# User Manual for Micro-A748

**V4.0.SI**

**www.microembedded.in**

# 1.0. Introduction.

## 1.1 Overview of ARM

Today we see a lot of new and amazing gadgets in the market. We see high end mobile phones, tablet PC, Net book PC and state of the art electronic devices. It will amaze us that all these devices are using processors which are having ARM Microprocessor inside it. ARM microprocessor technology is the most efficient technology available today.

Micro-A748 Microcontroller Kit is specifically designed keeping in mind the needs of students, to understand and learn the architecture of the ARM 7 microprocessor.

The Micro-A748development kit is based on LPC2148 aARM7 core based SOC(system on chip). Devices like LED's and LCD, I2C interface based memory devices, RTC, SD/MMC Card interface, Matrix Keypad, ADC, DAC, graphical LCD, Stepper motor, etc are interfaced on the board.

### ARM Controller.

The ARM Processor is a RISC(Reduced Instruction Set Computer) machine. The RISC instruction set is primarily known for smaller number of instructions and higher throughput at a lower power.

The ARM Architecture is characterized by the following;

- Load-store architecture.
- Fixed length 32-bit instructions in ARM Mode and 16-bit instructions in THUMB Mode.
- 3 address instruction format.

There are several modes of operation. In each mode there are 16 registers that are accessible to the user programs. A Current Program Status Register (CPSR) is also available which contains all the mode bits and the interrupt bit and the condition bits. The modes allow the user program restricted access to certain resources of the processor.

*Following are the modes in the ARM Processor.*

**User Mode:** This is the main operating mode for the processor. Program running in this mode can achieve isolation and protection for the resources.

**Fast Interrupt Processing Mode (FIQ):** This mode is entered when the Fast interrupt is received. This is called a fast interrupt as it has a dedicated vector address from where the program can immediately execute without any latency.

**Normal Interrupt Processing mode (IRQ):** Interrupt from other sources will make the processor enter this mode.

**Software Interrupt Mode:** when the processor encounters a Software Interrupt Instruction this mode is entered. Software Interrupts are a standard way to enter the privileged mode from User Mode.

**Undefined Instruction mode:** when a processor attempts to execute a instruction which neither its main core or the co-processor can execute then this mode is entered.

**System mode:** Privileged operating system task can be run from this mode.

**Abort Mode:** this mode is entered when a data fault is occurred.

The 3 Stage Pipeline.

There is a 3 stage pipeline in the ARM 7 processor.

- Fetch: Fetch the instruction from memory.
- Decode: Decode the instruction.
- Execute: Execute the instruction.

Due to the 3 stage pipeline the ARM processor is able to effectively execute 1 instruction for every clock cycle. Thus it has a very high throughput and performance.

The ARM Processor supports the following Data Types.

8 Bit signed and unsigned byte.

16 Bit signed and unsigned half word aligned on 2 byte boundary.

32 bit signed and unsigned Word aligned on 4 byte boundary.

The ARM Processor supports both the BIG Endean and Little Endean Memory alignment.

**Features:**

The Micro-A748 microcontroller board has been specifically designed keeping in mind the needs of students for learning the ARM architecture. The board gives a complete overview for interfacing various peripheral devices which are used in the industry and consumer devices alike. A hands-on with the board will develop in the student the experience to design and implement various devices and products based on the ARM Microcontroller.

*Following are the features of the Micro-A748 Microcontroller board.*

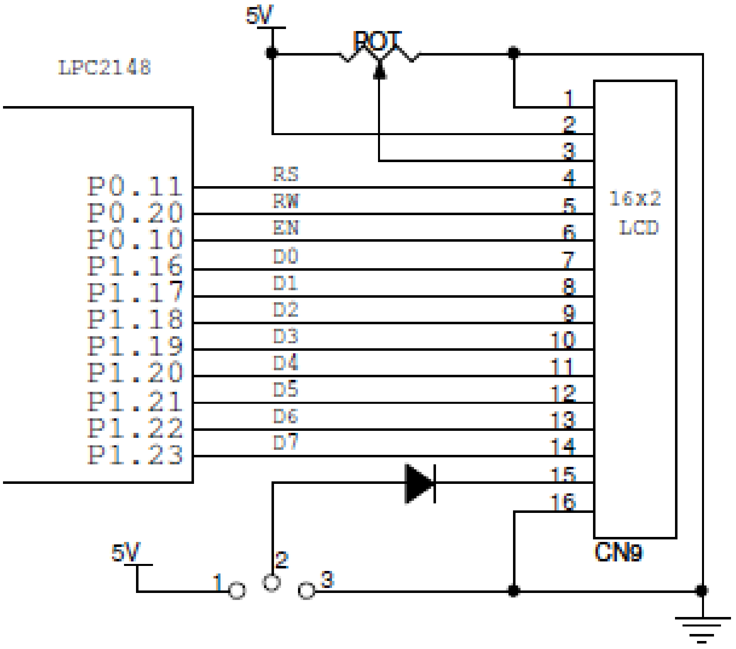
- Based on LPC2148 microcontroller from NXP running at maximum 60 MHz
- 12VPower Supply.
- UART-0 with USB-to-Serial and UART-1with RS-232 transceiver.
- 16x2 LCD module.
- 16 Key-(4x4) Matrix Keypad.
- 8 general purpose LED’s.
- RTC with power backup.
- EEPROM with I2C interface.
- SD/MMC card interface on SPI interface.
- Zigbee Module.
- 10 bit on Chip ADC interfaced to External voltage source and Temperature Sensor.
- 10 Bit DAC.
- Stepper Motor and DC Motor Driver (L293D).

# 2.0. Hardware: Functional Description and Interfacing.

## 2.1. Liquid Crystal Display (LCD).

A 16x2 character LCD is interfaced on the Micro-A748 board on **Port-0**.

Note: In order to use the LCD, switch **SW20** must be between position**1-2 (LCD ON)**.



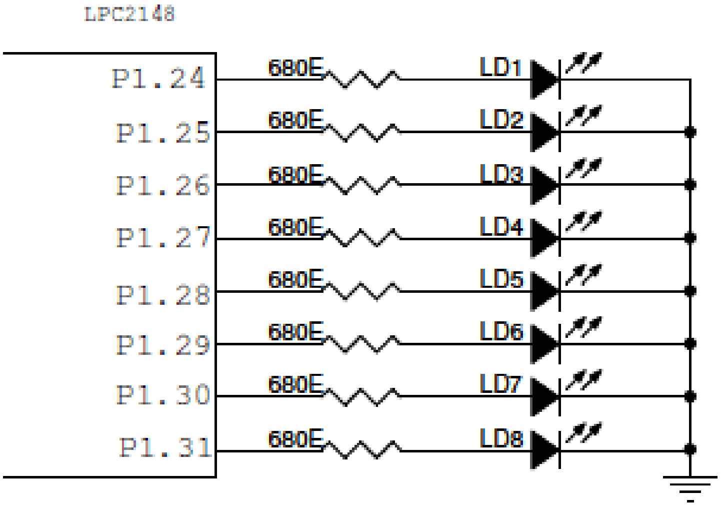
### LCD Interfacing Details.

