SCIENTIFIC CALCULATOR USING EDUARM BOARD

A M Ch Jyothi, N Santoshi
Department of Electronics and communications Engineering, GVPCOE(A), Vizag
malathijyothi@gvpce.ac.in
Santoshineelamsetty@gmail.com

Abstract-- **EduARM** *is an ARM Cortex-M3 core LPC1768 educational development board. It is suitable for beginner, intermediate and advanced embedded developers. EduARM has an extensive set of peripherals that allow users to design and build various applications. The most exciting feature of the board is the 2.8" touch screen TFT display. Scientific calculator is implemented using the EduARM development board and prime framework (suite of software libraries). The calculator will support various functions similar to a basic scientific calculator such as: basic arithmetic, square root, percentage, inverse, power, factorial, logarithmic, exponential & trigonometric.*

Keywords-- **Cortex-M3, EDUARM, LPC1768, TFT display, prime framework**

1. INTRODUCTION

EduARM is an ARM Cortex-M3 core LPC1768 educational development board. It is suitable for beginner, intermediate and advanced embedded developers. EduARM has an extensive set of peripherals that allow users to design and build various applications. The most exciting feature of the board is the 2.8" touch screen TFT display.The board may be powered by 5V external power supply or using USB. Along with these on-board peripherals, EduARM has 2 auxiliary ports which has variety of peripherals like GPIO, PWM, ADC, UART, I2C and external interrupt which allows the user to extend its functionality. Scientific calculator is implemented using the EduARM development board and prime framework (suite of software libraries). The calculator will support various functions similar to a basic scientific calculator such as: basic arithmetic, square root, percentage, inverse, power, factorial, logarithmic, exponential & trigonometric.

2. EDUARM

A. EduARM Kit contains:

- EduARM development tool consisting of EduARM development board, TFT LCD and ABS casing.
- Prime Framework, a suite of software libraries for faster application development.
- 4GB SD Card for ample data storage.
- Stylus for using touch screen.
- Micro USB Cable for programming and powering support.
- 5V/1A Power Adapter for power supply.

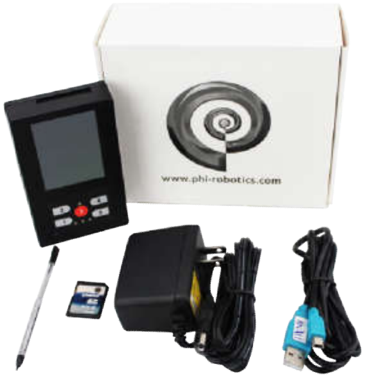


Figure 1: EDUARM KIT

B. Hardware/Software Requirements

The hardware/software required to execute the project:

- EduARM development board
- USB Type A to Mini B cable
- 5V DC adapter
- Stylus
- Seed Project
- MinGW
- ARM Tool chain
- Flash Magic
- FTDI USB-Serial Driver
- Programming using C

C. Specifications

- Microcontroller: LPC1768 (ARM Cortex-M3)
- On-board crystal for microcontroller: 12MHz
- On-board crystal for on-chip RTC: 32.768 KHz
- Input power supply options: USB, battery and 5V adapter
- TFT display: 320x240 pixels with real 262,144 colors display with ILI9320 display controller
- Touch screen controller: ADS7843
- Audio codec: SGTL5000
- Audio codec sampling frequency: 8 kHz to 96 KHz
- Accelerometer: 3 Axis digital output

