

STM32F107 ARM-CM3 Board User Guide

Rev. 1.0

Release: 4/20/2010



EMBEST CO., LIMITED

Address: Room 509, Luohu Science & Technology Building, No.85, Taining Road, Shenzhen, Guangdong, China 518020
Telephone: +86-755-25621715
Fax: +86-755-25616057
Sales Email: sales.en@embedinfo.com
Support Email: support.en@embedinfo.com
Website: http://www.embedinfo.com/en



Revision history	
------------------	--

Rev	Date	Description
1.0	4/20/2010	Initial version



Catalog

CHAPTER 1 OVERVIEW	4
1.1 The Microcontroller Introduction	4
1.2 Hardware resources list	4
1.3 Software resources list	5
CHAPTER 2 GETTING STARTED	7
2.1 Documents Description	7
2.2 Version Information	7
2.3 Hardware resource requirements	7
2.4 Preparations	7
CHAPTER 3 STM32F107 BOARD INTRODUCTION	9
3.1 Board Interface Overview	9
3.2 Jumpers Settings	10
3.3 Hardware Interface Introduction	10
3.3.1 JTAG	10
3.3.2 Micro SD Card	
3.3.3 <i>Ethernet</i>	
3.3.4 AUGIO	10
3.3.5 CAN	
3.3.7 I CD & Touch Screen Controller	
3.3.8 Mini USB Port	
3.3.9 EEPROM	
3.3.10 MEMS motion sensor	
3.3.11 JoyStick Button & others Buttons	11
<i>3.3.12</i> 7 sections of numerical code tubes	11
3.3.13 LED	12
CHAPTER 4 SOFTWARE RESOURCES INTRODUCTION	13
CHAPTER 5 SOFTWARE DEVELOPMENT AND EXAMPLES	14
5.1 MDK Introduction	14
5.2 Compiler routine	14
5.2.1 Open a routine	14
5.2.2 Compiler the routine	15
5.3 Debug and Download the routine using Emulator	16
5.3.1 Debug and Download the routine using ULINK2	
5.3.2 Download and Debug the routine Using Emilink JIAG Emulator	21
APPENDIX A: AFTER-SALE SERVICE	



Chapter 1 Overview

1.1 The Microcontroller Introduction

The STM32F107 incorporates the high performance ARM® Cortex[™]-M3 32-bit RISC core operating at a 72 MHz frequency, high speed embedded memories (Flash memory up to 256 Kbytes and SRAM up to 64 Kbytes), and an extensive range of enhanced I/Os and peripherals connected to two APB buses. All devices offer two 12-bit ADCs, four general-purpose 16-bit timers plus a PWM timer, as well as standard and advanced communication interfaces: up to two I2Cs, three SPIs, two I2Ss, five USARTs, an USB OTG FS and two CANs. Ethernet is available on the STM32F107.

The STM32F107 operates in the -40 to +105 °C temperature range, from a 2.0 to 3.6 V power supply. A comprehensive set of power-saving mode allows the design of low-power applications.

Peripherals ⁽¹⁾		STM32F105Rx		STM32F107Rx		STM32F105Vx		STM32F107Vx			
Flash memory in Kbytes		64	128	256	128	256	64	128	256	128	256
SRAM in Kbytes		20	32	64	48	64	20	32	64	48	64
Ethernet		No			Ye	s	No			Yes	
	General-purpose		4								
Timers	Advanced-control					1					
	Basic					2					
	SPI(I ² S) ⁽²⁾		3(2)		3(3	2)		3(2)		3(2)
	I ² C		2		1		2			1	
Communication interfaces	USART	5									
	USB OTG FS	Yes									
	CAN	2									
GPIOs		51							80		
12-bit ADC		2									
Number of chann	nels	16									
12-bit DAC		2									
Number of chann	nels	2									
CPU frequency		72 MHz									
Operating voltage		2.0 to 3.6 V									
Operating temperatures		Ambient temperatures: -40 to +85 °C /-40 to +105 °C Junction temperature: -40 to + 125 °C									
Package		LQFP64 LQFP100									

1.2 Hardware resources list

- STM32F107 ARM 32-bit Cortex[™]-M3 An RS232 connection socket (DB9) CPU, 72 MHz maximum frequency
- An Ethernet interface
- A 3.2-inch large-screen 320 * 240 LCD interface
- SPI interface
- IIC interface

TFT-LCD

- Resistive touch screen professional high-precision screen controller chip
- Low Power, Stereo w/Headphone & Speaker Amps
- A speaker
- A standard 3.5mm headphone jack
- A five-way joystick
- USB OTG
- Three GPIO buttons
- A RESET button
- 2 CAN connection sockets

