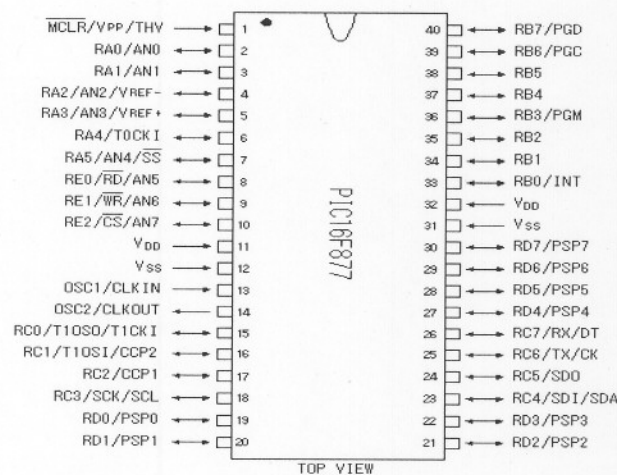


Fig. 2.2 Pin diagram of PIC16F877 micro-controller

Table 2.1 :The key features of PIC16F87x family Microcontrollers:

| Key Features | PIC16F873 | PIC16F874 | PIC16F876 | PIC16F877 |
|--------------------------------|-----------|-----------|-----------|-----------|
| MAX Operating Frequency | 20MHz | 20MHz | 20MHz | 20MHz |
| FLASH Program Memory | 4K | 4K | 8K | 8K |

| | | | | |
|--|-------------------------------------|---|-------------------------------------|---|
| (14-bit words) | | | | |
| Data Memory (bytes) | 192 | 192 | 368 | 368 |
| EEPROM Data Memory (bytes) | 128 | 128 | 256 | 256 |
| I/O Ports | RA0-5 (6) RB0-7 (8) RC0-7 (8) | RA0-5 (6) RB0-7 (8) RC0-7 (8) RD0-7 (8) RE0-2 (3) | RA0-5 (6) RB0-7 (8) RC0-7 (8) | RA0-5 (6) RB0-7 (8) RC0-7 (8) RD0-7 (8) RE0-2 (3) |
| Timers | 3 | 3 | 3 | 3 |
| CCP | 2 | 2 | 2 | 2 |
| Serial Communications | MSSP, USART | MSSP, USART | MSSP, USART | MSSP, USART |
| Parallel Communications | - | PSP | - | PSP |
| 10-bit Analog-to-Digital Module | 5 Channels | 8 Channels | 5 Channels | 8 Channels |
| Instruction Set | 35 Instructions | 35 Instructions | 35 Instructions | 35 Instructions |
| Pins (DIP) | 28 Pins | 40 Pins | 28 Pins | 40 Pins |

2.24 Analog-to-Digital Converter Module

A built in Analog-to-Digital converter is one of the major advantages of a PIC 16F877A chip.

The ADC offers these facilities :

- 10-bit resolution
- minimum conversion time $20\mu\text{S}$
- Interrupt on ADC conversion completion

The Analog-to-Digital (A/D) Converter module has eight inputs . The analog input charges a sample and hold capacitor. The output of the sample and hold capacitor is the input into the converter. The converter then generates a digital result of this analog level via successive approximation. The A/D conversion of the analog input signal results in a corresponding 10-bit digital number. The A/D module has high and low voltage reference input that is software selectable to some combination of VDD, VSS, RA2, or RA3. Where RA2 and RA3 can be used to have variable reference voltages for ADC , whereas VDD and VSS are used for fixed reference voltage. The A/D converter has a unique feature of being able to operate while the device is in SLEEP mode. To operate in SLEEP, the A/D clock must be derived from the A/D's internal RC oscillator. The A/D conversion time per bit is defined as TAD. The A/D conversion requires a minimum 12 TAD per 10-bit conversion.

For simplicity of the device, I have used only the 8 most significant bits from the 10bit digital value, as this allows a complete sample to be stored and transmitted in a single byte.

ACQUISITION TIME

$$\begin{aligned} \text{TACQ} &= \text{Amplifier settling time} + \\ &\quad \text{Hold capacitor Charging time} + \\ &\quad \text{Temperature coefficient} \\ &= T_{\text{AMP}} + T_c + T_{\text{COFF}} \\ &= 2\mu\text{s} + T_c + [(\text{Temperature} - 25\text{ C})(0.05\mu\text{s/C})] \\ T_c &= \text{CHOLD}(\text{RIC} + \text{RSS} + \text{RS})\ln(1/2047) \\ &= -120\text{pF}(1\text{K}\Omega + 7\text{K}\Omega + 10\text{K}\Omega)\ln(0.0004885) \\ &= 16.47\mu\text{s} \\ \text{TACQ} &= 2\mu\text{s} + 16.47\mu\text{s} + [(50\text{ C} - 25\text{ C})(0.05\mu\text{s/C})] = 19.72\mu\text{s} \end{aligned}$$

Note:

- 1:** The reference voltage (VREF) has no effect on the equation, since it cancels itself out.
- 2:** The charge holding capacitor (CHOLD) is not discharged after each conversion.
- 3:** The maximum recommended impedance for analog sources is 10 k Ω . This is required to meet the pin leakage specification.
- 4:** After a conversion has completed, a 2.0 TAD delay must complete before acquisition can begin again. During this time, the holding capacitor is not connected to the selected A/D input channel.

2.3 Display Unit- Graphical LCD

The display unit used for this project is a graphical LCD module with resolution of 240 x 128 pixel with inbuilt T6963C controller. The T6963C does all the initialization for the display, but all the data and control lines for its operation should be sent from the processing unit(Microcontroller) accordingly.

2.31 The Main Features T6963C controller

- Resolution up to 240x128 pixels
- Capable of displaying text and graphics
- Build-in character generator ROM (CG-ROM) with 128 pre-defined characters
- 8 bit parallel bus and signal lines for interfacing to a micro controller. Data and commands from and to the micro controller are multiplexed on this bus.
- Capable of controlling up to 64 KB of external display memory
- Duty Ratio for display multiplex driving in the range of 1/16 up to 1/128
- Current consumption 3-4 mA maximum
- Operating temperature -20 to +70 degrees Celsius

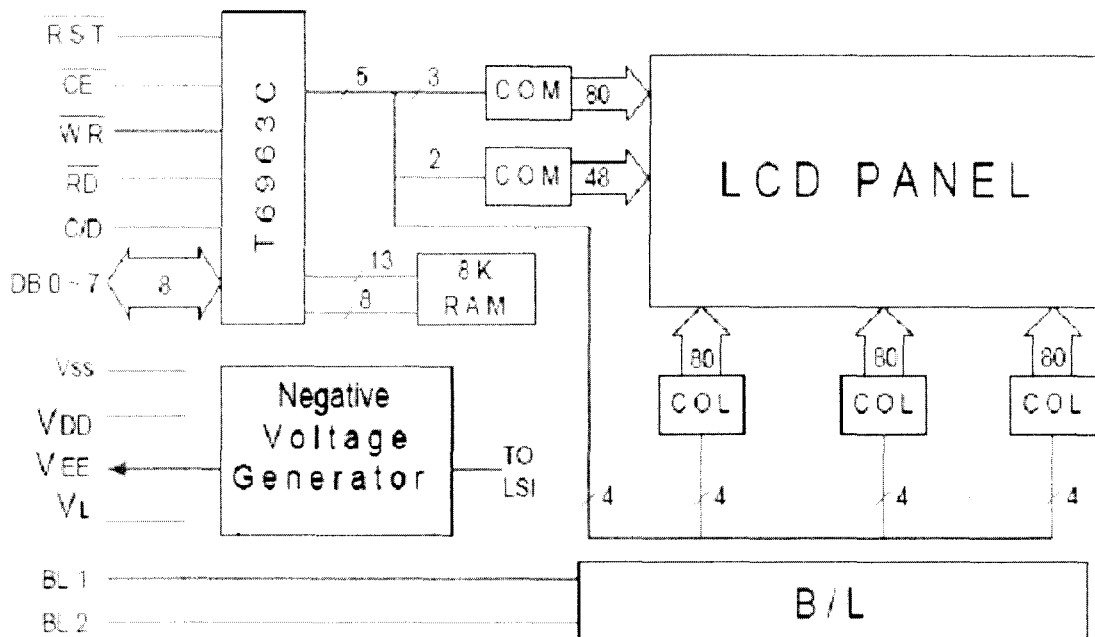


FIG: 2.3 Block diagram of Graphic LCD module

Although the T6963C is a controller for small LCDs, it has a lot of functions and options. The T6963C can not operate without a controlling element (usually a MPU). The T6963C is controlled by commands and data from the MPU. It is not possible, to load instructions or a program into the T6963C itself. Every operation and set-up parameter must be sent by the MPU. Every software consideration or command means a set of commands or data sent by the MPU.

The T6963C is used to control the row and column drivers of monochrome STN graphic displays with a resolution up to 240x128 pixels. The display row and column drivers are connected to the T6963C via a number of timing signals and one or two serial data lines. The T6963C is connected to a micro controller unit and external display memory, often called VRAM or display RAM.

2.32 Display Memory

The T6963C can control up to 64 KB display memory which must be of SRAM type memory (The T6963C can not refresh DRAM). The T6963C has 16 address lines, 8 data lines and several controlling lines to access the display memory. An option would be the direct mapping of the display memory in the microcontroller internal RAM. The advantage would be easy manipulation of the RAM by the microcontroller. (Without external address decoder logic).

| | |
|------------------------------|---|
| d0~d7 (Pins 22-26 and 28-30) | Data I/O pins for VRAM |
| ad0~ad15 (Pins 33-48) | Address lines for VRAM |
| R/W(Pin 31) | Read/Write signal for VRAM |
| /ce (Pin 32) | Chip Enable pin for VRAM of any address |
| /ce0 (Pin 49) | Chip Enable pin for VRAM in the address range 0000h to 07FFh |
| /ce1 (Pin 50) | Chip Enable pin for VRAM in the address range 0800h to 0FFFh |

Table 2.2 Pins for VRAM Communication

The T6963C differs from most other LCD controllers in its use of the display RAM. A fixed area of memory is normally allocated for text, graphics and the external character generator, but with the T6963C the size for each area can be set by software commands. This means that the area for text, graphics and external character generator can be freely allocated within the external memory, up to 64 KB

2.33 MPU Connection

The communication between the T6963C and a MPU is not complex and there are only few hardware considerations. The T6963C has an 8 bit data bus and several signals for communication with a MPU. The PIC 16F877A has five I/O Ports(A,B,C,D,E), any one of the port can be used as data port and any other as control ports. It is better to use any port among B,C,D ports as an I/O port , because Port A & Port E can be used for analog input.

Note:

If the MPU instruction execution time is less than 200 ns, wait states, NOPs or delay loops may be required to allow time for the T6963C to recognize the instruction.

| | |
|---|--|
| C/ \bar{D} (Pin 4) Command/Write selection. | To write or read data, it must be driven "low". To write a Command or read status, it must be driven "high". |
| \bar{RD} (Pin 5) Data Read. | It must be driven "low" to read data from VRAM via the T6963C. |
| \bar{WR} (Pin 6) Data Write. | It must be driven "low" to write data to VRAM via the T6963C. |
| D0~D7 (Pins 7- 14) | Data I/O pins for communications between MPU and T6963C |
| \bar{CE} (Pin 15) Chip Enable. | It must be driven "low" while MPU communicates with T6963C. |
| \bar{RESET} (Pin 16) Chip Enable. | It is driven 'low' initially, then driven 'high' as long as the display is on |

Table 2.3 Pins for MPU Communication

2.4 Setting up the T6963C

The T6963C must be set-up so that it operates correctly with the given display characteristics. There are both, hardware and software issues to consider. Hardware issues include selecting the correct pins which determine the size of the display (i.e. the resolution) or set the Font Size for example. Software issues include commands from the micro- controller for example to set the “software” size of the display

MDS, MD0, MD1 (Pins 3, 4, 5): Pins for selection of LCD vertical size, all possible combinations are given in Appendix-1. For example the combination MDS:MD1:MD0 = H:L:L would set 128 vertical dots .MD2, MD3 (Pins 6, 7): Pins used to select the number of columns. Columns in this instance refer to the number of displayable characters per row. There can be 4 combinations: 32 columns, 40 columns, 64 columns and 80 columns.

FS0, FS1 (Pins 8, 9): Pins for selection of Font Size. There are four Font Sizes available: 5x8, 6x8, 7x8 and 8x8 dots per font. (horizontal dots by vertical dots). The horizontal Font Size does not affect the driving signals, i.e. it does not change the horizontal resolution of the display when changing the Font Size.

Note:

Though there are several these pins available on the T6963C , But for the Graphic- LCD module I am using in this project , only MD2 and FS1 pins are accessible by the user. Datasheet for the LCD module is given in Appendix 2.

2.41 Powering sequence for T6963C

After powering on the whole system, a reset of the T6963C should be issued to ensure correct operation. This is achieved by holding the $\overline{\text{RESET}}$ pin “low” for at least 5-6 clock cycles. VDD must be stable at +5 V for the reset.

It is important to keep the correct power on sequence for the whole system because the T6963C generates the M-Signal (In the T6963C datasheet it is called FR-Signal) for the LCD row and column drivers. The M-Signal is used to change the polarity of the driving voltage so that in sum, no DC is applied to the Liquid Crystal material. To ensure the stability of the M-Signal, the T6963C must be fully initialized before applying the Voltage VEE (-15V) to the Liquid Crystal material. If T6963C is reset during normal operation, it must be ensured that the VEE (-15V) voltage and the display are turned off until the T6963C is fully reinitialized.

The correct power up sequence would be:

1. Apply power to MPU and T6963C. The display and the display's power (VEE) should be off.
2. Reset T6963C and wait until it is fully initialised. (5-6 clock cycles).
3. Apply power to the display (VEE on).
4. Apply signals to the display and switch it on

Note:

For powering off, the reverse applies, i.e. first, switch off the display, remove signals, remove VEE display power and then switch off the MPU and T6963C (with proper delays, too).

2.5 LCD screen

Due to the low power and size requirements of the display for the scope, it was immediately clear that only LCD-type screens would be suitable. As it is desirable to display sampled waveforms on the device, a graphics-based display would have been the best solution, especially as many graphic displays can also be programmed to display text as well. The LCD used in this project has 240 pixels horizontally and 128 pixels vertically. This is the Maximum resolution that a T6963C driver controller can support.

The Toshiba - T6963C protocol allows the screen controller to be read as well as written bytes of data to the screen memory. Two different types of data can be sent , the display character data, and control data. The control signals are sent from the PIC accordingly.

To aid readability, the LCD displays can be chosen with or without a backlight. Backlight option makes the display readable at night and in dark areas. Power for the backlight is separate from the main LCD power supply. The backlight for the LCD display I am using requires +5V, and also has a contrast option, provided through an external variable resistor. After the contrast is once set, it didn't require alteration during use. Powering the backlight uses a considerable amount of power, especially when the whole device will be running from batteries.

The following table summarizes the connections required to the LCD:

| | | | | | | | | | | | | | | | |
|---------|-----|------|------|-----|------------------|-----------------|-----------|-----------------|-----------------|-----|----------------|-----------------|-----------------|------------|------------|
| PIN NO. | 20 | 19 | 18 | 17 | 16 | 15 | 14 ~ 7 | 6 | 5 | 4 | 3 | 2 | 1 | BL 1 | BL 2 |
| SYMBOL | N/A | FS 1 | MD 2 | VEE | \overline{RST} | \overline{CE} | DB7 ~ DB0 | \overline{WR} | \overline{RD} | C/D | V _L | V _{DD} | V _{SS} | INV OUT | INV OUT |

The device is designed to be able to both sample, store and detect digital lines. The digital I/O lines are perhaps the easiest part of the device they are simply connected straight from the digital I/O port to a set of I/O lines on the microcontroller. All control (including setting whether a line is an input or output) is achieved through software.

2.6 PIC Programming

There are two different ways to program the FLASH memory on the microcontrollers by using a separate programmer, or by using the in-circuit programming facility. In the present Project I have used the external PIC Programmer to program the PIC 16F877A.

Figure 2.4 shows the process of programming PIC micro-controllers. The programming of PIC is done on the platform of Microchip developed **MPLAB** software.

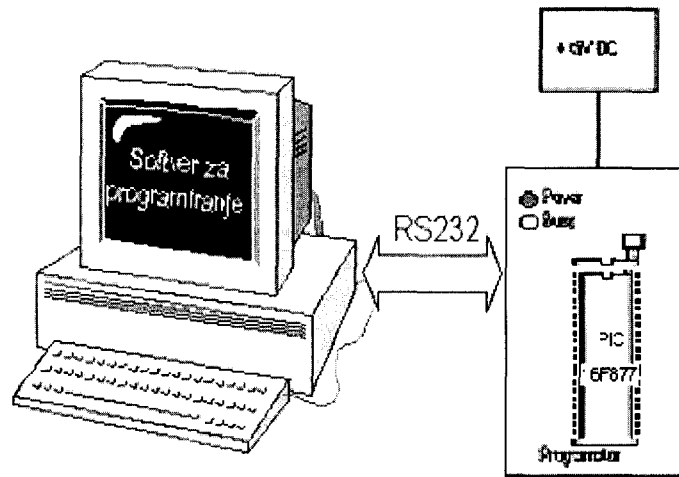


Fig. 2.4: Programming the PIC micro-controller using IC Programmer

IC programmer

- PIC IC PROGRAMMER is a microchip MCU development programmer that enables to program software into PIC MCU device.
- PIC IC PROGRAMMER system components include: RS-232 interface to connect to any standard PC serial port, 9V power supply.
- Device programmer includes program memory, configuration bits and ID locations.

The important features of IC PROGRAMMER:

- We can verify whether PIC MCUs are blank.
- We can verify that code in the target MCU matches our firmware.
- We can read code from an unprotected PIC MCU into MPLAB IDEs program memory window for debugging and programming into micro devices.

Before programming a device the following steps must be followed

1. Install PIC IC PROGRAM hardware,
2. Establish communication between the PIC IC PROGRAM and PC,
3. Next step is to set up the device programmer and configure bit dialogs. Then load hex file into program memory. Check whether device is blank before programming.

Chapter 3

Software: Mini- Oscilloscope

3.1 Introduction

The ability to communicate is of great importance in any field. However, it is only possible if both communicating partners know the same language, i.e., follow the same rules during communication. Using these principles, we can also define communication that occurs between microcontrollers and man. Language that microcontroller and man use to communicate is called "assembly language".

