Laboratory 4
Dr. Don Evans
CSE 2240 –Spring_2017_Lab_Assignments
Assembly Language Programming and Machine Organization
TA: Naseer Jan
Junkins 215

Section N11 - Tu 12:00PM to 1:50PM Junkins 215
Section N12 - Th 3:00PM - 4:50PM Junkins 215
Section N13 - Tu 8:00AM - 9:50AM Junkins 217

Laboratory 4: LED blinking through user specified numbers (done in lab#3), lookup table and innovative ideas

Make sure that your lab#3 works correctly. Start work on lab-4. Students are required to complete step II and step III before their lab sessions without evaluation boards. You may use evaluation boards to test your program during lab session. So that TA could update grade during lab session

Note: Students can work in a group for assigned task regarding innovative idea only.

STEP I : Board and project setup

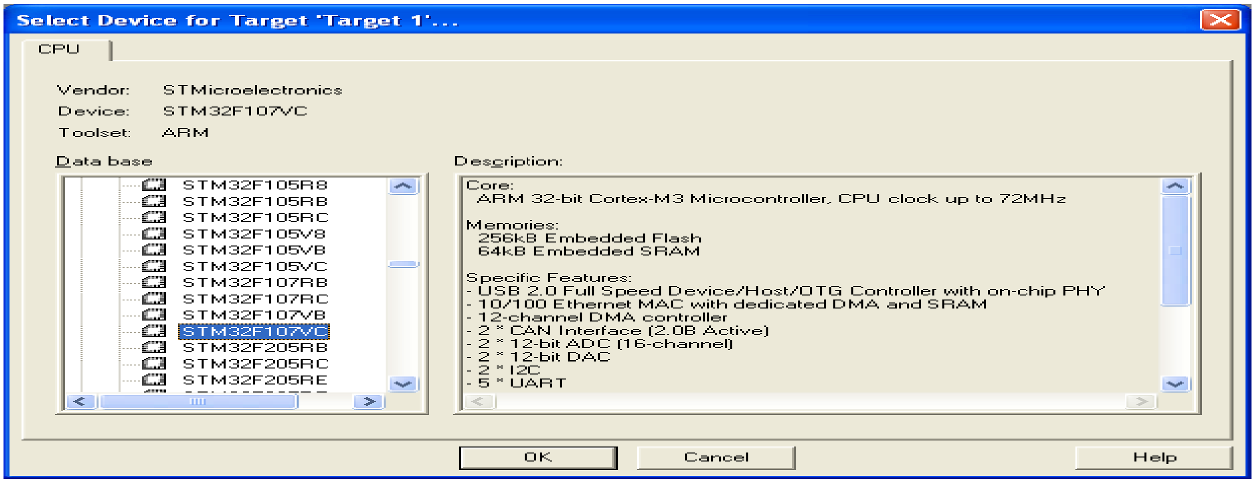
a) Open the ARM evaluation board user's guide
http://www.keil.com/support/man/docs/mcbstm32c/default.htm
b) Follow steps given in sub-topic 'setup' to setup your board
The MCBSTM32 board requires a power connection and a ULINK-USB adapter between your development PC and the JTAG connector.
To use the MCBSTM32C Evaluation Board with ULINK, you must:

- Connect the ULINK adapter to the JTAG connector of the MCBSTM32C board
- Connect the ULINK adapter to the PC via a standard USB cable
- Connect power from your PC to the board using either a USB A to Micro A or USB A to Micro B cable

c) Create new project under **Project → New → µVision Project**, name it 'led' and save it under project directory. Select the device STM32F107VC under STMicroelectronics. DO NOT add the default startup file. The default startup file is meant to work with C-programs. We have edited the default startup file so that it works with ASM code.