D. Setting up the Environment:

1. Seed Project
2. Setting workspace
3. Compile the Seed Project
4. Burn the Seed Project onto the microcontroller

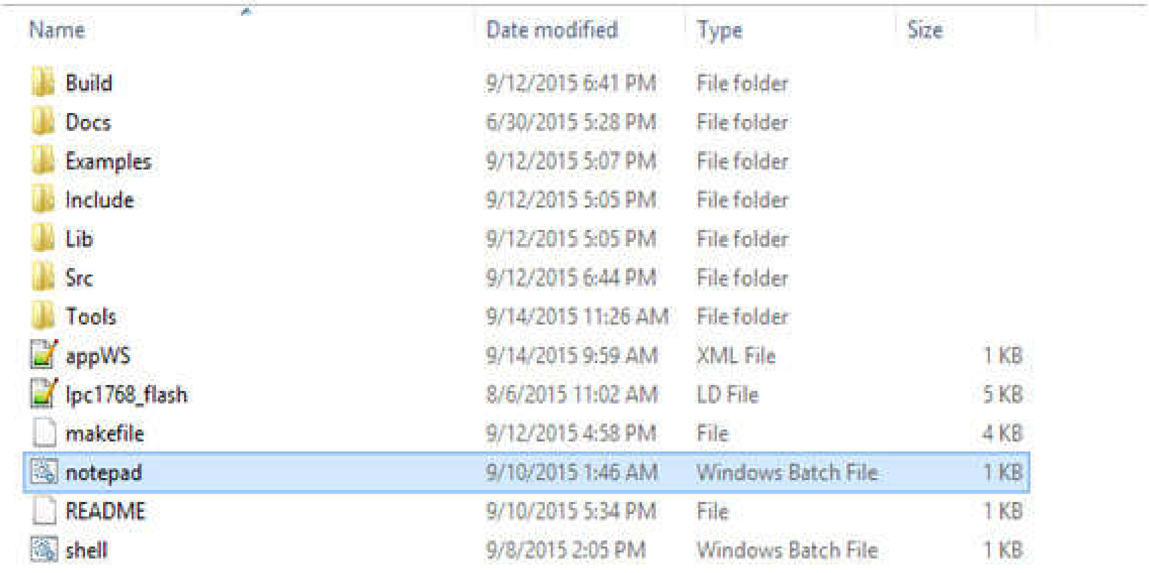
1. Seed Project

The seed project gives you a project template, containing various folders and files; that serve as a substantial starting point to build and manage your project. The seed project you have received includes the following folders:

- Build**:** The compiled program will be stored in this folder.
- Docs**:** Documentation will be stored in this folder
- Examples**:** Any example code provided by Phi-education will be stored in this folder
- Lib**:** The libraries and APIs that are provided by Phi Education will be stored in this folder
- Include**:** The header files will be stored in this folder
- Src**:** The source files for your project will be stored in this folder. We provide a file named app.c to help you start your development. This application only prints "Welcome to Phi Education" on the LCD. Your project code will go in this file.
- Tools**:** Contains various tools required for developing and building the project.

2. Setting workspace

- Open the project folder and double click on "notepad.bat".



Name	Date modified	Type	Size
Build	9/12/2015 6:41 PM	File folder	
Docs	6/30/2015 5:28 PM	File folder	
Examples	9/12/2015 5:07 PM	File folder	
Include	9/12/2015 5:05 PM	File folder	
Lib	9/12/2015 5:05 PM	File folder	
Src	9/12/2015 6:44 PM	File folder	
Tools	9/14/2015 11:26 AM	File folder	
appWS	9/14/2015 9:59 AM	XML File	1 KB
lpc1768_flash	8/6/2015 11:02 AM	LD File	5 KB
makefile	9/12/2015 4:58 PM	File	4 KB
notepad	9/10/2015 1:46 AM	Windows Batch File	1 KB
README	9/10/2015 5:34 PM	File	1 KB
shell	9/8/2015 2:05 PM	Windows Batch File	1 KB

Figure 2: notepad.bat file in seed project

- Notepad++ will open as shown in Figure 03.