The instruction set of PIC 16F877 consists of 35 single word instruction, there are three types of instruction: byte-oriented, bit-oriented and literal and control operation. The complete set of instructions is given in Appendix - 3. Each instruction is a 14-bit word, divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC 16F877 has 8Kx14 words of Flash memory, 256 bytes of EEPROM data memory and 368 bytes of Data memory (SRAM). The data memory in the PIC is partitioned into multiple banks which contain General Purpose Registers and the Special function Registers. There are four Banks in PIC 16F877A Microcontroller, the banks are selected by bits RP1(STATUS<6>) and RP0(STATUS<5>) is shown in Table 3.1.

| RP1:RP0 | Bank |
|---------|------|
| 00 | 0 |
| 01 | 1 |
| 10 | 2 |
| 11 | 3 |

Table 3.1: PIC 16F877 Bank selection

Each banks are of 128 bytes, the lower locations of each bank are reserved for Special Function Registers. Above the Special Function Registers are the General Purpose Registers. The banks 0 to 3 each have available 80 bytes of Data memory (General Purpose Registers) which are independent from each other. Banks 2 and 3 have further 16 bytes, which are also independent. However upper 16 bytes of each bank have a common root. Accessing any of these 16 bytes in any bank automatically accesses those same location in Bank 0 . So for the PIC16F877A, in Bank order, the available data memory locations total is $96+80+96+96 = 368$ bytes.

The Toshiba T6963C is a very popular LCD controller for use in small graphics modules. It is capable of controlling displays with a resolution up to 240×128 ^{pixels}. Because of its low power and small outline it is most suitable for mobile applications. Although this controller is small, it has the capability of displaying , merging text and graphics and it manages all the interfacing signals to the displays Row and Column drivers. Various functions and options can be selected through software commands. Complete command set is listed in the Appendix-1. The commands are sent from the microcontroller and affect the display, the

external memory and the T6963C functions. The sequence of data is sent to the Graphic LCD through the T6963C LCD Controller.

3.2 General Considerations of Commands and Data for the T6963C controller

While sending commands or data to the T6963C a certain sequence must be obeyed. First of all, before sending any command or data to the T6963C, a status check must be performed. The status check is necessary to find out, if the T6963C is ready to accept any commands or data.

The status check is done by driving,
RD pin of the T6963C controller "low",
WD pin to "high",
CE pin to "low",
and C/D pin to high".

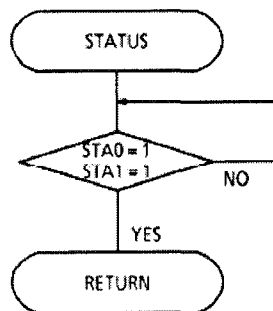
The T6963C is still busy when one of the STA0,STA1 bits are "low". To accept commands and data STA0 and STA1 must be "high". After a successful status check, the T6963C is ready to accept commands and/or data. A command is always one byte long. However, some commands requires some data to be sent along with it. There are commands, which need two data bytes, one data byte or none. The data must be sent before the Command, i.e. if a command needs two data bytes, the first data byte must be sent, then the next and finally the actual command. Status checks must be performed before sending each byte.

| MSB | | | | LSB | | | |
|------------|--|------------|------------|------------|------------|---------------------------------------|------------|
| STA7 D7 | STA6 D6 | STA5 D5 | STA4 D4 | STA3 D3 | STA2 D2 | STA1 D1 | STA0 D0 |
| STA0 | Check command execution capability | | | | | 0 : Disable 1 : Enable | |
| STA1 | Check data read/write capability | | | | | 0 : Disable 1 : Enable | |
| STA2 | Check Auto mode data read capability | | | | | 0 : Disable 1 : Enable | |
| STA3 | Check Auto mode data write capability | | | | | 0 : Disable 1 : Enable | |
| STA4 | Not used | | | | | | |
| STA5 | Check controller operation capability | | | | | 0 : Disable 1 : Enable | |
| STA6 | Error flag. Used for Screen Peek and Screen copy commands. | | | | | 0 : No error 1 : Error | |
| STA7 | Check the blink condition | | | | | 0 : Display off 1 : Normal display | |

Table 3.2 Status bits read from the T6963C

Status checking flow

a)



b)

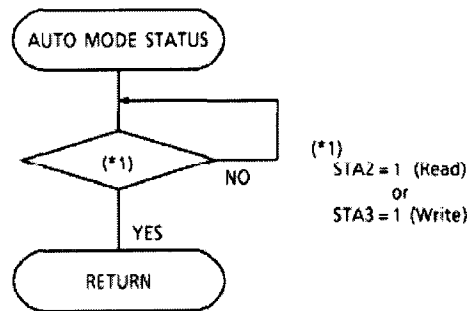
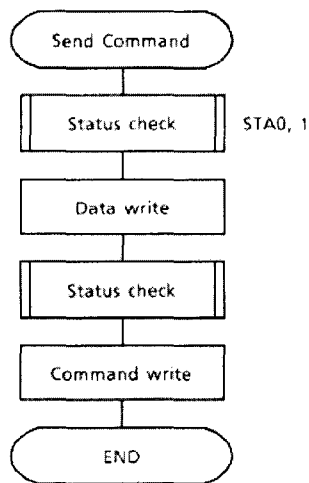


Figure 3.1: Flow chart for Status checking of T6963C controller.

Status byte can be read from the data lines D7~D0. Each bit (except bit 5) represents a certain state of the T6963C. The Important bits for the status check in normal mode are bits 0 (D0) and 1 (D1). of the status byte STA0 (D0) and STA1 (D1). The bits D2 and D3 has to be set for auto read and auto write respectively.

Procedure for sending a command

a) The case of 1 data



b) The case of 2 data

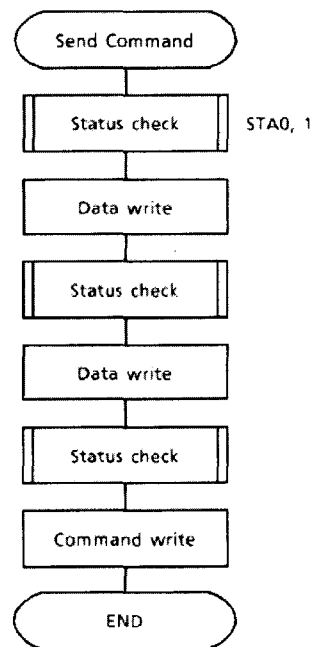


Figure 3.2: Flow chart for sending a command to T6963C controller.

3.3 Writing the Control Program

After the hardware is completely built, the final step is to write the Program that controls the project. There are many options by which we can write the Program, they include using machine code, assembly language, or a higher-level language. Which programming language you use depends on things like desired execution speed, program length, and convenience. In this project I have chosen assembly language to write the program. I have used MPLAB software for writing the assembly language program.

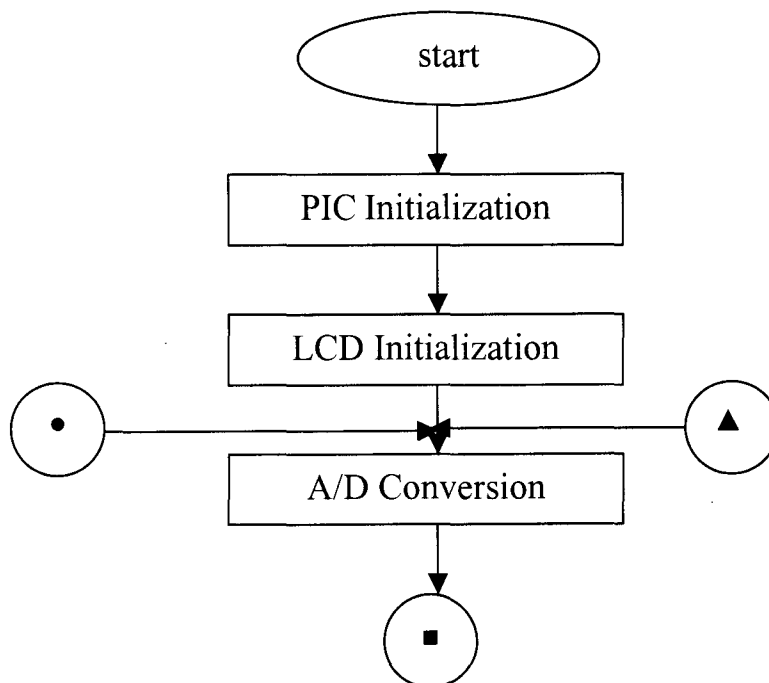
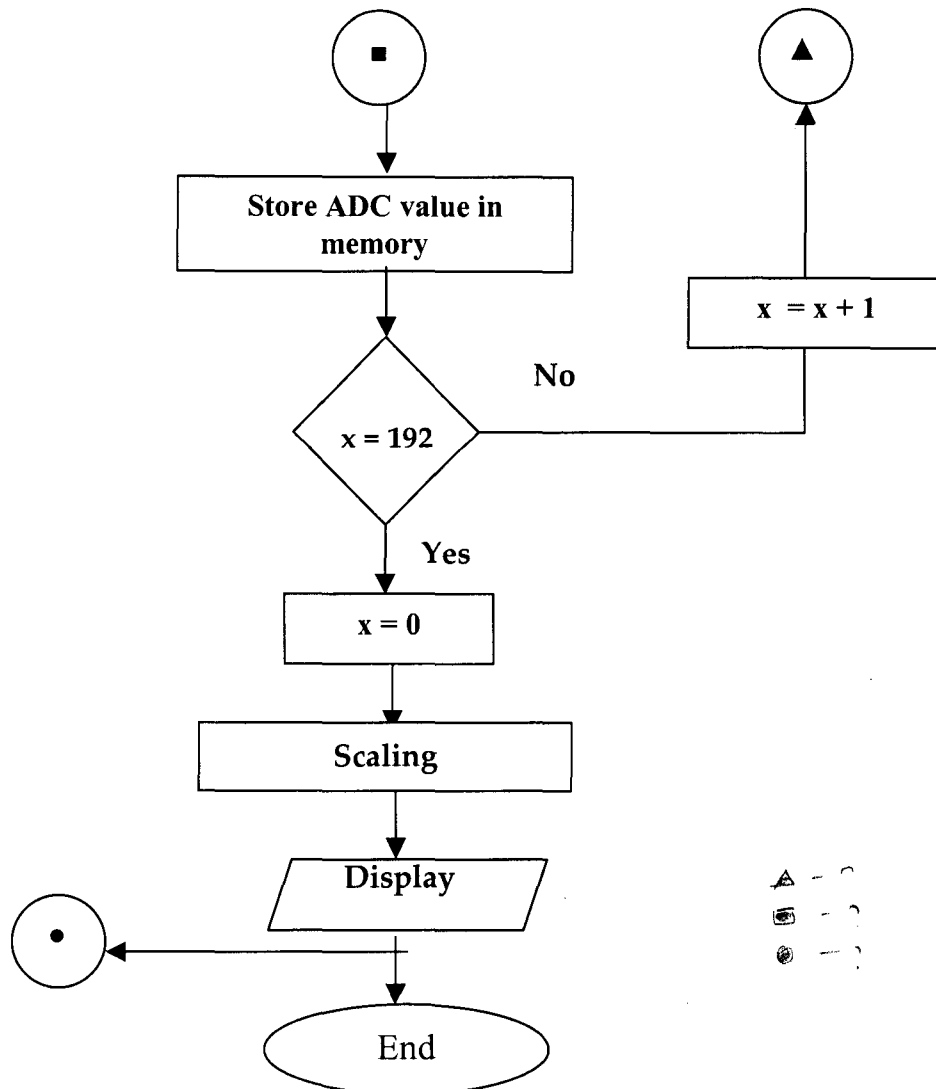


Figure 3.3: Flow chart of a single channel oscilloscope.



3.31 Testing and Debugging

After writing the program, it has to be tested and as necessary, find and correct the ^{error} mistakes to get it working properly. MPLAB is a windows based integrated development environment (IDE) for Microchip technology incorporated PIC MCU families. MPLAB allows writing, debugging and optimising PIC micro applications for firmware product designs. MPLAB also

supports the MPLAB ICE and PIC master emulators, PIC START PLUS, IC program etc.

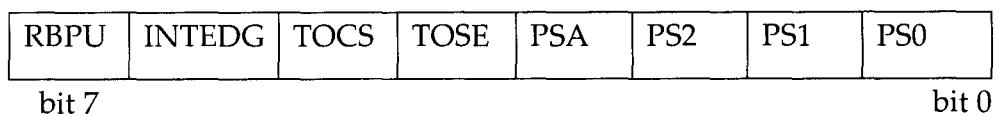
MPLAB IDE runs under Microsoft windows 3.1x, windows 95, 98, NT and 2000. It allows creating and editing of source code by providing a full-featured text editor. Using MPLAB IDE we can easily debug source code with the aid of build result window, that displays the error found by compiler, assembler and linker when generating executable files.

3.4 PIC Initialization

As soon as the PIC microcontroller starts running the program the PIC initialization has to be done. As a part of initialization the ports has to be assigned as an input or an output. There are five ports in PIC 16F877A controller, which user can assign as input or output. The ports once initialized can be changed anywhere in between the program.

The Special Function Registers are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. The special function register OPTION_REG is a readable and writable register, which contains various control bits to configure the timer TMR0 prescaler/WDT postscaler, the External interrupt and weak pull-ups on PORTB.

OPTION_REG REGISTER

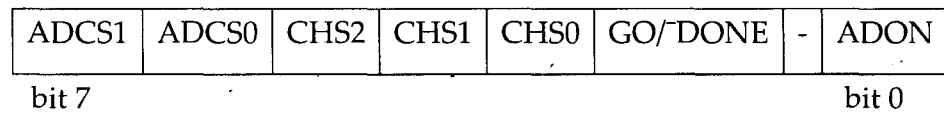


- bit 7 **RBPU**: PORTB pull-up Enable bit
 1 = PORTB pull-ups are disabled
 0 = PORTB pull-ups are enabled
- bit 6 **INTEDG**: Interrupt Edge select bit
 1 = Interrupt on rising edge of RB0/INT pin
 0 = Interrupt on falling edge of RB0/INT pin
- bit 5 **TOCS**: TMR0 clock source select bit
 1 = transition on RA4/TOCKI pin
 0 = Internal instruction cycle clock
- bit 4 **TOSE**: TMR0 source Edge select bit
 1 = increment on high-to-low transition on RA4/TOCKI pin
 0 = increment on low-to-high transition on RA4/TOCKI pin
- bit 3 **PSA**: Prescaler assignment bit
 1 = Prescaler is assigned to the WDT
 0 = Prescaler is assigned to the Timae0 module
- bit 2-0 **PS2:PS0** : Prescaler rate Select bits

| Bit value | TMR0 Rate | WDT Rate |
|-----------|-----------|----------|
| 000 | 1 : 2 | 1 : 1 |
| 001 | 1 : 4 | 1 : 2 |
| 010 | 1 : 8 | 1 : 4 |
| 011 | 1 : 16 | 1 : 8 |
| 100 | 1 : 32 | 1 : 16 |
| 101 | 1 : 64 | 1 : 32 |
| 110 | 1 : 128 | 1 : 64 |
| 111 | 1 : 256 | 1 : 128 |

The A/D module in the PIC is initialized using two registers ADCON0 and ADCON1. The ADCON0 register is used to assign conversion clock, analog channels, to check the status of the A/D conversion. The ADCON1 register is used to select to A/D result format, A/D port configuration i.e., selecting channels as either digital or analog inputs.

ADCON0 REGISTER



bit 7-6 **ADCS1:ADCS0:** A/D Conversion Clock select bits

00 = FOSC/2

01 = FOSC/8

10 = FOSC/32

11 = F_{RC} (clock derived from the internal A/D module RC oscillator)

bit 5-3 **CHS2:CHS0:** Analog channel select bits

000 = Channel 0

001 = Channel 1

010 = Channel 2

011 = Channel 3

100 = Channel 4

101 = Channel 5

110 = Channel 6

111 = Channel 7

bit 2 **GO/DONE:** A/D Conversion status bit

if ADON =1:

1 = A/D Conversion in progress

0 = A/D conversion not in progress

bit1 **Unimplemented**

bit0 **ADON: A/D on bit**

1 = A/D converter module is operating

0 = A/D converter module is shut-off

ADCON1 REGISTER

| | | | | | | | |
|-------|---|---|---|-------|-------|-------|-------|
| ADFM | - | - | - | PCFG3 | PCFG2 | PCFG1 | PCFG0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **ADFM : A/D Result Format select bit**

1 = Right justified .6 Most significant bits of ADRESH are read as '0'

0 = left justified . 6 Least significant bits of ADRESH are read as '0'

bit 6-4 **Unimplemented**

bit 3-0 **PCFG3:PCFG0: A/D port configuration control bits:**

| PCFG3: PCFG0 | AN7 ⁽¹⁾ RE2 | AN6 ⁽¹⁾ RE1 | AN5 ⁽¹⁾ RE0 | AN4 RA5 | AN3 RA3 | AN2 RA2 | AN1 RA1 | AN0 RA0 | VREF+ | VREF- | CHAN/ Refs ⁽²⁾ |
|-----------------|---------------------------|---------------------------|---------------------------|------------|------------|------------|------------|------------|-------|-------|------------------------------|
| 0000 | A | A | A | A | A | A | A | A | VDD | VSS | 8/0 |
| 0001 | A | A | A | A | VREF+ | A | A | A | RA3 | VSS | 7/1 |
| 0010 | D | D | D | A | A | A | A | A | VDD | VSS | 5/0 |
| 0011 | D | D | D | A | VREF+ | A | A | A | RA3 | VSS | 4/1 |
| 0100 | D | D | D | D | A | D | A | A | VDD | VSS | 3/0 |
| 0101 | D | D | D | D | VREF+ | D | A | A | RA3 | VSS | 2/1 |
| 011x | D | D | D | D | D | D | D | D | VDD | VSS | 0/0 |
| 1000 | A | A | A | A | VREF+ | VREF- | A | A | RA3 | RA2 | 6/2 |
| 1001 | D | D | A | A | A | A | A | A | VDD | VSS | 6/0 |
| 1010 | D | D | A | A | VREF+ | A | A | A | RA3 | VSS | 5/1 |
| 1011 | D | D | A | A | VREF+ | VREF- | A | A | RA3 | RA2 | 4/2 |
| 1100 | D | D | D | A | VREF+ | VREF- | A | A | RA3 | RA2 | 3/2 |
| 1101 | D | D | D | D | VREF+ | VREF- | A | A | RA3 | RA2 | 2/2 |
| 1110 | D | D | D | D | D | D | D | A | VDD | VSS | 1/0 |
| 1111 | D | D | D | D | VREF+ | VREF- | D | A | RA3 | RA2 | 1/2 |

A = Analog input ; D = Digital I/O

3.5 LCD Initialization

The initialization of the LCD includes setting the Graphic Home Address, Text Home Address, Graphic Display Area, Text Display Area, setting the Display Mode. Etc., Depending on the application, LCD can be used to display both Graphics and Text , or either of them.

