EMBEDDED SYSTEMS INTERNSHIP

EMERTXE INFORMATION TECHNOLOGIES

PROJECT TITLE: CAR BLACK BOX

PRESENTED BY

AYUSH GUPTA, SRMIST

PRE-INTERNSHIP TALKS

1) What are Embedded Systems?

2) Why we need them and what role do they play in our lives?

3) What is an MCU inside an Embedded System?

4) What are PIC Microcontrollers and how they work?

5) How to control them using programs written in C/Embedded C?

6) How to build amazing projects such as a Car Black Box using them?

KEY TAKEAWAYS FROM INTERNSHIP

- Overview of C Programming
- Keywords, Operators, Arrays, Functions and Pointers
- Strings and Pre-processor Directives
- Overview of Embedded Systems
- Types of Embedded Systems
- Microcontrollers and Microprocessors
- Car Black Box Project Overview
- MPLAB X IDE, XC8 Compiler and PICSIM Lab
- Interrupts [ISRs] and Timers
- PIC16F877A and its Peripherals
- Digital Keypad and CLCD
- Car Black Box Project Explanation and Demo

C PROGRAMMING OVERVIEW

The C programming language is a general-purpose, high-level language that was originally developed by Dennis M. Ritchie to develop the UNIX operating system at Bell

Labs. C was originally first implemented on the DEC PDP-11 computer in 1972. In 1978, Brian Kernighan and Dennis Ritchie produced the first publicly available description of C, now known as the K&R standard. The UNIX operating system, the C compiler, and essentially all UNIX applications programs have been written in C. The C has now become a widely used professional language for various reasons such as

- Easy to learn
- Structured language
- It produces efficient programs.

C KEYWORDS

The following list shows the reserved words in C. These reserved words may not be used as constant or variable or any other identifier names.

auto	else	Long	switch
break	enum	register	typedef
case	extern	return	union
char	float	short	unsigned
const	for	signed	void
continue	goto	sizeof	volatile
default	if	static	while
do	int	struct	_packed
double			

OPERATORS IN C

An operator is a symbol that tells the compiler to perform specific mathematical or logical manipulations. C language is rich in built-in operators and provides the following types of operators:

Arithmetic Operators (+ , - , / , * , %)

Relational Operators (== , != , > , < , >= , <=)

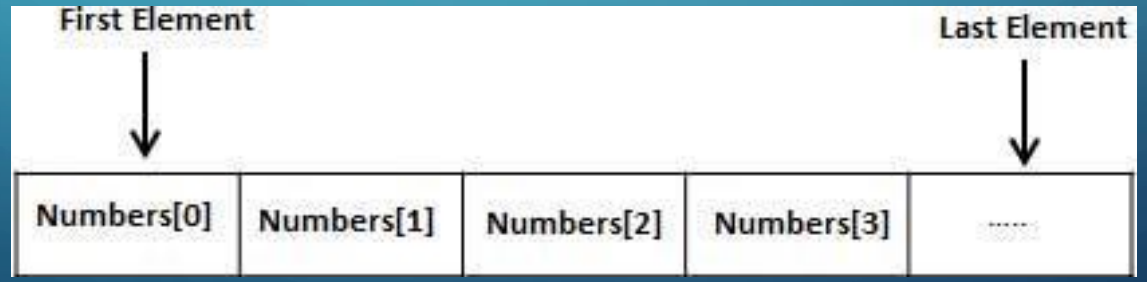
Logical Operators (&& , || , !)

Bitwise Operators (^ , & , | , ~ , >> , <<)

Assignment Operators (Short hand operators for all others such as += , >>= etc.)

ARRAYS IN C

C programming language provides a data structure called the array, which can store a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type. Instead of declaring individual variables, such as number0, number1, ..., and number99, you declare one array variable such as numbers and use numbers[0], numbers[1], and ..., numbers[99] to represent individual variables. A specific element in an array is accessed by an index. All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.