| 16x2 Character LCD | Pin Details | |
|---|---|---|
| | LPC2148 | Description |
| D0 | P1.16 | LCD Data Bit 0 |
| D1 | P1.17 | LCD Data Bit 1 |
| D2 | P1.18 | LCD Data Bit 2 |
| D3 | P1.19 | LCD Data Bit 3 |
| D4 | P1.20 | LCD Data Bit 4 |
| D5 | P1.21 | LCD Data Bit 5 |
| D6 | P1.22 | LCD Data Bit 6 |
| D7 | P1.23 | LCD Data Bit 7 |
| EN | P0.10 | LCD Enable Signal |
| RS | P0.11 | LCD Register Select |
| RW | P0.20 | LCD Read Write |

## 2.2. Light Emitting Diodes(LED).

Eight general purpose LED's are provided on the Micro-A748 Board.The LED's are connected to the GPIO pins in common cathode configuration.LED's **LD1 thru LD8** are interfaced to Port Pins **P1.24 thru P1.31**. All LED's are buffered using a current buffer IC 74244.

Note: In order to make the LED ON we have to give Logic "1"(HIGH).

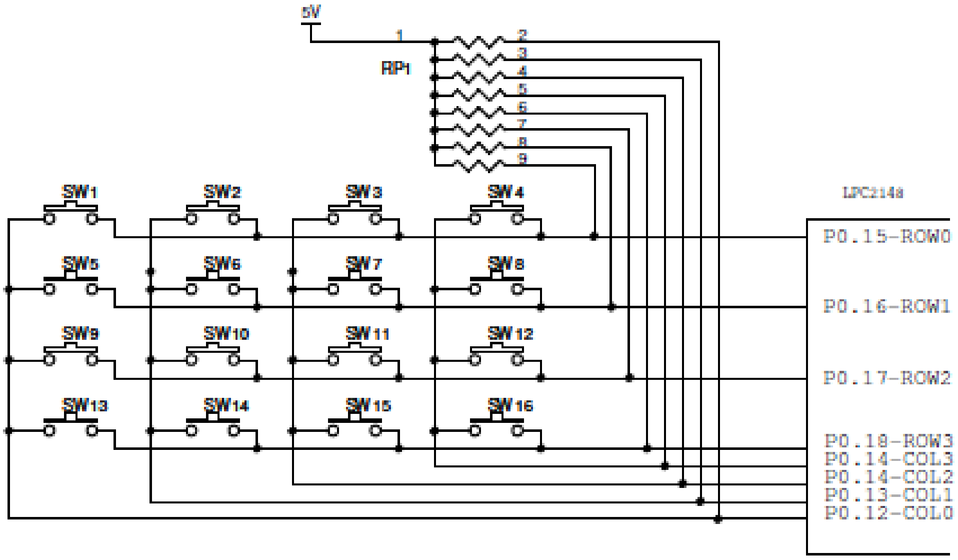


**LED interfacing details.**

| LED | Pin Details | |
|---|---|---|
| | **LPC2148** | **Description** |
| LD 0 | P1.24 | Common Cathode LED 0 |
| LD 1 | P1.25 | Common Cathode LED 1 |
| LD 2 | P1.26 | Common Cathode LED 2 |
| LD 3 | P1.27 | Common Cathode LED 3 |
| LD 4 | P1.28 | Common Cathode LED 4 |
| LD 5 | P1.29 | Common Cathode LED 5 |
| LD 6 | P1.30 | Common Cathode LED 6 |
| LD 7 | P1.31 | Common Cathode LED 7 |

## 2.3. Keypad.

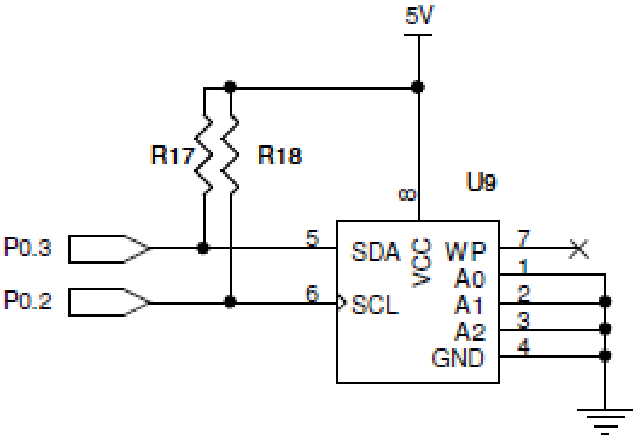
A 16 key Keypad in a 4x4 matrix circuit is provided on the Micro-A748 board. The Keypad is interfaced using the GPIO pins of the processor. The pins are all pulled up using a resistor network.



**Keypad interfacing details.**

| Keypad | Pin Details | |
|---|---|---|
| | **LPC2148** | **Description** |
| COL 1 | P0.12 | Keypad COL 1 Signal |
| COL 2 | P0.13 | Keypad COL 2 Signal |
| COL 3 | P0.14 | Keypad COL 3 Signal |
| COL 4 | P0.15 | Keypad COL 4 Signal |
| ROW 1 | P0.16 | Keypad ROW 1 Signal |
| ROW 2 | P0.17 | Keypad ROW 2 Signal |
| ROW 3 | P0.18 | Keypad ROW 3 Signal |
| ROW 4 | P0.19 | Keypad ROW 4 Signal |

## 2.4. I2C Interface (EEPROM).



The I2C interface from the processor has been brought out on the Micro-A748 board. The EEPROM (AT24Cxx) with I2C interface has been integrated on the board.

The Port pins have to be configured to work as I2C clock and Data lines.

Device Address for the i2c EEPROM is **0xA0**. Pull up resistor is also populated on the board.

The **I2C-0** pins are also available on **GPIO Header 1 (CN11)** for interfacing with external devices.

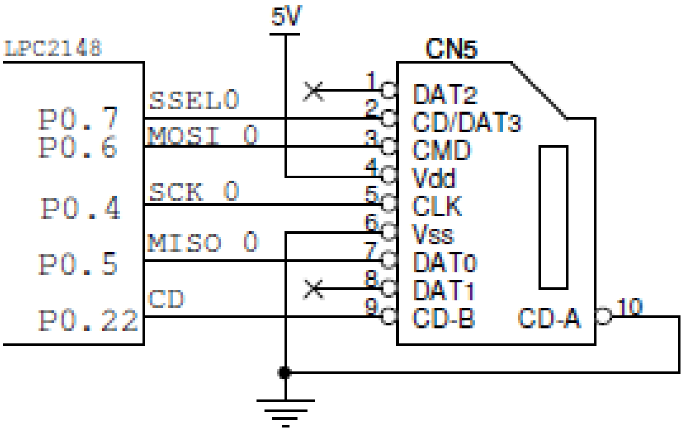
**I2C interfacing details.**

| I2C | Pin Details | |
|---|---|---|
| | **LPC2148** | **Description** |
| I2C0_SCL | P0.2 | I2C Clock |
| I2C0_SDA | P0.3 | I2C Data |

## 2.5.SD Card Interface.

The SPI interface from the processor is available on the Micro-A748 board. Devices such as SPI based SD/MMC Card can be interfaced on the board using the SPI.

