INTRODUCTION OF THE ARM LPC 2148

The LPC2148 microcontrollers are based on a 32 bit ARM7TDMI-S CPU with real-time emulation and embedded trace support, that combines the microcontroller with embedded high speed flash memory of 512 kB

FEATURES

32-bit ARM7TDMI-S microcontroller in a tiny LQFP64 package.

8 to 40 kB of on-chip static RAM and 32 to 512 kB of on-chip flash program memory.128 bit wide interface/accelerator enables high speed 60 MHz operation.

In-System/In-Application Programming (ISP/IAP) via on-chip boot-loader software. Single flash sector or full chip erase in 400 ms and programming of 256 bytes in 1 ms.

EmbeddedICE RT and Embedded Trace interfaces offer real-time debugging with the on-chip RealMonitor software and high speed tracing of instruction execution.

USB 2.0 Full Speed compliant Device Controller with 2 kB of endpoint RAM.

In addition, the LPC2146/8 provide 8 kB of on-chip RAM accessible to USB by DMA.

One or two (LPC2141/2 vs. LPC2144/6/8) 10-bit A/D converters provide a total of 6/14analog inputs, with conversion times as low as 2.44 ms per channel.

Single 10-bit D/A converter provides variable analog output.

Two 32-bit timers/external event counters (with four capture and four comparechannels each), PWM unit (six outputs) and watchdog.

Low power real-time clock with independent power and dedicated 32 kHz clock input.

Multiple serial interfaces including two UARTs (16C550), two Fast I2C-bus (400 kbit/s), SPI and SSP with buffering and variable data length capabilities.

Vectored interrupt controller with configurable priorities and vector addresses.

Up to 45 of 5 V tolerant fast general purpose I/O pins in a tiny LQFP64 package.

Up to nine edge or level sensitive external interrupt pins available

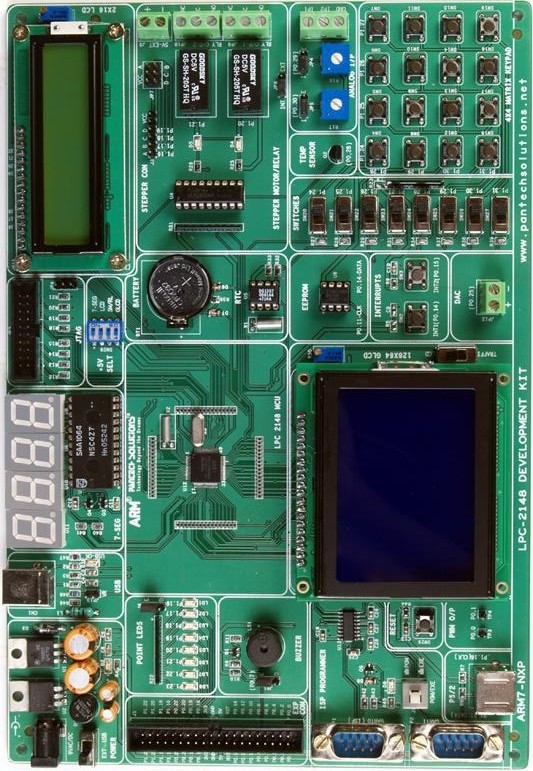
BLOCK DIAGRAM OF LPC 2148 ARM MICRO CONTROLLER