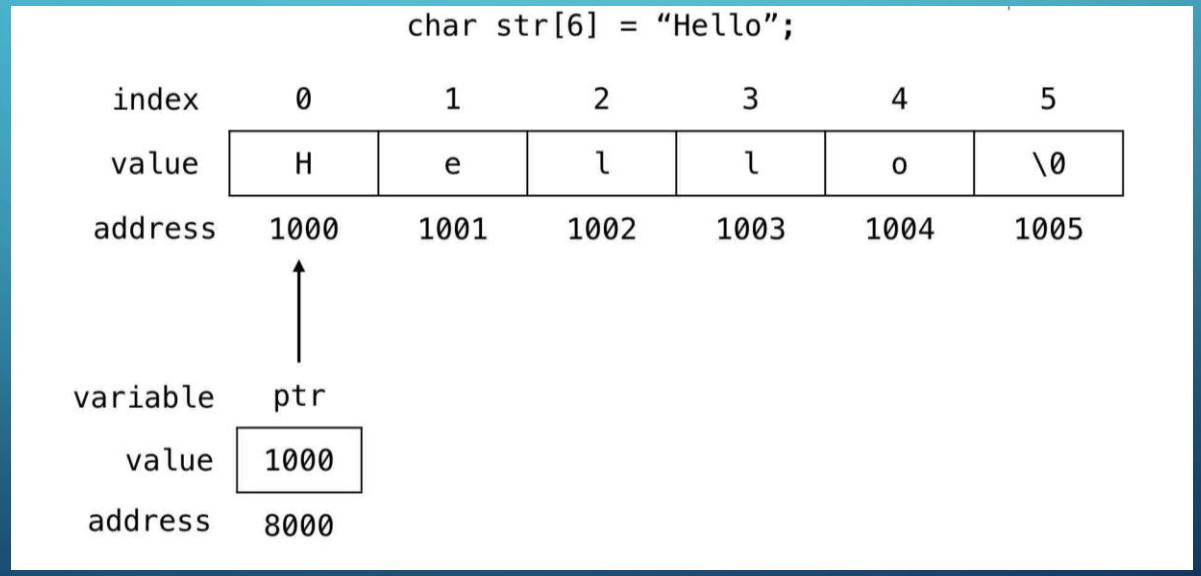
C FUNCTIONS

Function is a group of statements that together perform a task. Every C program has at least one function, which is main(), and all the most trivial programs can define additional functions. You can divide up your code into separate functions. How you divide up your code among different functions is up to you, but logically the division usually is so each function performs a specific task. A function declaration tells the compiler about a function's name, return type, and parameters. A function definition provides the actual body of the function

```
return_type function_name( parameter list )
{
body of the function
}
```

POINTERS IN C

Pointers in C are easy and fun to learn. Some C programming tasks are performed more easily with pointers, and other tasks, such as dynamic memory allocation, cannot be performed without using pointers. So it becomes necessary to learn pointers to become a perfect C programmer. Let's start learning them in simple and easy steps. As you know, every variable is a memory location and every memory location has its address defined which can be accessed using ampersand (&) operator, which denotes an address in memory.



STRINGS IN C

The string in C programming language is actually a one-dimensional array of characters which is terminated by a null character '\0'. Thus a null-terminated string contains the characters that comprise the string followed by a null.

The following declaration and initialization create a string consisting of the word "Hello". To hold the null character at the end of the array, the size of the character array containing the string is one more than the number of characters in the word "Hello".

char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};

If we follow the rule of array initialization then you can write the above statement as follows:

char greeting[] = "Hello";

Inside Memory →

PREPROCESSORS

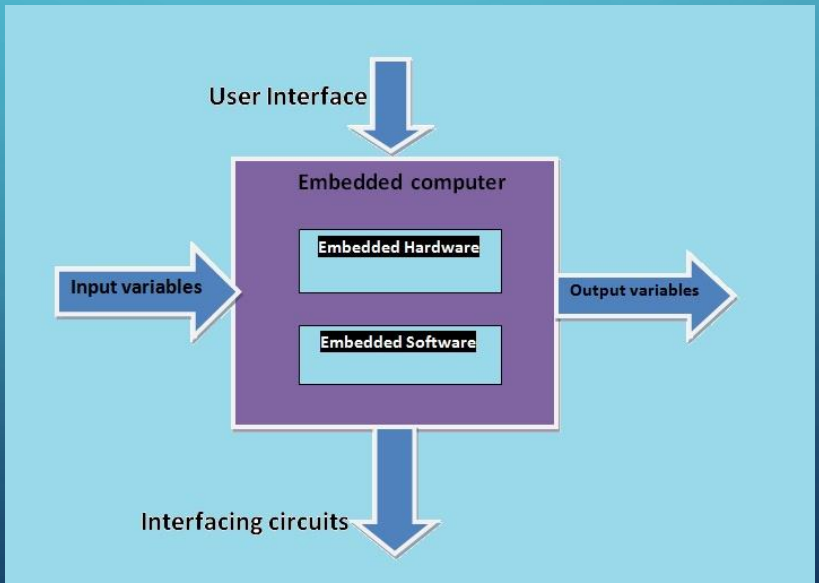
The C Preprocessor is not part of the compiler, but is a separate step in the compilation process. In simplistic terms, a C Preprocessor is just a text substitution tool and they instruct compiler to do required pre-processing before actual compilation. We'll refer to the C Preprocessor as the CPP. All preprocessor commands begin with a pound symbol (#).