The Graphic (Text) Home address represents the starting address in the external display RAM for Graphic (Text) display, both the Graphic and Text Home Address can be merged together. The Graphic (Text) Area can be set during the initialization, it is better that it matches with the display columns defined by the hardware settings, so that the addressing scheme will be simpler.

3.51 Setting the Display Size

The actual display size is set using software commands. Two commands “Set Text Area” and “Set Graphic Area” are important to set the software-display size. Both commands need one data byte. Their values (in HEX) indicate how many bytes are used for one complete row, i.e. how “far” it is from one byte to the same byte of the next row. It makes sense to assign the same value to both Text Area and Graphics Area. Scrolling can be done by altering the Text Home Address. The external character generator area is always 2 Kbytes and can only be set to certain addresses.

3.52 Setting different Modes of Display

There are many modes by which the Graphics and Text can be displayed. The logical OR, EXOR, AND operation on the text and graphic display can be performed.

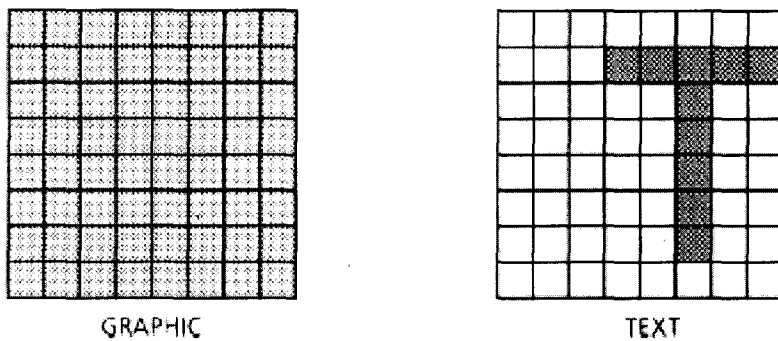


figure 3.4 Examples for graphic and text pattern

If the figure 3.4 patterns corresponds to the Graphic and Text respectively, then their display in different modes is shown in figure 3.5

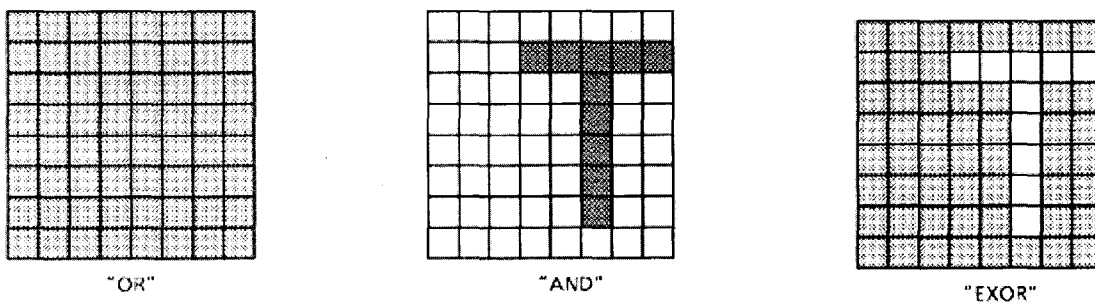


figure 3.5 Different display modes

The T6963C has the capability to generate characters in two ways. The first is with the built-in character generator ROM. The second way is using the external character generator. To ensure high flexibility, the T6963C can operate in two modes. The internal character generator ROM for the 128 pre-defined characters and the external character generator area for 128 user-defined characters. The other mode is using only the external character generator RAM for up to 256 user-defined characters. The two modes are called "Internal CGROM Mode" and "External CG-RAM Mode". It is possible to select either Internal Character Generator mode or External character Generator mode. If Internal CG ROM mode is selected character codes 00H to 7FH are assigned to built-in Character Generator ROM. The characters codes 80H to FFH are automatically assigned to the External Generator RAM.

The location of external CG-Area in VRAM is set by the command "Set Offset Register". The external CG-Area is always 2 Kbytes in size (2048 Bytes) and can store up to 256 user-defined characters. (Each character needs 8 bytes, $256 \times 8 \text{ bytes} = 2048 \text{ bytes}$). The command "Set Offset Register" needs two bytes of data to be sent before the command is sent to the T6963c controller. (The second data byte of the Command "Set Offset Register" must always be 00h) The five least significant bits of the first data byte sets the 2k external CG-Area within VRAM.

With five bits, 32 different memory segments can be selected ($2^5 = 32$). (There are 32 possible locations for a block of 2 Kbytes within 64 Kbytes of memory therefore 5 bits are enough to set the external CG-RAM area).

| Least significant data bits of first data byte | CG-Area (2 kByte) |
|--|-------------------|
| 00000 | 0000h – 07FFh |
| 00001 | 0800h – 0FFFh |
| 00010 | 1000h – 17FFh |
| | |
| 11101 | E800h – EFFFh |
| 11110 | F000h – F7FFh |
| 11111 | F800h – FFFFh |

Table:3.3 The five bit combination corresponds to a memory block addresses

In “Internal CG-ROM Mode” the first 1024 bytes (1024 = 400h) of the external character generator memory block are not used therefore the address of the first external character must have an offset of 400h to the start address (With a start address of 1000h for example the first character would be located at 1400h).

T6963C combines the Off Set Register value and character code to locate the physical memory location of the chosen character.

| | | | | | | | | | | | | | | | | |
|-----------------------|---|---|---|---|----|--|---|---|----|---|---|---|----|------------------|---|---|
| RAM Address (HEX) | 1h | | | | 4h | | | | 0h | | | | 0h | | | |
| VRAM Address (Binary) | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Actual Meaning | 5 Bits Offset Register Data (0 0010 = 02h) | | | | | 8 Bits Character Code (1000 0000 = 80h) | | | | | | | | 3 Bits Line Scan | | |

Each character in the external RAM is 8 bytes wide (A character can have a Font Size of up to 8 by 8 dots) and therefore the character with code 80h is located from 1400h to 1407h. (And therefore the data for this character must be written to these addresses. To show this character on the display, "Text Mode" must be enabled, the selected character is taken from the memory location in character RAM and stored in the TEXT area of VRAM prior to display on the screen. The address pointer must be set to the appropriate address in the Text Area of VRAM and then, the command "Write Data" must be executed with the data byte "80h". The T6963C knows automatically this is a character from the external CG-RAM and writes all 8 bytes from the CGRAM locations 1400h to 1407h to the appropriate text memory location

Note:

A character code of 7Fh represents the letter "f" for the common CG-ROM Type 0101. This character generator table is similar to a small area of the ASCII Table. The characters of the T6963C CG-ROM from 00h to 5Fh are the same as for ASCII 20h to 7Fh. The ASCII characters have an offset of 20h compared to the T6963C characters. If you want an ASCII character displayed, you need its hex code and subtract 20h. If you send this value as character code to the T6963C you have it.

3.6 Address Pointer Considerations

The T6963C address pointer has several functions. Basically, it is used to set the address in the VRAM where to read or to write data. The address pointer is set by the “Set Address Pointer” command 24h and needs two data bytes a low address byte and a high address byte. (The addresses in the VRAM are 16 Bits wide).

To write data from the MPU into the VRAM, the address pointer has to be set to the required address and a “Write Data” or “Auto Write Data” command executed. To read data from the VRAM, set the address pointer and execute a “Read Data” or “Auto Read Data” command. The “Write/Read Data” commands can be executed so that the address pointer is increased or decreased automatically. An “Auto Write/Read Data” command controls the increase of the address pointer. To show data on the display, set the address pointer to the appropriate address in the Text or Graphic Area in the VRAM, enable the display with the “Mode Set” commands and execute a “Write Data” Command. The data now stored in the VRAM area is shown automatically on the display.

3.7 A/D Conversion

The PIC does the A/D conversion by successive approximation technique and stores the 10-bit result in two registers ADRESH (Bank0) and ADRESL(Bank1).For simplicity I have opted the result to be Left justified by making the bit 7 of ADCON1 register ‘Low’, so that the ADC value in the register ADRESH can be considered as the final value, neglecting the two MSB bits of the register ADRESL. It will introduce a negligible error in the ADC value.

The ADC result in the ADRESH register is moved into the Banks successively, for the simplicity of the programming I have only stored 64 samples in each of the 3 Banks of data memory, the remaining one bank is used for scaling the ADC values. As the T6963C controller can maximum support LCD of the size 240 pixels Horizontally and 128 Pixels Vertically, the ADC value cannot be directly mapped on the screen, because the 8 bit ADC value will corresponds to 256 levels whereas vertically only 128 pixels are available on the LCD screen. Moreover as the T6963C is only byte addressable LCD controller, the scaling of the ADC value play an important role in displaying the smooth waveform.

3.8 Displaying the Waveforms on the LCD screen

The PIC first acquires the analog value, and the corresponding digital value is stored in the data memory. While displaying, the data from a bank is first displayed and simultaneously the displayed data is also erased from the memory, then the procedure is repeated for the data in the other banks. Only after the data in all the banks are displayed the PIC repeats procedure again and again, which makes it to function as an oscilloscope.

As the PIC 16F877A has got three banks for storing the ADC value, so it can be used as a single channel scope by storing the values simultaneously in all the three banks, or it can also be used utmost as a three channel scope by manipulating the register ADCON0 during the acquisition of the data. As mentioned earlier the register ADCON0 can be used to select the ADC channel.

Chapter 4

Building up of External Cavity Diode Laser

Diode lasers are very small source. They are characterized by very high efficiency, a long operating life and cost-efficient production processes. High-performance diode lasers therefore have great innovation potential and technical as well as economic advantages. The available wavelength range of diode lasers cover atomic transitions of many inorganic atoms such as Li, Na, Rb, Cs and Ca.

4.1 External Cavity Diode Laser (ECDL)

Laser cooling of atoms demands selective excitation of certain hyperfine transitions, which are separated by a few tens of MHz. This requires lasers with single mode output and linewidth less than the linewidth of the transition (~ few MHz). The laser should be tunable to access any transition of the hyperfine spectrum. In the free running Diode laser output is some tens of MHz wide with an output power of 80 -150 mW and can be continuously tuned only over certain limited region. The major drawbacks of the free running diode laser, when used for high resolution spectroscopy and laser cooling experiments, are multimode spectrum, large linewidth, difficulty of continuous tuning, mode hops and stringent requirements on the stability of temperature and injection current. As the laser power increases the spontaneous emission becomes less important at the higher photon densities and hence the linewidth decreases.

However the output power of the laser cannot be increased arbitrarily (large). A more efficient method to reduce the linewidth is to make the cavity longer, this can be achieved by the use of optical feedback to control the laser.

The external cavity device provides wavelength/frequency tuneability also. The extension of the laser cavity length and wavelength selection can be achieved by a wavelength dispersive element as one end of the cavity. Usually diffraction gratings are used. The lasing cavity is now determined by back facet of the diode and the grating as shown in Fig.4.1, the increased cavity length (~few cm) reduces intermode spacing.

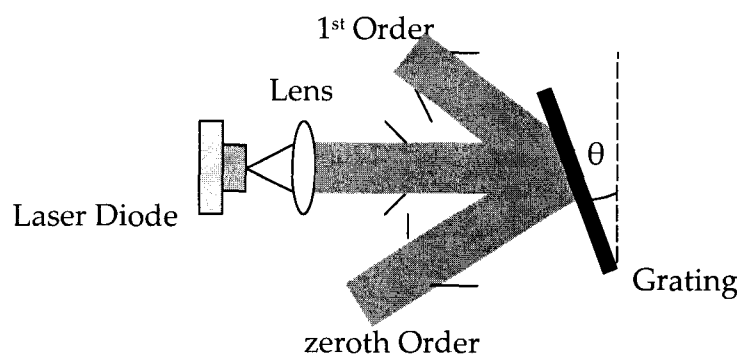


Figure 4.1 : Schematic of ECDL

The optical feedback from the grating narrows the linewidth of laser output to approximately 1MHz and provides a simple method for tuning the laser over a range of $\approx 25\text{nm}$. The laser has a frequency tuning range of 6nm for a temperature change of 30K. Note that grating feedback tuning range in the ECDL configuration is much larger than the temperature tuning range. The grating will select the lasing emission from the wide gain spectrum of a free running laser giving a narrow linewidth at a desired wavelength. Further more wavelength tuning of the output may be achieved by a mechanical rotation of the grating.

The main components of an ECDL are :

Base Plate: the plate on which all the components are mounted. This is made of material with low thermal coefficient of expansion.

The Diode Laser: we normally use Sanyo or Sharp diode, which delivers ~80mW power at ~100mA current.

The Collimating Lens: as the output of the laser is divergent ,this component is most essential. Typically, anti-reflection coated lenses,4-6mm focal length are used.

The Diffraction Grating : we usually use the grating of groove density 1200 lines/mm.

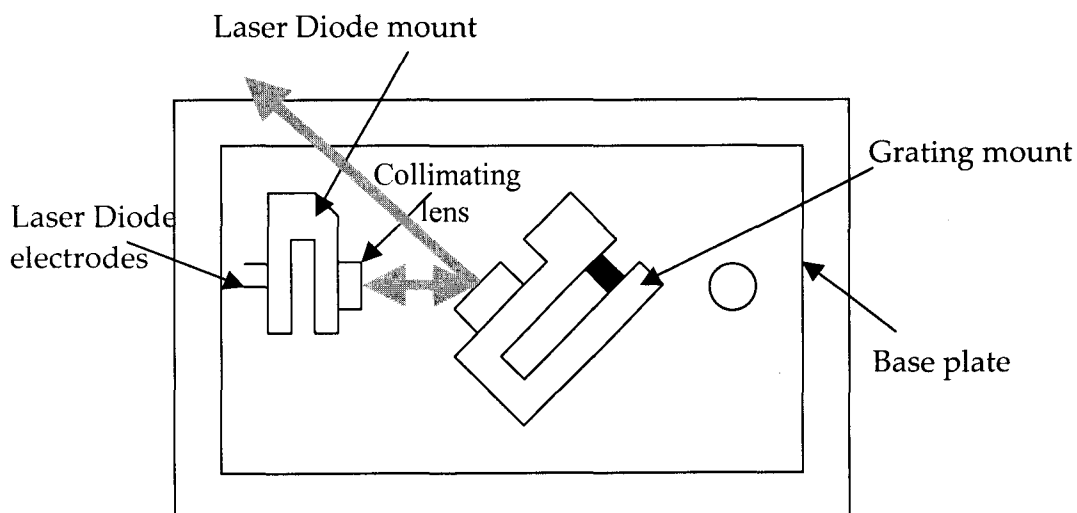


Figure 4.2 Components used in an ECDL

Two external cavity configurations are commonly used for narrowing linewidth of the diode laser.

- 1) Littrow configuration of grating
- 2) Littmann configuration of the grating

Note: We generally build ECDL with the Littrow configuration.

Littrow cavity

In this type the cavity is formed between the rear facet of the diode laser and the reflection type grating in such a way that the zeroth order beam goes out as output beam and first order beam is coupled back to the diode.

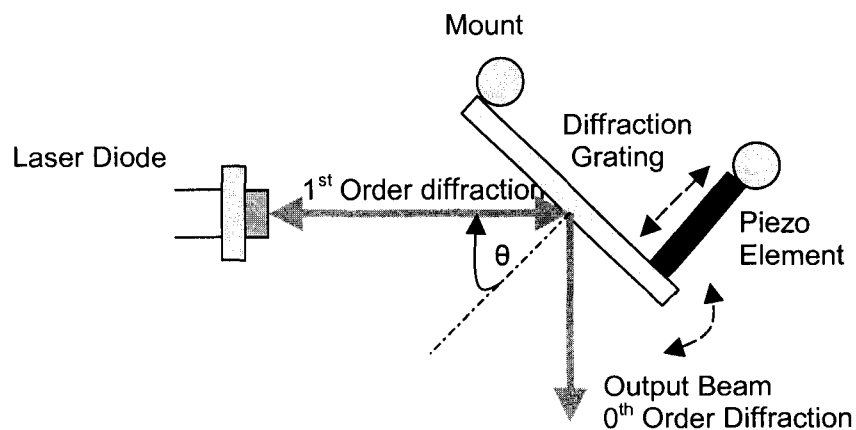


figure 4.3 Littrow configuration of Grating

In the Littrow configuration as shown in the Fig4. 3 the angle of incidence and the angle diffraction are the same.

The diffraction from a grating is described by the equation

$$d(\sin \theta_m - \sin \theta_i) = m\lambda$$

'd' the spacing of the grating

' θ_m ' the angle to the normal of the mth order diffracted beam

' θ_i ' the angle to the normal of the incident beam

'm' order of diffraction

' λ ' the wavelength of the incident/diffracted light

If the incident angle is such that the first order diffraction beam counter propagates the laser output,

the equation simplifies to

$$2d \sin \theta = m\lambda$$

where θ is the Littrow angle for a given wavelength λ . It is better to use a blazed grating to get maximum efficiency in the first order. The tuning of the cavity can be achieved by changing the position of grating by the adjustment of screws on a coarse scale and by PZT displacement on a fine scale. In Littrow configuration the grating mount permits fine adjustment about two perpendicular axes, one horizontal, one vertical. The grating angle is so adjusted that the +1st order reflection goes back, through the lens onto the diode, providing the feedback. The setting of the grating for feedback is the most crucial part in building the ECDL. Incorrect placement may lead to mode jumps or multimode operation. when the feedback is the best, the threshold current is the least.

4.2 Temperature controller

The laser has a frequency tuning range of 6nm for a temperature change of 30K. Thus to stabilize the frequency it is very important to stabilize the diode's temperature. Precise control of the temperature of both the laser diode and the base plate is essential for a long term reliable operation of the laser at a particular wavelength. The cooling of the base plate and Laser diode can be achieved using Peltier.

A peltier is an electronic package that consists of two sides say A and B, one of which 'A' is placed in contact with the sample whose temperature is to be controlled and the other side 'B' to a heat sink. Depending on the direction of current flow, one may heat A or cool it. The side B then gets cooled or heated.