The **SPI-0** pins are also available on **GPIO Header 1 (CN11)** for interfacing with external devices.



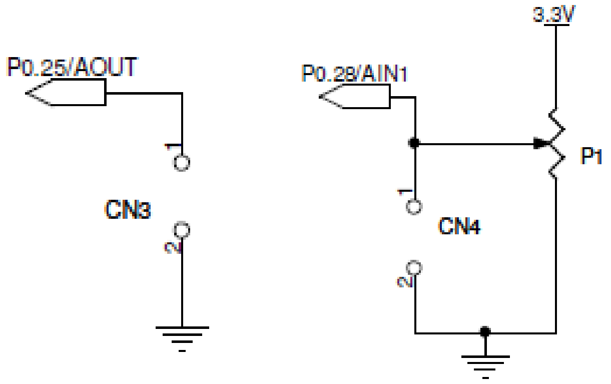
**SD Card interfacing details.**

| SD Card | Pin Details | |
|---|---|---|
| | **LPC2148** | **Description** |
| CLK (5) | P0.4 | SD Card SPI Clock |
| CD/DAT3 (2) | P0.7 | SD Card Chip Select |
| DAT 0 (7) | P0.5 | SD Card SPI Data OUT (MISO) |
| CMD (3) | P0.6 | SD Card SPI Data IN (MOSI) |
| CD (9) | P0.22 | SD Card Card Detect |

## 2.6. Analog to Digital Converter (ADC)&Digital to Analog Convertor (DAC).

The processor on the Micro-A748 contains a 10 bit ADC with 7 channels. The board allows the user with oneanalog channel. Analog voltage is provided using potentiometers **P1**. **ADC0.1** peripheral is connected to this input. The Analog voltage can be measured at connector **CN4.**

The Processor on Micro-A748 contains a 10 bit Digital to Analog Convertor. The Output of this DAC is brought out on pin **P0.25** on the board at connector **CN3**.



**ADC and DAC interfacing details.**

| Data Acquisition | Pin Details | |
|---|---|---|
| | **LPC2148** | **Description** |
| ADC | P0.28 | ADC Input (ADC0.1) |
| DAC | P0.25 | DAC output |

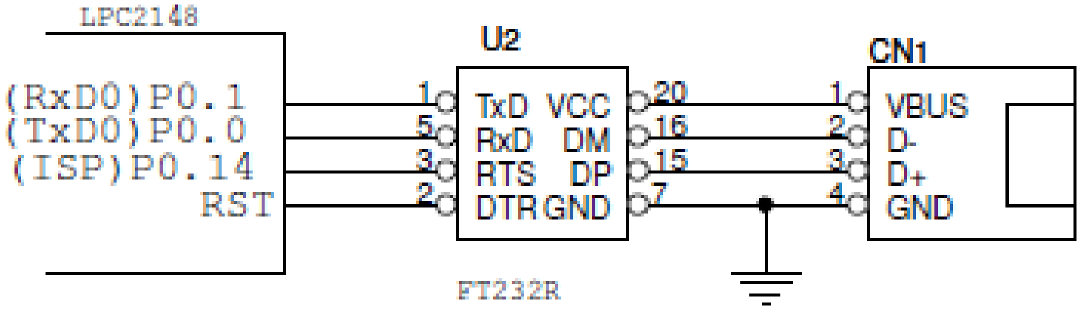
## 2.7. RTC

The Micro-A748 board uses the internal on-chip RTC on the processor LPC2148. A crystal of **32.768 KHz** is populated on the board. Backup power is provided by the lithium ion battery cell.

## 2.8. Serial Communication using UART.

Two UART interfaces from the processor are available for the user on the Micro-A748 board.

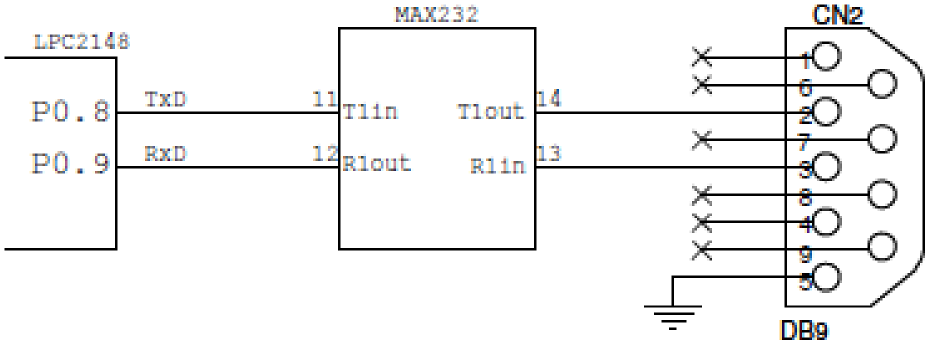
**UART0** is connected to **USB-to-Serial** on connector **CN1**.



**UART-0 interfacing details.**

| UART0<br>(USB-to-Serial)DBGU | Pin Details | |
|---|---|---|
| | **LPC2148** | **Description** |
| RxD0 | P0.1 | UART0 Receive |
| TxD0 | P0.0 | UART0 Transmit |

**UART1** is connected using**DB9**connector-**CN2**. UART1 has **RS232** transceivers.

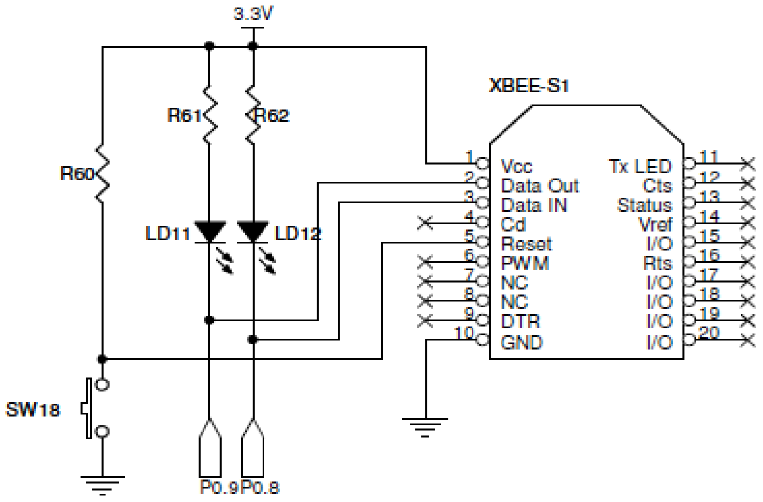


**UART-1 interfacing details.**

| UART1 (COM1) | Pin Details | |
|---|---|---|
| | **LPC2148** | **Description** |
| RxD1 | P0.9 | UART1 Receive |
| TxD1 | P0.8 | UART1 Transmit |

