# CS3691 EMBEDDEDSYSTEMS AND IOT

## SYLLABUS

## COURSE OBJECTIVES:

- To learn the internal architecture and programming of an embedded processor.
- To introduce interfacing, I/O devices to the processor.
- To introduce the evolution of the Internet of Things (IoT).
- To build a small low-cost embedded IoT system using Arduino / Raspberry Pi/ open platform.
- To apply the concept of the Internet of Things in real-world scenario.

## LIST OF EXPERIMENTS

1. Write 8051 Assembly Language experiments using simulator.
2. Test data transfer between registers and memory.
3. Perform ALU operations.
4. Write Basic and arithmetic Programs Using Embedded C.
5. Introduction to Arduino platform and programming
6. Explore different communication methods with IoT devices(Zigbee,GSM, Bluetooth)
7. Introduction to Raspberry PI platform and python programming
8. Interfacing sensors with Raspberry PI
9. Communicate between Arduino and Raspberry PI using any wireless medium
10. Setup a cloud platform to log the data
11. Log Data using Raspberry PI and upload to the cloud platform
12. Design an IOT based system

## OUTCOMES:

CO1: Explain the architecture of embedded processors.

CO2: Write embedded C programs.

CO3: Design simple embedded applications.

CO4: Compare the communication models in IoT

CO5: Design IoT applications using Arduino / Raspberry Pi/open platform.

| EXP NO: | 8051 Assembly Language program using Keil simulator |
|---|---|
| DATE | |

**AIM:**

To write 8051 Assembly Language Program for an 8-bit addition using Keil simulator and Execute it.

**SOFTWARE REQUIRED:**

| S.No | Software Requirements | Quantity |
|---|---|---|
| 1 | Keilμvision5IDE | 1 |

**INTRODUCTIONTO8051SIMULATORS:**

A simulator is software that will execute the program and show the results exactly to the program running on the hardware, if the programmer finds any errors in the program while simulating the program in the simulator, he can change the program and re-simulate the code and get the expected result, before going to the hardware testing. The programmer can confidently dump the program in the hardware when he simulates his program in the simulator and gets the expected results.

8051 controller is a most popular 8-bit controller which is used in a large number of embedded applications and many programmers write programs according to their application. So testing their programs in the software simulators is a way. Simulators will help the programmer to understand the errors easily and the time taken for the testing is also decreased.

These simulators are very useful for students because they do need not to build the complete hardware for testing heir program and validate their program very easily in an interactive way.

**LIST OF 8051 SIMULATORS:**

The list of simulators is given below with their features:

**1. MCU 8051:** MCU 8051 is an 8051 simulator that is very simple to use and has an interactive IDE (Integrated Development Environment). It is developed by Martin Osmera and most important of all is that it is completely free. There are many features for this IDE they are

- ✓ It supports both C and assembly language for compilation and simulation

- ✓ It has an in-built source code editor, graphical notepad, ASCII charts, Assembly symbol viewer, etc. It also supports several 8051 ICs like at89c51, A89S52, 8051, 8052, etc.
- ✓ It will support certain electronic simulations like LED, 7segment display, LCD etc. which will help in giving the output when you interface these things to the hardware directly.
- ✓ It has tools like hex decimal editors, base converters, special calculator, file converters, inbuilt hardware programmers, etc.
- ✓ It has syntax validation, pop base auto-completion etc.

You can download this tool from https://sourceforge.net/projects/mcu8051ide/files/.

**2. EDSIM 51:** This is a virtual 8051 interfaced with virtual peripherals like 7 segment display, motor, keypad, UART etc. This simulator is exclusively for students developed by James Rogers,. The features of this simulator are

- ✓ Have virtual peripherals like ADC, DAC with scope to display comparator etc.
- ✓ Supports only assembly language
- ✓ IDE is completely written in JAVA and supports all the OS.
- ✓ Completely free and with user guide, examples, etc.

You can download this simulator from the https://www.edsim51.com/index.html.

**3. 8051 IDE:** This simulation software is exclusively for the Windows operating system (98/xp). The features of this simulator are

- ✓ Text editor, assembler, and software simulate in one single program.
- ✓ Has facilities like Breakpoint setter, execute to break point, predefined simulator watch window, etc.
- ✓ It is available in both free version and paid version.

You can download this tool from https://www.acebus.com/win8051.htm

**4. KEILµVision:**KEIListhemostpopularsoftwaresimulator.Ithasmanyfeatureslike interactive IDE and supports both C and assembly languages for compilation and simulation.

You can download and get more information from https://www.keil.com/c51/.