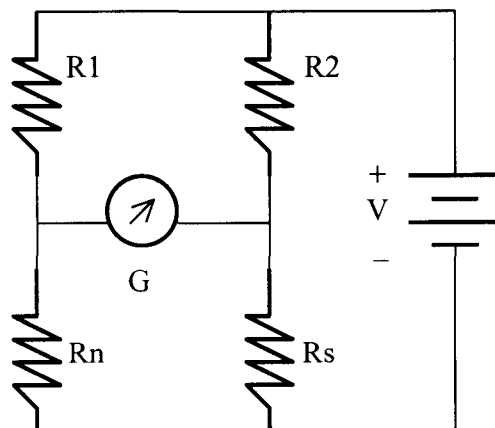


Figure 4.4: Wheatstone Bridge used in Temperature Controller

The temperature controller is a wheatstone bridge, with one of its arms consisting of a Negative Temperature Co-efficient (NTC) thermistor as shown in

Figure 4.4. Initially, the thermistor is calibrated to find its temperature equivalent voltage, from which the desired set temperature resistors are used.

'Rn' NTC thermistor resistance

'Rs' Set temperature equivalent resistance

The equation for a balanced bridge is given by $R_n / R_1 = R_s / R_2$

For a desired temperature, we select the value of R_s , which unbalances the bridge resulting in a voltage across the voltmeter. This voltage is fed back to the Peltier as an error signal. Whenever there is any change in the temperature of the ECDL box, the resistance of the NTC changes. This change results in a voltage developing across the wheatstone bridge. This voltage is taken as an error signal by the temperature controller and a feedback signal is generated which is fed to a peltier situated in the ECDL box. Depending on the polarity of the feedback signal, the peltier either cools or heats the ECDL box.

4.3 Laser Diode protection circuit

Laser diodes are very sensitive to voltage spike. The active region is very narrow ($3\mu\text{m}$) and an overvoltage of just a volt will result in extremely large electric fields. The purpose of protection circuit is

1. To protect laser from over voltages or spikes.
2. To protect the laser from reverse voltages
3. To maintain steady current if there is a momentary reduction in voltage
4. To limit the current through the laser diode.

As the pin configuration and the characteristic parameters are different for different laser diodes, separate protection circuits have to be built for each laser diode. Each Laser diode package has three electrodes, one common electrode and two free electrodes, one from laser diode and the other from photo diode. Therefore, according to the electrode configurations the laser diodes can be divided into two major categories, namely "common anode" and "common cathode".

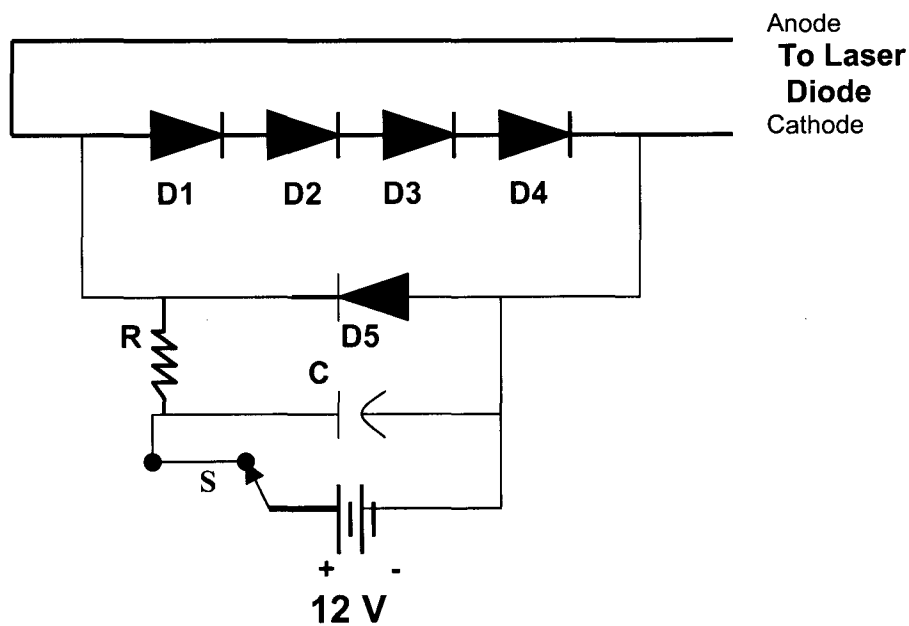


Figure 4.5: Laser Diode Protection Circuit

In the protection circuit figure 4.5 , the current flows through the resistance(R) and then can split into four parallel arms. one of the arms contains the laser diode which is Forward biased , another contains 'n' number of forward biased

fast diodes in series. Another arm contains a reverse biased fast diode and the last arm has a capacitor (~100nF)

The forward biased voltage drop for each diode is 0.7 V, 'n' should satisfy the condition.

$$n \times (0.7) \leq V_{op}$$

where V_{op} is the maximum operating voltage of the laser diode, and 'n' is an greatest integer that satisfies the inequality. If there is an over voltage the diodes in series come into conduction and ensure that no excess voltage drops across laser diode. If the protection circuit gets connected to laser driver wrongly (LDA and LDC get exchanged) then only reverse biased diode will conduct. When the voltage suddenly drops, the capacitor acts as source, maintaining the laser current.

Based on the principles discussed above, an external cavity diode laser was built and adjusted for optimum feedback. The laser was then tuned to the required wavelength-in this case the $5S_{1/2} F=3$ to $5P_{3/2} F^1$ transitions in ^{85}Rb .

4.4 Saturation Absorption spectroscopy

The simplest spectroscopy that can be performed with the laser diode is to observe the absorption and Doppler-free saturated absorption of Rubidium atom. This can be easily done in a small glass vapor cells which are at room

Say Rubidium

temperature. It also provides the simplest way to determine the short and long term frequency stability and tuning behavior of the laser frequency.

The Doppler width for a gas can be estimated from $\Delta\nu = 2k v_{mp}$, where v_{mp} = most probable velocity at the temperature of the gas, and $k = 2\pi/\lambda$. At room temperature both for ^{85}Rb and ^{87}Rb $v_{mp} \approx 230\text{m/s}$ and the Doppler width is $\approx 2\text{ GHz}$. The natural line width of a hyperfine transition on the other hand, $\sim 2\pi \times 6\text{MHz}$, and the spacing between various hyperfine levels are few tens to few hundreds of MHz. These, therefore, can not be resolved as the Doppler linewidth is much larger. The technique of Doppler free saturation absorption is used to select out from a collection of atoms moving with Maxwellian distribution of velocities, those atoms that have zero velocity component in the direction of the input beam. This then yields absorption lines, which have widths equaling their natural linewidth.

Saturated absorption spectroscopy is a kind of nonlinear spectroscopy. In the linear spectroscopy a single propagating wave is incident on the sample, some photons are absorbed and only some fraction of the wave reaches the detector. Here the radiation reaching the detector is proportional to the radiation incident on the sample. In nonlinear spectroscopy there are two counter propagating beams which interact with the same atoms in the region where they intersect. In this case the radiation reaching the detector is dependent on both the beams. Saturated absorption eliminates Doppler broadening to uncover hyperfine structure. Doppler free saturated absorption spectroscopy is insensitive to this effect and thus allows the splitting between energy levels of a

sample to be measured with much greater precision. The fundamentals of Doppler free saturated absorption spectroscopy were explored for a rubidium sample. Rubidium sample can be theoretically treated as a hydrogen like atom because of its single valence electron. Rubidium can absorb radiation when the frequency corresponds to a difference in energy levels exciting the valence electrons to a higher state. The absorption spectra in traditional spectroscopy are characterized by broad signals around each absorbed frequency. This is a result of Doppler effect. When the laser tuned to resonant frequency the difference between the probe beam that passes through the pump beam and the probe beam that is less effected by the pump beam gives the Doppler free emission spectrum.

4.41 Doppler broadening

Random thermal motion of atoms or molecules creates a Doppler shift in the emitted or absorbed radiation. The spectral lines of such atoms or molecules are said to be Doppler broadened since the frequency of radiation emitted or absorbed depends on atomic velocities. If an atom is moving towards or away from a laser radiation, then it sees radiation that is blue or red shifted respectively. If an atom at rest, relative to the laser absorbs radiation of frequency ' ν_0 ', then when the atom is approaching the laser it will see blue shifted radiation, hence for absorption to occur the frequency of the laser must be less than ' ν_0 ' in order for it to be blue shifted to the resonance value of ' ν_0 '. Similarly, if the atom is receding from the laser, the frequency must be greater than ' ν_0 ' for absorption to occur.

In a quantitative way, let us consider ν_0 is the atomic resonant frequency when

the atom is at rest in the frame of the laser. Consider the atom is moving along the x axis with a velocity v_x , then the frequency of the absorbed radiation in the rest frame of the laser will be ν_L ,

$$\text{where } \nu_L = \nu_0 (1 + v_x/c).$$

If v_x is negative, moving towards the laser, then $\nu_L < \nu_0$, that is the atom moving toward the laser observes radiation that is blue shifted from ν_L upto ν_0 . If v_x is positive, moving away from the laser, then $\nu_L > \nu_0$, that is, the atom absorbs radiation that is red shifted from ν_L down to ν_0 . Therefore an ensemble of atoms having a distribution of speeds will absorb light over a range of frequencies. The atoms in the cell are moving with velocities that have a Maxwell Boltzmann distribution.

The probability of having a particle in the velocity range v_x and $v_x + d v_x$ is given by

$P(v) \propto \exp(-mv^2_x / 2kT)$, where v_x is the particle velocity and k is the Boltzmann constant and T is the temperature in kelvin.

The full width half maximum amplitude of the Doppler broadened signal is given by

$$\Delta\nu_{1/2} = 2 \frac{\nu_0}{c} \sqrt{\frac{2kT \ln 2}{M}}$$

M is the mass of absorbing atom in kilograms. This $\Delta\nu_{1/2}$ is about 500MHz for Rubidium at 300K.

In Doppler free saturated absorption spectroscopy the absorption spectra of two probe beams are measured. The pump beam which is counter propagating with one of the probe beams changes the ground state population of the rubidium atoms. Since these intersecting pump and probe beams are propagating in opposite directions, only a narrow velocity class of atoms centered about $V = 0$ will interact with both probe and pump. Other velocity classes of atoms will interact either with the probe beam or with the pump beam, depending on the laser detuning from the main transition and on the magnitude of the Doppler shift by the atom's velocity. The $V = 0$ class of atoms are saturated by the strong beam. This means that almost half the atoms are promoted to the excited state of the transition so that the weak probe beam is less attenuated by the Rb vapour. In other words there are less atoms in the ground state to absorb the probe light.

Consider the frequency $\nu_L \gg \nu_0$ or $\nu_L \ll \nu_0$ The velocity classes of atoms $v_x = -c(\nu_0 / \nu_L - 1)$ will partially absorb the pump beam in the case $\nu_L < \nu_0$. These atoms see the frequency of the pump beam shifted upto their transition frequency. The probe beam will be partially absorbed by the velocity class $v_x = c(\nu_0 / \nu_L - 1)$ at the same time. Since these are completely independent velocity classes of atoms, the pump does not affect the absorption of the probe beam. Only when the laser frequency is equal to the transition frequency both beams are absorbed by the same velocity class of atoms by those centered about $v_x = 0$. Subtracting from the crossed probe beam signal from the unaffected probe beam signal will result in sharp peaks at the resonant frequencies.

4.42 Saturation Absorption Spectroscopy: Experimental

set- up

The experimental set up for the saturation absorption spectroscopy is shown in the Fig 4.6 Using mirrors and beam splitters, the beam from a diode laser was split into three parts. The beam splitter reflects a fraction of the laser light on each surfaces (' 4% at each reflection)to get two weak parallel probe beams. These probe beams were sent through a cell containing rubidium atoms. After decreasing their intensity so as not to saturate the detectors, they were measured by the two photodiode detectors connected to an oscilloscope. The remaining high intensity is reflected by two mirrors and is aligned to counter propagate over one of the probe beams in the Rb cell. Since these intersecting beams are opposite in directions the atoms having velocity $v = 0$ will interact with both pump and probe. The $v = 0$ class of atoms are saturated by the strong pump beam. So the weak probe beam is less attenuated by the Rb vapor. Two different photodiodes detect the two probe beams separately. The photodiode currents are converted into voltages by means of I/V converters in both the cases. Both the photodiodes have their own separate gain stages and offset stages. Gain stage help to make the intensities and shapes of the two beams equal before the introduction of the pump beam. Offset control is to eliminate the effects of dark currents in the photo detector signal. When the Doppler profile of the unsaturated second probe beam is electronically subtracted from the signal obtained from the saturated probe beam using a differential amplifier, we get a Doppler free hyperfine spectrum. Here we have selected only the brightest fluorescence and ramped the fluorescence by using the piezo. A triangular waveform drives the piezo at a frequency of 1Hz and at a voltage of 0. 5Volt,

hence sweeps the laser frequency. Ramping the frequency up and down, one obtains a partial mirror image of the spectrum.

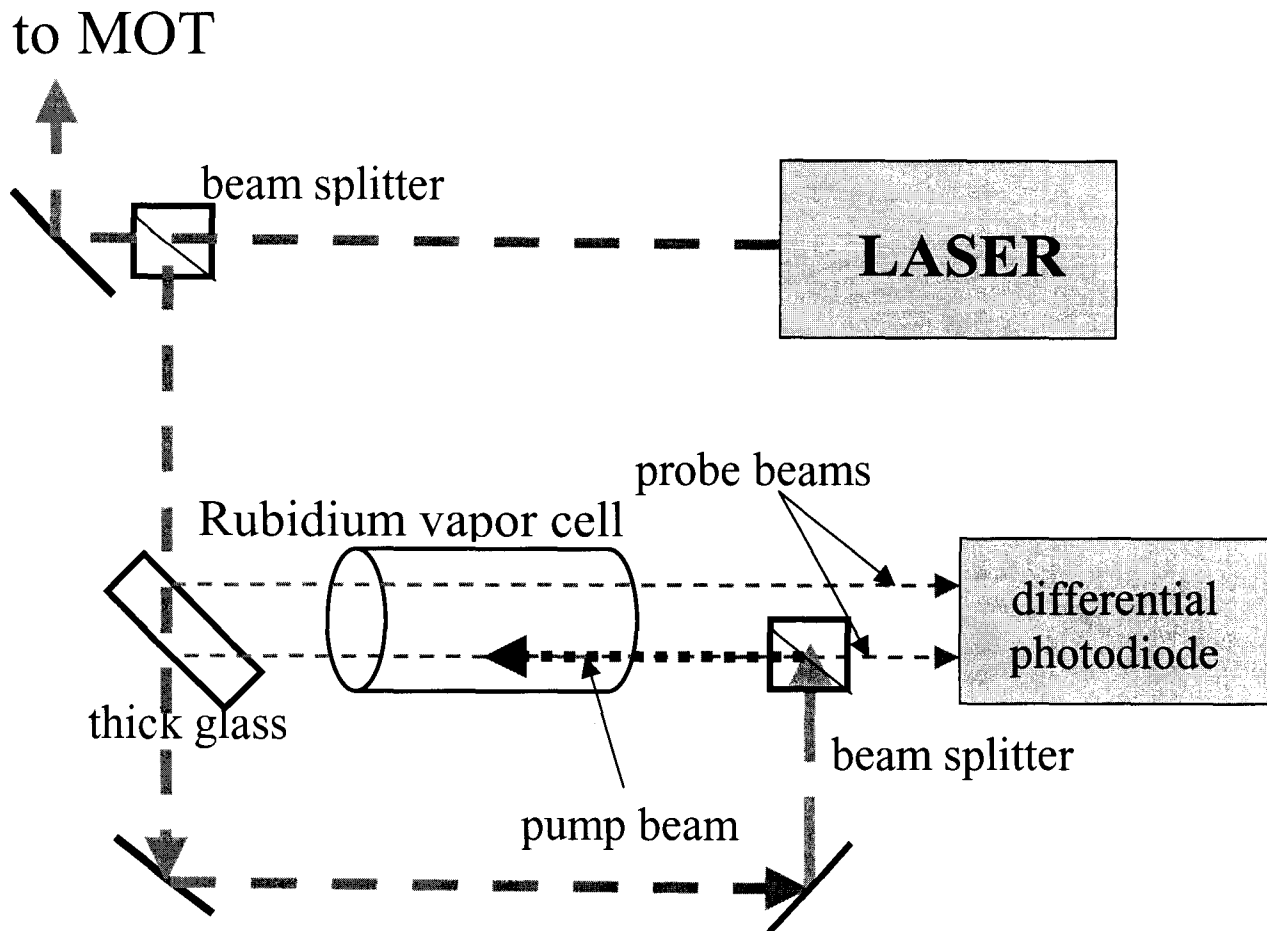


Figure 4.6: Experimental set up of saturation absorption spectroscopy

The Fig 4.7 gives the Doppler broadened signal and saturation absorption signal of Rb85 riding on the Doppler broadened signal. The next Fig4.8 gives the difference signal, the Rb85 cooling spectrum, along with the ramp signal. These are the signals that are desired to be displayed on the Mini-Oscilloscope. The laser can be locked at a particular frequency by monitoring these signals.