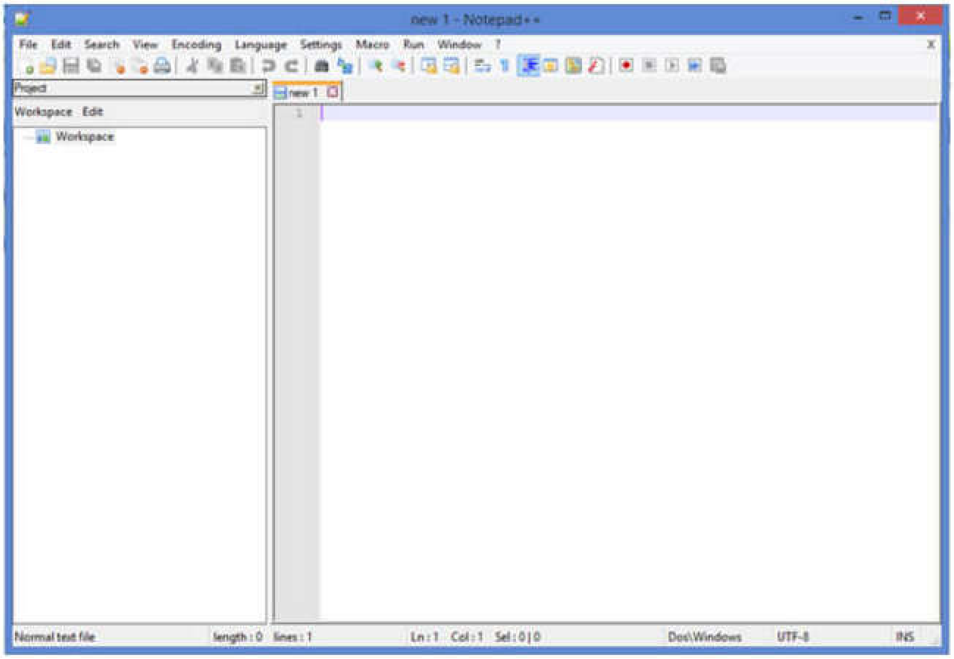
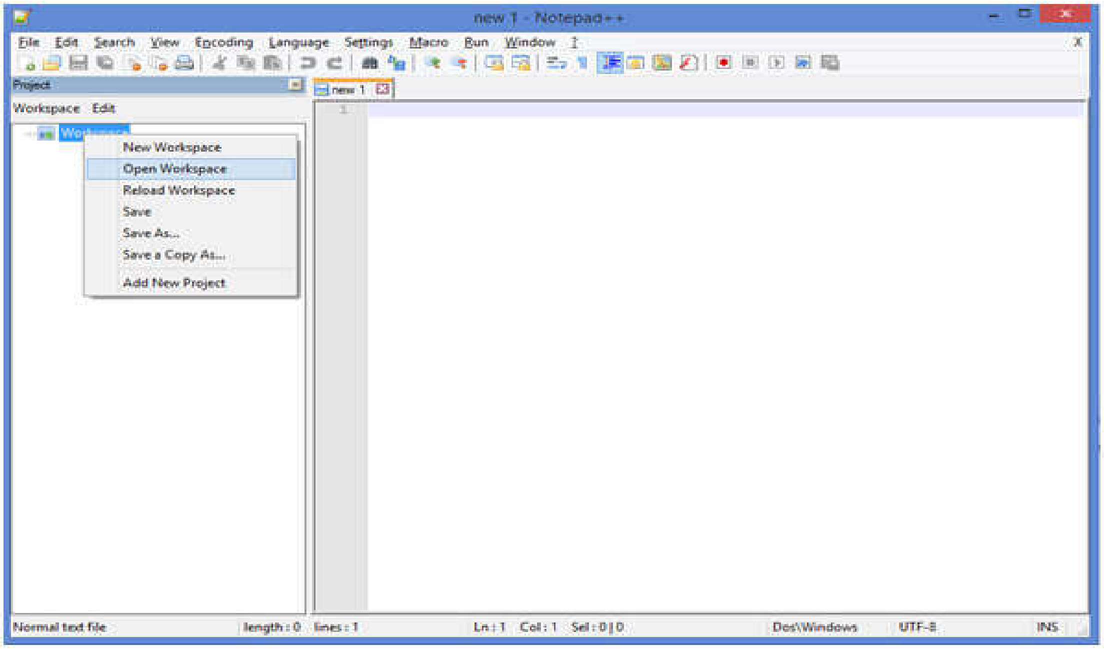