Directive	Description
#define	Substitutes a preprocessor macro
#include	Inserts a particular header from another file
#undef	Undefines a preprocessor macro
#ifdef	Returns true if this macro is defined
#ifndef	Returns true if this macro is not defined
#if	Tests if a compile time condition is true
#else	The alternative for #if
#elif	#else an #if in one statement
#endif	Ends preprocessor conditional
#error	Prints error message on stderr
#pragma	Issues special commands to the compiler, using a standardized method

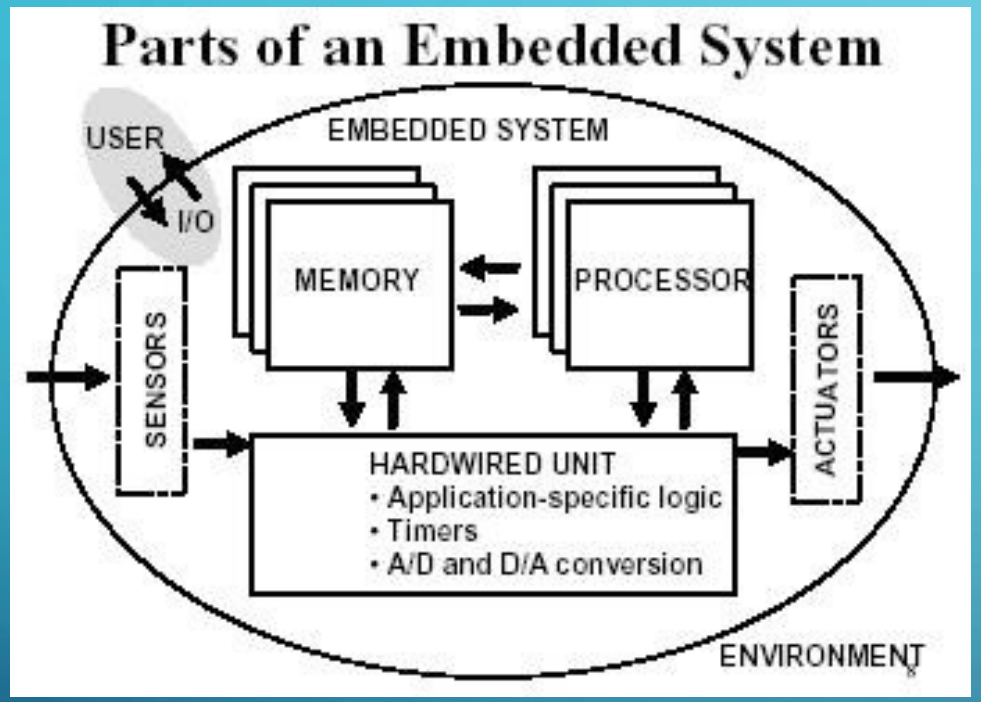
OVERVIEW OF EMBEDDED SYSTEMS

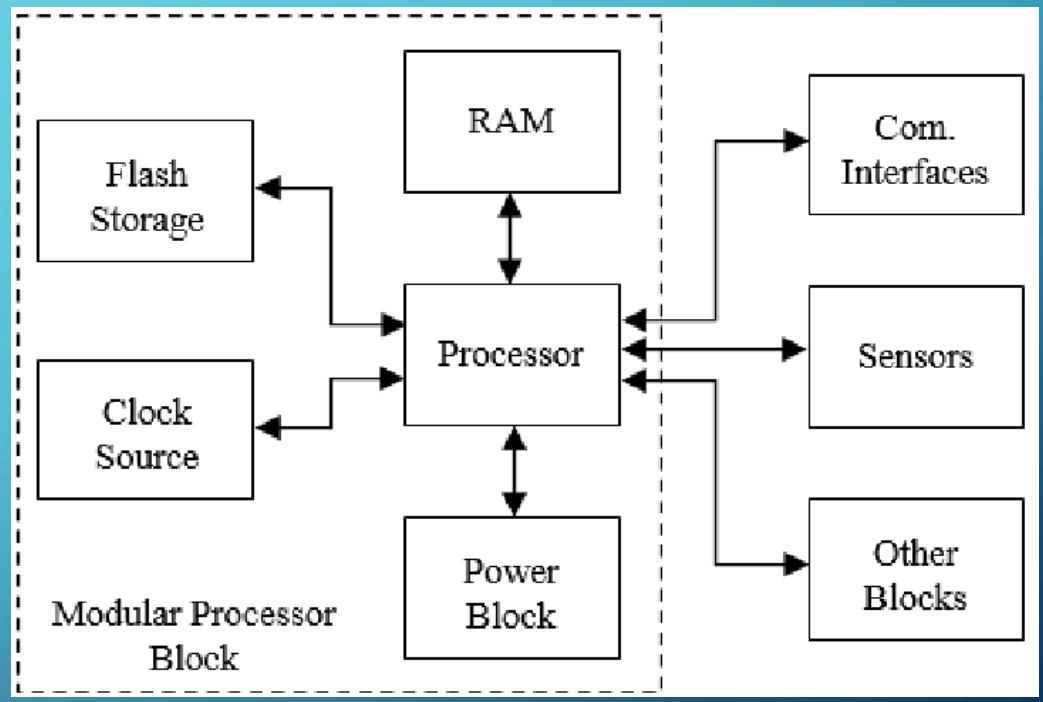
Embedded systems are a type of computer system that usually forms part of a larger system, device or piece of electronic equipment. They consist of a microcontroller that performs a specific function within a product and can be found in almost every modern piece of technology that we use today.

They are the systems which integrate computer hardware with software to perform any modern day task we can think of!



A TYPICAL EMBEDDED SYSTEM





TYPES OF EMBEDDED SYSTEMS

- Stand-alone/Independent Embedded System

An embedded system that is described as independent or stand-alone works by itself and does not require a host system, such as a computer, to carry out its function. Examples include temperature measurement devices.

- Real-Time Embedded System

The majority of embedded systems operate with real-time software, meaning that they give their required output function within a specified time interval.

- Network Embedded System

A network embedded system works in a device or machine that is connected to a network that provides output to other systems. They're frequently found in home security systems where more complex system connected by a network.

- Mobile Embedded System

Mobile embedded systems are any embedded systems used in small devices that are designed to be portable.

MICROCONTROLLERS AND MICROPROCESSORS

A microcontroller (μC) is a small computer on a single integrated circuit consisting of a relatively simple central processing unit (CPU) combined with peripheral devices such as memories, I/O devices, and timers. It acts as the heart of the embedded system. Since on-chip memory and I/O output component is available. Therefore the circuit is less complex.

Microprocessor is generally a central processing unit on a single integrated circuit chip containing millions and millions of very small components called transistors that work together and act as the heart of a computer system. Since memory and I/O output is to be connected externally. Therefore the circuit is more complex.