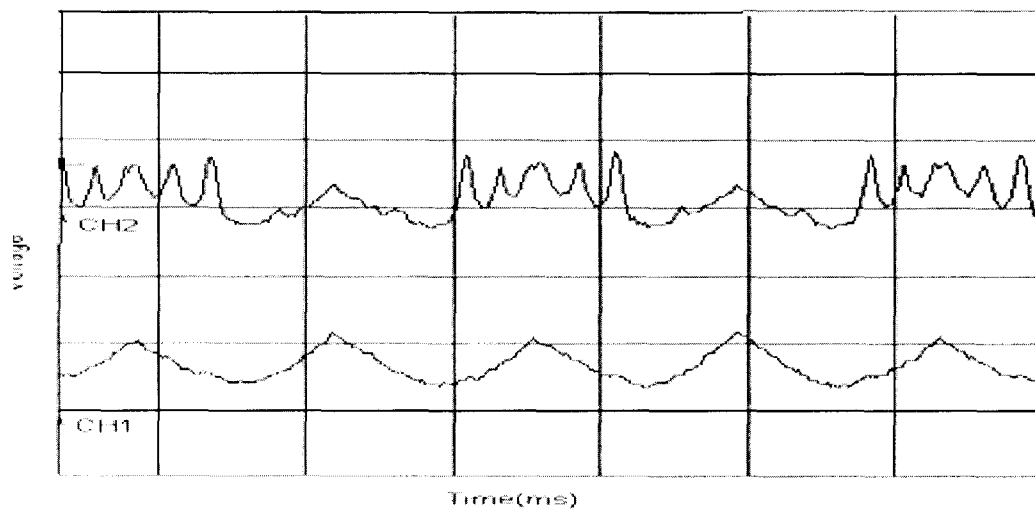


Figure 4.7: (A) Doppler broadened signal (B) Doppler along with Rb85 cooling

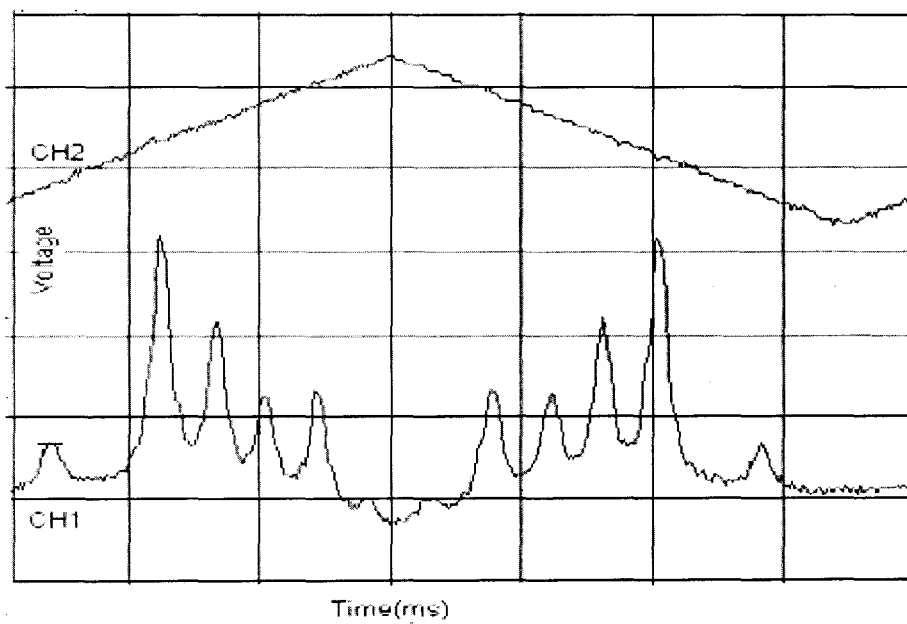


Figure 4.8: Cooling spectrum of Rb85 *hot*

4.5 Frequency Locking

The frequency of the diode laser can be locked to the saturated absorption component with the help of an electronic servo control, where any deviation from the locked frequency is taken as the error signal which is applied to the external cavity component for controlling the cavity length so as to pull the laser frequency to the locked position. One method of locking the laser frequency is through lock-in amplifier. In this technique, the PZT is modulated sinusoidally at a reference frequency (800Hz) and the resulting modulated saturation absorption signal is phase sensitively detected at the reference frequency and the laser frequency is locked at the peak. Any deviation from the locked position again generates an error signal which after smoothing and amplification is applied to PZT.

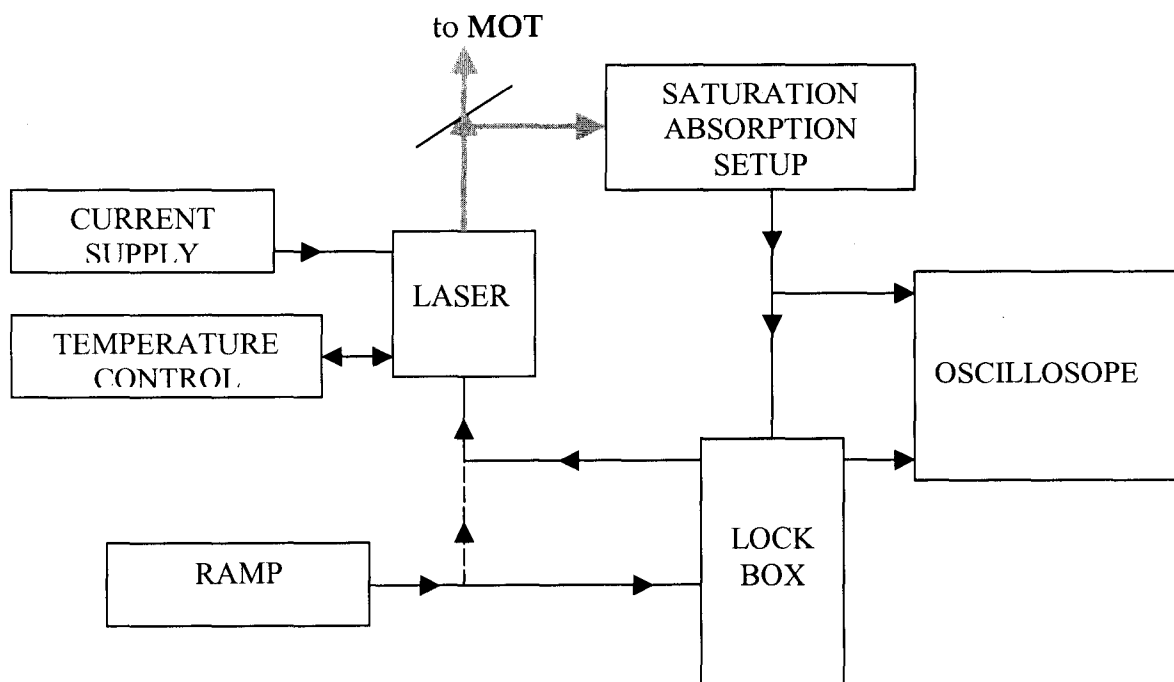


Figure 4.9 lock box connected to ECDL.

The laser is scanned across the hyperfine lines by sending a ramp signal to the piezo-electric transducer (PZT) attached to the grating mount. The ramp signal sent to the PZT is drawn from the "Lock-in". We then reduce the ramp amplitude so that the frequency scanning gives rise to only one of the peaks of the saturated absorption spectrum, the one to which we want to lock the laser frequency. As the ramp signal sent to the PZT is modulated using sinewave, the detector output, which earlier gave smooth peak, is now distorted due to modulation. The new signal is multiplied in a multiplier stage in the lock-in circuit with the same sine wave which was used to modulate the ramp signal. The output of this stage is passed through a low pass filter. A dc signal is obtained which is derivative corresponds to the peak of the hyperfine absorption where we want to lock the frequency. The error signal produced (its the derivative of the actual signal) crosses zero exactly at the position of the peak of the spectrum, where the laser should be locked, and is positive at one side and negative at the other side. Because of this it is a suitable feedback signal to lock the frequency of the laser exactly at the top of the spectrum. The error signal monitor thus becomes an excellent indicator of the magnitude and spectral characteristics of the compensated noise.

The drift rate of the unlocked laser is normally under 5 MHz/min when the system is properly stabilized, and this slow drift is eliminated by locking. The short-term jitter amplitude of the unlocked laser frequency is typically ± 3 MHz on a 1-s time scale, the short-term intensity variation are much smaller than 1%. When locked, the laser frequency is stabilized to 1MHz or better.

Chapter 5

Design and Results

The microcontrollers currently available are not fast enough to simulate in software, test equipment such as a 50MHz oscilloscope. However, they are capable of easily simulating lower speed devices, such as volt-meters, and low speed (perhaps around 1-5MHz) sampling devices. These type of applications still have a very wide use, and thus any device based on current microcontroller technology could provide a solution with real benefits.

The microcontroller based Mini- oscilloscope is integrated to the external cavity diode laser, the figure 5.1 shows block diagram of the interconnection of entire system .

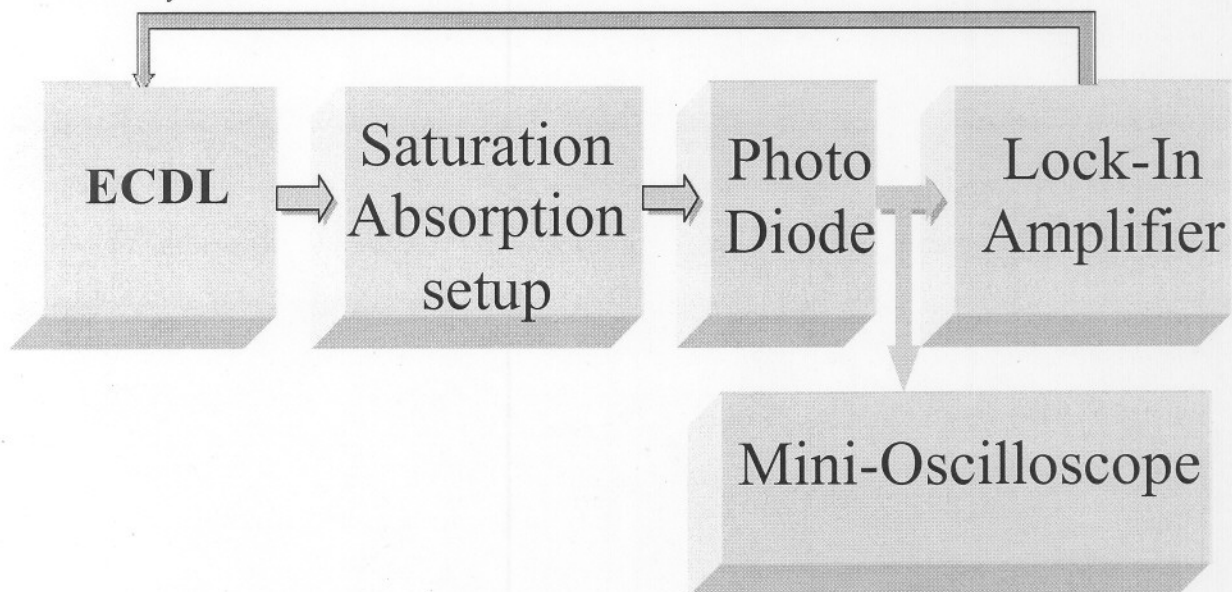
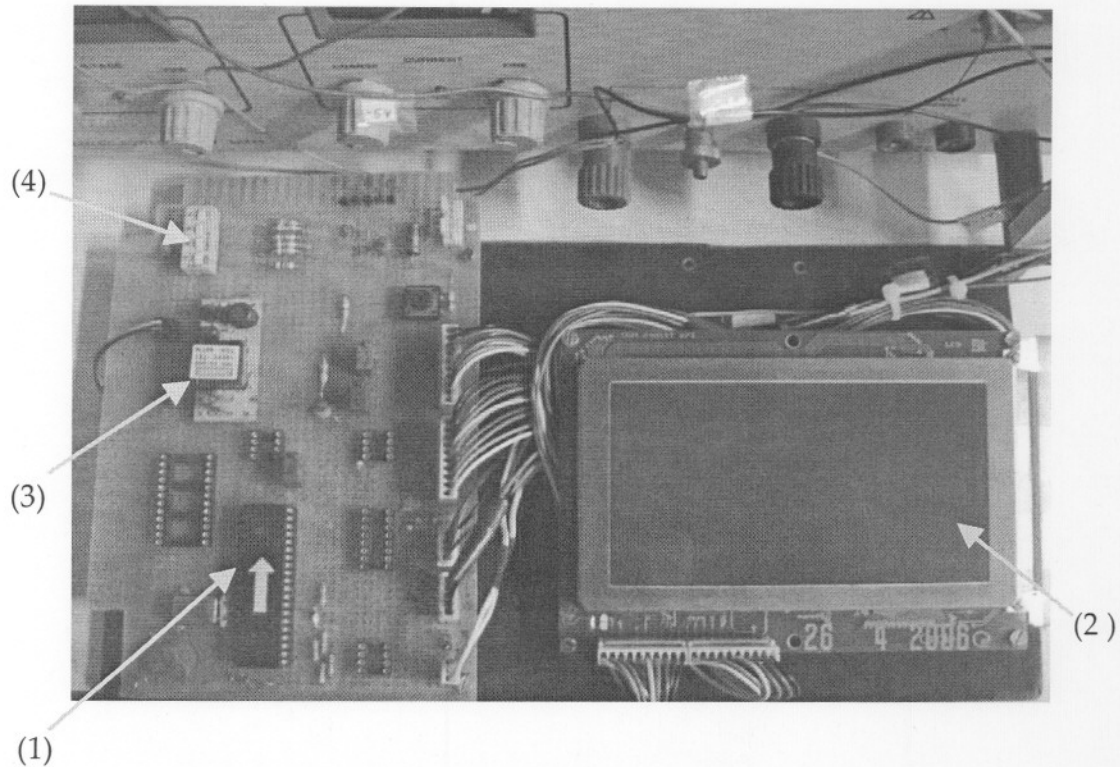


Figure 5.1 Block diagram of the Mini-oscilloscope connected to the ECDL

The Mini-oscilloscope built for this project is shown in the figure 5.2



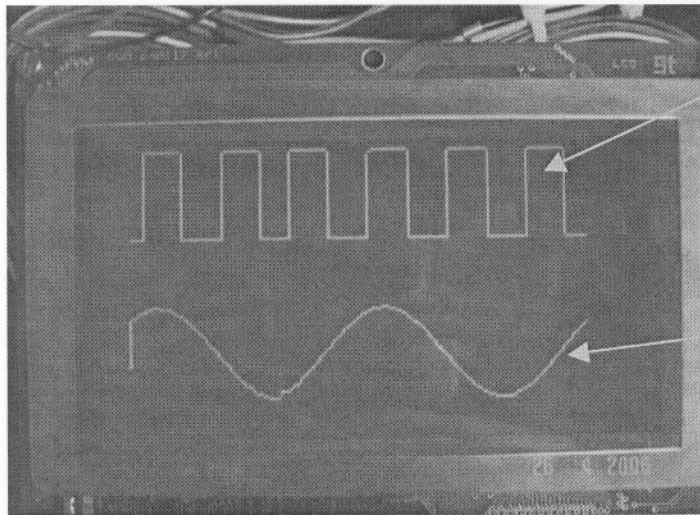
- (1) The PIC 16F877A Microcontroller, (2) The Graphical LCD screen
(2) The CCFL Inverter, (4) Dip switches to provide different sampling rates.
(3)

Figure 5.2 Circuit built for Mini-Oscilloscope

Features of Mini-oscilloscope

- Can be used to monitor low frequency signals
- Two channel scope
- Different sampling rates are provided
- Variable voltage division for ADC

The figure 5.3 (A) , 5.3 (B) shows the display of the signal fed from the signal generator on the LCD screen .



Channel 1

Signal Frequency = 30 Hz

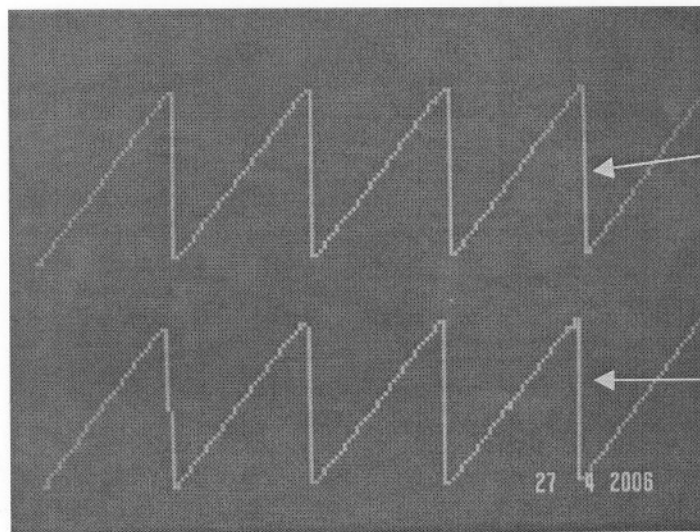
P - P Voltage = 3.5 volts

Channel 2

Signal Frequency = 10 Hz

P - P Voltage = 3.5 volts

(A)



Channel 1

Signal Frequency = 50 Hz

P - P Voltage = 3.8 volts

Channel 2

Signal Frequency = 50 Hz

P - P Voltage = 3.8 volts

(B)

Figure 5.3 Mini-oscilloscope as two channel display

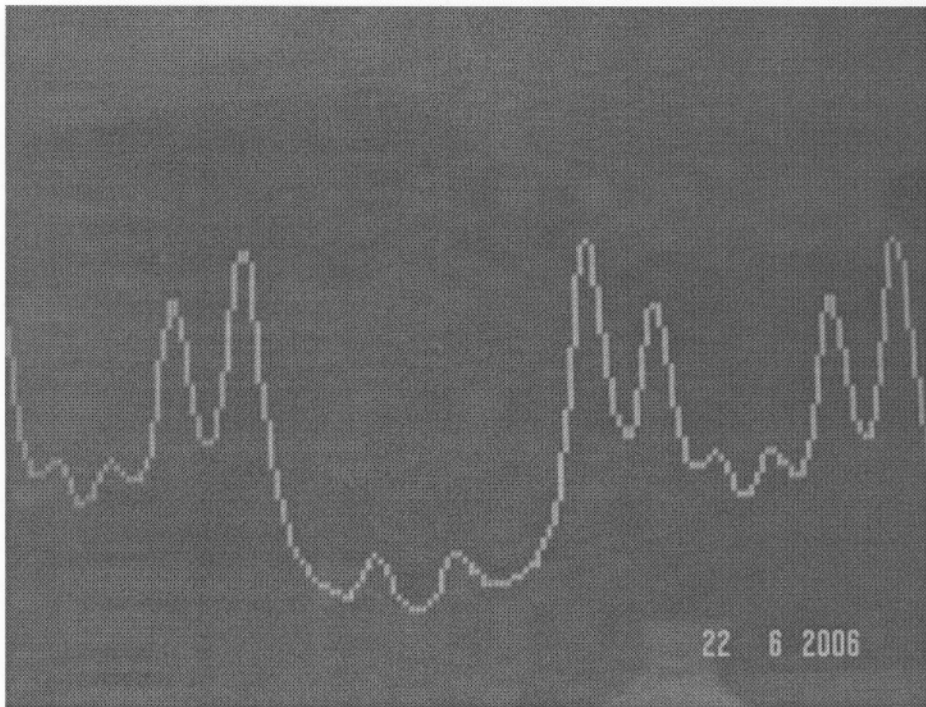


Figure 5.4(a) ^{85}Rb cooling signal on Mini-oscilloscope.