Figure 3: Workspace in notepad++

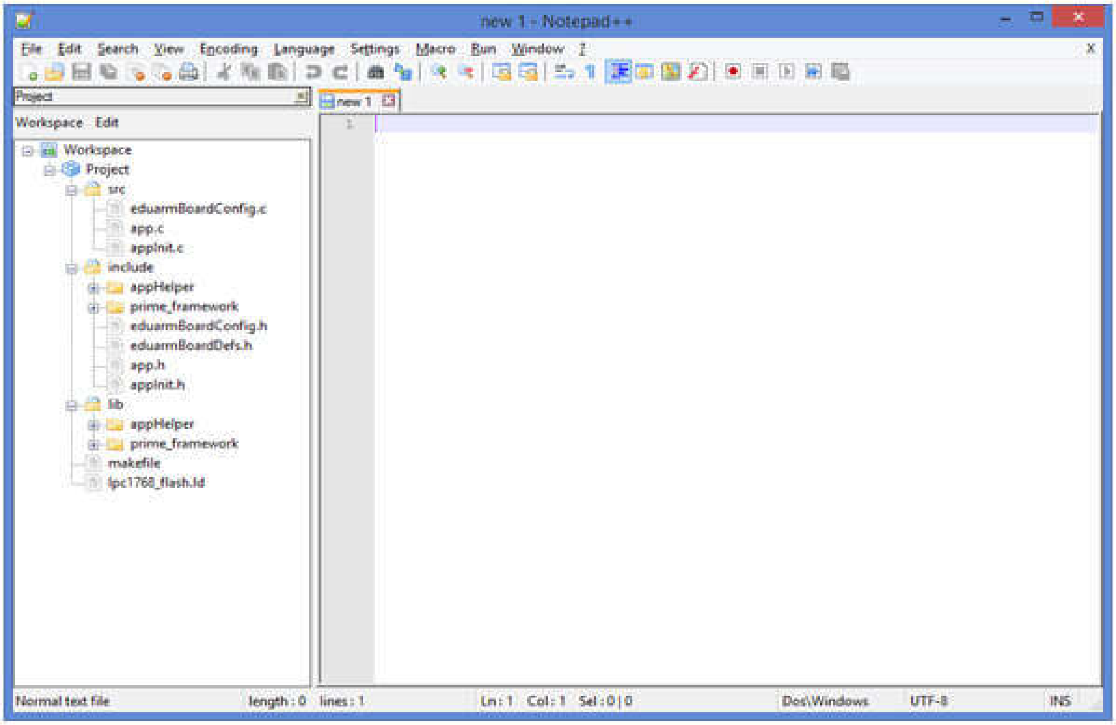
- If the project is not opened, Right click **"Workspace"**. Click **"Open Workspace"**as shown in Figure 04.



Normal text file length : 0 lines : 1 Ln : 1 Col : 1 Sel : 0 | 0 Dos\Windows UTF-8 INS

Figure 4: Right click on Workspace

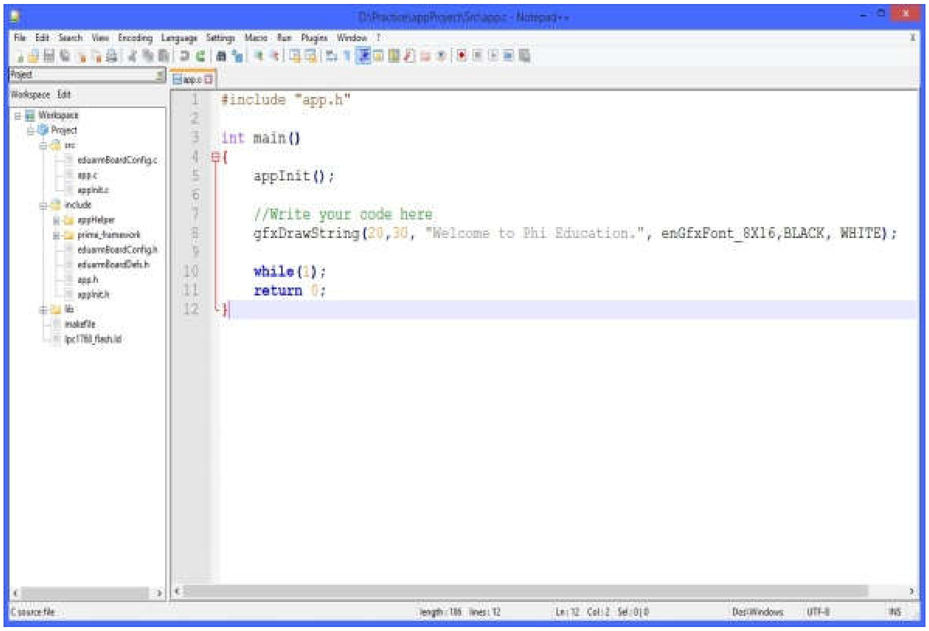
- Select **"appWS.xml"** file from seed project and click open. Project folder appears. On expanding Project folder, contents as shown in Figure 5 are displayed.