GENERAL BLOCK DIAGRAM



LPC 2148 ARM Micro Controller Development Board

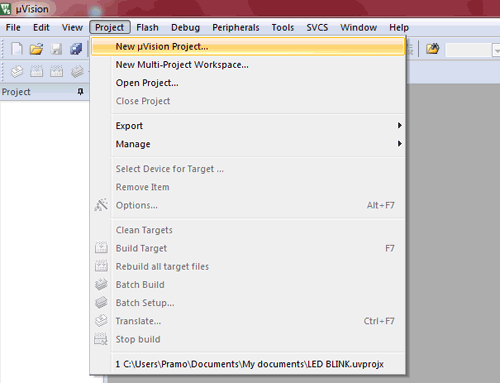


Steps involved to Create a New Project in KeiluVision 5 for ARM7 LPC2148

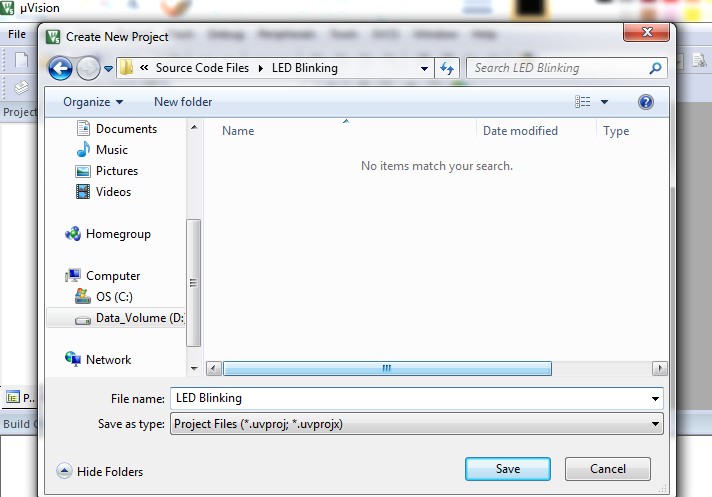
Step 1: Open KeiluVision 5



Step 2: Now open Project –> New uVision Project

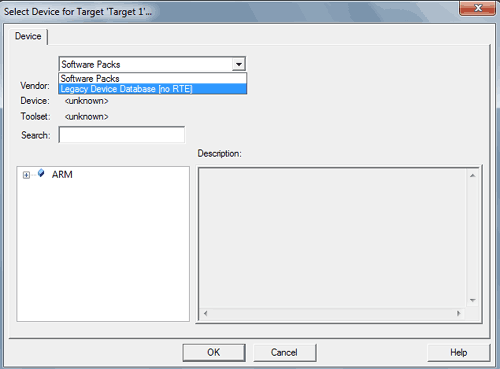


Step 3: Give Name to Project e.g. “LED blinking” and save it.

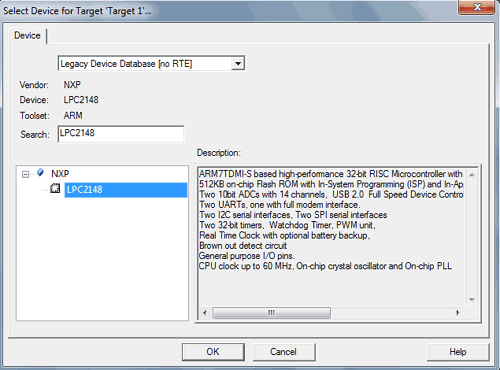


Step 4: then appears the popup box Select Device for Target “Target1”. Click the drop down menu where you need to select Legacy Device Database [no RTE].

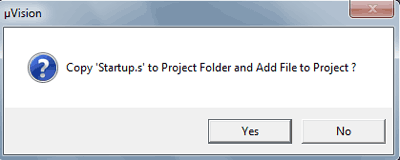
NOTE: If you need to install ARM7 packages this Legacy device database packages.



Step 5: Select our Device name LPC2148 under the NXP category and click OK

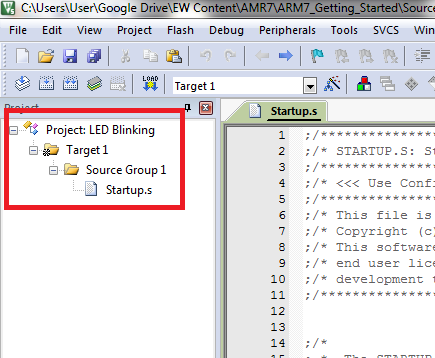


Step 6: A dialogue box appears to copy *Startup.s* to project folder, just click yes

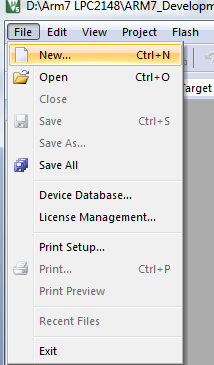


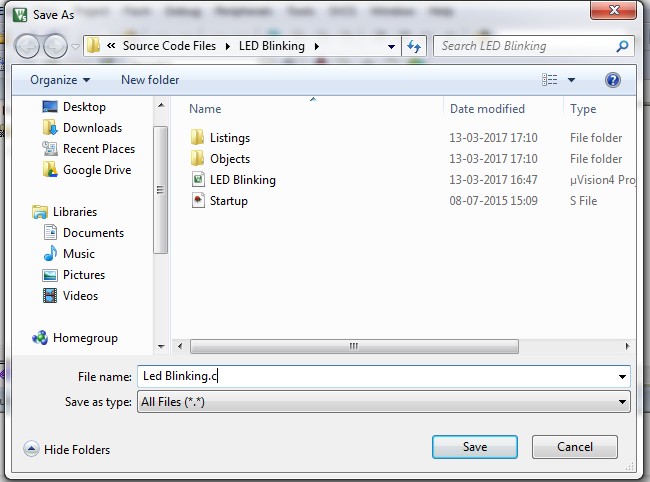
Step 7: Now it appears like the below image.