- A Micro SD card connector
- with IIC interface connects to external EEPROM
- touch ADC and DAC Module
 - One way analog input potentiometer
- 4 LED light tube, a power supply LED Four 7 sections of numerical code tubes
 - CODEC An ultra compact low-power three axes linear accelerometer
 - Support Touch Screen Controller Port expansion
 - A JTAG/SWD debug interface(20pin)
- A mini-type USB socket, support for Own CPU power measurement circuit specifically designed for low-power applications
 - Power Supply: USB powered

Project Name	Function Description
8leds	Use the 8leds to display different digital.
	Use the ADC1 and DMA to transfer continuously converted
	data from ADC1 to memory.
BKP Backup Data	This example shows how to store user data in the Backup
	data registers.
	Configure CAN2 to send and CAN1 to receive CAN frames
CAN_Normal_single_board	in normal mode. The sent frames are used to control LEDs
	by pressing key push button.
CAN Normal two board	CAN peripheral to send and receive frames in normal
	mode(need two boards).
DAC TwoChannels Triangle	This example describes how to use two DAC channels to
Wave	generate two different signals with triangle waves on each
	DAC Channel output.
EMAC_Test	Network interface functional test
FYTI	This example shows how to configure an external
	interrupt line.
ELASH Program	This example provides a description of how to program the
	STM32F10x FLASH.
	This example describes how to use GPIO BSRR (Port bit
GPIO_IOToggle	set/reset register) and BRR (Port bit reset register) for IO
	toggling.
	This example provides a basic example of how to use the
I2C_EEPROM	I2C software library and an associate I2C EEPROM driver
	to communicate with an I2C EEPROM device

1.3 Software resources list

IWDC	This example shows how to reload at regulate period the				
IWDG	IWDG counter using the SysTick interrupt.				
LCD	Using the LCD display pictures and test touchscreen				
	This example demonstrates how to declare dynamic				
	peripherals pointers used for Debug mode.				
NV/IC VactorTable Balacatio	This example describes how to set the CortexM3 vector				
	table in a specific address other than default using the				
	NVIC_SetVectorTable function				
	This example shows how to enter the system to STANDBY				
PWR_STANDBY	mode and wake-up from this mode using: external RESET,				
	RTC Alarm or WKUP pin.				
	This example shows how to configure the System				
RCC	clock(SYSCLK) to have different frequencies:24MHz,				
	36MHz, 48MHz, 56MHz and 72MHz				
	it demonstrates how to setup the RTC peripheral, in terms				
RTC_Calendar	of prescale and interrupts, to be used to keep time and to				
	generate Second interrupt.				
	Traffic program based on a RTX Kernel that controls a				
OS_Test	traffic light.				
	UCOSII 2.8.6				
	This example provides a description of how to initialize the				
SDcard	SD card on Embest_STM3210C board then write and read				
	512 bytes from the SD card, then verify them.				
	This example shows how to configure the SysTick to				
SysTick	generate a time base equal to 1 ms. The system clock is				
	set to 72 MHz, the SysTick is clocked by the AHB clock				
	(HCLK).				
USART HyperTerminal Inte	This example provides a description of how to use the				
rrupt	EVAL_COM1 transmit and receive interrupts to				
•	communicate with the Hyper Terminal.				
USB Test	It Contains four sub-routines, Audio_Streaming				
	JoyStickMouse, Mass_Storage and Virtual_COM_Port				
WWDG	This example shows how to update at regulate period the				
	WWDG counter using the Early Wakeup interrupt (EWI).				



Chapter 2 Getting Started

2.1 Documents Description

File name / Item	Description	Attribute	
STM32F107_Board_	This document		
UserManualV1.0.pdf		M 0/1KB	
EM_STM3210C Board	Development beard schematic	1 100VD	
Schematic.pdf		M 128KB	
STM32F105_107xx	Datashoot of STM22E105_107yy	1 091KB	
Datasheet.pdf		M 901KD	
STM32F105_107xx Reference	Poference Manual of STM32E105 107vv		
Manual.pdf			
Other PDE decuments	Introduce other modules in the board,		
	such as Audio, CAN		

2.2 Version Information

- > The version of the development tools: MDK4.01
- > The version of the ST Library: V3.1.2

2.3 Hardware resource requirements

When we test STM32F107 BOARD, PC recommended the following configuration:

- 2.0GHz (or higher) of the CPU
- 512M RAM
- 2 USB interfaces
- A serial interface
- Windows XP operating system
- KEIL Integrated Development Environment installed