Normal text file length : 0 lines : 1 Ln : 1 Col : 1 Sel : 0 | 0 Dos\Windows UTF-8 INS

Figure 5: Project Folder Structure

- Open the **"app.c"**. Start writing your code here.



```
#include "app.h"

int main()
{
    appInit();

    //Write your code here
    gfxDrawString(20,30, "Welcome to Phi Education.", enGfxFont_8X16,BLACK, WHITE);

    while(1);
    return 0;
}
```

Figure 6: app.c file opened for editing

- **Programming in C:**

  These are different snippets for writing the program code

  Code Snippet 1: Writing a character with a Specific Font and Color at a Specific Position on the LCD screen

```
PFEnStatus gfxDrawChar(PFdword x, PFdword y, PFbyte ch, PFEnGraphicFonts fontSize, PFdword fontColor, PFdword backColor);
```

Code Snippet 2: Writing a String with a Specific Font and Color at a Specific Position on the LCD screen.

```
PFEnStatus gfxDrawString(PFdword x, PFdword y, PFchar *string,EnGraphicFonts fontSize, PFdword fontColor, PFdword backColor);
```

x,y : (x,y)coordinate of pixel to place the character on the display.

Ch**:** Character to be displayed.

fontSize**:** Font size of the character.

font Colour**:** Colour of the character.

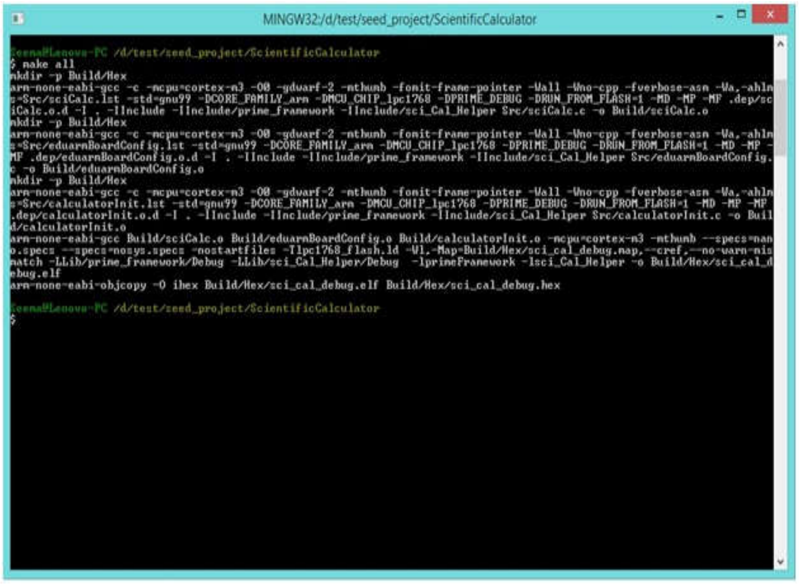
back Colour**:** Background colour for the character.

Code Snippet 3:

```
PFEnBoolean touchAvailable(PFdword *xPos, PFdword *yPos);
```

(xPos**.** yPos**):**On return, it holds the(X, Y)coordinate of where the touch panel was touched.

- *Compiling the seed project:*
- Go to that location where you extracted the seed project and double click batch file named shell.bat
- At the command prompt, type the make all command and press ENTER.



MINGW32:/d/test/seed_project/ScientificCalculator

```
Seema@Lenovo-PC /d/test/seed_project/ScientificCalculator
$ make all
mkdir -p Build/Hex
arm-none-eabi-gcc -c -mcpu=cortex-m3 -O0 -gdwarf-2 -mthumb -fomit-frame-pointer -Wall -Wno-cpp -fverbose-asm -Wa,-ahlm
s=Src/sciCalc.lst -std=gnu99 -DCORE_FAMILY_arm -DMCU_CHIP_lpc1768 -DPRIME_DEBUG -DRUN_FROM_FLASH=1 -MD -MP -MF .dep/sc
iCalc.o.d -I . -IInclude -IInclude/prime_framework -IInclude/sci_Cal_Helper Src/sciCalc.c -o Build/sciCalc.o
mkdir -p Build/Hex
arm-none-eabi-gcc -c -mcpu=cortex-m3 -O0 -gdwarf-2 -mthumb -fomit-frame-pointer -Wall -Wno-cpp -fverbose-asm -Wa,-ahlm
s=Src/eduarmBoardConfig.lst -std=gnu99 -DCORE_FAMILY_arm -DMCU_CHIP_lpc1768 -DPRIME_DEBUG -DRUN_FROM_FLASH=1 -MD -MP -
MF .dep/eduarmBoardConfig.o.d -I . -IInclude -IInclude/prime_framework -IInclude/sci_Cal_Helper Src/eduarmBoardConfig.
c -o Build/eduarmBoardConfig.o
mkdir -p Build/Hex
arm-none-eabi-gcc -c -mcpu=cortex-m3 -O0 -gdwarf-2 -mthumb -fomit-frame-pointer -Wall -Wno-cpp -fverbose-asm -Wa,-ahlm
s=Src/calculatorInit.lst -std=gnu99 -DCORE_FAMILY_arm -DMCU_CHIP_lpc1768 -DPRIME_DEBUG -DRUN_FROM_FLASH=1 -MD -MP -MF
.dep/calculatorInit.o.d -I . -IInclude -IInclude/prime_framework -IInclude/sci_Cal_Helper Src/calculatorInit.c -o Buil
d/calculatorInit.o
arm-none-eabi-gcc Build/sciCalc.o Build/eduarmBoardConfig.o Build/calculatorInit.o -mcpu=cortex-m3 -mthumb --specs=nan
o.specs --specs=nosys.specs -nostartfiles -Tlpc1768_flash.ld -Wl,-Map=Build/Hex/sci_cal_debug.map,--cref,--no-warn-mis
match -LLib/prime_framework/Debug -LLib/sci_Cal_Helper/Debug  -lprimeFramework -lsci_Cal_Helper -o Build/Hex/sci_cal_d
ebug.elf
arm-none-eabi-objcopy -O ihex Build/Hex/sci_cal_debug.elf Build/Hex/sci_cal_debug.hex

Seema@Lenovo-PC /d/test/seed_project/ScientificCalculator
$
```