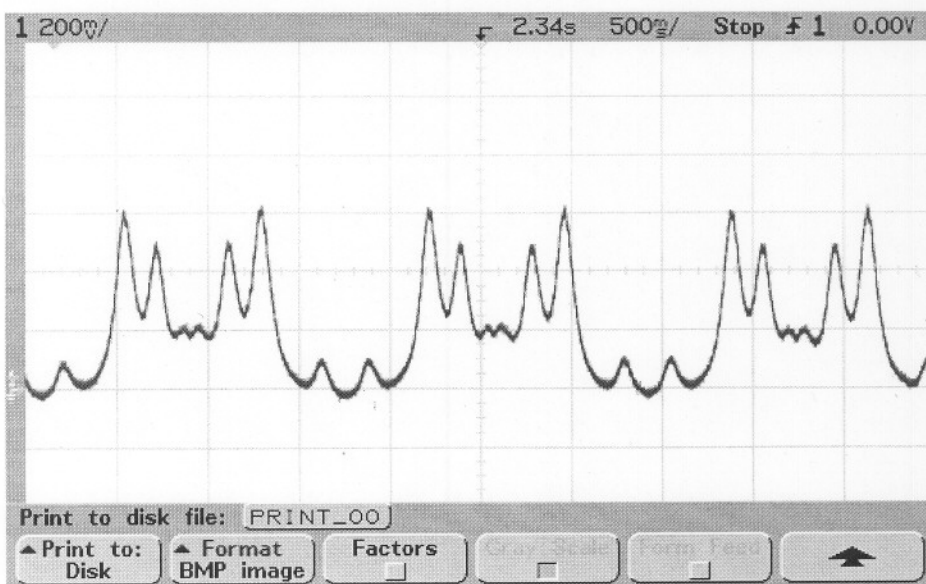


Figure 5.4(b) ^{85}Rb cooling signal on Agilent oscilloscope.

Signal Frequency ~ 0.6 Hz
P-P voltage ~ 625 mV

The figure 5.4(a), gives the display of the Rb cooling signals on the Mini-Oscilloscope. It can be compared with display on Agilent oscilloscope, shown in figure 5.4 (b).

Limitations

There are several limitations in using PIC microcontroller. The PIC 16F877A has only 35 instructions that can be used for all the logical, arithmetic, data transfer operations.. etc., So there may be some difficulty while writing the program , and the program can become slightly lengthier. As PIC 16F877A can have the maximum of 5 Volts as the Reference for the A/D conversion, the signals with peak to peak voltage higher than 5 Volts cannot be displayed.

If the proper timing is not provided to the LCD, synchronization problem can arise. The consequences of this is that 'One command or data byte is sent, and the LCD interpret as some other way'. So the synchronization plays a major role while interfacing the LCD to a microcontroller. Proper delay should be provided while programming the PIC, so that the all the commands and data bytes are sent in a correct sequence.

Note :

The assembly language program written for this project is given in Appendix 4

Chapter 6

Conclusion

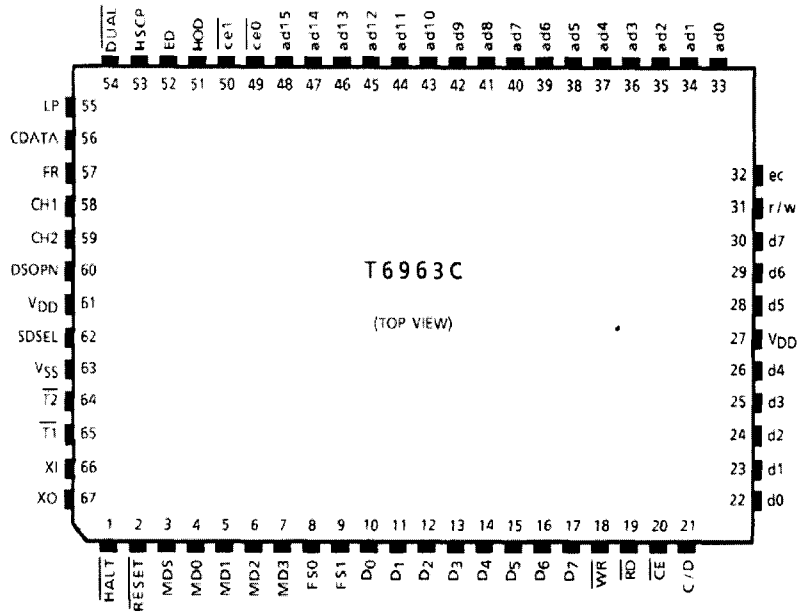
Built the circuit for microcontroller based Mini-Oscilloscope. Used PIC- 16F877A microcontroller as the processing unit and Graphical LCD module with T6963C controller as display panel. Written the assembly language program on the MPLAB software. Initially the Mini-Oscilloscope was tested feeding the signals from signal generator. Provided facility for having different sampling rate and variable voltage division. Built one external cavity diode laser(ECDL)using a Sanyo IR laser diode which has a lasing wavelength in the range of 775 to 783nm. The temperature of the ECDL is properly stabilized. Then integrated the Mini-Oscilloscope to the ECDL and observed the fluorescence spectrum using saturation absorption setup.

Bibliography

1. PIC16F87X Data Sheet
2. Application Notes for the T6963C LCD Graphics Controller Chip
3. T6963C datasheet by Steve Lawther
4. Everyday Practical Electronics (EPE) Magazine, February 2001
supplement
5. *Frequency Locking of ECDL*
Mrs. Preethi T.M.
6. Carl. E. Wieman, Leo Hollberg, "Using diode lasers for atomic physics",
Rev. Sci. Instrum. 62(1), 1-20, 1991.
7. C. J. Hawthorn, K. P. Weber, and R. E. Scholten, "Littrow configuration
tunable external cavity diode laser with fixed direction output beam", Rev. Sci.
Instrum. 72(12), 2001
8. Advanced optics laboratory, "Doppler-free saturated absorption spectroscopy:
Laser spectroscopy".

Appendix –1
T6963C Datasheet

PIN ASSIGNMENT



PIN FUNCTIONS

| PIN NAME | I/O | FUNCTIONS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------------------|----------|--|------|-----|----|----|-----|-----|-----------|----|----|-----|---------|-----|-----|-----|-----|----|---|-----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------|---|---|---|---|----|----|----|----|---|---|----|----|----|----|----|----|--------|----|----|----|----|----|----|-----|-----|----|----|----|-----|-----|-----|-----|-----|--|----------|--|--|--|--|--|--|--|-----------|--|--|--|--|--|--|--|
| MDS MD0 MD1 | Input | Pins for selection of LCD size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <tr> <td>DUAL</td> <td>H</td> <td>H</td> <td>H</td> <td>H</td> <td>H</td> <td>H</td> <td>H</td> <td>H</td> <td>L</td> <td>L</td> <td>L</td> <td>L</td> <td>L</td> <td>L</td> <td>L</td> <td>L</td> </tr> <tr> <td>MDS</td> <td>L</td> <td>L</td> <td>L</td> <td>L</td> <td>H</td> <td>H</td> <td>H</td> <td>H</td> <td>L</td> <td>L</td> <td>L</td> <td>L</td> <td>H</td> <td>H</td> <td>H</td> <td>H</td> </tr> <tr> <td>MD1</td> <td>H</td> <td>H</td> <td>L</td> <td>L</td> <td>H</td> <td>H</td> <td>L</td> <td>L</td> <td>H</td> <td>H</td> <td>L</td> <td>L</td> <td>H</td> <td>H</td> <td>L</td> <td>L</td> </tr> <tr> <td>MD0</td> <td>H</td> <td>L</td> <td>H</td> <td>L</td> <td>H</td> <td>L</td> <td>H</td> <td>L</td> <td>H</td> <td>L</td> <td>H</td> <td>L</td> <td>H</td> <td>L</td> <td>H</td> <td>L</td> </tr> <tr> <td>LINES</td> <td>2</td> <td>4</td> <td>6</td> <td>8</td> <td>10</td> <td>12</td> <td>14</td> <td>16</td> <td>4</td> <td>8</td> <td>12</td> <td>16</td> <td>20</td> <td>24</td> <td>28</td> <td>32</td> </tr> <tr> <td>V-DOTS</td> <td>16</td> <td>32</td> <td>48</td> <td>64</td> <td>80</td> <td>96</td> <td>112</td> <td>128</td> <td>32</td> <td>64</td> <td>96</td> <td>128</td> <td>160</td> <td>192</td> <td>224</td> <td>256</td> </tr> <tr> <td></td> <td colspan="8">1 SCREEN</td> <td colspan="8">2 SCREENS</td> </tr> </table> | DUAL | H | H | H | H | H | H | H | H | L | L | L | L | L | L | L | L | MDS | L | L | L | L | H | H | H | H | L | L | L | L | H | H | H | H | MD1 | H | H | L | L | H | H | L | L | H | H | L | L | H | H | L | L | MD0 | H | L | H | L | H | L | H | L | H | L | H | L | H | L | H | L | LINES | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | V-DOTS | 16 | 32 | 48 | 64 | 80 | 96 | 112 | 128 | 32 | 64 | 96 | 128 | 160 | 192 | 224 | 256 | | 1 SCREEN | | | | | | | | 2 SCREENS | | | | | | | |
| | | DUAL | H | H | H | H | H | H | H | H | L | L | L | L | L | L | L | L | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | MDS | L | L | L | L | H | H | H | H | L | L | L | L | H | H | H | H | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | MD1 | H | H | L | L | H | H | L | L | H | H | L | L | H | H | L | L | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | MD0 | H | L | H | L | H | L | H | L | H | L | H | L | H | L | H | L | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | LINES | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| V-DOTS | 16 | 32 | 48 | 64 | 80 | 96 | 112 | 128 | 32 | 64 | 96 | 128 | 160 | 192 | 224 | 256 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 1 SCREEN | | | | | | | | 2 SCREENS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MD2 MD3 | Input | Pins for selection of number of columns | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <tr> <td>MD2</td> <td>H</td> <td>L</td> <td>H</td> <td>L</td> </tr> <tr> <td>MD3</td> <td>H</td> <td>H</td> <td>L</td> <td>L</td> </tr> <tr> <td>Columns</td> <td>32</td> <td>40</td> <td>64</td> <td>80</td> </tr> </table> | MD2 | H | L | H | L | MD3 | H | H | L | L | Columns | 32 | 40 | 64 | 80 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | MD2 | H | L | H | L | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MD3 | H | H | L | L | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Columns | 32 | 40 | 64 | 80 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| FS0 FS1 | Input | Pins for selection of font | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <tr> <td>FS0</td> <td>H</td> <td>L</td> <td>H</td> <td>L</td> </tr> <tr> <td>FS1</td> <td>H</td> <td>H</td> <td>L</td> <td>L</td> </tr> <tr> <td>Font</td> <td>5x8</td> <td>6x8</td> <td>7x8</td> <td>8x8</td> </tr> </table> | FS0 | H | L | H | L | FS1 | H | H | L | L | Font | 5x8 | 6x8 | 7x8 | 8x8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | FS0 | H | L | H | L | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| FS1 | H | H | L | L | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Font | 5x8 | 6x8 | 7x8 | 8x8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| D0 to D7 | I/O | Data I/O pins between CPU and T6963C (D7 is MSB) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| WR | Input | Data Write. Write data into T6963C when WR = L. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RD | Input | Data Read. Read data from T6963C when RD = L. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CE | Input | Chip Enable for T6963C. CE must be L when CPU communicates with T6963C. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| PIN NAME | I/O | FUNCTIONS | | | | | | | | | | |
|--------------|---------|--|--------------|---------|----|---------|----|--------------|---|---|---------|----|
| C/D | Input | WR = L C/D = H : Command Write C/D = L : Data Write RD = L C/D = H : Status Read C/D = L : Data Read | | | | | | | | | | |
| HALT | Input | H Normal, L Stops the oscillation of the clock | | | | | | | | | | |
| RESET | Input | H Normal (T6963C has internal pull-up resistor) L Initialize T6963C. Text and graphic have addresses and text and graphic area settings are retained. | | | | | | | | | | |
| DSPON | Output | Control pin for external DC/DC. DSPON is L when HALT is L or RESET is L. (When DSPON goes H, the column drivers are cleared.) | | | | | | | | | | |
| DUAL | Input | H Single-Scan L Dual-Scan <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>DUAL</td> <td>H</td> <td>H</td> <td>L</td> <td>L</td> </tr> <tr> <td>SDSEL</td> <td>H</td> <td>L</td> <td>H</td> <td>L</td> </tr> </table> | DUAL | H | H | L | L | SDSEL | H | L | H | L |
| DUAL | H | H | L | L | | | | | | | | |
| SDSEL | H | L | H | L | | | | | | | | |
| SDSEL | Input | H Sending data by odd/even separation L Sending data by simple serial method <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>Upper screen</td> <td>HOD, ED</td> <td>ED</td> <td>HOD, ED</td> <td>ED</td> </tr> <tr> <td>Lower screen</td> <td>—</td> <td>—</td> <td>LOD, ED</td> <td>ED</td> </tr> </table> | Upper screen | HOD, ED | ED | HOD, ED | ED | Lower screen | — | — | LOD, ED | ED |
| Upper screen | HOD, ED | ED | HOD, ED | ED | | | | | | | | |
| Lower screen | — | — | LOD, ED | ED | | | | | | | | |
| ce0 (LOD) | Output | ce0 at DUAL = H Chip enable pin for display memory in the address range 0000H to 07FFH LOD at DUAL = L Serial data output for odd columns in lower area of LCD | | | | | | | | | | |
| ce1 (LSCP) | Output | ce1 at DUAL = H Chip enable pin for display memory in the address range 0800H to 0FFFH LSCP at DUAL = L Shift clock pulse output for column drivers in lower area of LCD | | | | | | | | | | |
| ce | Output | Chip enable pin for display memory of any address | | | | | | | | | | |
| d0 to d7 | I/O | Data I/O pins for display memory | | | | | | | | | | |
| ad0 to ad15 | Output | Address outputs for display memory (ad15 = L : for upper area of LCD, ad15 = H : for lower area of LCD) | | | | | | | | | | |
| R/W | Output | Read/Write signal for display memory | | | | | | | | | | |
| ED | Output | SDSEL = H : Data output for even columns in both upper and lower areas of LCD SDSEL = L : Data output for columns in both upper and lower areas of LCD | | | | | | | | | | |
| HOD | Output | Data output for odd columns in upper area of LCD | | | | | | | | | | |
| CDATA | Output | Synchronous signal for row driver | | | | | | | | | | |
| HSCP | Output | Shift clock pulse for column driver of upper area of LCD | | | | | | | | | | |
| LP | Output | Latch pulse for column driver. Shift clock pulse for row driver | | | | | | | | | | |
| FR | Output | Frame signal | | | | | | | | | | |
| XI | Input | Crystal oscillator input | | | | | | | | | | |
| XO | Output | Crystal oscillator output | | | | | | | | | | |
| CH1, CH2 | Output | Check signal | | | | | | | | | | |
| T1, T2 | Input | Test input. Usually open | | | | | | | | | | |
| VDD | — | Power supply (5.0V) | | | | | | | | | | |
| VSS | — | Power supply (0V) | | | | | | | | | | |

COMMAND DEFINITIONS

| COMMAND | CODE | D1 | D2 | FUNCTION |
|------------------------|----------|-------------|--------------|--------------------------------|
| REGISTERS SETTING | 00100001 | X address | Y address | Set Cursor Pointer |
| | 00100010 | Data | 00H | Set Offset Register |
| | 00100100 | Low address | High address | Set Address Pointer |
| SET CONTROL WORD | 01000000 | Low address | High address | Set Text Home Address |
| | 01000001 | Columns | 00H | Set Text Area |
| | 01000010 | Low address | High address | Set Graphic Home Address |
| | 01000011 | Columns | 00H | Set Graphic Area |
| MODE SET | 1000X000 | — | — | OR mode |
| | 1000X001 | — | — | EXOR mode |
| | 1000X011 | — | — | AND mode |
| | 1000X100 | — | — | Text Attribute mode |
| | 10000XXX | — | — | Internal CG ROM mode |
| | 10001XXX | — | — | External CG RAM mode |
| DISPLAY MODE | 10010000 | — | — | Display off |
| | 1001XX10 | — | — | Cursor on, blink off |
| | 1001XX11 | — | — | Cursor on, blink on |
| | 100101XX | — | — | Text on, graphic off |
| | 100110XX | — | — | Text off, graphic on |
| | 100111XX | — | — | Text on, graphic on |
| CURSOR PATTERN SELECT | 10100000 | — | — | 1-line cursor |
| | 10100001 | — | — | 2-line cursor |
| | 10100010 | — | — | 3-line cursor |
| | 10100011 | — | — | 4-line cursor |
| | 10100100 | — | — | 5-line cursor |
| | 10100101 | — | — | 6-line cursor |
| | 10100110 | — | — | 7-line cursor |
| | 10100111 | — | — | 8-line cursor |
| DATA AUTO READ / WRITE | 10110000 | — | — | Set Data Auto Write |
| | 10110001 | — | — | Set Data Auto Read |
| | 10110010 | — | — | Auto Reset |
| DATA READ / WRITE | 11000000 | Data | — | Data Write and Increment ADP |
| | 11000001 | — | — | Data Read and Increment ADP |
| | 11000010 | Data | — | Data Write and Decrement ADP |
| | 11000011 | — | — | Data Read and Decrement ADP |
| | 11000100 | Data | — | Data Write and Nonvariable ADP |
| | 11000101 | — | — | Data Read and Nonvariable ADP |
| SCREEN PEEK | 11100000 | — | — | Screen Peek |
| SCREEN COPY | 11101000 | — | — | Screen Copy |

X : invalid

CHARACTER CODE MAP
ROM code 0101

| MSB \ LSB | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|-----------|---|---|---|---|----|---|---|---|---|---|---|---|---|---|---|---|
| 0 | | ! | " | # | \$ | % | & | ' | (|) | * | + | , | - | . | / |
| 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | < | = | > | ? |
| 2 | @ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| 3 | P | Q | R | S | T | U | V | W | X | Y | Z | [| \ |] | ^ | _ |
| 4 | ^ | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
| 5 | p | q | r | s | t | u | v | w | x | y | z | { | | } | ~ | |
| 6 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |

ABSOLUTE MAXIMUM RATINGS (Ta = 25°C)

| ITEM | SYMBOL | RATING | UNIT |
|-----------------------|------------------------|-------------------------------|------|
| Supply Voltage | V _{DD} (Note) | -0.3 to 7.0 | V |
| Input Voltage | V _{IN} (Note) | -0.3 to V _{DD} + 0.3 | V |
| Operating Temperature | T _{opr} | -20 to 70 | °C |
| Storage Temperature | T _{stg} | -55 to 125 | °C |

(Note) Referenced to V_{SS} = 0V.

