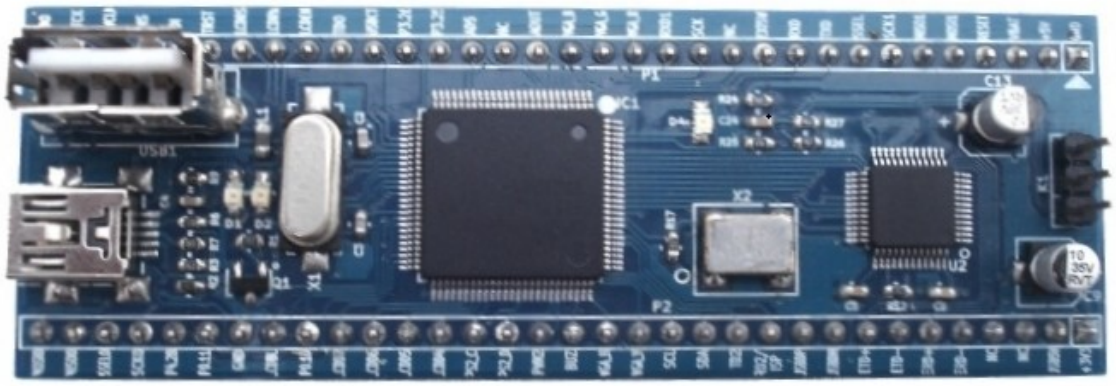
mX-LPC1768-S



USER MANUAL

Table of Contents

Introduction

The mX-LPC1768-S is an add on board to the mX-BaseBoard. The mX-S signifies it as a stamp module for mX-BaseBoard. The stamp board features LPC1768 from NXP, based on CORTEX-M3.

Features

The mX-LPC1768-S is mounted on a board with connecting pins.

Hardware

- 50 Mhz oscillator for Ethernet
- 12 Mhz for Controller
- 2x16 with contrast control & back light
- SD Card connector
- Power Jack
- Power Switch
- Reset Button
- ISP Button
- External interrupt Button
- Buzzer
- Audio Jack
- 20 pin JTAG header
- PS/2 keyboard
- VGA connector
- Serial Connector 0
- Serial Connector 1
- Preset for ADC
- On board EEPROM
- GPIO brought to male header pins

Note: Features are dependent on the stamp.
Following are the salient features of the mX-LPC1768 Stamp board:

- Dimensions: 27x84 mm2
- On-board Test Led
- 12MHz crystal
- 50Mhz oscillator for Ethernet
- On board USB Host / Device support
- On board Ethernet Transceiver
- On board dual port USB power switch and Over-current protection

Software

Precompiled firmware to test the peripherals on the mX-BaseBoard with stamp is preloaded and available *here*.

Getting Started

Before starting you would need the following things handy and ready.

Requirement

The requirement is put in two sections.

Hardware

- Power adapter – rating 7.5 V, 1 AMP
- SD card
- USB mini cable
- Serial cable
- PS/2 Keyboard
- Headphones / Speaker

Software

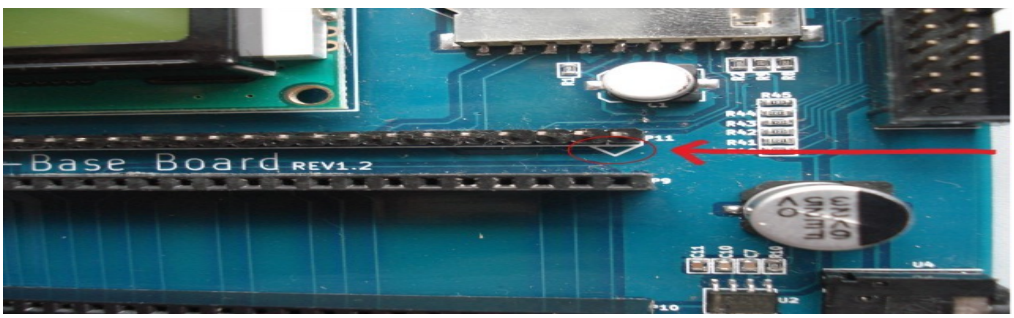
- PC with Linux OS / Windows XP OS
- Use minicom as terminal software / HyperTerminal

Setup

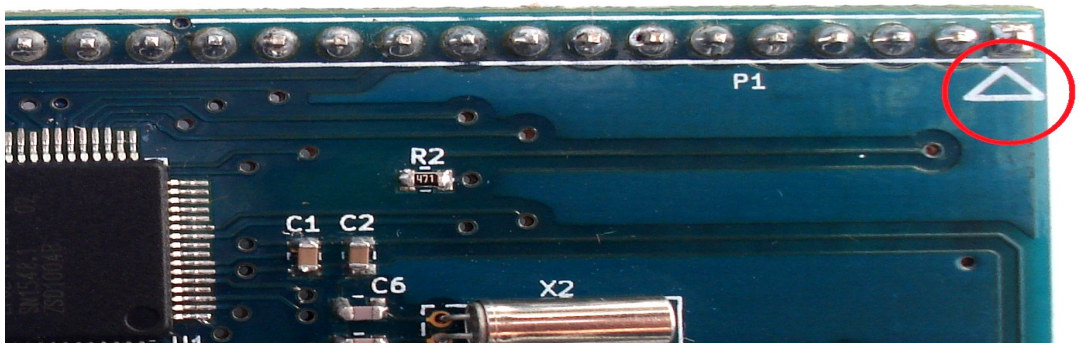
Hardware

The mX-LPC1768-S board should be mounted on the board with a particular alignment.