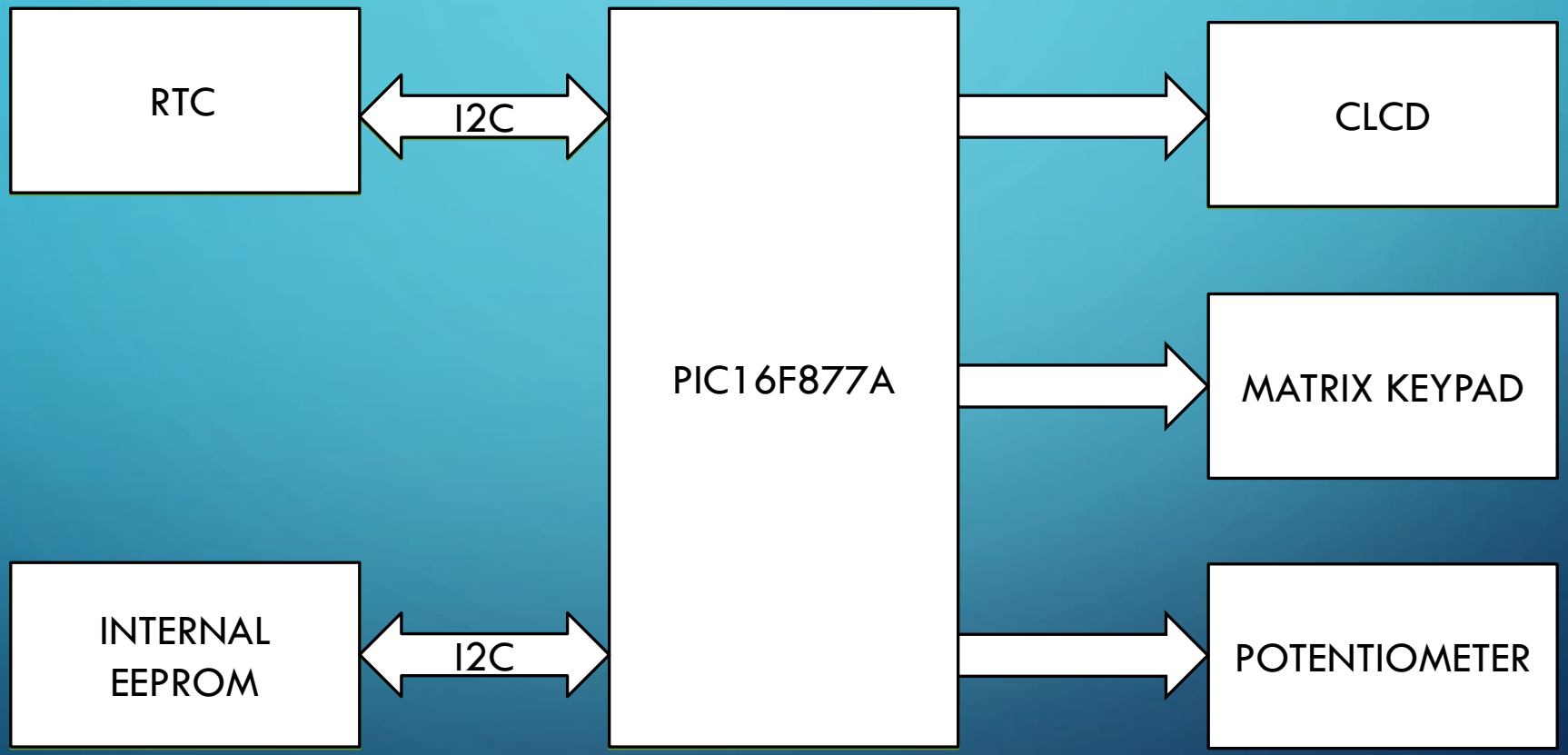
CAR BLACK BOX PROJECT

A Car Black Box is a device which is installed in an automobile vehicle to monitor certain events happening in and around the car such as how the car is being drove, the speed of the car, gear change, acceleration but MOST IMPORTANTLY it is used to record events like accidents.

A black box serves the purpose of a device that can record the events/things that happen before an accident/crash to investigate the cause of the same.

Hence, it is also sometimes referred to as the event data recorder.