Device selection

Select the microcontroller from the Device Database
Vendor: STMicroelectronics
Device: STM32F107VC



```
Vendor:   STMicroelectronics
Device:   STM32F107VC
Toolset:  ARM

Data base
  STM32F105R8
  STM32F105RB
  STM32F105RC
  STM32F105V8
  STM32F105VB
  STM32F105VC
  STM32F107RB
  STM32F107RC
  STM32F107VB
  STM32F107VC
  STM32F205RB
  STM32F205RC
  STM32F205RE

Description:
Core:
 ARM 32-bit Cortex-M3 Microcontroller, CPU clock up to 72MHz

Memories:
 256kB Embedded Flash
 64kB Embedded SRAM

Specific Features:
- USB 2.0 Full Speed Device/Host/OTG Controller with on-chip PHY
- 10/100 Ethernet MAC with dedicated DMA and SRAM
- 12-channel DMA controller
- 2 * CAN Interface (2.0B Active)
- 2 * 12-bit ADC (16-channel)
- 2 * 12-bit DAC
- 2 * I2C
- 5 * UART
```

d) Please check the appendix to verify if your project settings are correct.
e) Create groups 'startup' and 'source' to organize all project files
f) Copy the file ***system_stm32f10x_cl.s*** in the project directory and in Keil add under group 'system'
g) Add a new source file under 'source'. You can now start writing assembly codes for your lab.

STEP II: Lookup table

Majority of students are done with Lab-3 blinking LED. Lab-3 enables you that how to control LEDs, LEDs ON/OFF and introduce delay among LEDs. For this lab, use **lab-3 source code** and complete following tasks which are as follows:

In this step, you will have to perform check between 1 and 8 on the MCBSTM32C LEDs. Your program contains a lookup table in data memory that contains the appropriate values to display from 1 to 8. You need to use loop that searches for value in look up table and then display then turn on LEDs according to the value. For example, if program finds value 4 in look up table then it should turn on LED8, LED9, LED10 and LED10. If your program finds 0 it should turn off all LEDs. Moreover, if your program finds value greater than 8 then it should turn off all LEDs.

Later, implement this idea by using character values such as A, B, C, D, E, F, G, H etc.,

STEP III: Innovative Ideas:

Since we haven't covered sound effects. So visual representations in form of lighting would be an appropriate task for innovative ideas. Different **innovative ideas demo** will be presented to students during lab session. You may choose any option from the following as shown below or you may think about any innovative idea.

i. Ripple effects: Since we will be using much of the source code, use the process to turn on the middle two LEDs (i.e. LED-11 and LED-12) for certain time then turn on the light to the left (i.e. LED-10) and right (led-13) of the middle simultaneously and turn off middle two, and vice versa. This will create ripples starting in the middle and rushing outwards
ii. Convergence effects to a significant point
iii. Morse code: https://en.wikipedia.org/wiki/Morse_code

iv. Switch between odd and even: Turn on LED-8, LED-10, LED-12, LED-14 then turn off LED-8, LED-10, LED-12, LED-14 simultaneously, Turn on: LED-9, LED-11, LED-13, LED-15 then turn off LED-9, LED-11, LED-13, LED-15 simultaneously

v. Any an artistic piece for example, ;LEDs to the tune of "Twinkle, Twinkle Little Star"

vi. Simple Palindrome

vii. Traffic light sequence

viii. Longest palindrome sub sequence

ix. Leap Year

x. Odd and Even programs

xi. An ARM Assembly program that asks the user for a positive integer x and sum up the integers from 1 to x

xii. Factorial of the positive integer

xiii. Longest alphabetically increasing sub-sequence

xiv. the greatest common divisor GCD by using the following different ways such as : Euclid algorithm for the Greatest Common Divisor (gcd), Binary normalization shift-and-subtract, Binary parity “right-normalize” and subtract, “Ternary parity right-normalize” and subtract using decimal

xv. Matching substring in a binary substring

xvi. The power of two and three in 30!.

xvii. The prime factorization of 20!