## INSTALLATIONOFKEILSOFTWARE

### Setup Keil IDE for Programming

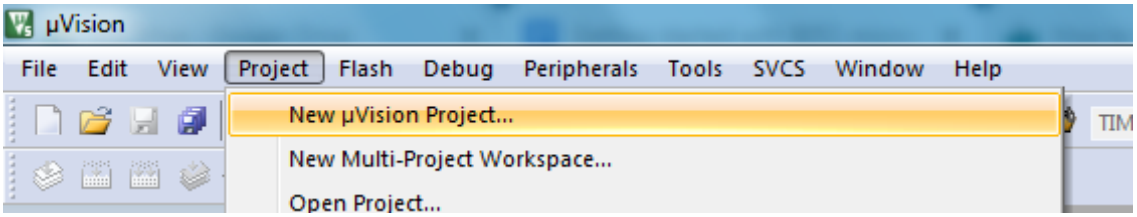
Keilµ Vision IDE is a popular way to program MCUs containingthe8051 architectures. It supports over 50 microcontrollers and has good debugging tools including logic analyzers and watch windows.

In this article, we will use the AT89C51ED 2microcontroller,which has:

- 64KBFLASHROM
- On-chip EEPROM
- 256BytesRAM
- In-System programming for uploading the program
- 3Timer/counters
- SPI,UART,PWM

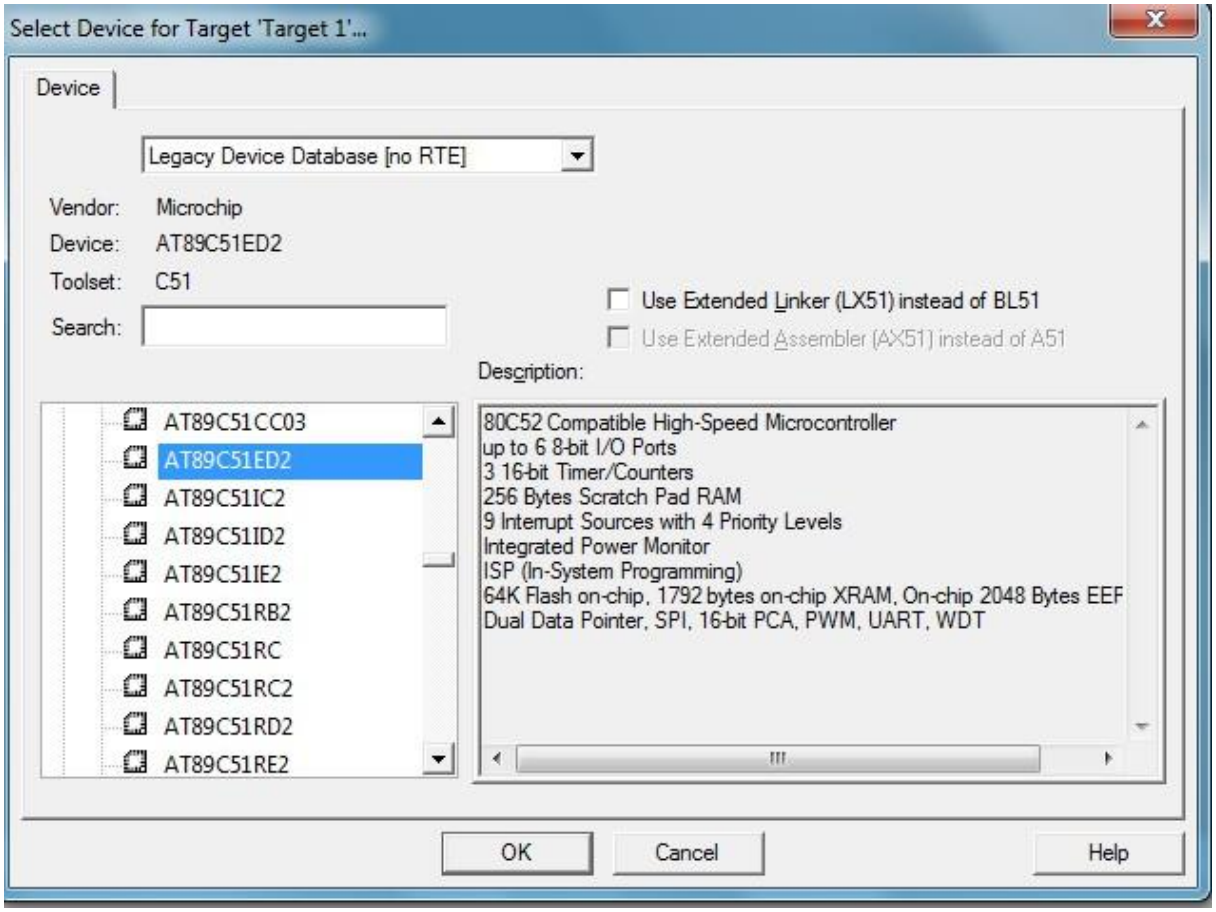

The Keilµ Visionicon.