2.4 Preparations

- Jumper Settings: Jumper use the default sets, no need to change.
- > Serial Connection: Connect com of board and the com of PC through serial port cable.
- > LCD Connection: The LCD screen inserted in the LCD interface of board.
- USB Connection: Using USB cable, one end plugged into the USB port on the board, the other end connected to PC.
- SD Card Connection: Connect Micro SD to SD socket on the board.
- JTAG Debugger Connection: One end connected to JTAG interface on the board, the other end connected to PC.
- > Serial Port Receive Settings: In the PC, run HyperTerminal serial communication



program, select the serial port used and set the following parameters (to set status: Baud rate (115200), data bits (8 bits), stop bits (1 bit), parity bit (no), data flow control (no)).

Network Connection: Through the crossover cable provided connect J3 interface on the board and the network interface of PC side.



Chapter 3 STM32F107 Board Introduction



J1	Micro SD Card	U6	Lower-power CODEC chip
J2	JTAG Interface	U7	CAN Transceivers
J3	NET Interface	U8	CAN Transceivers
]4	Audio Line Out	U9	USB very low capacitance ESD
			protection
J5	Audio Analog IN	U10	Enhanced single channel power
			switches
J6	CAN1	U11	Touch Screen Controller
J7	UART	U12	2.5V Voltage adjustor
J8	CAN2	U13	3.3V Voltage adjustor
J9	LCD	U14	7 sections of numerical code tubes
J10	USB MINI AB Interface	D1	LED1
J11	Processor External Round	D2	LED2
	Cable		
J12	Processor External Round	D3	LED3
	Cable		



J13	Control GND Jumper	D4	LED4
U1	STM32F107 Chip	K1	User Button
U2	Serial I ² C BUS EEPROM	K2	Tamper Button
U3	MEMS motion sensor	K3	Wakeup Button
U4	JoyStick Button	K4	Reset Button
U5	Ethernet Transceiver		

3.2 Jumpers Settings

ID	Name	Default Settings	Note
JP1	BOOT0	Close (1-2)	Choose Start-up mode cooperating with JP2
JP2	BOOT0	Close (1-2)	Choose Start-up mode cooperating with JP1
JP3	MICBIAS	Close	Microphone DC bias voltage control signal corresponding to AIN3A
JP4	MICBIAS	Close	Microphone DC bias voltage control signal corresponding to AIN3B
JP5	SPKR/HP	Close	Speaker / Headphone output options

3.3 Hardware Interface Introduction

3.3.1 JTAG

A standard 20-pin JTAG connector is implemented on the STM32F107 BOARD for any ARM JTAG Emulator, such as ULINIK2, JLink, CoLink...

3.3.2 Micro SD Card

A Mini SD Card Interface is implemented on this Board; it can only be used by Mini SD Card and it transports data with SPI mode.

3.3.3 Ethernet

A Physical Layer Transceiver DP83848J (U5) and an integrated RJ45 interface are implemented in this Board, and it supports both 10BASE-T and 100BASE-TX Ethernet protocol, which ensures compatibility and interoperability with all other standard based Ethernet solutions.

3.3.4 Audio

The STM32F107 Board through a CS42L52 low-power stereo audio codec chip to connect the I2C port and a DAC channel in STM32F107. The CS42L52 is a highly integrated, low power stereo CODEC with headphone and Class D speaker amplifiers.

STM32F107 BOARD supports stereo audio playback, and CS42L52 sets the parameters through I2C bus. The speaker and headphone can automatically switch between through corresponding pin.



In addition, CS42L52 can also support the MIC input, encodes the input sound signal into the digital signal microcontroller can identify.

3.3.5 CAN

There are 2 controller area network (CAN) transceivers in this board, and they both use SN65HVD230D controller chip. Each CAN transceiver is designed to provide differential transmit capability to the bus and differential receive capability to a CAN controller a speeds up to 1 Mbps.

3.3.6 UART

The Universal Asynchronous Receiver Transmitter features a 9-pin UART that can be used for communication and trace purposes. It offers an ideal channel for ISP downloading.

3.3.7 LCD & Touch Screen Controller

STM32F107 BOARD provides a 320*240 size TFT LCD with a Touch Screen Controller. This LCD uses a ILI9325 driver IC and represents a pixel with 16 bits. Touch Screen uses a 8-bit STMPE811 control chip and we can get the touch-screen data through I2C interface.

3.3.8 Mini USB Port

A USB Mini AB interface is implemented to transport USB data, and it also supports USB-OTG full speed.