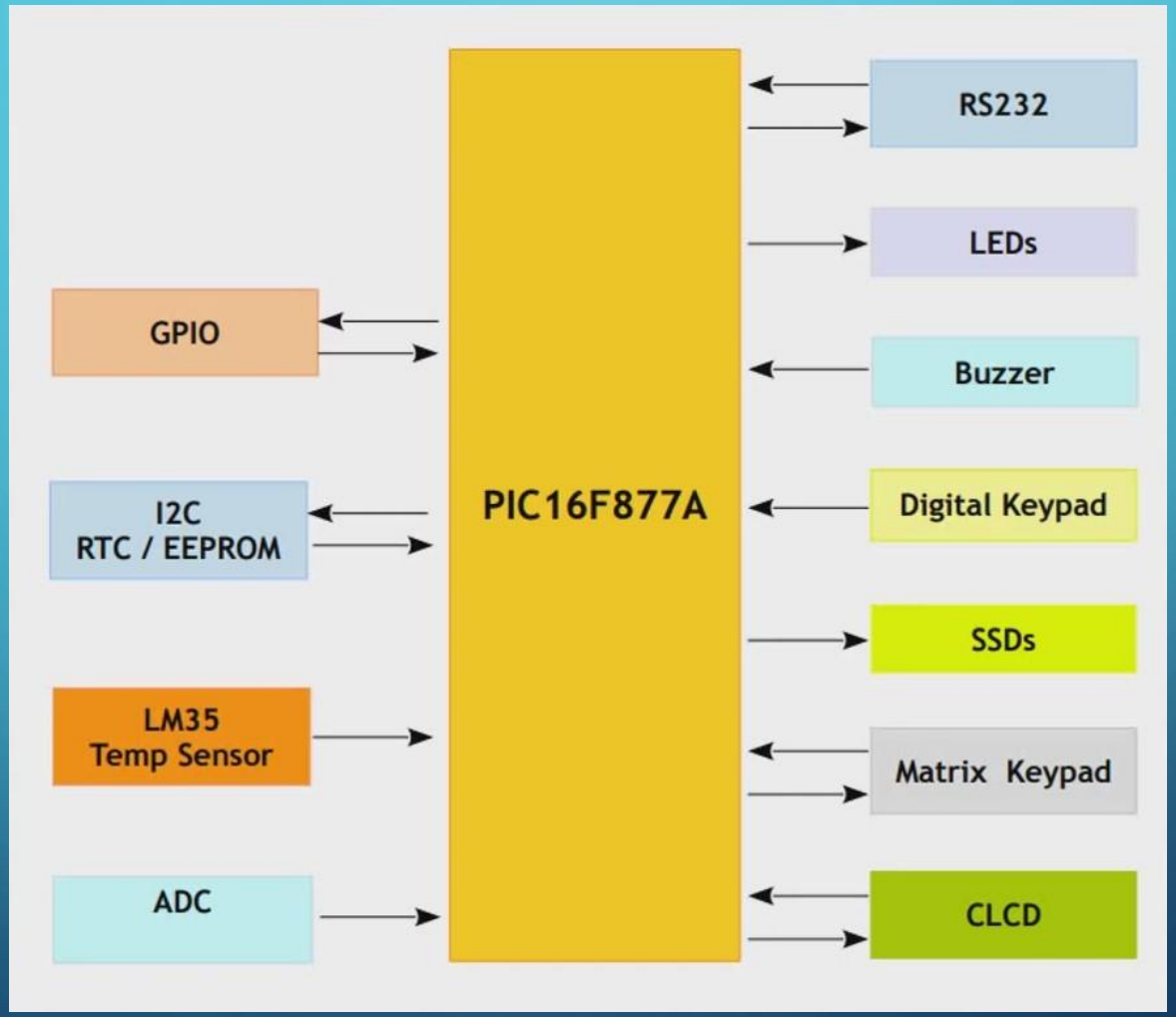
BLOCK DIAGRAM OF CAR BLACK BOX



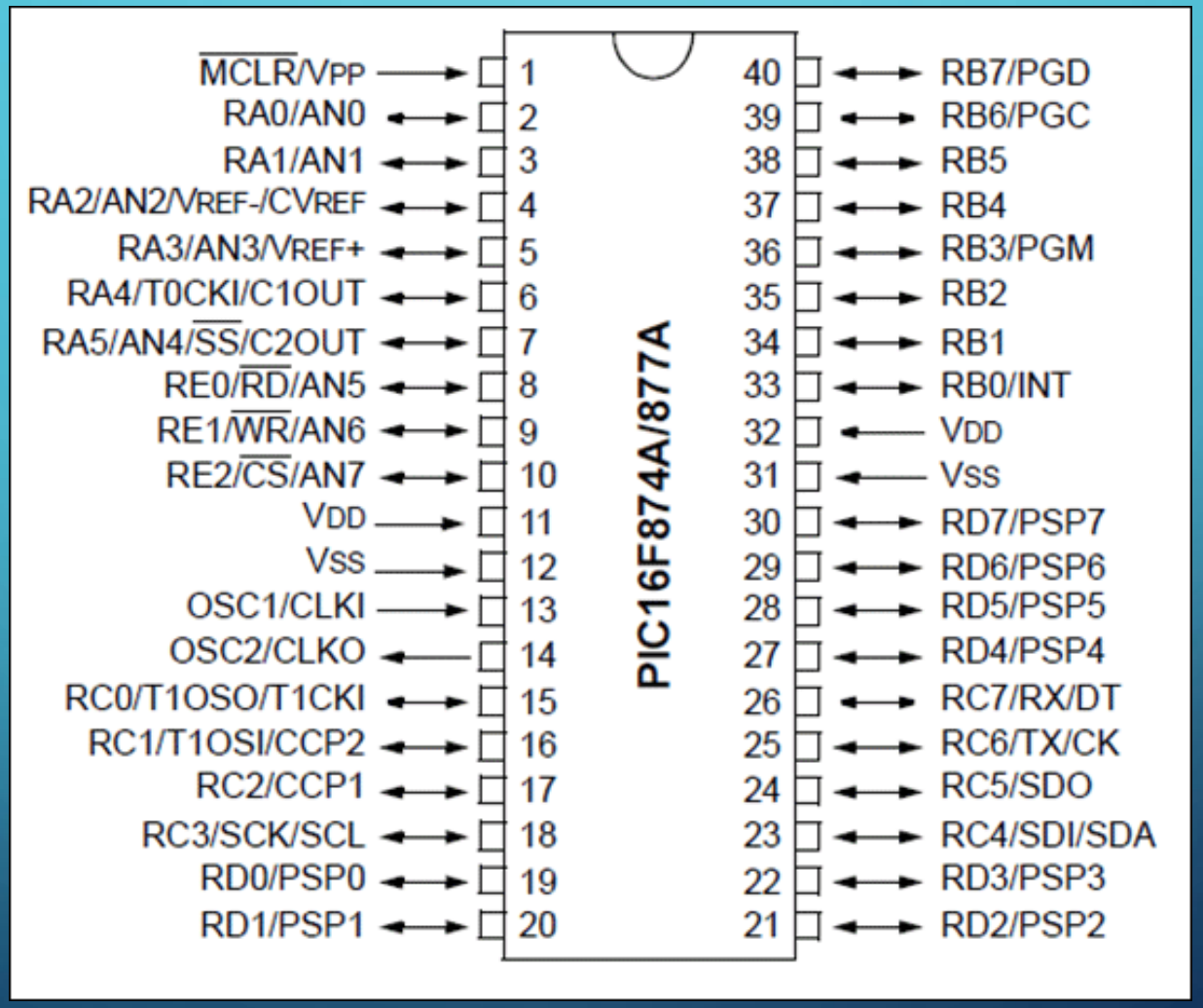
PROJECT REQUISITES

- Embedded C Coding Skills
- MPLAB X IDE
- XC8 Compiler
- PICSIM Lab
  - PIC16F877A MCU
  - 16x2 CLCD Display
  - Matrix Keypad
  - Tactile Switches
  - Timer2 Module
  - RTC, EEPROM and POT

PICSIM LAB ARCHITECTURE



PIC16F877A MCU PIN DIAGRAM



INTERRUPTS and ISRs

As the name suggests Interrupts are special events that require immediate attention by stopping a microcontroller/microprocessor from the current task. It is that signal to the processor emitted by hardware or software indicating an event that needs immediate attention. Whenever an interrupt occurs, the controller completes the execution of the current instruction and starts the execution of an Interrupt Service Routine (ISR) or Interrupt Handler. ISR tells the processor or controller what to do when the interrupt occurs. The interrupts can be either hardware interrupts or software interrupts.