xviii. Number less than and relatively prime to 28

xix. Etc.,

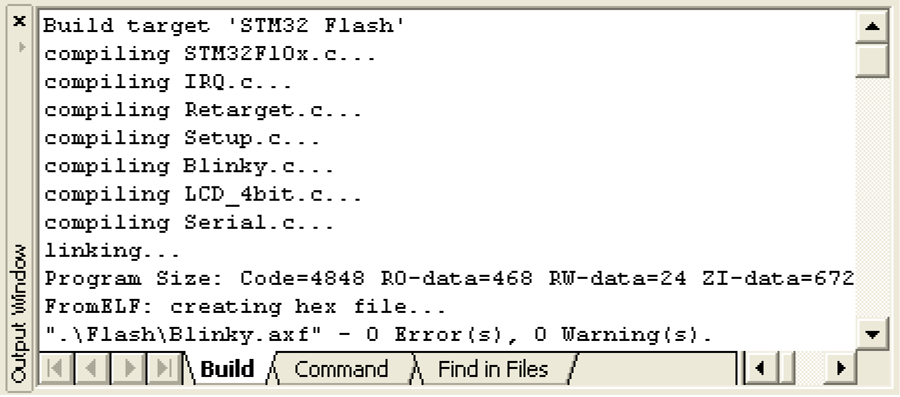
xx. Etc.,

Build and Download to Flash

Build the Traffic example program following this procedure. The executable files are placed in an output folder, ready for downloading.

When you are ready to compile and link your project, use the Build Target command from the Project menu or the Build toolbar. On the Build toolbar, select the appropriate target for your program.

µVision translates and links the source files and creates an absolute object module that you may load into the µVision debugger for testing. The status of the build process displays in the **Build** page of the **Output Window**



```
Build target 'STM32 Flash'
compiling STM32F10x.c...
compiling IRQ.c...
compiling Retarget.c...
compiling Setup.c...
compiling Blinky.c...
compiling LCD_4bit.c...
compiling Serial.c...
linking...
Program Size: Code=4848 RO-data=468 RW-data=24 ZI-data=672
FromELF: creating hex file...
".\Flash\Blinky.axf" - 0 Error(s), 0 Warning(s).
```

Click the **Download to Flash** toolbar button. This executes the Flash download program (**Target Driver** or **External**) selected in **Project — Options for Target — Utilities** and downloads the application program into the STM32F10x device

Start the Debugger

Use the Start/Stop Debug Session toolbar button to start debugging the program.

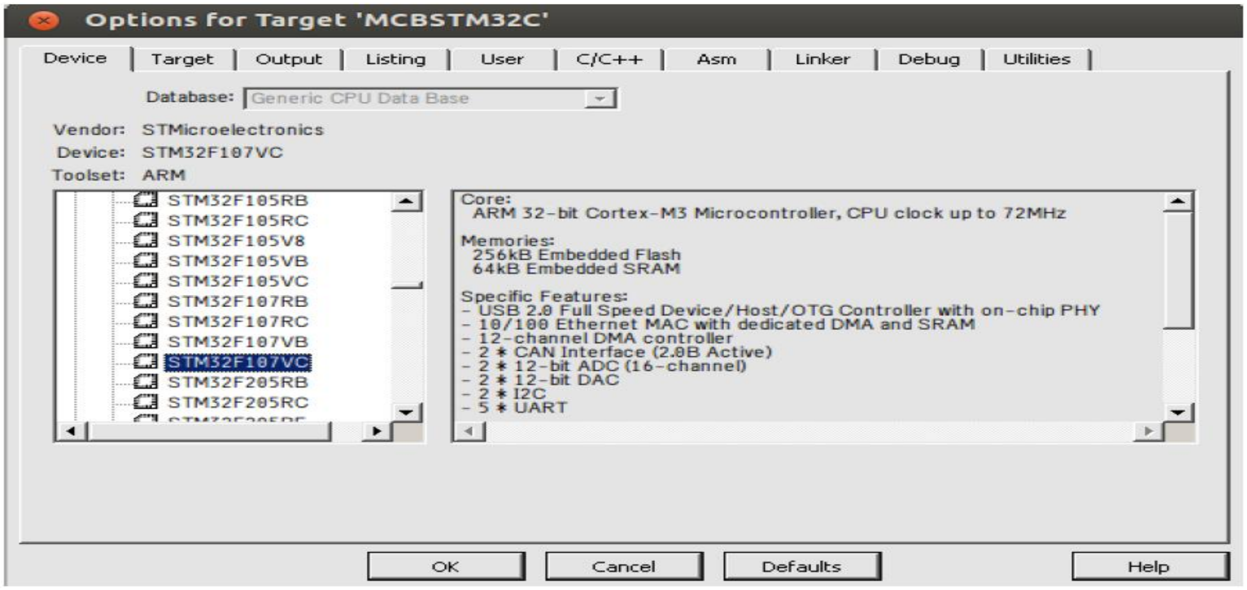
Or

Use the µVision command, Debug — Start/Stop Debug Session, to start debugging the program

Demonstrate result to the lab instructor

APPENDIX

Select STM32F107VC under devices.