Figure 7: make all

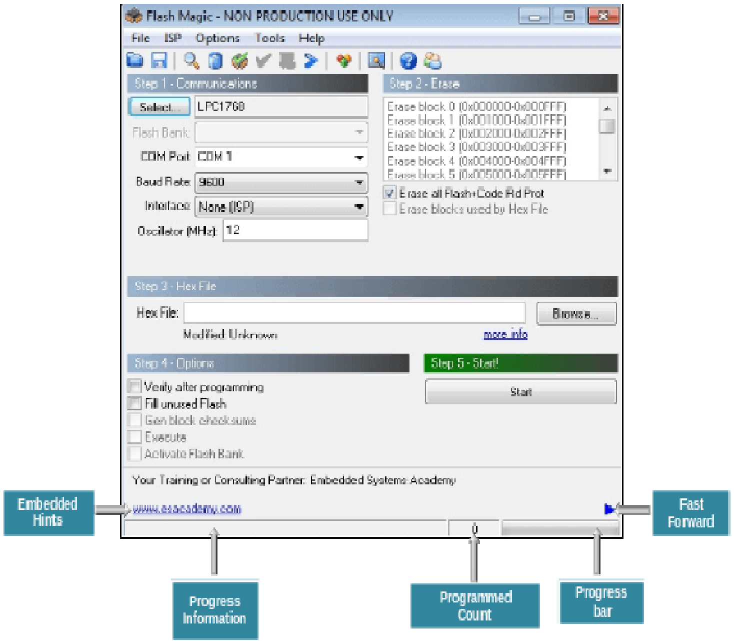
The hex file app_debug.hex is generated in the Build/Hex directory or folder and is ready to be burned on the microcontroller.

For more information about debugging, refer to the Debugging the Project section.

- Note: You can run the make clean command at the command prompt if you want to delete all the previous hex code and object files.
- When the project is built, serial debugging is enabled by default. To disable debugging when building the project, run the make PRIME_DEBUG=0 all command. This command creates a hex file named app_release.hex

Burning the seed project

Now you need to burn the hex file you generated in the previous section on to EduArm. To do so you can refer the lesson Burning the hex file onto EduARM section. Remember to load the newly generated hex file and not the demo project



LPC1768
Flash Bank:
COM Port: COM 1
Baud Rate: 9600
Interface: None (ISP)
Oscillator (MHz): 12

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

Hex File:
Modified: Unknown

☐ Verify after programming
☐ Fill unused Flash
☐ Gen block checksums
☐ Execute
☐ Activate Flash Bank

Your Training or Consulting Partner: Embedded Systems Academy
www.esacademy.com
0

Figure 8: burning the seed project

On the Flash Magic Interface:

- The progress messages are displayed at the bottom left.
- The progress bar is located at the bottom right.
- In between the progress messages and progress bar, the program count is displayed.
- The Programmed Count displays the number of times the hex file has been programmed since it was last modified or selected.
- The Embedded Hints are displayed above the progress information. Embedded links are rotating links that allow you to click and browse web pages. Click the Fast forward icon to quickly browse through the links.