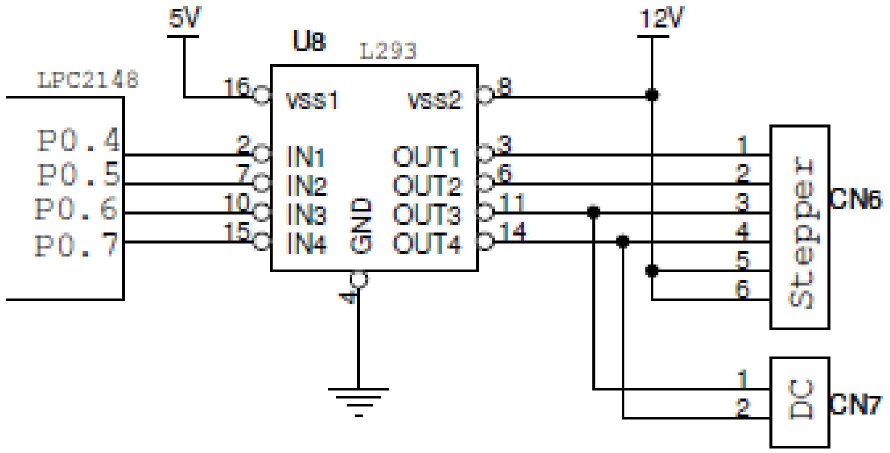
### 2.8.1. Zigbee.

Awireless Zigbee module **XBee-S1**(CN15) is populated on the Micro-A748 board.



### 2.9. Stepper Motor and DC Motor.

A stepper motor driver (L293D) is provided on the Micro-A748 board for the user.
The user can connect the stepper motor using connector **CN6**.
The user can connect the DC motor to connector **CN7**.Pin **P0.7 (PWM 2)** is connected to DC motor and can be used to vary speed of the motor using PWM functionality.
In order to use the motor interface connect switch **SW19** in position **1-2 (MTR_ON)**.

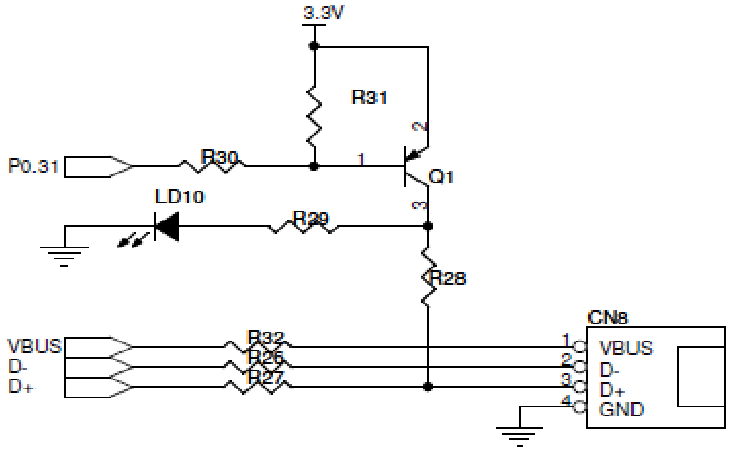


**Motor interfacing details.**

| Stepper Motor | Pin Details | |
|---|---|---|
| | **LPC2148** | **Description** |
| MTR1 | P0.4 | Stepper |
| MTR2 | P0.5 | Stepper |
| MTR3 | P0.6 | Stepper /DC Motor |
| MTR4 | P0.7 | Stepper/DC Motor |

## 2.10. USB Device.

The processor on the Micro-A748 (LPC2148) has on chip USB device controller and driver interface. The Micro-A748 has connected this interface to a USB device connector (USB-B type connector).



**USB Device interfacing details.**

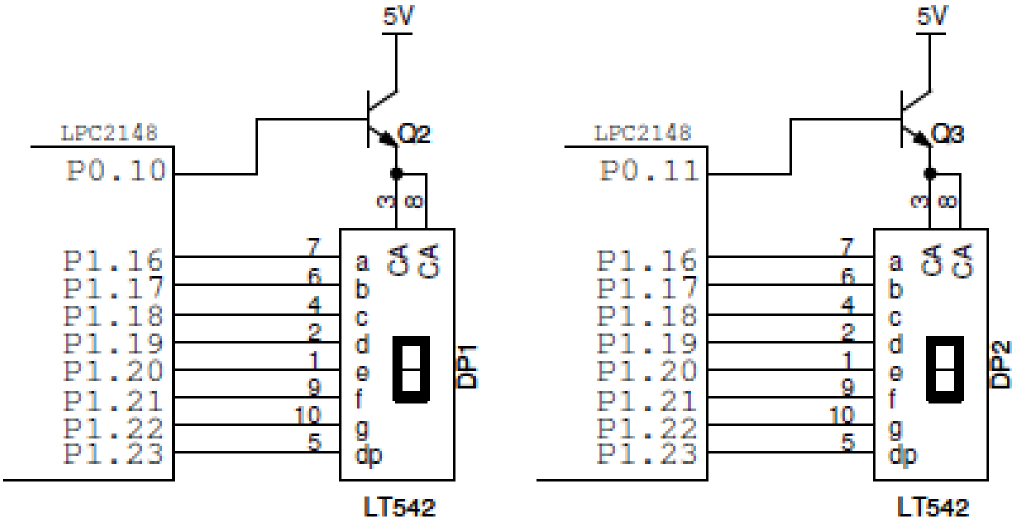
| USB | Pin Details | |
|---|---|---|
| | **LPC2148** | **Description** |
| VBUS | P0.23 | USB Power Line |
| D+ | D+ (10) | Data Bus + |
| D- | D- (11) | Data Bus - |
| USB_CONN | P0.31 | USB Connected signal |
| UP_LED | P1.18 | Link LED |

## 2.11. Seven Segment.

Two Seven segment devices are interfaced on the Micro-A748 Board in multiplexed connection. The Seven-Segment are interfaced to the GPIO pins in common anode configuration.

In order to use the DP1 Seven-Segment you need to set **SEG_SEL1 (P0.10)** to ‘1’ (HIGH) and in order to use the DP2 Seven-Segment you need to set **SEG_SEL2 (P0.11)** to ‘1’ (HIGH).

Note: in order to use Seven-Segment connect switch **SW20** in position **2-3 (7 Seg_ON).**



**Seven Segment interfacing details.**

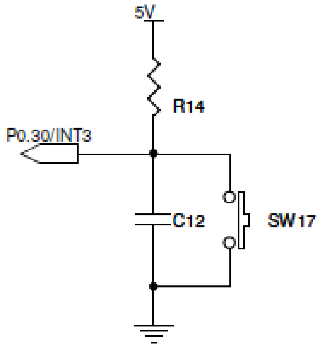
| Seven Segment | Pin Details | |
|---|---|---|
| | LPC2148 | Description |
| A | P1.16 | Seven Segment A |
| B | P1.17 | Seven Segment B |
| C | P1.18 | Seven Segment C |
| D | P1.19 | Seven Segment D |
| E | P1.20 | Seven Segment E |
| F | P1.21 | Seven Segment F |
| G | P1.22 | Seven Segment G |
| DP | P1.23 | Seven Segment DP |
| SEG_SEL1 | P0.10 | Seven segment select 1 |
| SEG_SEL2 | P0.11 | Seven segment select 2 |

## 2.12.Miscellaneous Interfacing.

Other interfacing included on the Micro-A748 board is the **External Interrupt** switch connected to the **P0.30 (EINT3)** pin of the microcontroller.