Options for Target 'MCBSTM32C'

Database: Generic CPU Data Base

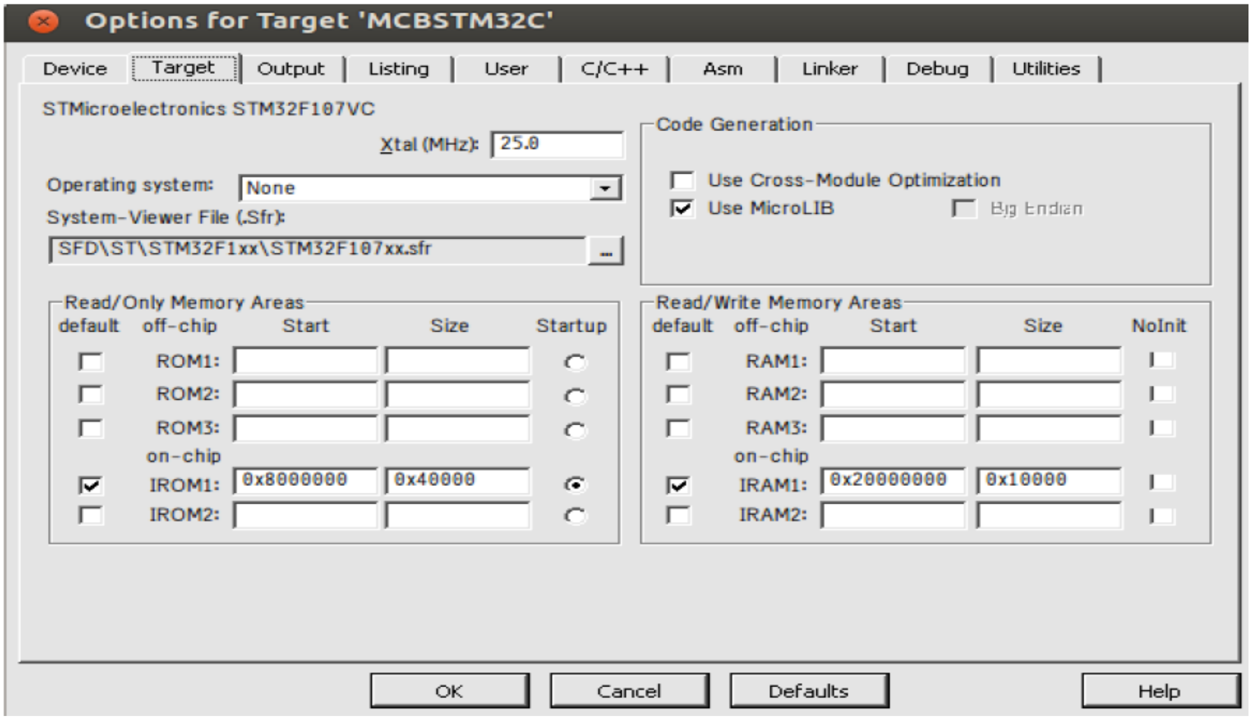
Vendor: STMicroelectronics
Device: STM32F107VC
Toolset: ARM

```
Core:
 ARM 32-bit Cortex-M3 Microcontroller, CPU clock up to 72MHz

Memories:
 256kB Embedded Flash
 64kB Embedded SRAM

Specific Features:
- USB 2.0 Full Speed Device/Host/OTG Controller with on-chip PHY
- 10/100 Ethernet MAC with dedicated DMA and SRAM
- 12-channel DMA controller
- 2 * CAN Interface (2.0B Active)
- 2 * 12-bit ADC (16-channel)
- 2 * 12-bit DAC
- 2 * I2C
- 5 * UART
```

Select appropriate range for read/write memory areas. Under 'Code Generation' make sure that 'Use MicroLIB' is checked