ISR attends to the request of an interrupting source by clearing the interrupt flag and should save register contents that may be affected by the code in the ISR. When an interrupt occurs MPU completes the instruction being executed then it disables the global interrupt enable. Places the address from the program counter on the slack.

TIMERs

These are used to measure the time or generate the accurate time delay. The microcontroller can also generate/measure the required time delays by running loops, but the timer relieves the CPU from that redundant and repetitive task, allowing it to allocate maximum processing time for other tasks.

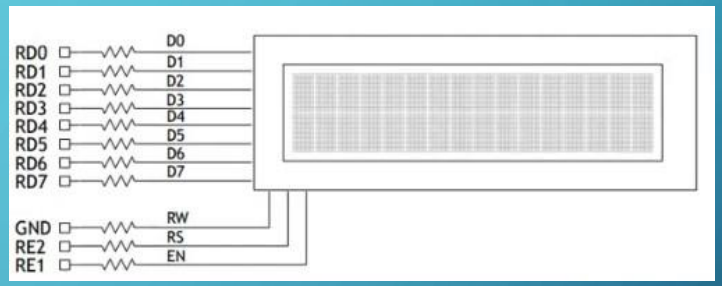
The timer can also be simply a binary counter that can be configured to count clock pulses(Internal/External). Once it reaches the max value, it will roll back to zero setting up an Overflow flag and generates the interrupt if enabled.

PIC16F877A has three timers:

- Timer0 (8-bit timer)
- Timer1 (16-bit timer)
- Timer2 (8-bit timer)

16X2 CLCD

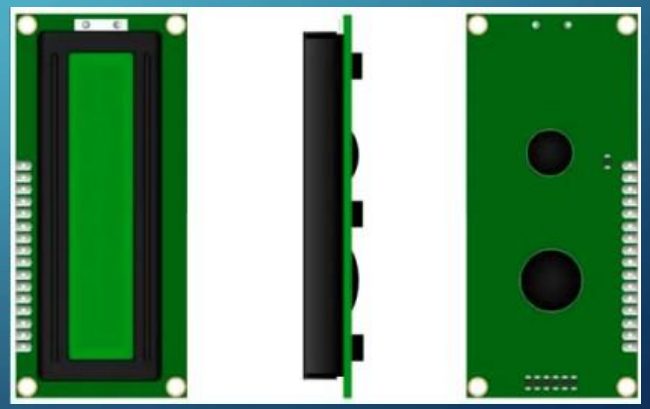
CLCD stands for Character Liquid Crystal Display which is used for displaying ASCII characters and some special symbols.