Note: Improper mounting of the stamp board on mX-BaseBoard may damage the stamp board and the mX- BaseBoard.



The GND pin on stamp board should be aligned with the pin 1 of P9 female header on BaseBoard. Refer to the marking as shown in the image below.



Software

- PC with WINDOWS XP
- Hyperterminal

The mX-LPC1768-S comes with a Hex files and the binaries from NGX Technologies to validate the mX-BaseBoard.

Validating mX-LPC1768-S Board

Once you have all these accessories connected to the mX-BaseBoard you can run a simple test to verify the proper working of all the peripherals. It is highly recommended that you test all the peripherals as soon you receive the BaseBoard. The mX-LPC1768-S is shipped with the preloaded bootloader firmware and binaries to test all the peripherals.

When you receive the mX-LPC1768-S board, it will be loaded will firmware for all peripherals. On connecting the power supply a default message will appear on LCD. The interfaces are activated in the following manner: LCD ,LED D4 on stamp,I2C EEPrROM, DAC,UART0, UART 1,USB HID, Ext Int, ADC, Keyboard.The other peripherals like USB HOST,SD Card and Ethernet are given as individual examples and hex files to test.Refer Read me text for testing with individual hex files.

LED

The LED D4 on the stamp board blinks for a few times.

User Interface Switch (Ext Int)

The Switch SW6 is connected to one of the external interrupt line of controller. To test this interface simply press the switch and you should hear the beep sound on the buzzer. This confirms

that both the interrupt line and the buzzer module are working fine.

LCD display

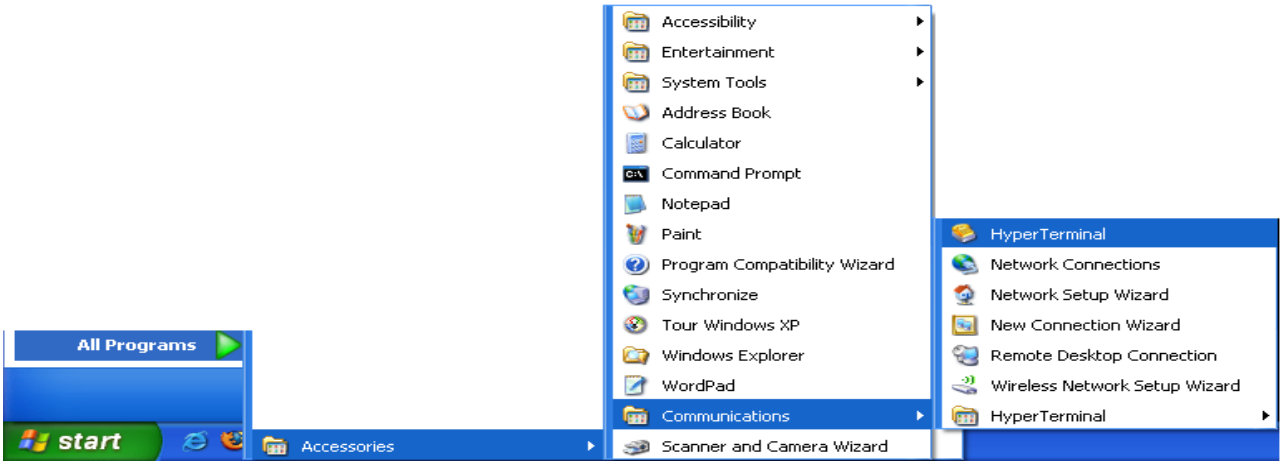
A default message "NGX TECHNOLOGIES" on line one and "mX-LPC1768 Stamp" on line two will be displayed and later status of I^2C EEPROM and ADC is displayed. The back light of LCD can be controlled by connecting jumper to appropriate pins of JP5. The contrast of LCD can be varied using the POT R19 and ADC can be varied using the POT R14.

PS/2 keyboard

To enable PS/2 connect jumper to J2. Connect a PS/2 keyboard to this connector. Now press any key on the keyboard. The user can see which key he/she has pressed on the LCD.