Pin **PWM5 (P0.21)** from the microcontroller is connected to the LED. The intensity of the LED can be changed by varying the duty cycle using PWM.

A temperature sensor **LM35** is interfaced to the Micro_748 board on pin **P0.29**. The ADC channel **AIN2** is used to read the digital value. LM35 sensor gives 10 mv change for a rise in 1 ºC.



**Miscellaneous interfacing details.**

| General | Pin Details | |
|---|---|---|
| | LPC2148 | Description |
| Interrupt (EINT3) | P0.30 | Interrupt |
| PWM | P0.21 | PWM LED |
| Temperature Sensor | P0.29 | LM35 Temp sensor |

## 2.12. Port Extensions (GPIO Headers).

### 2.12.1 GPIO Header 1

| GPIO Header 1 | Pin Details |
|---|---|
| | LPC2148 |
| CN11.1 | 5 V |
| CN11.2 | NC |
| CN11.3 | P0.8/TxD1 |
| CN11.4 | NC |
| CN11.5 | P0.0/TXD0 |
| CN11.6 | P0.1/RXD0 |
| CN11.7 | P0.2/SCL0 |
| CN11.8 | P0.3/SDA0 |
| CN11.9 | P0.4/SCK0 |
| CN11.10 | P0.5/MISO0 |
| CN11.11 | P0.6/MOSI0 |
| CN11.12 | P0.7/SSEL0 |
| CN11.13 | P0.9/RXD1 |
| CN11.14 | GND |

### 2.12.2 GPIO Header 2

| **GPIO Header 2** | **Pin Details** |
|---|---|
| | **LPC2148** |
| CN12.1 | 5 V |
| CN12.2 | NC |
| CN12.3 | P0.10 |
| CN12.4 | NC |
| CN12.5 | P0.12 |
| CN12.6 | P0.13 |
| CN12.7 | P0.14 |
| CN12.8 | P0.15 |
| CN12.9 | P0.16 |
| CN12.10 | P0.17 |
| CN12.11 | P0.18 |
| CN12.12 | P0.19 |
| CN12.13 | P0.11 |
| CN12.14 | GND |

### 2.12.3 GPIO Header 3

| **GPIO Header 3** | **Pin Details** |
|---|---|
| | **LPC2148** |
| CN13.1 | 5 V |
| CN13.2 | NC |
| CN13.3 | P0.29/AIN2 |
| CN13.4 | NC |
| CN13.5 | P1.16 |
| CN13.6 | P1.17 |
| CN13.7 | P1.18 |
| CN13.8 | P1.19 |
| CN13.9 | P1.20 |
| CN13.10 | P1.21 |
| CN13.11 | P1.22 |
| CN13.12 | P1.23 |
| CN13.13 | P0.30/EINT3 |
| CN13.14 | GND |

### 2.12.4 GPIO Header 4

| GPIO Header 4 | Pin Details |
|---|---|
| | **LPC2148** |
| CN14.1 | 5 V |
| CN14.2 | NC |
| CN14.3 | P0.20/AOUT |
| CN14.4 | NC |
| CN14.5 | P1.24 |
| CN14.6 | P1.25 |
| CN14.7 | P1.26 |
| CN14.8 | P1.27 |
| CN14.9 | P1.28 |
| CN14.10 | P1.29 |
| CN14.11 | P1.30 |
| CN14.12 | P1.31 |
| CN14.13 | P0.28/AIN1 |
| CN14.14 | GND |

## 2.13. Graphic LCD extension Header (CN18).

| Graphical LCD | Pin Details | | |
|---|---|---|---|
| | LPC2148 | Connector CN18 | Description |
| D0 | P1.16 | CN18.11 | GLCD Data Bit 0 |
| D1 | P1.17 | CN18.12 | GLCD Data Bit 1 |
| D2 | P1.18 | CN18.13 | GLCD Data Bit 2 |
| D3 | P1.19 | CN18.14 | GLCD Data Bit 3 |
| D4 | P1.20 | CN18.18 | GLCD Data Bit 4 |
| D5 | P1.21 | CN18.17 | GLCD Data Bit 5 |
| D6 | P1.22 | CN18.16 | GLCD Data Bit 6 |
| D7 | P1.23 | CN18.15 | GLCD Data Bit 7 |
| RESET | P1.24 | CN18.6 | Reset |
| D/I | P1.25 | CN18.5 | Data / Instruction Signal |
| CS1 | P1.26 | CN18.8 | GLCD Chip Select 1 |
| CS2 | P1.27 | CN18.7 | GLCD Chip Select 2 |
| EN | P0.10 | CN18.10 | GLCD Enable Signal |
| R/W | P0.11 | CN18.9 | GLCD Read / Write |
| VCC | | CN18.1, CN18.3, CN18.19 | Power Supply |
| GND | | CN18.2, CN18.4, CN18.20 | Power Supply |

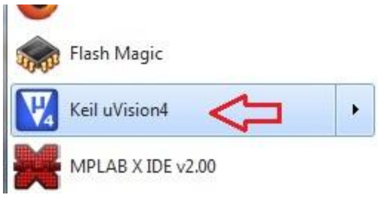
# 3.0. Software: Functional Description and Interfacing.

## 3.1. PC Connection Setup.

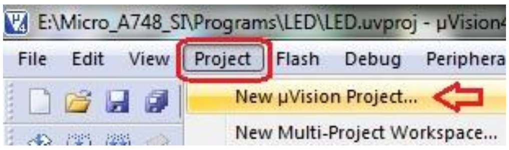
The Micro-A748 Board connects to the PC via the USB to Serial port (CN1). The Hex file generated by the IDE can be downloaded into the LPC2148 flash memory using the ISP feature on the microcontroller via the serial port. The Micro-A748 Board development kit has a USB cable included in the package. Connect the USB-A side of the connector to the PC and the USB-B side to the Micro-A748 Board.

## 3.2. Creating a new Project in Keil IDE.

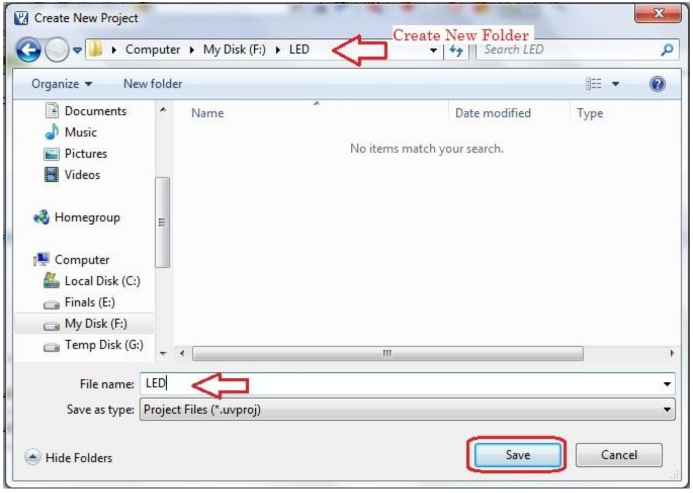
1. Start the KeilµVision 4 IDE.