STMicroelectronics STM32F107VC

Xtal (MHz): 25.0

Operating system: None

System-Viewer File (.Sfr):
SFD\ST\STM32F1xx\STM32F107xx.sfr

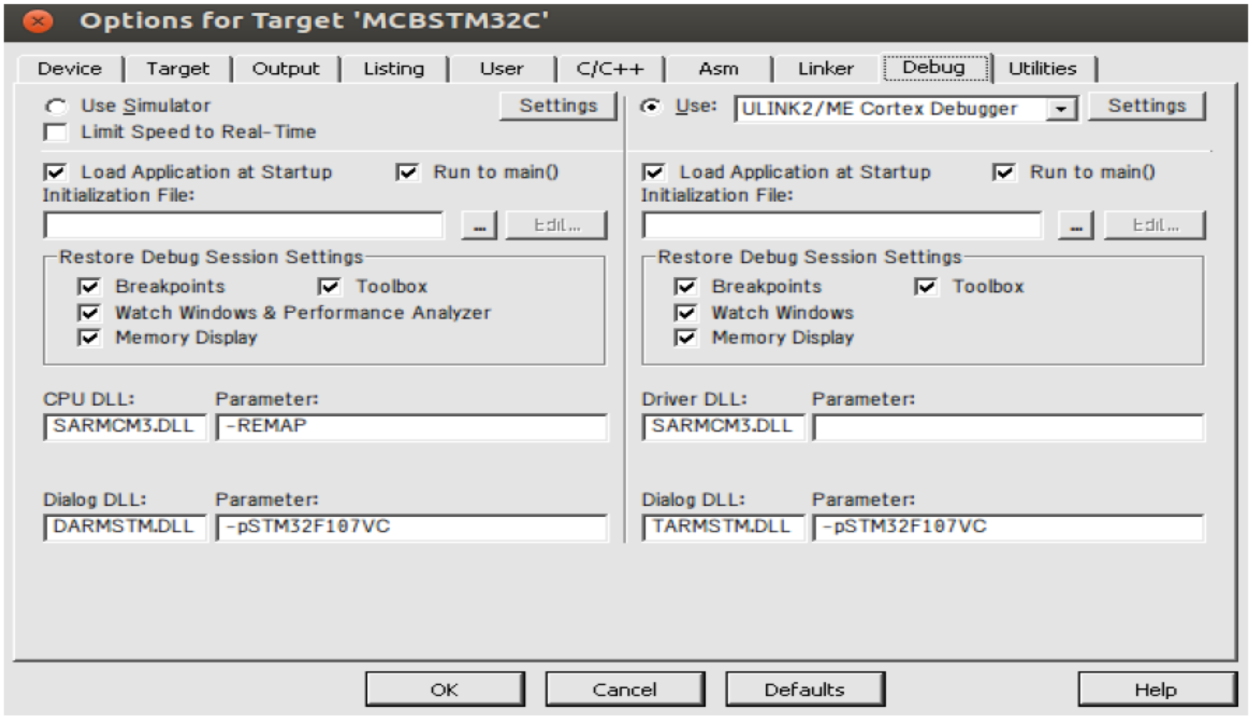
Code Generation
□ Use Cross-Module Optimization
☑ Use MicroLIB □ Big Endian

Read/Only Memory Areas

default	off-chip	Start	Size	Startup
□	ROM1:			○
□	ROM2:			○
□	ROM3:			○
	on-chip			
☑	IROM1:	0x8000000	0x40000	●
□	IROM2:			○

Read/Write Memory Areas

default	off-chip	Start	Size	NoInit
□	RAM1:			□
□	RAM2:			□
□	RAM3:			□
	on-chip			
☑	IRAM1:	0x20000000	0x10000	□
□	IRAM2:			□



○ Use Simulator
□ Limit Speed to Real-Time

☑ Load Application at Startup ☑ Run to main()
Initialization File:

Restore Debug Session Settings
☑ Breakpoints ☑ Toolbox
☑ Watch Windows & Performance Analyzer
☑ Memory Display

CPU DLL: SARMCM3.DLL Parameter: -REMAP

Dialog DLL: DARMSTM.DLL Parameter: -pSTM32F107VC

● Use: ULINK2/ME Cortex Debugger

☑ Load Application at Startup ☑ Run to main()
Initialization File:

Restore Debug Session Settings
☑ Breakpoints ☑ Toolbox
☑ Watch Windows
☑ Memory Display

Driver DLL: SARMCM3.DLL Parameter:

Dialog DLL: TARMSTM.DLL Parameter: -pSTM32F107VC