UART0 & UART1

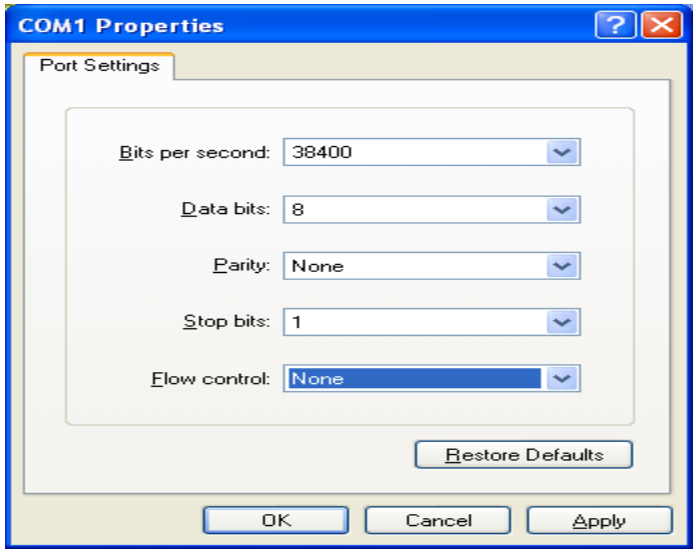
Open the hyper terminal as shown in the below image. To test the UART you can use either a full modem or half modem cable.



Click on hyper terminal a "Connection Description" window opens. Enter a name under the name tab e.g. BlueBoard and click OK.

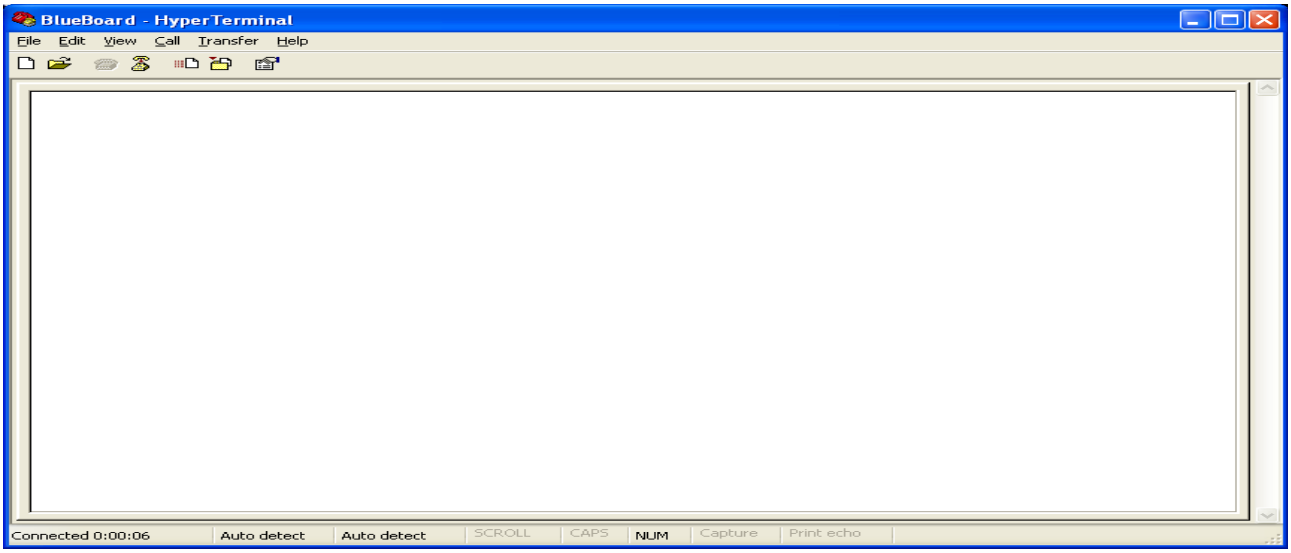
A “Connect To” window opens where you have to select the COM port. In this example it is COM1.Click OK.

A “COM1Properties” window appears. Set the values as shown below.Click OK.

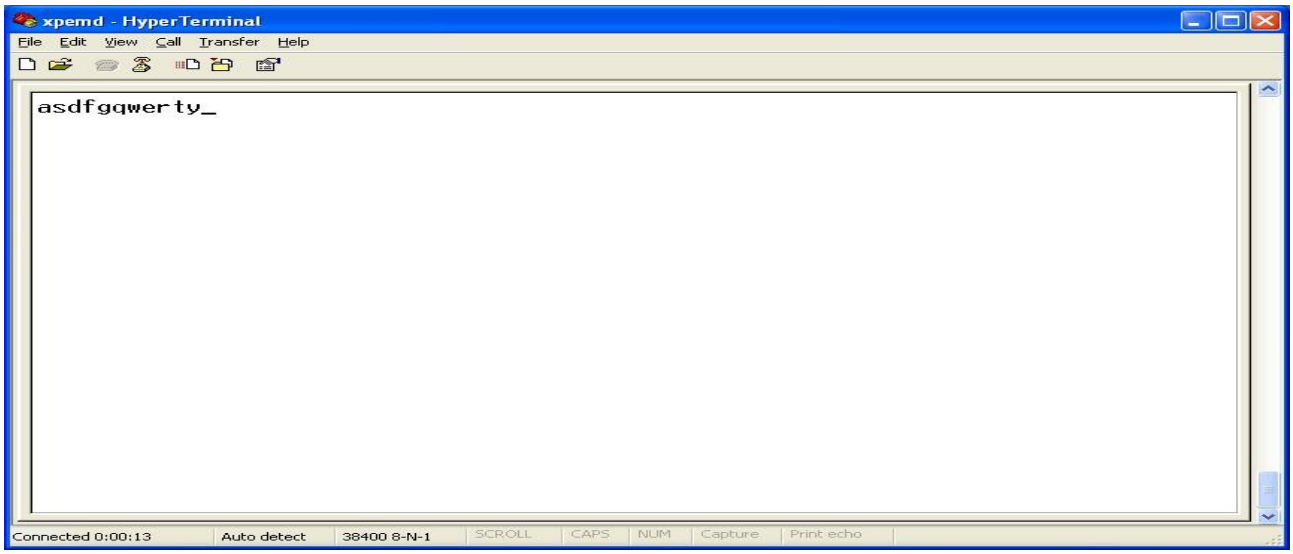


Click OK.