In addition, supply 5V voltage to the Board through this USB port.

3.3.9 EEPROM

A 64kbit EEPROM is connected to the I2C bus in STM32F107 BOARD.

3.3.10 MEMS motion sensor

There is a MEMS motion sensor which uses the LIS302DL chip. The LIS302DL is an ultra compact low-power three axes linear accelerometer. It includes a sensing element and an IC interface able to provide the measured acceleration to the external world through I2C serial interface.

3.3.11 JoyStick Button & others Buttons

STM32F107 BOARD provides a JoyStick button, and it has the UP, DOWN, LEFT, RIGHT and ENSURE function.

In addition, the board also provides RESET, WAKEUP, TAMPER and USER buttons. RESET button is used to reset the processor; WAKEUP connects with PA0 pin, it is used to wakeup the processor when it is sleep; TAMPER and USER respectively connect with PC13 and PB7 pins, they can be used for user input.

3.3.12 7 sections of numerical code tubes

The Board has four 7 sections of numerical code tubes. They can be used to display time, data or other numbers.



3.3.13 LED

STM32F107 BOARD provides 4 LEDs D1, D2, D3 and D4, they respectively connect with PE2, PE3, PE4 and PE5 IO pins, and they can be used for user output.



Chapter 4 Software Resources Introduction

There are 25 test examples in the CD-ROM provided which are built under the Keil RealView MDK environment. The directory of storing these test examples is 03-software\examples. You can find 27 folders in this directory. The Libraries and Utilities are common source files and head files. The other 25 items are all test cases used to test the peripherals or functions of the STM32F107 Board.

Test cases' functions, most of those are easy to understand, but need to explain just that 24-USB_Test. There are four projects in the directory 03-software\Examples\24-USB_Test\USB_DEVICE\Project. They are Audio Streaming, JoyStick Mouse, Mass Storage and Virtual COM Port, all of which are used to test USB function of STM32F107 Board. You can get the details about testing introductions and phenomenon from the readme.txt files in the projects or get them by view the STM32F107 BOARD UserManual V1.0.doc documents.

Note: The evaluation edition of RealView MDK we provided in the CD-ROM of the board might be not able to compile all of test programs we offer, because it has only 32KB code limitation. If you would like to purchase the authorized edition of RealView MDK of Keil, please contact Keil Inc. directly or his distributors in the world.



Chapter 5 Software development and examples

5.1 MDK Introduction

RealView MDK Development Suite is the latest software development tool of ARM Limited Corporation for Embedded Processors. MDK4.01 integrates the most advanced technology in this industry, including μ Vision4 IDE and RealView RVCT4.0 Compiler; RealView MDK supports ARM7, ARM9 and the latest Cortex-M3 Core Processor. It has a configuration wizard for startup code and integrates flash program module, powerful device simulation, performance analyzer and so on. You can obtain the free evaluation edition of MDK software from the CD released with STM32F107 Board in the directory of 04-tools\RealviewMDK, or you can download the latest evaluation version of RealView MDK for free of charge from Keil website www.keil.com. Double click the installation file setup.exe; finish Keil μ Vision4 installation under the guidance of the installation wizard.

Note: If you want to purchase the authorized RealView MDK, please contact Keil Inc. directly or his distributors in the world.

The installation interface as follows:

Setup RealView Microcontroller Development Ki	t ¥4.01 🛛 🔀
Welcome to Keil µ Vision3 Release 10/2009	An ARM [®] Company
This SETUP program installs:	
RealView Microcontroller Development Kit V4.01	
This SETUP program may be used to update a previous product installa However, you should make a backup copy before proceeding.	tion.
It is recommended that you exit all Windows programs before continuing	with SETUP.
Follow the instructions to complete the product installation.	
— Keil µVision3 Setup ————————————————————————————————————	
<< Back	Cancel

5.2 Compiler routine

5.2.1 Open a routine

Enter in the folder of 03-software\Examples; Open the Embest_STM3210C.uvproj file in theprojectdirectoryofoneroutine.Forexample:\03-software\Examples\01-8leds\project\Embest_STM3210C.uvproj.



5.2.2 Compiler the routine

1) If you need the .hex format file, then you can configure MDK to build the hex file, Click Select Folder for Object... to specify the hex file's output directory. Or you can skip this step.

Options for Target 'Embest_STE3210C'	×
Device Target Output Listing User C/C++ Asm Linker Debug Utilities	
Select Folder for Objects <u>Name of Executable</u> : Embest_STM3210C	
Create Executable: .\obj\Embest_STM3210C Debug Information Create Batch File Create HEX File	
 ✓ Browse Information ✓ Create Library: .\obj\Embest_STM3210C.LIB 	
OK Cancel Defaults Help	

2) If you need the .bin format file, then you can configure MDK to build the bin file. Or you can skip this step.