To start writing a new program, you need to create a new project. Navigate to **project —> New µVision project**. Then save the new project in a folder.

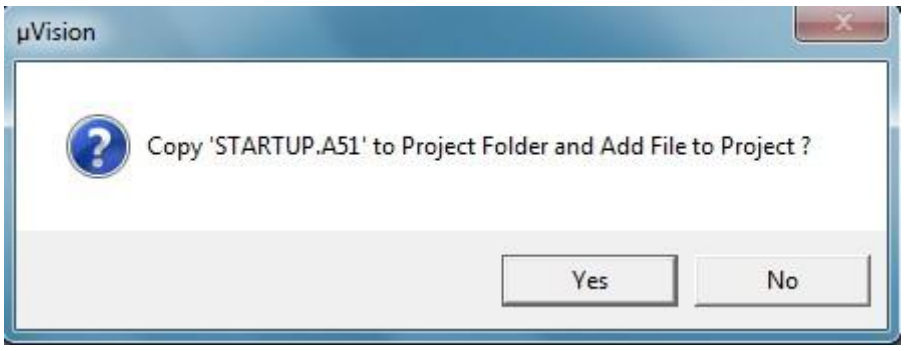



After saving the file, new windows will popup asking you to select your microcontroller.

As discussed, we are using AT89C51/AT89C51ED2/AT89C52, so select this controller under the Microchip section (as Atmel is now a part of Microchip).

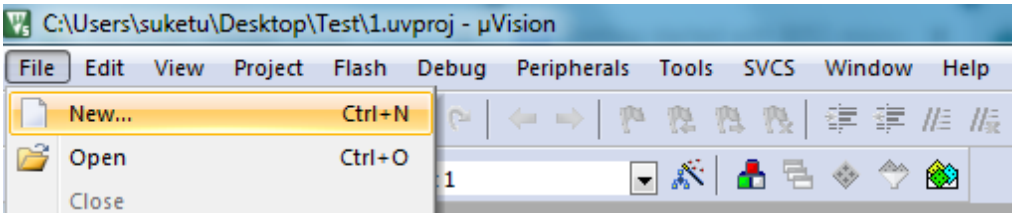



Select '**Yes**' in the next pop-up, as we do not need this file in our project.