Next an empty “BlueBoard-Hyper Terminal” window opens as shown. Now make sure that the BlueBoard is powered and the serial port is connected to the respective port to be tested (UART0 or UART1). By pressing any key from keyboard the same will be echoed back on the terminal.



For UART0 -- (J4)
For UART1 – (J5)



UART0 can also be used for serial programming. If the selected bootloader mode is Manual then Half modem cable should be used. Auto mode is not available with mX-Baseboard .

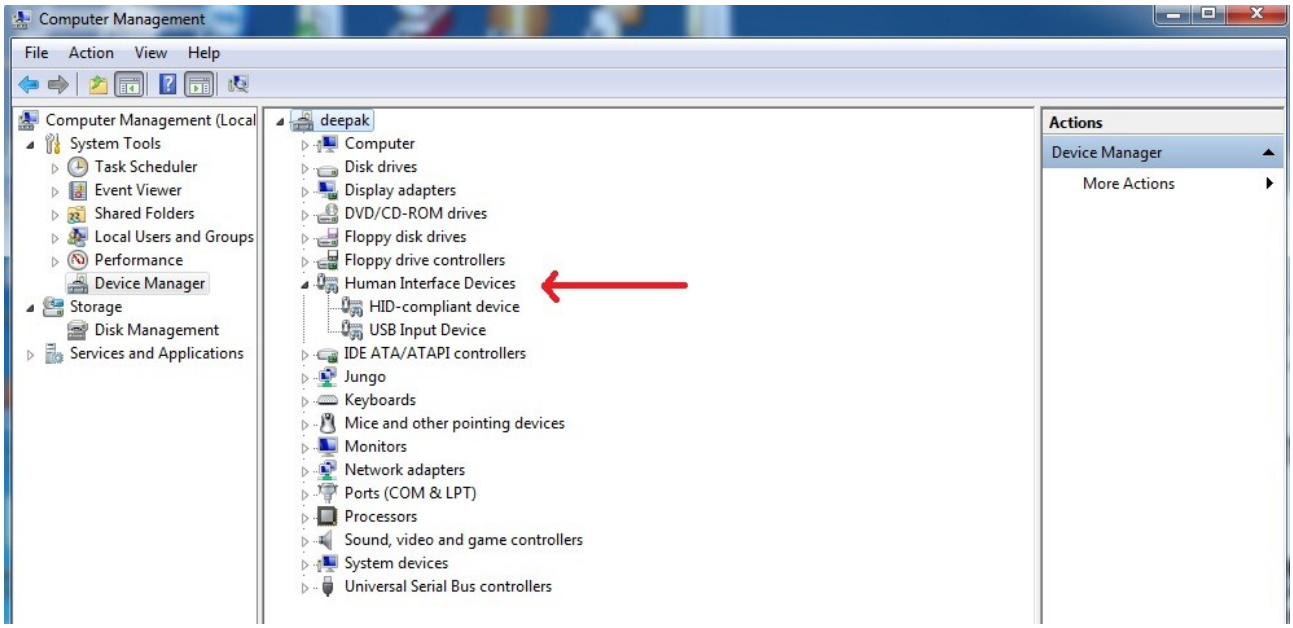
Audio jack (DAC)

Connect a headset to the audio jack (U4) connector. You should hear some random sound. The sound is heard only for few seconds after power ON or RESET.

ADC

The ADC is connected to a potentiometer (R14). To test the ADC turn the potentiometer, as the position varies the output number on the LCD varies.

USB Device



Connect the USB cable to USB connector. The USB enumeration can be checked in device manager. The mX-LPC1768-S enumerates as a USB HID as shown in the fig above.

JTAG connector

To enable debugging on the board connect jumper to P8 and connect the JTAG to debug port. We have successfully tested the mX-LPC1768-S with JTAG interface using a Wiggler Clone JTAG. To test this feature you need to have the necessary software support on your PC.

Buzzer

Connect jumper to JP1, when the board is turned on or RESET you will hear a beep after pressing switch SW6 as external interrupt. This is how the user can confirm the status of the Buzzer.

SD/MMC connector

Test SD/MMC connector with mX_SD_card_LPC1768.bin.This binary tests SD Card, when SD card is FAT formatted and inserted to the board and pressed reset , a text saying hello is written to the SD card.Observe output by reading the SD card.

Note: The SD/MMC card being tested should be formatted with FAT file system (Not FAT32 or NTFS format).