ELECTRICAL CHARACTERISTICS

DC CHARACTERISTICS

TEST CONDITIONS (Unless otherwise noted, V_{SS} = 0V, V_{DD} = 5.0V ± 10%, Ta = -20 to 75°C)

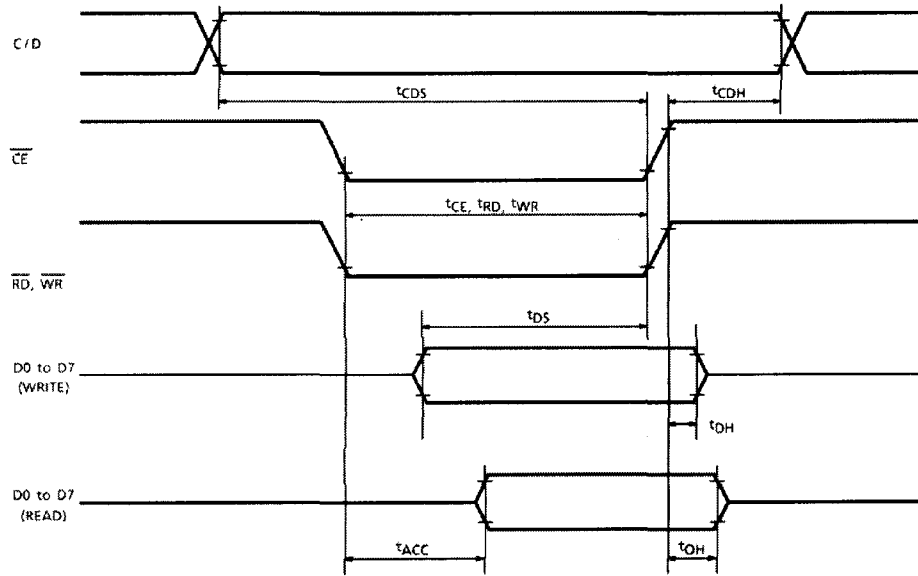
| ITEM | SYMBOL | TEST CIR-CUIT | TEST CONDITIONS | MIN | TYP. | MAX | UNIT | PIN NAME |
|---------------------------------|---------------------|-----------------|--|-----------------------|------|-----------------|------|-----------------|
| Operating Voltage | V _{DD} | — | — | 4.5 | 5.0 | 5.5 | V | V _{DD} |
| Input | H Level | V _{IH} | — | V _{DD} - 2.2 | — | V _{DD} | V | Input pins |
| | L Level | V _{IL} | — | 0 | — | 0.8 | V | Input pins |
| Output Voltage | H Level | V _{OH} | — | V _{DD} - 0.3 | — | V _{DD} | V | Output pins |
| | L Level | V _{OL} | — | 0 | — | 0.3 | V | Output pins |
| Output Resistance | H Level | R _{OH} | V _{OUT} = V _{DD} - 0.5V | — | — | 400 | Ω | Output pins |
| | L Level | R _{OL} | V _{OUT} = 0.5V | — | — | 400 | Ω | Output pins |
| Input Pull-up Resistance | R _{PU} | — | — | 50 | 100 | 200 | kΩ | (Note 1) |
| Operating Frequency | f _{OSC} | — | — | 0.4 | — | 5.5 | MHz | |
| Current Consumption (Operating) | I _{DD} (1) | — | V _{DD} = 5.0V (Note 2) f _{OSC} = 3.0MHz | — | 3.3 | 6 | mA | V _{DD} |
| Current Consumption (Halt) | I _{DD} (2) | — | V _{DD} = 5.0V | — | — | 3 | μA | V _{DD} |

(Note 1) Applied $\overline{T1}$, $\overline{T2}$, \overline{RESET}

(Note 2) MDS = L, MD0 = L, MD1 = L, MD2 = H, MD3 = H, FS0 = L, FS1 = L, \overline{SDSEL} = L, \overline{DUAL} = H,
D7 to D0 = LHLHLHLH

• Switching Characteristics (2)

Bus Timing



Appendix –2

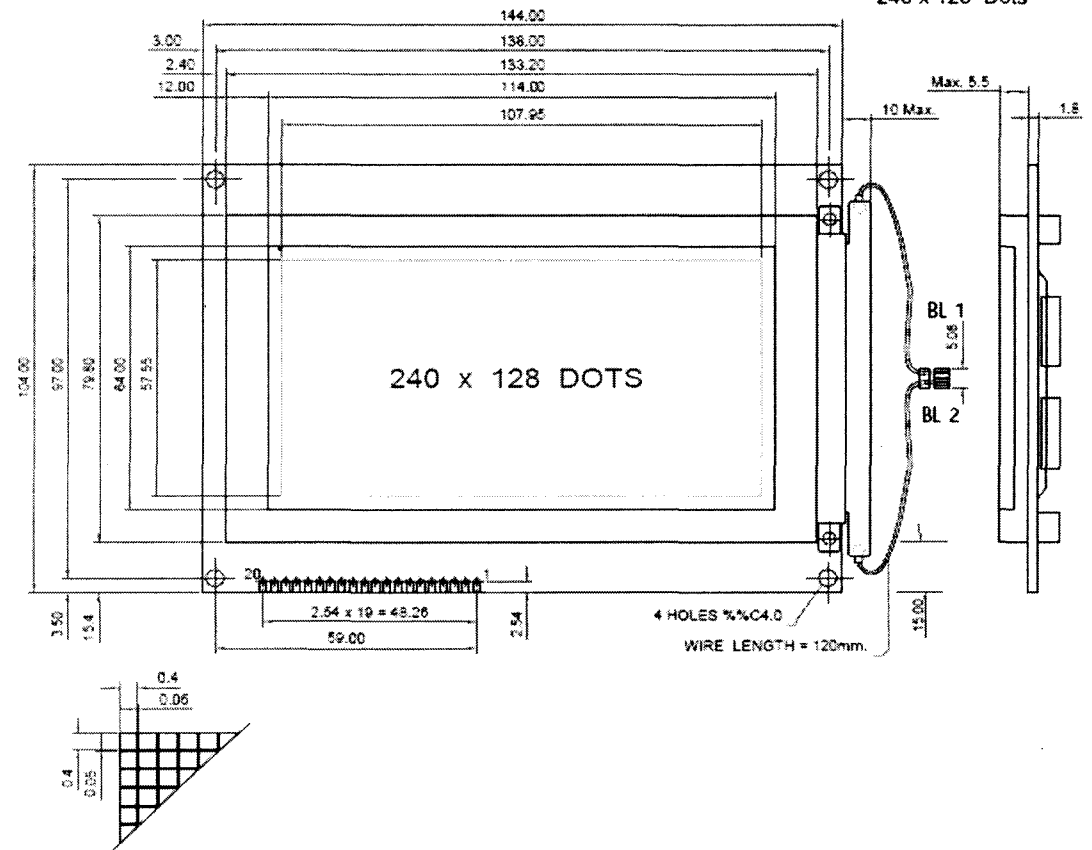
Oriole Electronics-OGMT24011 Datasheet

Graphic LCD Module

EXTERNAL DIMENSIONS :

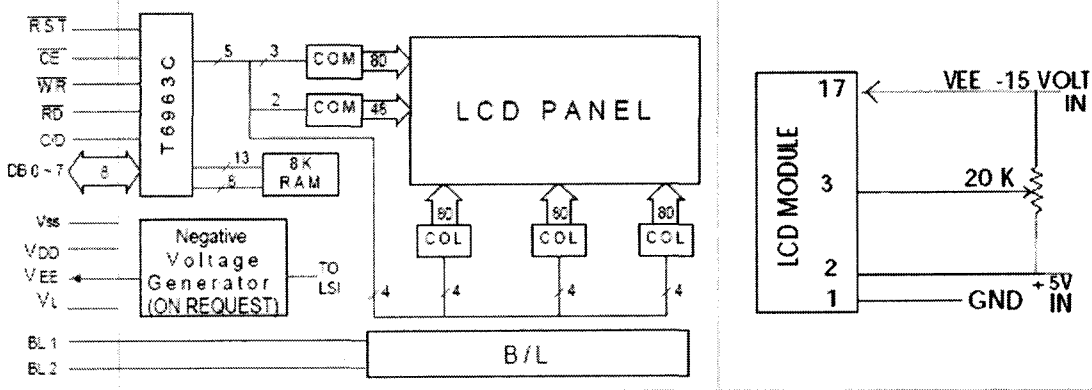
OGMT 24011

240 x 128 Dots



*BLOCK DIAGRAM

| PIN NO. | 20 | 19 | 18 | 17 | 16 | 15 | 14 ~ 7 | 6 | 5 | 4 | 3 | 2 | 1 | BL 1 | BL 2 |
|---------|-----|------|------|-----|-----|----|-------------|----|----|-----|----|-----|-----|--------|--------|
| SYMBOL | N/A | FS 1 | MD 2 | VEE | RST | CE | DB 7 ~ DB 0 | WR | RD | C/D | VL | VDD | VSS | IN/OUT | IN/OUT |



Appendix – 3

PIC 16F877A Instruction Set

PIC16F87X

TABLE 13-2: PIC16F87X INSTRUCTION SET

| Mnemonic, Operands | Description | Cycles | 14-Bit Opcode | | | Status Affected | Notes | |
|---|--------------------------------------|--------|---------------|------|-------|--------------------|--------|-------|
| | | | MSb | LSb | | | | |
| BYTE-ORIENTED FILE REGISTER OPERATIONS | | | | | | | | |
| ADDWF | f, d Add W and * | 1 | 00 | 0111 | ffff | ffff | C,DC,Z | 1,2 |
| ANDWF | f, d AND W with f | 1 | 00 | 0101 | ffff | ffff | Z | 1,2 |
| CLRF | * | 1 | 00 | 0001 | 1fff | ffff | Z | 2 |
| CLRWF | - | 1 | 00 | 0001 | 0xxx | xxxx | Z | |
| COMF | f, d Complement f | 1 | 00 | 1001 | ffff | ffff | Z | 1,2 |
| DECF | f, d Decrement f | 1 | 00 | 0011 | ffff | ffff | Z | 1,2 |
| DECFSZ | f, d Decrement f, Skip if 0 | 1(2) | 00 | 1011 | ffff | ffff | | 1,2,3 |
| INCF | f, d Increment f | 1 | 00 | 1010 | ffff | ffff | Z | 1,2 |
| INCFSZ | f, d Increment f, Skip if 0 | 1(2) | 00 | 1111 | ffff | ffff | | 1,2,3 |
| IORWF | f, d Inclusive OR W with f | 1 | 00 | 0100 | ffff | ffff | Z | 1,2 |
| MOVF | f, d Move * | 1 | 00 | 1000 | ffff | ffff | Z | 1,2 |
| MOVWF | * | 1 | 00 | 0000 | 1fff | ffff | | |
| NOP | - | 1 | 00 | 0000 | 0xxx0 | 0000 | | |
| RLF | f, d Rotate Left f through Carry | 1 | 00 | 1101 | ffff | ffff | C | 1,2 |
| RRF | f, d Rotate Right f through Carry | 1 | 00 | 1100 | ffff | ffff | C | 1,2 |
| SUBWF | f, d Subtract W from f | 1 | 00 | 0010 | ffff | ffff | C,DC,Z | 1,2 |
| SWAPF | f, d Swap nibbles in f | 1 | 00 | 1110 | ffff | ffff | | 1,2 |
| XORWF | f, d Exclusive OR W with f | 1 | 00 | 0110 | ffff | ffff | Z | 1,2 |
| BIT-ORIENTED FILE REGISTER OPERATIONS | | | | | | | | |
| BCF | f, b Bit Clear f | 1 | 01 | 00bb | bfff | ffff | | 1,2 |
| BSF | f, b Bit Set f | 1 | 01 | 01bb | bfff | ffff | | 1,2 |
| BTFSZ | f, b Bit Test f, Skip if Clear | 1(2) | 01 | 10bb | bfff | ffff | | 3 |
| BTFSZ | f, b Bit Test f, Skip if Set | 1(2) | 01 | 11bb | bfff | ffff | | 3 |
| LITERAL AND CONTROL OPERATIONS | | | | | | | | |
| ADDLW | k Add literal and W | 1 | 11 | 111x | kkkk | kkkk | C,DC,Z | |
| ANDLW | k AND literal with W | 1 | 11 | 1001 | kkkk | kkkk | Z | |
| CALL | k Call subroutine | 2 | 10 | 0kkk | kkkk | kkkk | | |
| CLRWDT | - | 1 | 00 | 0000 | 0110 | 0100 | TO,PD | |
| GOTO | k Go to address | 2 | 10 | 1kkk | kkkk | kkkk | | |
| IORLW | k Inclusive OR literal with W | 1 | 11 | 1000 | kkkk | kkkk | Z | |
| MOVLW | k Move literal to W | 1 | 11 | 00xx | kkkk | kkkk | | |
| RETFIE | - | 2 | 00 | 0000 | 0000 | 1001 | | |
| RETLW | k Return with literal in W | 2 | 11 | 01xx | kkkk | kkkk | | |
| RETURN | - | 2 | 00 | 0000 | 0000 | 1000 | | |
| SLEEP | - | 1 | 00 | 0000 | 0110 | 0011 | TO,PD | |
| SUBLW | k Subtract W from literal | 1 | 11 | 110x | kkkk | kkkk | C,DC,Z | |
| XORLW | k Exclusive OR literal with W | 1 | 11 | 1010 | kkkk | kkkk | Z | |

Appendix – 4
Assembly language program

Single Channel
D:\~~Project~~ & Project\Revised Prog
Single-ch.asm

/** program for the Mini-Oscilloscope**//

2) 2ch

```
list p=PIC16F877A,r=dec
include P16F877A.inc
ERRORLEVEL -302
```

```
INDF      EQU 00H
TMR0      EQU 01H
PCL EQU 02H
STATUS    EQU 03H
FSR EQU 04H
PORTA     EQU 05H
PORTB     EQU 06H
PORTC     EQU 07H
PORTD     EQU 08H
PORTE     EQU 09H
PCLATH    EQU 0AH
INTCON    EQU 0BH
ADRESH    EQU 1EH ;page 0
ADCON0    EQU 1FH ;page 0
```

```
OPTION_REG EQU 81H
TRISA      EQU 85H
TRISB      EQU 86H
TRISC      EQU 87H
TRISD      EQU 88H
TRISE      EQU 89H
ADRESL     EQU 9EH ;page 1
ADCON1     EQU 9FH ;page 1
```

;** VARIABLES ONLY ACCESSABLE THROUGH BLOCK 0, PAGE 0 **

```
ADC0      EQU 20H ; ADC storage set, extends to $5F (64 samples)
ADC01     EQU 6BH
LOOPB     EQU 60H ; general loop used in SETUP
LOOPC     EQU 61H ; general loop used in SETUP
TEMPA     EQU 63H
ADC11     EQU 64H
ADCTEMP1  EQU 66H
```



```

R1      EQU 6EH
R2      EQU 6FH
R3      EQU 65H

```

```

;***** VARIABLES ACCESSED THROUGH ALL PAGES & BLOCKS *****

```

```

LOOPA   EQU 70H ; waveform byte sample counter
ADCTEMP EQU 71H ; temp store used in LCD output routines
LOOPD   EQU 72H ; general loop used in SETUP
LOOPD1  EQU 73H ; general loop used in SETUP
LOOPE   EQU 74H ; sample/line counter
ADCADR  EQU 75H ; ADC block address
ADRLSB  EQU 77H ; LCD low address
ADRMSB  EQU 78H ; LCD high address
PEAKLO1 EQU 79H ; waveform sample min for graph
XYZ     EQU 7AH
COLUMN  EQU 7BH ; column length holder
PEAKLO  EQU 7CH ; waveform sample min for graph
PEAKHI  EQU 7DH ; waveform sample max for graph
LOBIT   EQU 7EH ; rotating bit for 8-bit column

```

```

;***** variables in BLOCK 0, PAGE 1 *****

```

```

ADC1    EQU 0xA0 ; ADC storage set for 2nd SET of 64 samples
          ; extends from AO to DF

```

```

;***** variables in BLOCK 2, PAGE 2 *****

```

```

MEM0    EQU 20H ; start of memory for 64 screen line samples
          ; extends from 20 to 5F
ADC2    EQU 0xA1 ; ADC storage set for 3rd SET of 64 samples
          ; extends from A1 to F0

```

```

;***** FIXED VALUES FOR COMMANDS *****

```

```

TXHOME   EQU 40H ; text home address command
TXAREA   EQU 41H ; text area (columns) address command
GRHOME   EQU 42H ; graphics home address command
GRAREA   EQU 43H ; graphic area (columns) address command
AWRON    EQU 0xB0 ; autowrite on command
AWROFF   EQU 0xB2 ; autowrite off command

```

```

OFFSET      EQU 22H ; offset command
ADPSET      EQU 24H ; address set command
PEEK        EQU 0xE0 ; screen peek command
CSRPOS      EQU 21H ; set cursor position command

```

```

;***** BIT VALUES *****

```

```

W          EQU 00H ; working reg
F          EQU 01H ; file reg
Z          EQU 02H ; ZERO status
C          EQU 00H ; CARRY status
DC         EQU 01H ; DIGIT CARRY status
RP0        EQU 05H ; STATUS BLOCK0 (BANKS0/1) reg
RP1        EQU 06H ; STATUS BLOCK1 (BANKS2/3) reg
GIE        EQU 07H ; INTCON reg
GO         EQU 02H ; ADCON0 reg
ADIF       EQU 06H

```

```

ORG 00H ; Interrupt vector address
GOTO START ; Jump to interrupt routine on interrupt
ORG 0x0005 ; Start of program memory

```

```

DELAY      MOVLW 0X01 ;
            MOVWF R2
            MOVLW 0X01 ; MAX freq hz
            MOVWF R1
            btfsc PORTC,0
            goto BB
            MOVLW 0X01
            MOVWF R2
            MOVLW 0X40 ;
            MOVWF R1
            btfsc PORTC,1
            goto BB
            MOVLW 0X02 ;
            MOVWF R2
            btfsc PORTC,2
            goto AA