The project name and its folders can be seen on the left side in the project window.

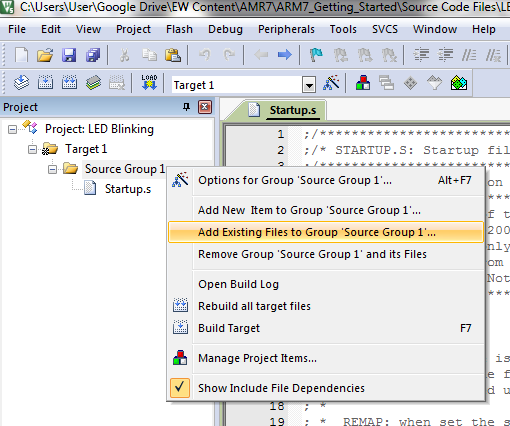


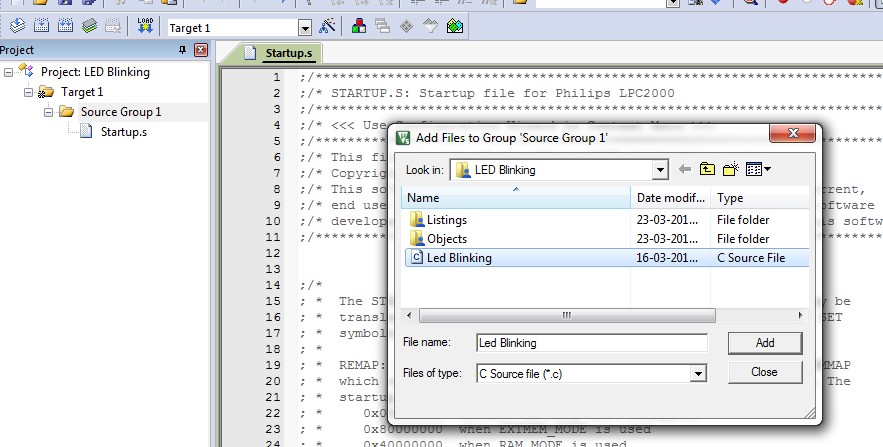
Now go to File tab and add New file from the menu.



Save the file from the previous step with a specific name. Add .c extension to the file name.Eg.(ledblinking.c)

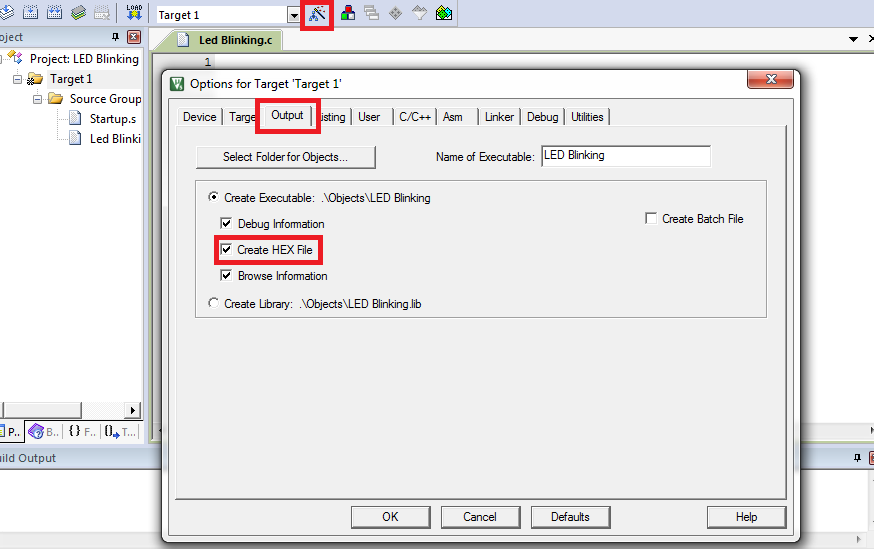
Add this .c extensionfile to Source Group folder in the project window by right clicking on Source Group1 folder and selecting Add Existing Files to Group ‘Source Group1’...