I^2C (EEPROM)

On the board at U8 a EEPROM with I^2C interface is present. The firmware tests it and diplays "I^2C EEPROM PASS".

USB HOST

Test USB HOST (USB1 connector) with USB_HOST_mXLPC1768.hex .After downloading hex to the stamp.Create a text naming MSREAD.txt and write anything and save it to the USB stick.Plug the USb stick onto stamp and press reset.Another file named MSWRITE.txt will be copied to USB stick with the data of MSREAD.txt.

Ethernet

Test Ethernet using Ethernet.hex. This tests the ethernet on board.When connected to LAN ,and configured to the local router IP address,opens UIP webserver page for configured host IP address.

Programming mX-LPC1768-S

mX-LPC1768-S can be programmed through NGX prallel port JTAG, NGX USB JTAG or through serial port using 'Flash Magic'. 'Flash Magic' is a freeware windows utility used download the hex file format onto the board. Flash Magic can be downloaded from here http://www.flashmagictool.com/. If your PC does not have a serial port; use a USB to serial converter to download the hex file using the Flash Magic utility. For programming with JTAG your system should have a parallel port or you can use the USB to JTAG from NGX Technologies and the supporting IDE which can communicate to the processor core over JTAG interface. We have successfully tested mX-LPC1768-S with wiggler clone JTAG and USB JTAG with CrossWorks IDE and KEIL IDE. A LINUX utility to download the hex file can be found here http://www.pjrc.com/arm/lpc2k_pgm/.

Programming mX-LPC1768-S Through ISP:
The mX-LPC1768-S can be programmed through ISP in Manual Mode

To program in Manual mode you need a half serial cable (which just has TX, RX and GND wire connected). connect the serial cable to UART0 (J4) and power the board.

To make the board enter programming mode

- Hold down SW5 (ISP) and SW4 (RESET), then release SW5 first and finally SW4
- The controller enters the bootloader mode if during reset the SW5 pin is low

Programming and Debugging with JTAG:

The mX-LPC1768-S can be programmed and debugged either using parallel port or USB JTAG.

- NGX Parallel Port JTAG using H-JTAG (refer *here*)
- NGX USB JTAG

USB Boot-loader

The mX-LPC1768-S comes pre-loaded with the USB boot-loader firmware. With the USB boot-loader you can simply drag and drop your compiled binary file onto the device as you would with any USB flash drive and it will update itself accordingly. Serial port is not required for flashing. To make the board enter boot-loader mode follow the steps below.