```

```

        MOVLW 0X04      ;
        MOVWF R2
        btfsc PORTC,3
        goto  AA
        MOVLW 0X0D      ; approx 0.606hz
        MOVWF R2
AA      MOVLW 0X80
        MOVWF R1
BB      NOP
I5      NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        DECFSZ R1,1
        GOTO I5
        DECFSZ R2 ,1
        GOTO AA
        RETURN

START  bsf  STATUS,5
        bcf  STATUS,6
        movlw 0xFF
        movwf TRISA ; PORTA as input
        clrf TRISC ; PORTC as input
        clrf TRISB ; PORTB as output GRAPHIC LCD D0-D7
        clrf TRISD ; PORTD as output GRAPHIC LCD control
        clrf TRISE ; PORTE as output
        movlw 0x06
        movwf OPTION_REG
        movlw 0x0B ; one analog pin RA0, +ref Vdd, -ref Vss ; left justified
        movwf ADCON1
        bcf STATUS,RP0
        clrf PORTC
        movlw 0x0f ; RST low, CE WR RD CD high
        movwf PORTD
        clrf PORTB
        clrf PORTE
        movlw 0x1f ; RST CE WR RD CD high
        movwf PORTD

```



```

movlw  MEM0      ; set address for 1st affected LCD store memory
addwf  PEAKLO,W
movwf  FSR      ;
movf   PEAKLO,W ; set loop to val of PEAKLO
movwf  LOOPE
movf   LOBIT,W
iorwf  INDF,F

```

```

OSC_SUB3: movf   LOOPE,W ; is loop = peakhi?
xorwf  PEAKHI,W
btfsc  STATUS,Z
goto   OSC_SUB4
movf   LOBIT,W ; no
iorwf  INDF,F
incf   LOOPE,F
btfsc  LOOPE,6 ;limit to 64 max
goto   OSC_SUB4
incf   FSR,F
goto   OSC_SUB3

```

```

OSC_SUB4: BCF   STATUS,7 ;BLKO
return

```

```

;***** send 8-bit column to screen

```

```

SHOWMEM: movlw MEM0
addlw  0X3F
movwf  FSR
movlw  0X40
movwf  LOOPE ; set loop val (number of lines)
movlw  0X07 ; set graphic base address ($02xx)
movwf  ADDRMSB
movf   LOOPD,W ; set column
addlw  0x66
movwf  ADRLSB

```

```

OSC_SUB5: call   SCREENADR ; set screen write address
movlw  AWRON ; AUTO WRITE ON
call   SENDCMD ; send command
call   CHECK3 ; read status for DA0/DA1 = 3
BSF   STATUS,7 ;BLK1
movf  INDF,W ; get value, clear store
clrf  INDF

```

```

BLK111: movlw    0X80 ; indicates which bit of column is affected
        movwf   LOBIT
        call    OSC_SUB11 ; config & send third set of samples to LCD
        incf   LOOPD,F    ; inc to next column
        decfsz XYZ          ;has column 24 reached?
        goto   BLK111     ; no
        goto   MAIN       ; start again at screen left

```

```

;***** START OF OSCILLOSCOPE SUB-ROUTINES *****

```

```

OSC_SUB1: movf   LOOPA,W
        addwf   ADCADR,W
        movwf   FSR
        rrf    INDF,W    ; divide by 2
        movwf   ADCTEMP
        rrf    ADCTEMP,W ; divide by 2 again and clear bits 6&7
        andlw   0X3F
        movwf   ADCTEMP ; store for later use
        movwf   PEAKHI ; and move into PEAKHI
        subwf   PEAKLO,W ; is PEAKHI > PEAKLO ? (subtract hi from lo)
        btfss  STATUS,C ; is there a borrow?(if carry flag is set then PEAKLO less then
PEAKHI)
        goto   OSC_SUB2 ; c=0
        movf   PEAKLO,W ; c=1, so swap lo & hi
        movwf   PEAKHI
        movf   ADCTEMP,W
        movwf   PEAKLO

```

```

OSC_SUB2: call   SETMEM ; config LCD memory store..
        movf   ADCTEMP,W ; store current val into PEAKLO
        movwf   PEAKLO
        incf   LOOPA,F    ; inc wavform counter
        bcf   LOOPA,6    ; limit to 64 max
        bcf   STATUS,C
        rrf    LOBIT,F
        btfss STATUS,C ; is CARRY set? (stating all 8 bits of col done)
        goto   OSC_SUB1 ; no
        call   SHOWMEM ; display results on screen
        return

```

```

;***** config LCD memory store

```

```

SETMEM: BSF     STATUS,7 ;BLK1

```

```

BCF     STATUS,7    ;BLKO
decf   FSR,F
call   OUTDATA    ;
movlw  AWROFF     ; AUTO WRITE OFF
call   SENDCMD    ; send command
decf   LOOPE,F
btfsc  STATUS,Z
return
movf   COLUMN,W   ; set for next line
addwf  ADRLSB,F
btfsc  STATUS,0   ; add CARRY (if any) to MSB
incf   ADRMSB,F
goto   OSC_SUB5

```

```

OSC_SUB11: BSF     STATUS,7
movf   LOOPA,W    ; copy ADC loop val into PEAKHI
addwf  ADCADR,W
movwf  FSR        ;
rrf    INDF,W     ; divide by 2
movwf  ADCTEMP
rrf    ADCTEMP,W  ; divide by 2 again and clear bits 6&7
andlw  0X3F
movwf  ADCTEMP    ; store for later use
movwf  PEAKHI     ; and move into PEAKHI
subwf  PEAKLO,W   ; is PEAKHI > PEAKLO ? (subtract hi from lo)
btfss  STATUS,C   ; is there a borrow?(if carry flag is set then PEAKLO less then
PEAKHI)
goto   OSC_SUB21  ; c=0

movf   PEAKLO,W   ;c=1, so swap lo & hi
movwf  PEAKHI
movf   ADCTEMP,W
movwf  PEAKLO

```

```

OSC_SUB21: call    SETMEM1    ; config LCD memory store..
BSF    STATUS,7
movf   ADCTEMP,W    ; store current val into PEAKLO
movwf  PEAKLO
incf   LOOPA,F     ; inc wavform counter
bcf    LOOPA,6     ; limit to 64 max

bcf    STATUS,C

```

```

    rrf      LOBIT,F
    btfss   STATUS,C ; is CARRY set? (stating all 8 bits of col done)
    goto    OSC_SUB11 ; no
    call    SHOWMEM1 ; display results on screen
    return

```

,***** config LCD memory store

```

SETMEM1: BSF      STATUS,7 ;BLK1
    movlw  MEM0 ; set address for 1st affected LCD store memory
    addwf  PEAKLO,W
    movwf  FSR
    movf   PEAKLO,W ; set loop to val of PEAKLO
    movwf  LOOPE
    movf   LOBIT,W
    iorwf  INDF,F

```

```

OSC_SUB31: movf   LOOPE,W ; is loop = peakhi?
    xorwf  PEAKHI,W
    btfsc  STATUS,Z
    goto   OSC_SUB41
    movf   LOBIT,W ; no
    iorwf  INDF,F
    incf   LOOPE,F
    btfsc  LOOPE,6 ;limit to 64 max
    goto   OSC_SUB41
    incf   FSR,F
    goto   OSC_SUB31

```

```

OSC_SUB41: BCF      STATUS,7 ;BLKO
    return

```

,***** send 8-bit column to screen

```

SHOWMEM1: movlw    MEM0
    addlw  0X3F
    movwf  FSR
    movlw  0X40
    movwf  LOOPE ; set loop val (number of lines)
    movlw  0X06 ; set graphic base address
    movwf  ADRMSB
    movf   LOOPD,W ; set column
    addlw  0x76
    movwf  ADRLSB

```



```
BCF STATUS,7
```

```
OSC_SUB51: call SCREENADR ; set screen write address
movlw AWRON ; AUTO WRITE ON
call SENDCMD ; send command
call CHECK3 ; read status for DA0/DA1 = 3
BSF STATUS,7 ;BLK1
movf INDF,W ; get value, clear store
clrf INDF
BCF STATUS,7 ;BLKO
decf FSR,F
call OUTDATA ;
movlw AWROFF ; AUTO WRITE OFF
call SENDCMD ; send command
decf LOOPE,F
btfsc STATUS,Z
return
movf COLUMN,W ; set for next line
addwf ADRLSB,F
btfsc STATUS,0 ; add CARRY (if any) to MSB
incf ADRMSB,F
goto OSC_SUB51
```

This part of program puts data from ADC into Bank (all 4).

```
SAMPLE1: movlw ADC0 ; set store address at ADC0 BLOCK0
movwf FSR ; initialize pointer, to RAM
movlw 0X40
movwf LOOPE ; no of samples value (64 from 20-5F)
movlw 0x41 ; set AD on, Fosc/2, RA1 analog input
movwf ADCON0
nop
nop
```

```
ADVAL1: btfsc ADCON0,GO ; start data conversion
goto ADVAL1
movf ADRESH,W
bsf ADCON0,GO
movwf INDF ; put it out to mem
call DELAY
incf FSR,F ; inc store address
```

```

decfsz LOOPE,F
goto    ADVAL1

movlw   ADC1    ;set store address at ADC1 BLOCK0
movwf   FSR          ; initialize pointer,to RAM
movlw   0X40
movwf   LOOPE      ;no of samples value (64 from A0-DF BLOCK0)

```

```

ADVAL2:  btfsc  ADCON0,GO ; start data conversion
        goto   ADVAL2
        movf   ADRESH,W
        bsf    ADCON0,GO
        movwf  INDF    ;put it out to mem
        call   DELAY
        incf   FSR,F    ;inc store address
        decfsz LOOPE,F
        goto   ADVAL2

        bsf    STATUS,7
        movlw  ADC2    ;set store address at ADC2 BLOCK1
        movwf  FSR          ; initialize pointer,to RAM
        movlw  0X40
        movwf  LOOPE      ;no of samples value (64 from A1-F0 BLOCK1)
        BCF STATUS,7
ADVAL3:  btfsc  ADCON0,GO ; start data conversion
        goto   ADVAL3
        movf   ADRESH,W ;get & invert ADC val
        bsf    ADCON0,GO
        BSF STATUS,7
        movwf  INDF    ;put it out to mem
        incf   FSR,F    ;inc store address
        bcf    STATUS,7
        call   DELAY
        decfsz LOOPE,F
        goto   ADVAL3
        return

```

AUTOWRITE: ; ** AUTO WRITE ROUTINE **

```

movwf   TEMPA    ; temp store value brought in on W(ooH INTO TEMPA)

```

```

call    CHECK8    ; read status for DA3 = 8
movf    TEMP,W    ; WRITE DATA
call    OUTDATA   ;
return  ;

```

;/.....

CMDADR: ; ** SET ADDRESS FOR COMMAND SENDING **

```

call    CHECK3    ; read status for DA0/DA1 = 3
movf    ADRLSB,W  ; WRITE DATA D1
call    OUTDATA   ;
call    CHECK3    ; read status for DA0/DA1 = 3
movf    ADRMSB,W  ; WRITE DATA D2
call    OUTDATA   ;
return

```

;/.....

SCREENADR: ; ** SET ADDRESS FOR WRITE/READ TO/FROM SCREEN

```

call    CHECK3    ; read status for DA0/DA1 = 3
movf    ADRLSB,W  ; WRITE ADDRESS LSB
call    OUTDATA   ;
call    CHECK3    ; read status for DA0/DA1 = 3
movf    ADRMSB,W  ; WRITE ADDRESS MSB
call    OUTDATA   ;
movlw   ADPSET    ; SET ADDRESS POINTER
call    SENDCMD   ; send command
return

```

GRAPHHOME: ; ** SET GRAPHIC HOME ADDRESS **

```

movlw   0X00     ; GRAPHIC HOME ADDRESS $0200
movwf   ADRMSB
clrf    ADRLSB
call    CMDADR   ; send command address
movlw   GRHOME  ;
call    SENDCMD  ; send command
return

```

GRAPHAREA: ; ** SET GRAPHIC AREA **

```

clrf    ADRMSB  ;

```

```

movf    COLUMN,W    ; columns length
movwf   ADRLSB
call    CMDADR      ; send command address
movlw   GRAREA      ; graphic area command
call    SENDCMD     ; send command
return

```

;/.....

SETMODE: ; ** SET MODE **

```

movlw   0X80        ; (OR mode, Internal CG mode)
call    SENDCMD     ; send command
return

```

;/.....

SETDISPLAY: ; ** DISPLAY MODE **

```

movlw   0X98        ; text off, graphic on, cursor & blink off
call    SENDCMD     ; send command
return

```

CLRGRAPH: ; ** CLEAR GRAPH AREA **

```

movlw   00H
movwf   ADRMSB
clrf    ADRLSB      ;
call    SCREENADR   ; set screen write address
movlw   AWRON       ; AUTO WRITE ON
call    SENDCMD     ; send command
movlw   80H         ; number of lines
movwf   LOOPC

```

```

CLRG2: movf    COLUMN,W    ; column length
movwf   LOOPB

```

```

CLRG3: movlw   00H        ; write 0

```

```

call    AUTOWRITE   ; auto write and increment
decfsz LOOPB,F
goto    CLRG3
decfsz LOOPC,F
goto    CLRG2
movlw   AWROFF      ; AUTO RESET OFF
call    SENDCMD     ; send command
return

```

;/.....

CHECK3: BSF STATUS,05 ; STATUS CHECK for DA0/DA1 = 3

```

movlw 0XFF
movwf TRISB ; set PORT B as inputs
BCF STATUS,5 ; RST CE WR RD CD
movlw 0X15 ; 1 0 1 0 1
movwf PORTD ; set CE, RD low
nop
CK3: btfss PORTB,0 ; PORT B bit 0 set?
goto CK3 ; no
CK3A: btfss PORTB,1 ; PORT B bit 1 set?
goto CK3A ; no RST CE WR RD CD
movlw 0X1F ; 1 1 1 1 1
movwf PORTD ; set controls
nop
BSF STATUS,5
clrf TRISB ; set PORT B as outputs
BCF STATUS,5
return

```

;.....

CHECK8: BSF STATUS,5 ; STATUS CHECK for DA3 = 8

```

bsf TRISB,3 ; set PORT B bit 3 as input
BCF STATUS,5 ; RST CE WR RD CD
movlw 0X15 ; 1 0 1 0 1
movwf PORTD ; set CE, RD low
nop
CK8: btfss PORTB,3 ; is PORT B,3 high?
goto CK8 ; no RST CE WR RD CD
movlw 0X1F ; 1 1 1 1 1
movwf PORTD ; set controls
nop
BSF STATUS,5
bcf TRISB,3 ; set PORT B bit 3 as output
BCF STATUS,5
return

```

;.....

OUTDATA: ; ** SEND DATA ROUTINE **

```

movwf TEMPA ; temp store val brought in on W
; RST CE WR RD CD
movlw 0X1E ; 1 1 1 1 0
movwf PORTD ; set CD low

```

```

movf    TEMPA,W    ; get stored data
movwf   PORTB     ; send data
nop     ; RST CE WR RD CD
movlw   0X12      ; 1 0 0 1 0
movwf   PORTD     ; set CD, CE, WR low
nop     ; RST CE WR RD CD
movlw   0X1E      ; 1 1 1 1 0
movwf   PORTD     ; set CE, WR high
nop     ; RST CE WR RD CD
movlw   0X1F      ; 1 1 1 1 1
movwf   PORTD     ; set CD high
return

```

;/.....

```

SENDCMD: ; ** COMMAND WRITE ROUTINE **
movwf   TEMPA     ; temp store val brought in on W
call    CHECK3    ; read status for DA0/DA1 = 3
movf    TEMPA,W   ; WRITE COMMAND
movwf   PORTB     ; send stored data
nop     ; RST CE WR RD CD
movlw   0X13      ; 1 0 0 1 1
movwf   PORTD     ; set CE, WR low
nop     ; RST CE WR RD CD
movlw   0X1F      ; 1 1 1 1 1
movwf   PORTD     ; set all high
return

```

```

SETUP:   ; RST CE WR RD CD GENERAL SETUP
movlw   0X1F      ; 1 1 1 1 1
movwf   PORTD     ; set controls high (off)
call    GRAPHHOME ; SET GRAPHIC HOME ADDRESS
call    GRAPHAREA ; SET GRAPHIC AREA
call    SETMODE   ; SET MODE (INT/EXT/AND-OR-XOR etc)
call    SETDISPLAY ; DISPLAY MODE (text, graph on/off etc)
call    CLRGRAPH  ; clear graph screen
return

```

END

MPU Interface Pin Functions

| Pin Name | I/O | Pin Function |
|------------------|-----|--|
| V _{SS} | - | Ground (0V) |
| V _{DD} | - | Logic Supply (+5V) |
| $\overline{C/D}$ | I | Register Select: "1" = Command Register, "0" = Data Register |
| \overline{RD} | I | Read Select (Active Low) - MPU \Leftarrow T6963C |
| \overline{WR} | I | Write Select (Active Low) - MPU \Rightarrow T6963C |
| \overline{CE} | I | Chip Enable (Active Low) |
| DB0 | I/O | Bi-directional Data Bus Line 0 |
| DB1 | I/O | Bi-directional Data Bus Line 1 |
| DB2 | I/O | Bi-directional Data Bus Line 2 |
| DB3 | I/O | Bi-directional Data Bus Line 3 |
| DB4 | I/O | Bi-directional Data Bus Line 4 |
| DB5 | I/O | Bi-directional Data Bus Line 5 |
| DB6 | I/O | Bi-directional Data Bus Line 6 |
| DB7 | I/O | Bi-directional Data Bus Line 7 |
| \overline{RES} | I | Reset (Active Low) |
| FS1 | I | Font Select 6x8 font: FS1="H" 8x8 font FS1="L" |
| MD2 | I | Mode Select 40 columns MD2="H" 32 columns MD2="H" |

LCD Interface Pin Functions

For LCD modules with on-board controllers, these signals are not generally accessible; they are accessible on the SGI-01 serial interface card with the on-board T6963C option installed.

| Pin Name | I/O | Pin Function |
|-----------------|-----|------------------------------------|
| D1 | O | Serial Data Line 1 |
| FLM (CDATA) | O | First Line Marker (Start of Frame) |
| M (FR) | O | Control Signal for AC drive of LC |
| CL1 (LP) | O | Latch clock |
| CL2 (SCP) | O | Shift clock for serial data |
| D2 | O | Serial Data Line 2 |
| V _{DD} | - | Power supply for logic circuits |
| V _{SS} | - | Ground |
| V _{EE} | - | Power supply for LC drivers |