Experiment no -01

**Study of Keil Micro Vision**

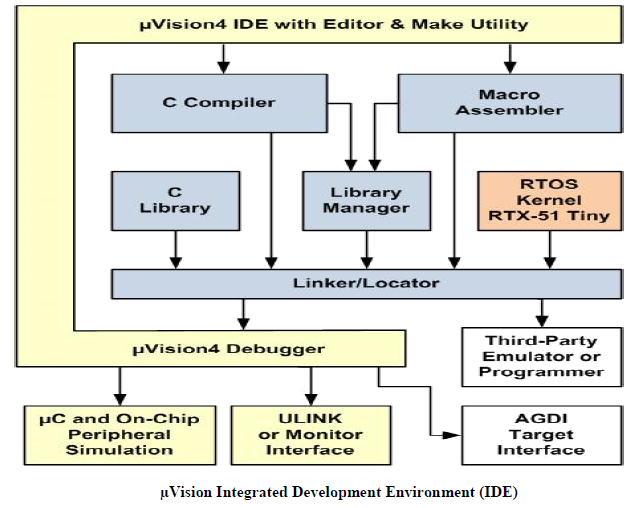
**Introduction to Keil:-**

Embedded system means some combination of computer hardware and programmable software which is specially designed for a particular task like displaying message on LCD. It involves hardware (8051 microcontroller) and software (the code written in assembly language). Some real life examples of embedded systems may involve ticketing machines, vending machines, temperature controlling unit in air conditioners etc. Microcontrollers are nothing without a Program in it.The μVision IDE is the easiest way for most developers to create embedded applications using the Keil development tools. The new Keil μVision4 IDE has been designed to enhance developer's productivity, enabling faster, more efficient program development.

**Keil MicroVision** is a free software which solves many of the main points for an embedded program developer. This software is an integrated development environment (IDE), which integrated a text editor to write programs, a compiler and it will convert your source code to hex files too. μVision4 introduces a flexible window management system, enabling us to drag and drop individual windows anywhere on the visual surface including support for Multiple Monitors.

**Development Tools**

The Keil C51 development tools offer numerous features and advantages that help you to develop embedded applications quickly and successfully. Find out more about the supported devices and the possible tool combinations available for the different 8051 variants. The following block diagram shows the components involved in the build process. The μVision IDE is a window-based software development tool that combines project management and a rich-featured editor with interactive error correction, option setup, make facility, and on-line help. Use μVision to create source files and organize them into a project that defines your target application.



**C Compiler**

The Keil Cx51 Compiler is a full ANSI implementation of the C programming language and supports all standard features of the C language. In addition, numerous extensions have been included to directly support the 8051 and extended 8051 architecture.

**Macro Assembler**

The Keil Ax51 Macro Assembler supports the complete instruction set of the 8051 and all 8051 derivatives.

**Library Manager**

The LIBx51 Library Manager allows you to create the object library from object files created by the compiler and assembler. Libraries are specially formatted, ordered program collections of object modules that may be used by the linker at a later time. When the linker processes a library, only those object modules necessary to create the program are used.

**Linker/Locater**

The Lx51 Linker/Locater creates the final executable 8051 program and combines the object files created by the compiler or assembler, resolves external and public references, and assigns absolute addresses. In addition, it selects and includes the appropriate run-time library modules.

**μVision Debugger :**The μVision Debugger is ideally suited for fast and reliable program debugging. The debugger includes a high-speed simulator capable of simulating an entire 8051 system including on-chip peripherals and external hardware.

**RTOS Kernel**

The RTOS Kernel, describes the advantages of using a real-time kernel like the Keil RTX51 Tiny in embedded systems.

**Creation of HEX File**

Some applications require a HEX file to download the application software into the physical device using a Flash programming utility. μVision creates HEX files with each build process when Create HEX File is enabled in the dialog Options for Target Output.

**Start Debugging**

μVision provides several ways to invoke debugging commands:

* Commands used from the menu Debug or the **Debug Toolbar**.
* Commands entered manually in the Command Window.
* Commands available from the **Context Menu** of the **Editor** or **Disassembly** window.
* Debug Functions executed from an initialization file.

**Start the Debugger**

* Use the **Start/Stop Debug Session** button from the **Debug Toolbar** to start or stop a debugging session.
* The current instruction or high-level statement (the one about to execute) is marked with a yellow arrow. For each step-command, the arrow moves to reflect the new current line or instruction.
* Depending on the **Options for Target — Debug** configuration, μVision loads the application program and runs the startup code (**Run to main ()**).
* μVision saves the editor screen layout and restores the screen layout of the last debug session. When program execution stops, μVision opens an Editor window with the source text or shows MCU instructions in the Disassembly Window.

**Execute Commands**

* + Run the program to the next break point, or type **GO** in the **Command Line**.
  + Halt the program, or press **Esc** while in the **Command Line**
  + Click **Reset** from the **Debug Toolbar** or from the **Debug — Reset CPU Menu** or type **RESET** in the **Command Line** to reset the CPU.

**Single-Stepping Commands**

* To step through the program and into function calls. Alternatively, you can enter **TSTEP** in the **Command Line**, or press **F11**.
* To step over the program and over function calls. Alternatively, you can enter **PSTEP** in the **Command Line**, or press **F10**.
* To step out of the current function. Alternatively, you can enter **OSTEP** in the **Command Line**, or press **Ctrl+F11**.

**On-Chip Peripherals**

There are a number of techniques you must know to create programs that can use the various on-chip peripherals and features of the 8051 family. Use the code examples provided here to get started working with the 8051.

* There is no single standard set of on-chip peripherals for the 8051 family. Instead, 8051 chip vendors use a wide variety of on-chip peripherals to distinguish their parts from each other. The code examples demonstrate how to use the peripherals of a particular chip or family. Be aware that there are more configuration options available than are presented in this text.

Follow the links to the on-chip peripherals:

1. **Header Files** - use the include files to define peripheral registers of the device in use.
2. **Startup Code** - initializes the microcontroller and transfers control to the **main** function.
3. **Special Function Registers** - explains how to use Special Function Registers (SFRs).
4. **Register Bank**s **-** explains how to use Register Banks.
5. **Interrupt Service Routines** - lists the different interrupt variants on 8051 devices.
6. **Interrupt Enable Registers** - shows how to enable the interrupts.
7. **Parallel Port I/O** - explains how to use standard I/O ports.
8. **Timers/Counters** - explains standard timers and counters.
9. **Serial Interface** - explains the implementation of serial UART communication.
10. **Watchdog Timer -** use a watchdog timer to recover from hardware or software failures.
11. **D/A Converter** - convert a digital output voltage to an analog output value.
12. **A/D Converter** - convert an analog input voltage to a digital value.
13. **Power Reduction Modes** - put the device into IDLE or POWER DOWN mode.