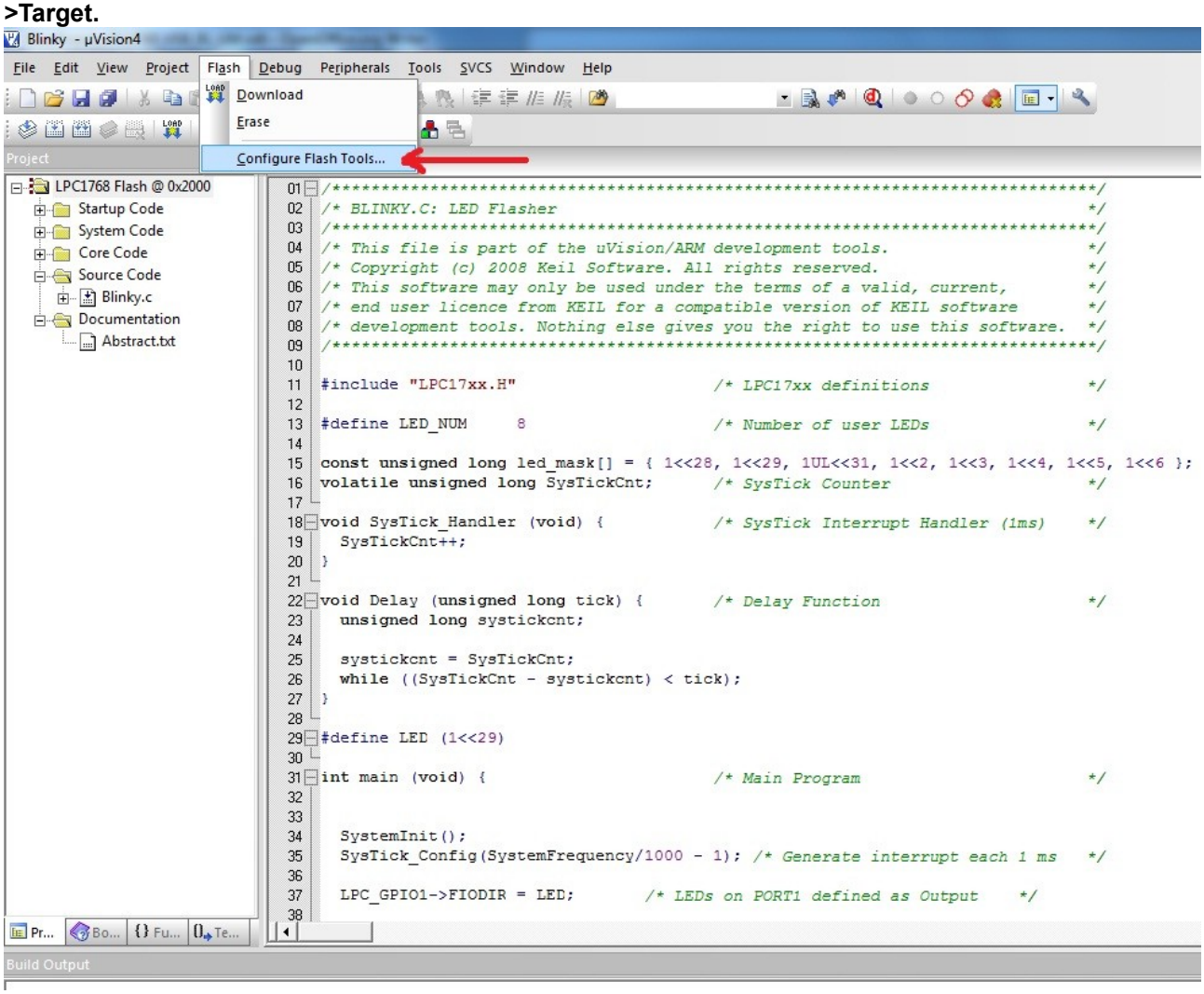
How To Enter USB Boot Loader Mode

After loading USB boot-loader firmware to the mX-LPC1768-S, hold down SW6 (USBBL) and SW4 (RESET), then release SW6 first and finally SW4. Now the pre-loaded USB boot-loader allows you to enumerate the board as a Mass Storage Device. Drag the compiled binary file and drop onto the device and reset the board using SW4 and the firmware executes.

Note: USB boot-loader firmware is provided as NGX_17xx_Bootloader_mXbase.hex only .The blinky test bin and the boot-loader hex can be found here.

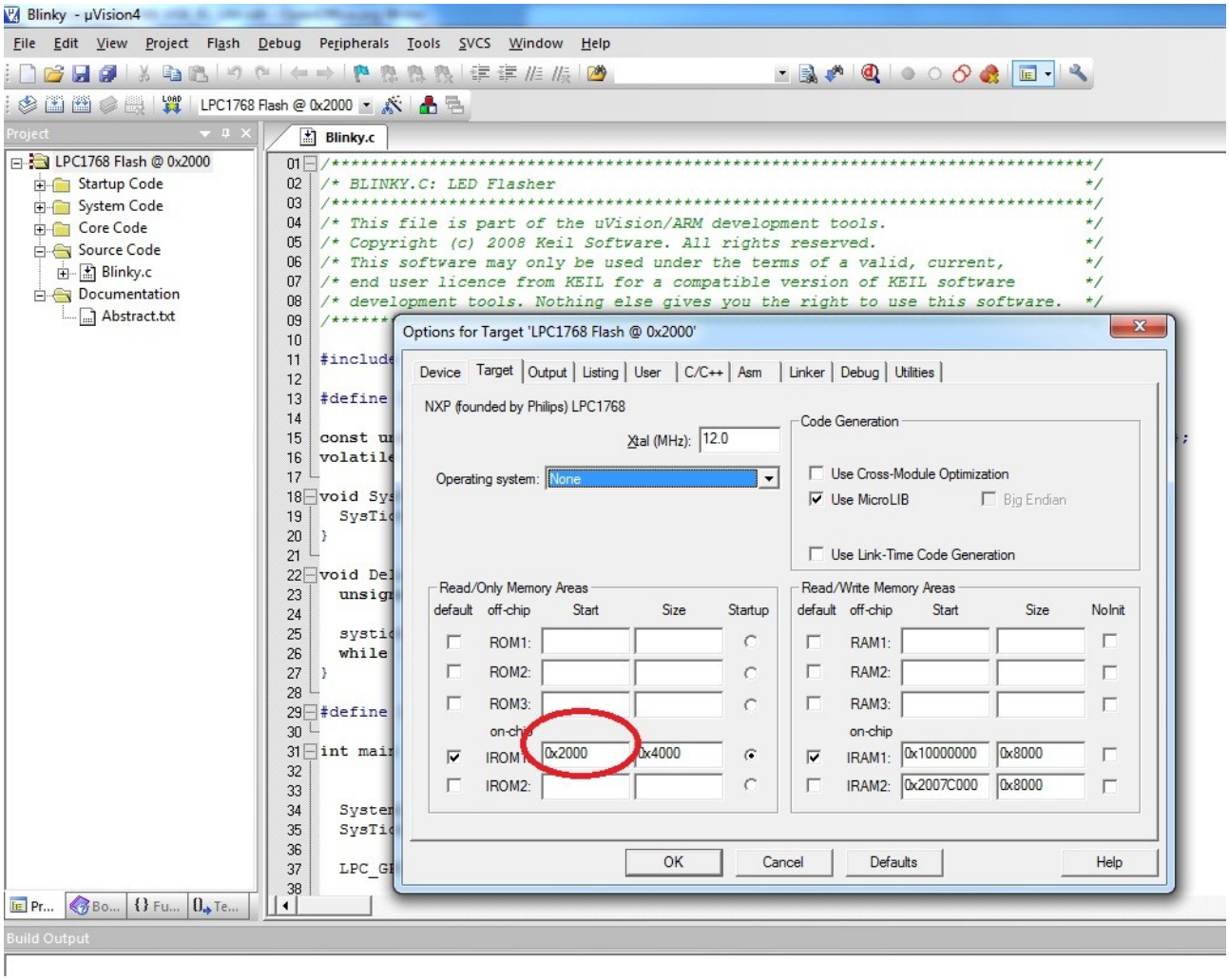
Steps involved in making your KEIL project compatible with the pre-loaded USB boot-loader in mX-LPC1768-S