3. Using the EDUARM

To use EduARM, you will need to power up EduARM by connecting it to the 5V DC power adapter and switching on the device. Once EduArm is powered up, the sample scientific calculator application that you have burnt on to EduARM will start.

A. Using the sample handheld Scientific Graphic Calculator:

The handheld scientific graphing calculator user interface appears as shown in Figure 9a and 9b:

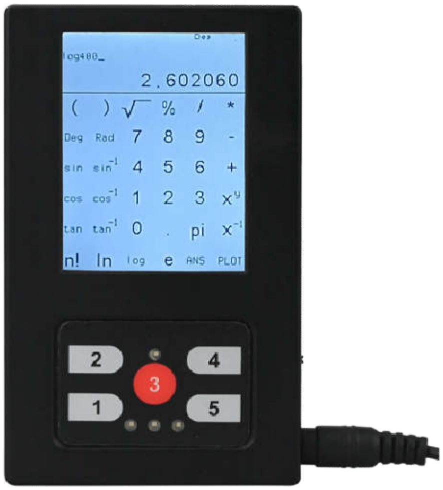


Figure 9a: Handheld Scientific Graphing Calculator User Interface – Normal Calculator Window

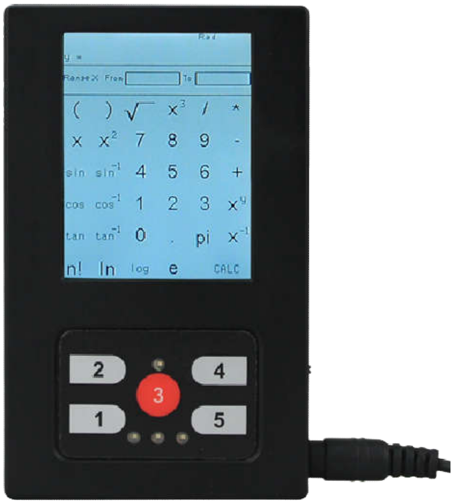


Figure 9b: Handheld Scientific Graphing Calculator User Interface – Graph Plotting Window

B. All the buttons on the user interface are similar to any scientific calculator, except for the following buttons:

- ANS**:** Use to get answer of previous calculation.
- PLOT**:** Use to switch to the window which prompts to input an equation for plotting the graph.
- CALC**:** Use to switch to normal calculator window. When the PLOT button is pressed, it changes to CALC button.

C. EduARM also has the following keys (shown in figures above) that you need to use to work with the calculator:

- Key 2**:** Use to clear the input section. It also clears previous answer stored in the ANS key.
- Key 3**:** Use to evaluate an expression. This key is also used to navigate to the next region in the graph input section, such as from equation input region to X-lower range input region and from X-lower range input region to X-upper range input region and display the graph.
- Key 4**:** Use to delete the last input character.

4. CONCLUSIONS

Scientific calculator is implemented using the EduARM development board and prime framework (suite of software libraries). The calculator performs various basic functions such as basic arithmetic, square root, percentage, inverse, power, factorial, logarithmic, exponential & trigonometric.

This project can further enhanced by adding additional programs for differentiation, definite integrals and number system conversions such as binary to octal etc. We can make this calculator visually more impressive by adding images of buttons instead of drawing them and adding a zoom feature to a graph for better understanding of the function domain.

ACKNOWLEDGMENT

We are thankful to technical team of phi-education society for their support. We are especially thankful to Dr.N Bala subramanyam, professor, ECE dept, GVPCOE (A), Visakhapatnam for providing support and facilities.

REFERENCES

1. *Phi_education.com/store/index.php?route=product/ pro uct&product_id=118.*
2. *LPC1768 User Manual NXP Semiconductors-UM10360.*
3. *Design and Development of Sensor-Based Mini Projects for Embedded System Laboratory Using ARM Cortex-M3(LPC1768).*
4. *SYSTEM IDENTIFICATION USING CHAOS THEORY ON ARM CORTEX M3 PROCESSOR DEVELOPMENT BOARD.*
5. *Aruna.kommu and Raghavendra Rao Kanchi, "ARM based Temperature Measurement and Processing to Remote Computer using Fiber Optic Cable." IEEE International Conference on Communication and Signal Processing", (ICCSP), 2013, April 3-5.*
6. *J. Yiu, "The Definitive Guide to ARM Cortex-M0". USA, Elsevier Science, (2011).*