2. Go to **Project Tab** and select**NewµVision Project.**


E:\Micro_A748_SI\Programs\LED\LED.uvproj - µVision4
File
Edit
View
Project
Flash
Debug
Periphera
New µVision Project...
New Multi-Project Workspace...

3. Store the Project in the directory of your choice and give a name.

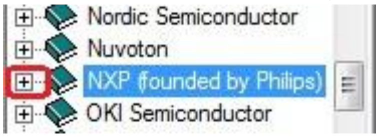


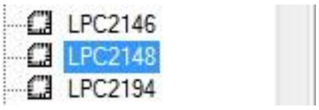
No items match your search.

File name: LED

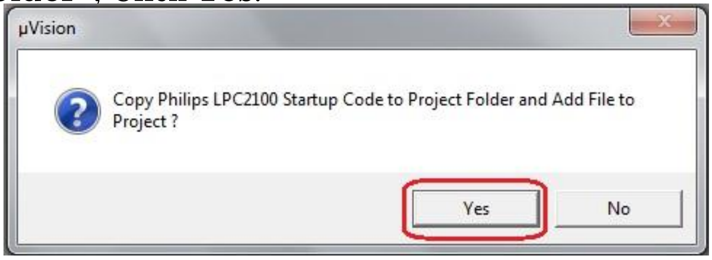
Save as type: Project Files (*.uvproj)

4. Select the Device as LPC2148 (NXP).



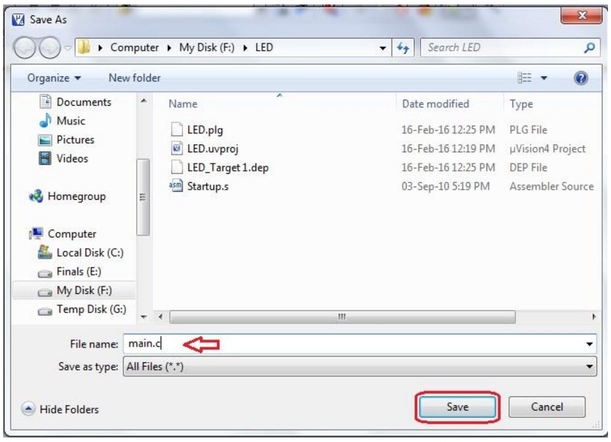


5. It will ask you to **"Copy NXP LPC2148 Startup Code to Project Folder and Add file to Folder"**, Click **Yes**.



Copy Philips LPC2100 Startup Code to Project Folder and Add File to Project ?

6. Go to **File** tab and select **New**. This will be your source file.
7. Go to **File** tab and click **Save As** ->save the fileas **main.c.**(or any other filename with the .c extension)

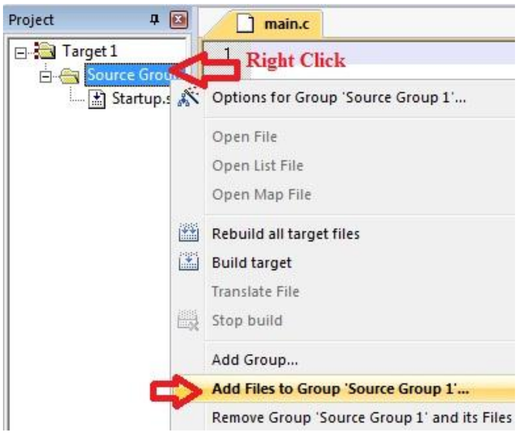


| Name | Date modified | Type |
|---|---|---|
| LED.plg | 16-Feb-16 12:25 PM | PLG File |
| LED.uvproj | 16-Feb-16 12:19 PM | µVision4 Project |
| LED_Target 1.dep | 16-Feb-16 12:25 PM | DEP File |
| Startup.s | 03-Sep-10 5:19 PM | Assembler Source |

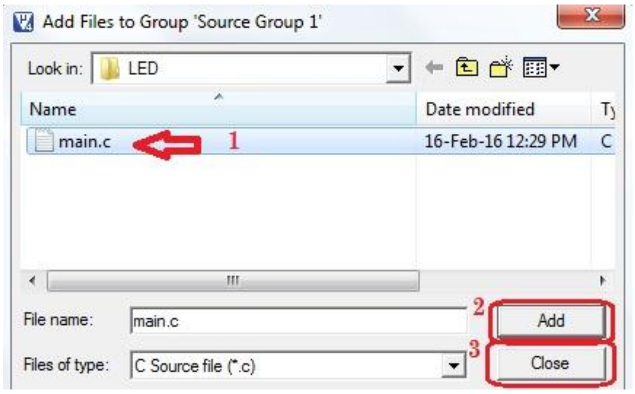
File name: main.c

Save as type: All Files (*.*)

8. Expand **Target1.** Right-click on **Source Group1** and select **Add files to group "Source Group1".**

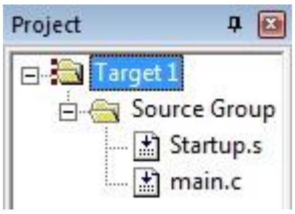


9. Select your source file (**main.c**) then click on **Add,** and then **Close.**

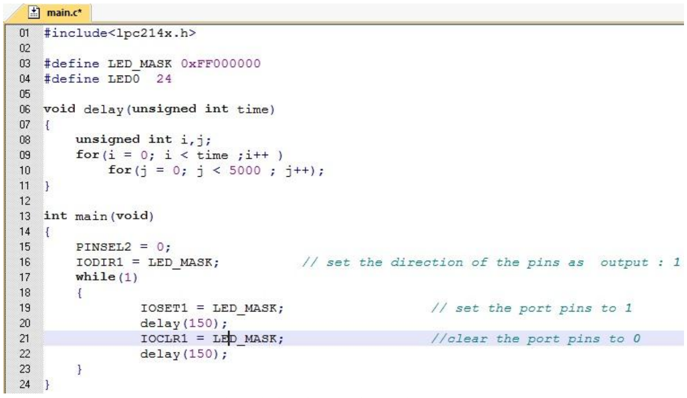


File name: main.c

Files of type: C Source file (*.c)

