

Development Tools

Keil μ Vision IDE

Keil μ Vision is an integrated development environment (IDE) for editing, compiling, running, and debugging programs for the 8051 microcontrollers. It is incorporated in the sense that the same environment is used to perform all these tasks. We will need the FLIP tool to load programs into the 8051 microcontrollers.

A *non-integrated* workflow would be where different tools are used for different tasks. For example, Visual Studio Code could be used for editing the program and a command line compiler could be used to compile the program.

At a high-level description, the Keil IDE workflow is as follows:

1. Create a new project.
2. Specify the target device while creating the project.
3. Create a new source file in the project.
4. Build the project.
5. Fix any compilation errors and rebuild.
6. Once the program compiles without errors, use the debugger to check the correctness of execution.

See the slide deck and video in the following section for more detailed instructions.

Download and Installation

To download the Keil IDE, you will need to register at <https://www.keil.com/demo/eval/c51.htm>

Windows users can install Keil by simply double-clicking the downloaded file.

Linux users will need to install Keil using [Wine](#).

On Ubuntu, install Wine by running the following command. Use the relevant package manager for other Linux distributions.

```
sudo apt install wine-stable
```

Run the following command in the folder where c51v960a.exe was downloaded.

```
wine c51v960a.exe
```

Getting Started with Keil μ Vision IDE

- The following slide deck has an introduction to using the Keil μ Vision IDE. The pdf can be downloaded by clicking [here](#).
- A video walk-through of the features of Keil μ Vision IDE is [here](#):

Debugging in Keil

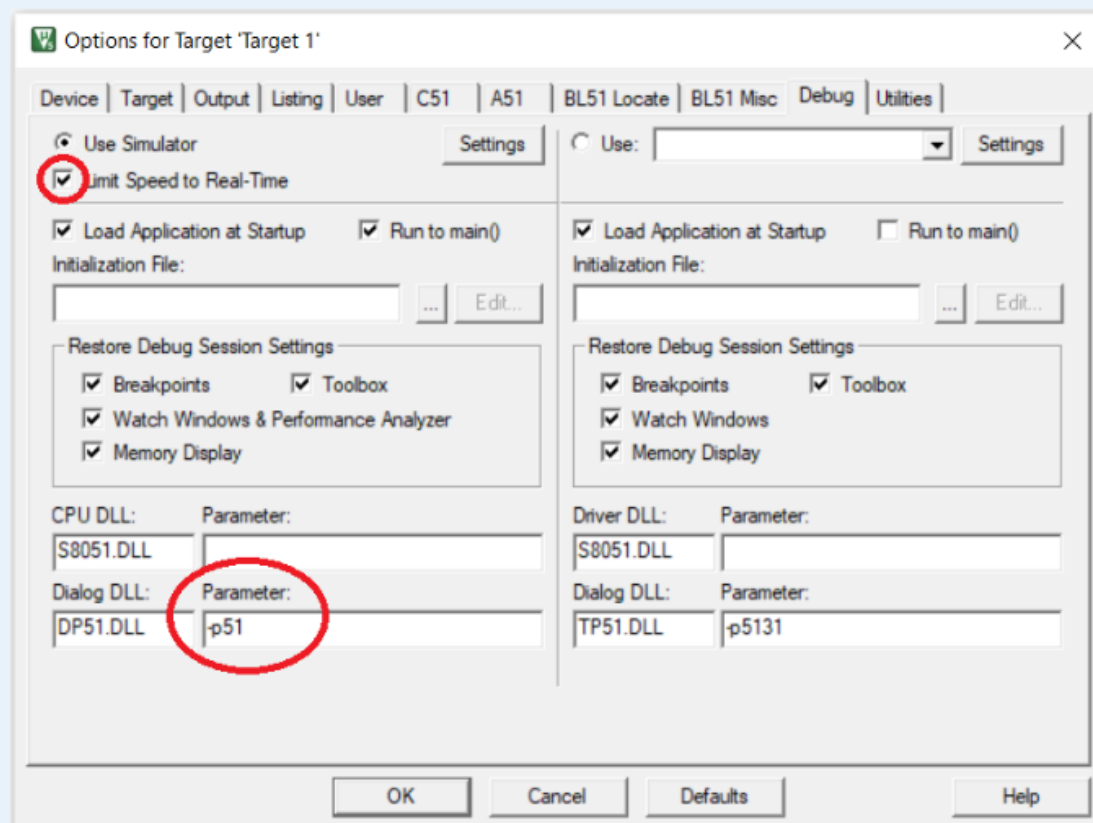
The following slide deck has instructions on debugging programs with the Keil μ Vision IDE. The pdf can be downloaded by clicking [here](#):

Note

If Keil μ Vision keeps crashing on pressing the **Start\stop debug session**, then follow these steps.

1. Right click on the target folder (at the left pane).
2. Select **Options for Target '<TargetName>'**.
3. Select the **Debug** pane.
4. Change the **Parameter** field to the right of **Dialog DLL** with text **DP51.DLL** to **-p51**.
5. Tick the **Limit Speed to Real-Time** checkbox.

See the below screenshot.



FLIP

Pt-51 board device driver installation instructions. The pdf of the above slide deck is available [here](#).

DFU Programmer Installation and Use

- **Installation steps for Linux**

- Run `sudo apt-get update -y`
- Run `sudo apt-get install -y dfu-programmer`

- **Installation steps for Mac**

- Press `Command+Space` and type `Terminal` and press `enter/return` key.
- Run the following in the Terminal app: `ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)" < /dev/null 2> /dev/null`
- If the screen prompts you to enter a password, please enter your Mac's user password to continue. When you type the password, it won't be displayed on screen, but the system would accept it. So just type your password and press `enter/return` key. Then wait for the command to finish.
- Run: `brew install dfu-programmer`

- **Use (common for Linux and Mac):**

- Run the following command for erasing flash: `sudo dfu-programmer at89c5131 erase.`
- Run the following command for programming flash: `sudo dfu-programmer at89c5131 flash <hex_file.hex>`. Make sure you are in the directory where your hexfile is present when you run it.