1)As shown in below fig. Open the µVision IDE and go to **Flash** --> **Configure Flash Tools-->Target.**

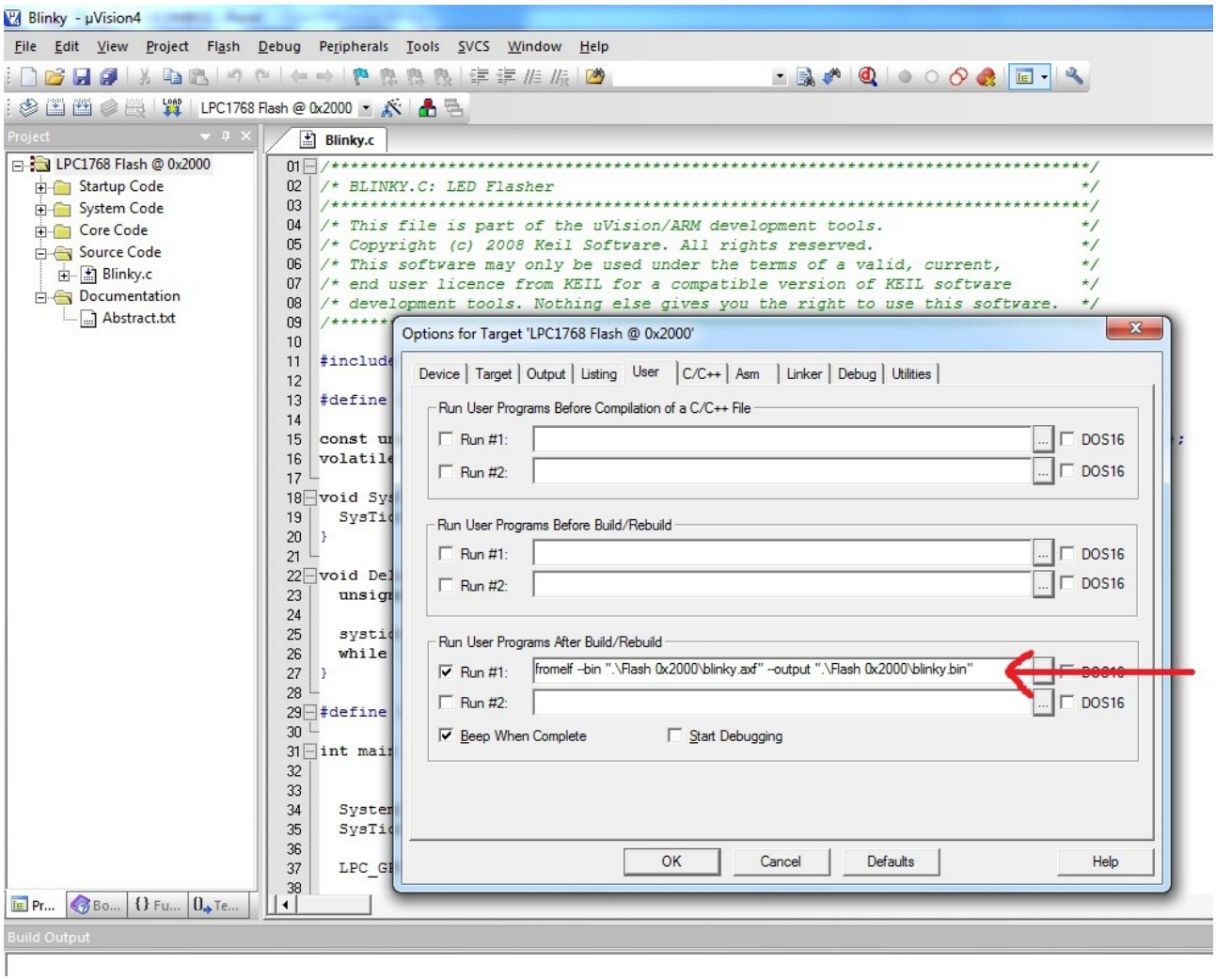



```
/*****************************************************************************/
/* BLINKY.C: LED Flasher                                                     */
/*****************************************************************************/
/* This file is part of the uVision/ARM development tools.                   */
/* Copyright (c) 2008 Keil Software. All rights reserved.                    */
/* This software may only be used under the terms of a valid, current,       */
/* end user licence from KEIL for a compatible version of KEIL software      */
/* development tools. Nothing else gives you the right to use this software. */
/*****************************************************************************/

#include "LPC17xx.H"                    /* LPC17xx definitions               */

#define LED_NUM     8                   /* Number of user LEDs               */

const unsigned long led_mask[] = { 1<<28, 1<<29, 1UL<<31, 1<<2, 1<<3, 1<<4, 1<<5, 1<<6 };
volatile unsigned long SysTickCnt;      /* SysTick Counter                   */

void SysTick_Handler (void) {           /* SysTick Interrupt Handler (1ms)   */
  SysTickCnt++;
}

void Delay (unsigned long tick) {       /* Delay Function                    */
  unsigned long systickcnt;

  systickcnt = SysTickCnt;
  while ((SysTickCnt - systickcnt) < tick);
}

#define LED (1<<29)

int main (void) {                       /* Main Program                      */


  SystemInit();
  SysTick_Config(SystemFrequency/1000 - 1); /* Generate interrupt each 1 ms   */

  LPC_GPIO1->FIODIR = LED;       /* LEDs on PORT1 defined as Output    */
```