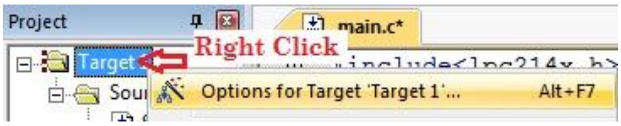


10. Start writing your code.

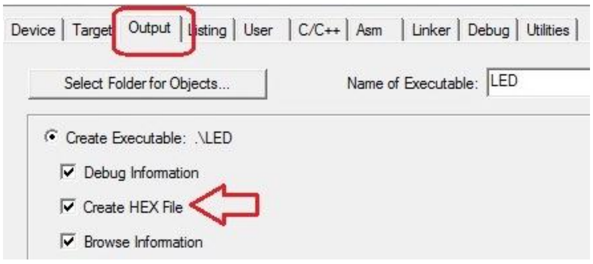


```c
#include<lpc214x.h>

#define LED_MASK 0xFF000000
#define LED0  24

void delay(unsigned int time)
{
    unsigned int i,j;
    for(i = 0; i < time ;i++ )
        for(j = 0; j < 5000 ; j++);
}

int main(void)
{
    PINSEL2 = 0;
    IODIR1 = LED_MASK;          // set the direction of the pins as  output : 1
    while(1)
    {
            IOSET1 = LED_MASK;                  // set the port pins to 1
            delay(150);
            IOCLR1 = LED_MASK;                  //clear the port pins to 0
            delay(150);
    }
}
```

11. Click on **Save.** Then right-click on **Target1**, select **Options for Target1.**



12. Go to **Output** tab and tick on **Create Hex File.** Then go to **Linker** tab and select **Use Memory Layout From Target Dialog**. Then click **OK.**



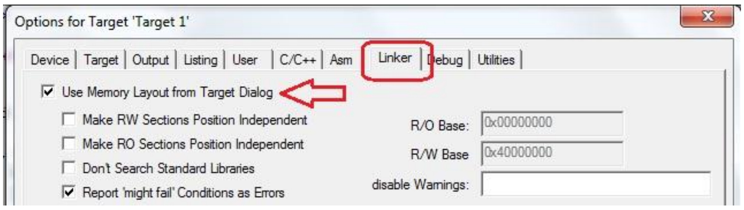
Name of Executable: LED

● Create Executable: .\LED

☑ Debug Information

☑ Create HEX File

☑ Browse Information



☑ Use Memory Layout from Target Dialog

□ Make RW Sections Position Independent

□ Make RO Sections Position Independent

□ Don't Search Standard Libraries

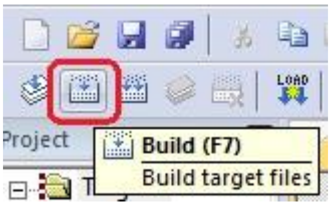
☑ Report 'might fail' Conditions as Errors

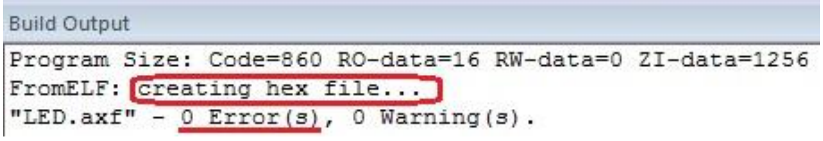
R/O Base: 0x00000000

R/W Base 0x40000000

disable Warnings:

13. Click on **Build** to compile the project and generate **Hex** file. If errors are present, then resolve the errors by double clicking on the error message.

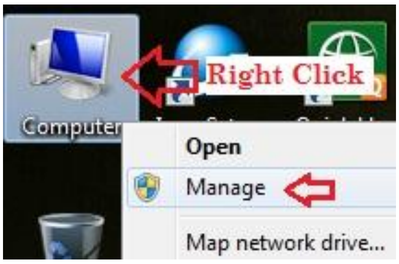




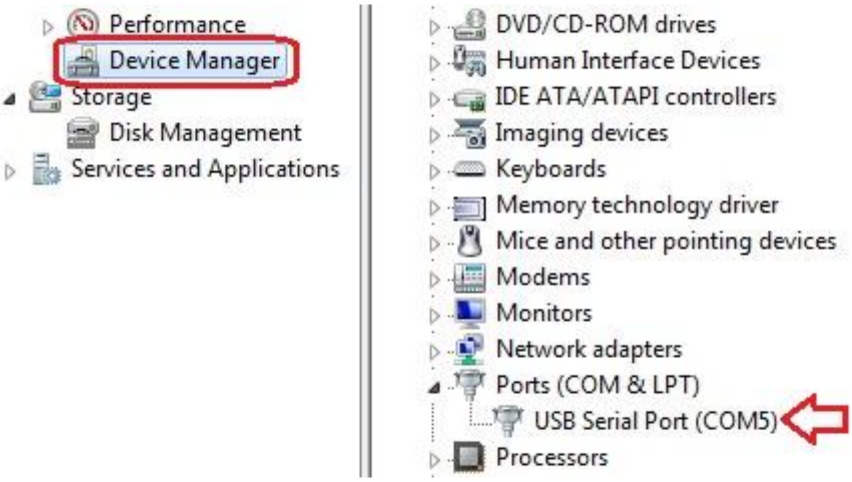
```
Program Size: Code=860 RO-data=16 RW-data=0 ZI-data=1256
FromELF: creating hex file...
"LED.axf" - 0 Error(s), 0 Warning(s).
```

### 3.3. To view the COM port number.

1. Right-click on **My Computer.** Then click on **Manage.**

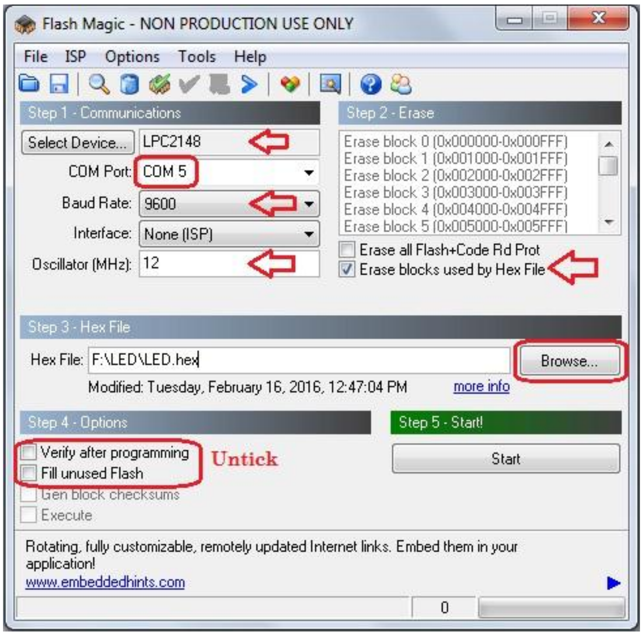


2. Select **Device Manager.** Then expand the **Ports** option and note down the COM port number of the **USB Serial Port.**



### 3.4. Flash Magic: Hex File Flashing Tool.

Flash Magic Software is used to program the microcontroller on the Micro-A748 Board.



Flash Magic - NON PRODUCTION USE ONLY

File ISP Options Tools Help

Step 1 - Communications

Select Device... LPC2148

COM Port: COM 5

Baud Rate: 9600

Interface: None (ISP)

Oscillator (MHz): 12

Step 2 - Erase

Erase block 0 (0x000000-0x000FFF)
Erase block 1 (0x001000-0x001FFF)
Erase block 2 (0x002000-0x002FFF)
Erase block 3 (0x003000-0x003FFF)
Erase block 4 (0x004000-0x004FFF)
Erase block 5 (0x005000-0x005FFF)

□ Erase all Flash+Code Rd Prot
☑ Erase blocks used by Hex File

Step 3 - Hex File

Hex File: F:\LED\LED.hex Browse...