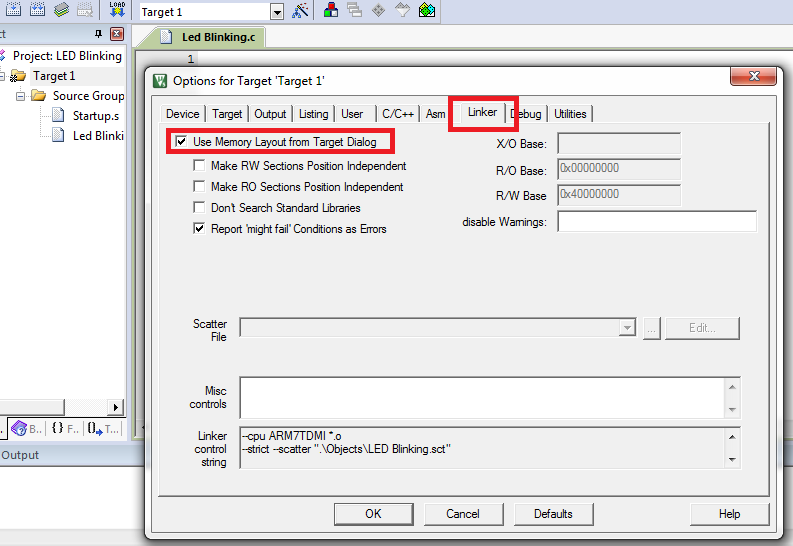


Select the previously saved file from the window that pops up and add it to the Source Group1. In our case, LED Blinking.c

Now click on the Options for Target ‘Target1’... symbol shown in red box in the image below or press Alt+F7 or right click on Target1 and click on Options for Target ‘Target1’....

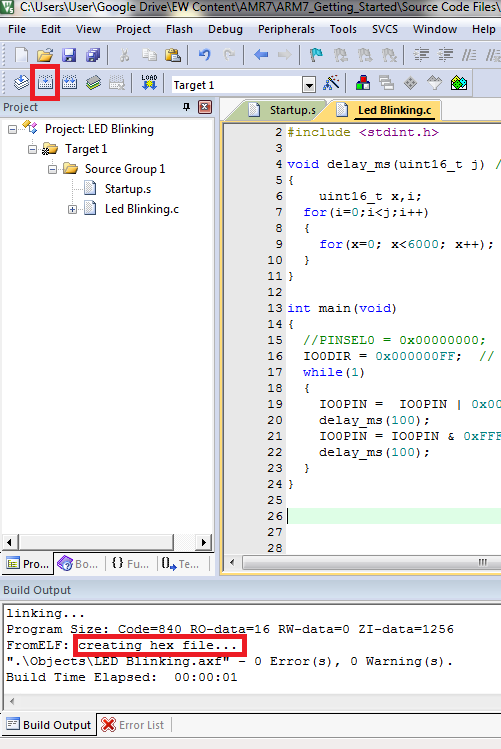
Options for target window will open. Go to the Output tab in that window. Tick ‘√’ Create HEX File option. Which will be burn into the microcontroller.

In the options for target window, go to the Linker tab. Select the Use Memory Layout from Target Dialogue option.



Then click on OK.

Now Type the source code for LED Blinking

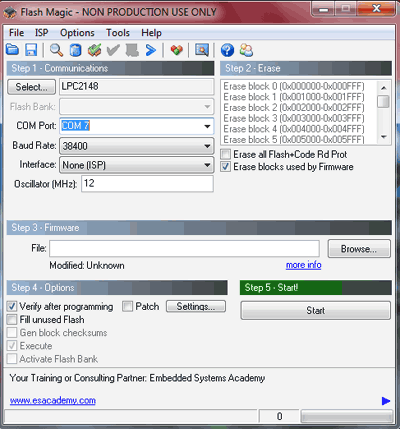
Once the code is typed, Build the code by clicking on the button shown in red in the image below. You can also build the project from the Build Target option in the Project tab or by pressing F7 on the keyboard.

You can see creating hex file ... in the Build Output window as shown in the image.

Once the project is built, a hex file is created in the Objects folder inside the folder of your project.

Use Flash Magic software to burn this hex file in your microcontroller. So open Flash Magic

The Flash magic tool appears as above.

Below are the steps for flashing the ARM LPC2148:

Select the LPC2148

Give the COM port number according to Device Manager (COM1)

Give baud rate as 9600

Oscillator as 12.000Mhz

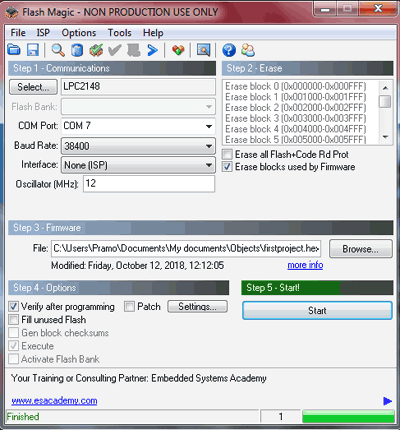
Tick mark the “Erase blocks used by firmware”

Now select the hex file path

Tick verify after programming checkbox.

And click START

After successfully burned the code , Finished (In green Colour) appears at the bottom as shown in image below



Now we can see the that LED starts blinking on the development board