2) Change the start address of on-chip IROM1 memory from 0 to 0x2000 as shown in fig. Below .This is the area where user program is placed and has to run when the controller is reset.



3) **Here in fig as below** Go to **Flash** --> **Configure Flash Tools-->User.**



Insert the user command in Run #1 ticking the check box. After compile and build the entire project,from elf it creates hex file along with the hex file there will be another file with the extension .axf. This user command is used to convert .axf file to .bin. Here in fig. Above blinky.axf is converted to blinky.bin which is used for drag drop programming with the USB boot-loader that allows you to enumerate the board as a Mass Storage Device.

Note:The location of .axf file in the user command must be same where its located in the project folder. In the above fig the default location of .axf file is folder Flash 0x2000 in the project folder so the user commands goes like .\Flash 0x2000 \blinky.axf.

Appendix

mX-LPC1768-S Board Utilities

For working with mX-LPC1768-S board there are certain tools that need to be installed. The tools required are:

Flash Magic Tool.
The flash magic tool can be downloaded from the following link:
http://www.flashmagictool.com/

For LINUX machines you may use http://www.pjrc.com/arm/lpc2k_pgm/

Tool chain

To be able to generate the hex or the binary file the user needs to install the tool chain for ARM based microcontrollers. Any toolchain can be used as long as it is able to generate the necessary files for downloading onto the mX-LPC1768-S. Here are few toolchain suggestions:
GNUARM Toolchain: http://winarm.scienceprog.com/winarm-tools/prepare-gnuarm-compiler-toolchain-for-windows.html
Crossworks IDE: http://www.rowley.co.uk/arm/
IAR Systems: http://www.iar.com

KEIL uvision4 IDE a 32K code limited version could be used .For those who wish to program (hex files) using **parallel port programmer** could consider **H-JTAG**

Schematics

Refer to the documentation here.

Sample Code

Download the zipped file here.

Information

Revision History

version: v1.0 author: Deepak S Patil

Legal

NGX Technologies Pvt. Ltd. provides the enclosed product(s) under the following conditions:

This evaluation board/kit is intended for use for ENGINEERING DEVELOPMENT, DEMONSTRATION, EDUCATION OR EVALUATION PURPOSES ONLY and is not considered by NGX Technologies Pvt. Ltd to be a finished end-product fit for general consumer use. Persons handling the product(s) must have electronics training and observe good engineering practice standards. As such, the goods being provided are not intended to be complete in terms of required design-, marketing-, and/or manufacturing-related protective considerations, including product safety and environmental measures typically found in end products that incorporate such semiconductor components or circuit boards. This evaluation board/kit does not fall within the scope of the European Union directives regarding electromagnetic compatibility, restricted substances (RoHS), recycling (WEEE), FCC, CE or UL and therefore may not meet the technical requirements of these directives or other related directives.

The user assumes all responsibility and liability for proper and safe handling of the goods. Further, the user indemnifies NGX Technologies from all claims arising from the handling or use of the goods. Due to the open construction of the product, it is the user's responsibility to take any and all appropriate precautions with regard to electrostatic discharge.

EXCEPT TO THE EXTENT OF THE INDEMNITY SET FORTH ABOVE, NEITHER PARTY SHALL BE LIABLE TO THE OTHER FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES.

NGX Technologies currently deals with a variety of customers for products, and therefore our arrangement with the user is not exclusive. NGX Technologies assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein.

Please read the User's Guide and, specifically, the Warnings and Restrictions notice in the User's Guide prior to handling the product. This notice contains important safety information about temperatures and voltages.

No license is granted under any patent right or other intellectual property right of NGX Technologies covering or relating to any machine, process, or combination in which such NGX Technologies products or services might be or are used.

Disclaimers

Information in this document is believed to be reliable and accurate. However, NGX Technologies does not give any representations or warranties, expressed or implied, as to the completeness or accuracy of such information and shall have no liability for the consequences of use of such information.

NGX Technologies reserves the right to make changes to information published in this document, at any time and without notice, including without limitation specifications and product descriptions. This document replaces and supercedes all information supplied prior to the publication hereof.

Trademarks

All referenced trademarks, product names, brands and service names are the property of their respective owners.