Modified: Tuesday, February 16, 2016, 12:47:04 PM more info

Step 4 - Options

□ Verify after programming
□ Fill unused Flash
□ Gen block checksums
□ Execute

Untick

Step 5 - Start!

Start

Rotating, fully customizable, remotely updated Internet links. Embed them in your application!
www.embeddedhints.com

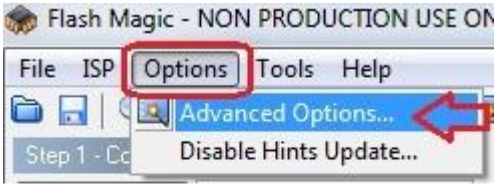
0

The following settingare to be done to correctly use Flash Magic with Micro-A748 board.

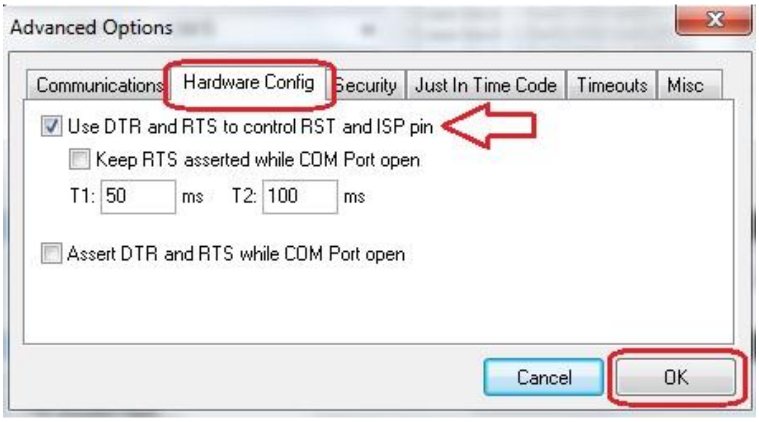
- Select Device as **LPC2148**.
- Select the correct COM Port. (See Section 3.3).
- Set baud Rate of your choice between **9600 - 38400**.
- Interface: **None(ISP)**.
- Oscillator: **12**MHz.
- Check the box: **Erase Blocks Used By hex File**.
- Uncheck the boxes: **Verify after programming** and **Fill Unused Flash.**

### Hardware Settings for Flash Magic.

- Click**Options→ Advanced Options** and go to**Hardware Config** tab.
- Check**USE DTR and RTS to control RST and ISP Pin** option.



Flash Magic - NON PRODUCTION USE ON
File ISP Options Tools Help
Advanced Options...
Disable Hints Update...
Step 1 - Co



Advanced Options

Communications | Hardware Config | Security | Just In Time Code | Timeouts | Misc

☑ Use DTR and RTS to control RST and ISP pin

□ Keep RTS asserted while COM Port open

T1: 50 ms T2: 100 ms

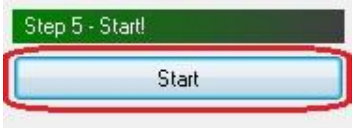
□ Assert DTR and RTS while COM Port open

Cancel OK

### Flashing the Micro-A748 Board.

- Following are the steps to be done while programming the HEX file to the microcontrollerFlash Memory.
- Power "ON" the board using the ON/OFF Switch.

- Browse the Hex file to be downloaded.
- Press Start button on Flash Magic.

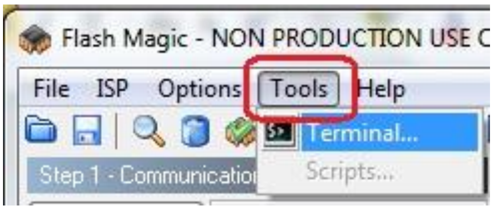


- Check if the program is downloading.
- The program will start running on the board as soon as the download is complete.
- Press the Reset Button on the board to restart running the program on the microcontroller.

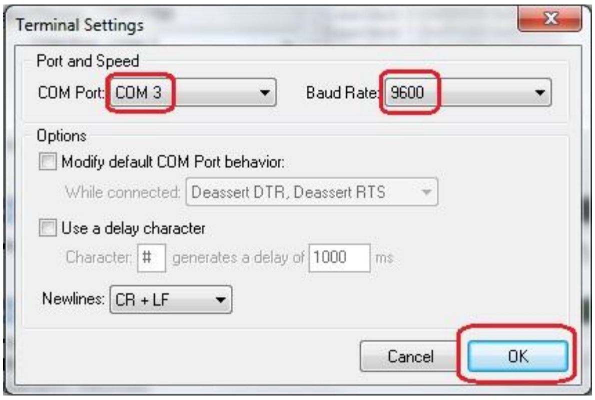
### 3.5. To view output on the terminal.

The output of the program given by using the **printf( )** standard library function can be viewed on the terminal of the PC. To open the terminal, follow these steps:

- In Flash Magic, click on **Tools → Terminal.**



- Select the proper **COM port** and **BaudRate** values. This is the application baudrate and is defined in the program by the UART_init( ) function.


Terminal Settings

Port and Speed

COM Port: COM 3

Baud Rate: 9600

Options

□ Modify default COM Port behavior:

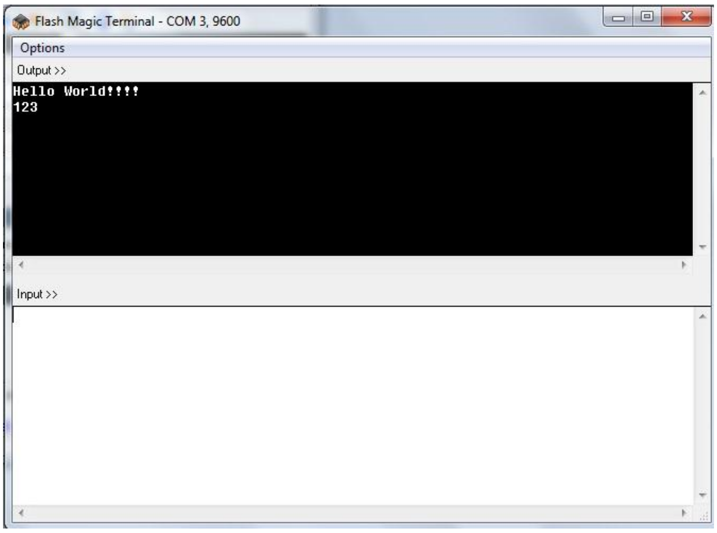
While connected: Deassert DTR, Deassert RTS

□ Use a delay character

Character: # generates a delay of 1000 ms

Newlines: CR + LF

- Click **OK** to open the terminal.


Flash Magic Terminal - COM 3, 9600

```
Hello World!!!!
123
```