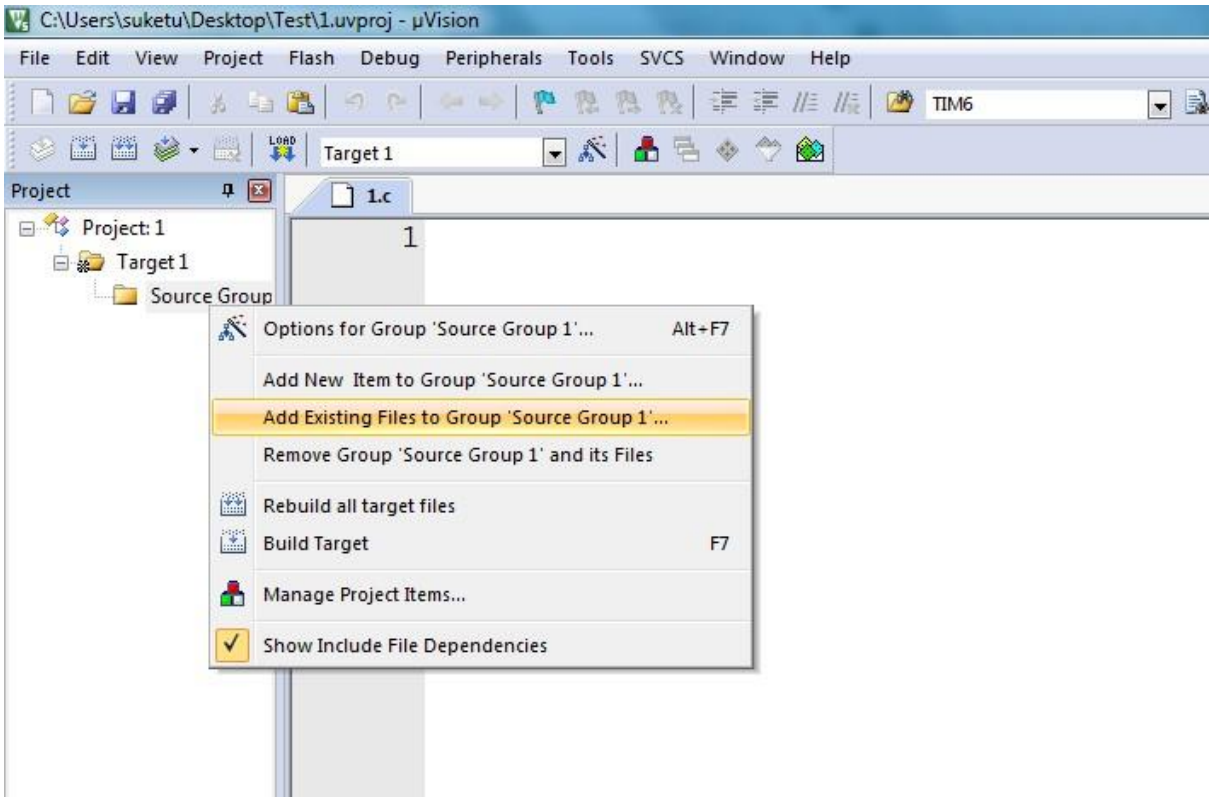



Our project work space is now ready!

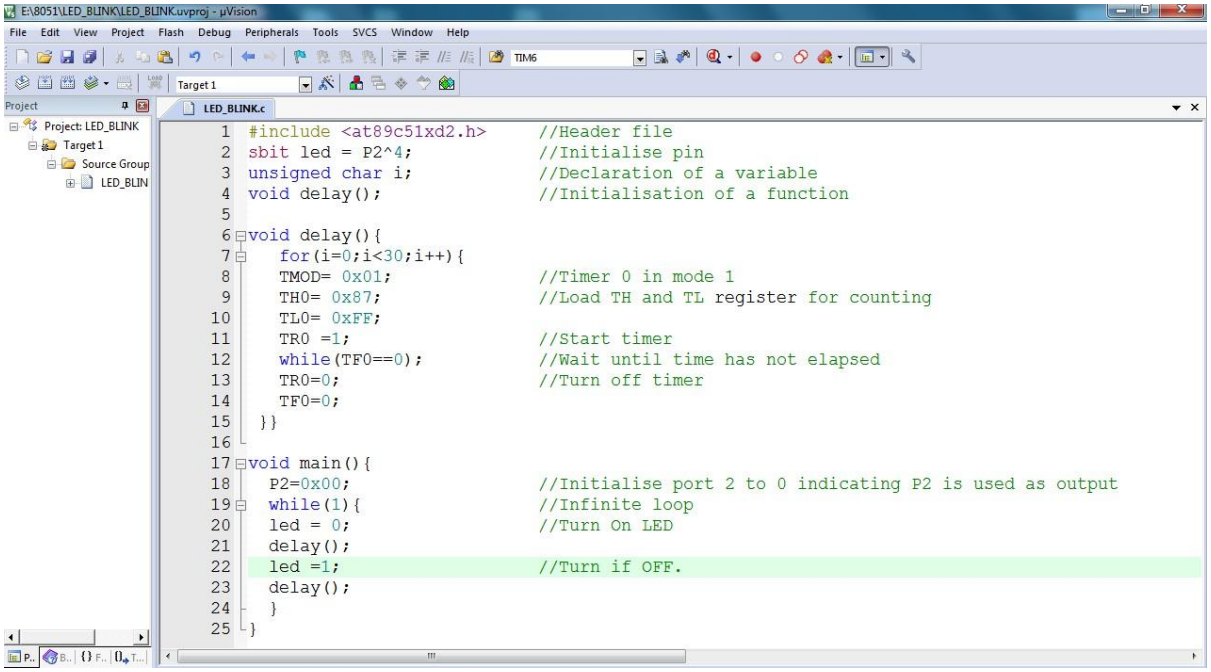
From here, we need to create file where we can write our C code. Navigate to **File—>New**. Once the file is created, save it with .c extension in the same project folder.



Next, we have to add that .c or .asm file to our project workspace. Select **Add Existing** Files and then select the created .c or .asm file to get it added.




The work space and project file are ready.



**PROCEDURE**

1. Create a new project, go to “Project” and close the current project“Close Project”.
2. Next Go to the Project New μ Vision Project and Create New Project Select Device for Target.
3. Select thedeviceAT89C51ED2orAT89C51orAT89C52
4. Add Startup file next go to “File” and click “New”.
5. Write a program on the editor window and save it with .asm extension.
6. Add this source file to Group and click on“BuildTarget”orF7.
7. Go to debugging mode to see the result of simulation clicking Run or step run.8.

**PROGRAM:**

```
ORG 0000H
CLR C
MOVA, #20H
ADD A, #21H
MOV R0, A
END
```

**OUTPUT:**

**RESULT:**