It is the most commonly used display in embedded systems.

It has 2 communication modes:

- 4-Bit Mode
- 8-Bit Mode



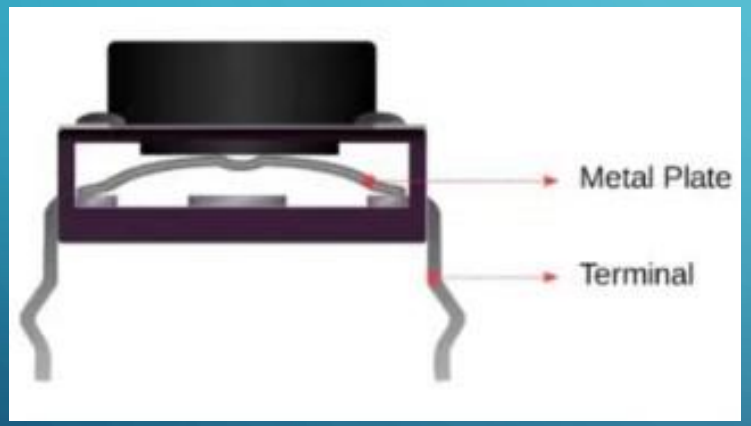
In our PICSIM Lab the two major types of display are:

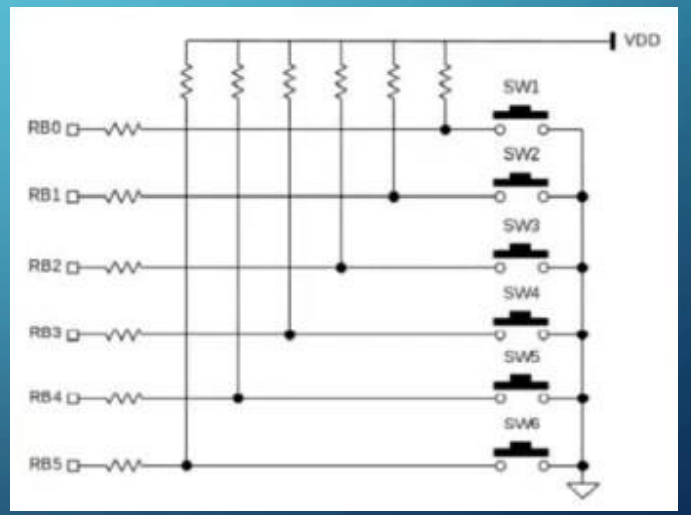
- 16x2 Display
- 16x4 Display

TACTILE SWITCHES

Tactile switches are an electromechanical device used to start or stop the flow of current through a circuit.

They provide simple and cheap interfacing and are preferable if the number of user inputs are very less.



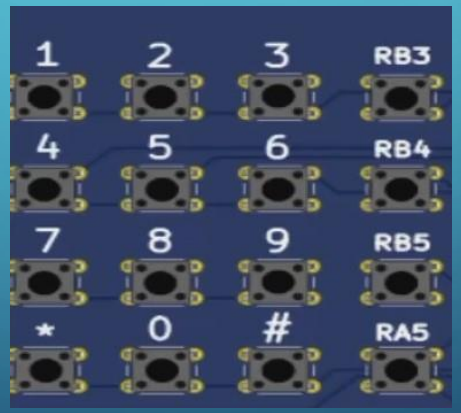


MATRIX KEYPAD [MKP]

This is a keypad in which the number of tactile switches are connected in rows and columns format.

This is used when more number of user inputs are required and still want to save some controller I/O lines.

These keypads are most commonly used in telephones, calculators, Digital Lockers etc.



RTC, EEPROM AND POTENTIOMETER

RTC stands for Real time clock which is an electronic device used to keep a measure on time with respect to the real world date and time clocks. Here in this project we've used a DS1307 RTC.

EEPROM stands for Electrically Erasable Programmable Read-Only Memory. This type of memory allows users to erase and reprogram stored data repeatedly in an application. In the project, it is being used to store events data.

Potentiometer is simply a variable resistor which is used in this project to increase/decrease the speed that can been seen on the 16x2 CLCD.

THANK YOU