Options for Target 'Embest_STE3210C'
Device Target Output Listing User C/C++ Asm Linker Debug Vtilities
Run User Programs Before Compilation of a C/C++ File
🗖 Run #1: 🛄 🗖 DOS16
🗖 Run #2:
Run User Programs Before Build/Rebuild
🗖 Run #1: 🗖 DOS16
🗖 Bun #2:
Run User Programs After Build/Rebuild
🔽 Run #1: D:\Keil\ARM\BIN40\fromelf.exebin -o ./obj/Embest_STM3210C.bin ./obj/Embest_S
Run #2: 🗆 DOS16
Beep When Complete Start Debugging
OK Cancel Defaults Help

- D:\Keil\ARM\BIN40\fromelf.exe specify the path of fromelf.exe, it will convert axf file to bin file
- --bin -o export bin file
- ./obj/Embest_STM3210C.bin the directory and file name of the bin file you want to create
- ./obj/Embest_STM3210C.axf the directory and file name of the axf file you want to convert
- 3) Click project->build, or click the shortcut button to build the routine.

🔣 Enbest_STI3:	210C	– μV	lision	4				
<u>File E</u> dit <u>V</u> iew [Project	Fl <u>a</u> sh	<u>D</u> ebug	Peripherals	<u>T</u> ools	<u>s</u> vcs	<u>W</u> indow	<u>H</u> elp
i 🗋 💕 层 🦪 i	New	/ µ⊻ision	Project					
🔗 🏼 🎆 🥔 I	New	/ Multi-Pi	roject <u>W</u> o	rkspace				
Project	Ope	n Projec	:t					
🖃 🛅 Embest_STM	⊆los	e Projec	t					
🗄 💼 Startup	Exp	ort						
+ CMSIS	Man	age						
🛨 🧰 Vser	Sele	ct Devic	e for Tarc	et 'Embest SI	M3210C			
⊕ 📄 Read_me	Der	ove The		jet Embest_51	102100			
	Non	ions for l	'' Target 'En	oboct STM221	00'			
		ions for	raryet En	nbest_510321	00			
	Clea	an <u>t</u> argel	t					
	🚵 <u>B</u> uik	l target						
Ĺ	🛗 <u>R</u> eb	uild all ta	arget files					

5.3 Debug and Download the routine using Emulator

The precondition for the next steps is that you have bought or owned a corresponding hardware Emulator.

5.3.1 Debug and Download the routine using ULINK2

- 1. Debug the routin using ULINK2
- 1) Choose Emulator

-	
mbesit	EMBEST CO., LIMITED

Options for Target 'Embest_STE3210C'	
Device Target Output Listing User C/C++ A	Asm Linker Debug Utilities
C Use <u>S</u> imulator <u>Settings</u> ☐ Limit Speed to Real-Time ✓ Load Application at Startup ✓ Run to main()	USe: ULINK Cortex Debugger ULINK ABM Debugger ULINK Cortex Debugger ULINK Cortex Debugger ULINK Cortex Debugger Load RDI Interface Driver So main()
Initialization File:	Initializatid Luminary Eval Board J-LINK / J-TRACE Cortex-M3 J-LINK
Restore Debug Session Settings Breakpoints I Toolbox Watchpoints & PA Memory Display	Restore Debug Session Settings Breakpoints Watchpoints Memory Display
CPU DLL: Parameter: SARMCM3.DLL	Driver DLL: Parameter: SARMCM3.DLL
Dialog DLL: Parameter: DARMSTM.DLL -pSTM32F107VC	Dialog DLL: Parameter: TARMSTM.DLL -pSTM32F107VC
OK Can	cel Defaults Help

2) Check the ULINK2, optional.

When ULINK2 connects to the Development Board, if the RUN and COM indicator lights first change to blue and then go out, and the USB indicator light has always been red, this proves that ULINK2 is no problem.

In addition, you can use the next way to check ULINK2. Click the Settings button in the Debug TAB, if the red marked part appears, it proves that ULINK2 is no problem.

Cortex-I Target Driver Set	tup	
Debug Trace Flash Download		1
ULINK USB - JTAG/SW Adapter	JTAG Device Chain	
Serial No: V0049S9E	IDCODE Device Name IR Ien	vlove
ULINK Version: ULINK2	TD0 O 0x3BA00477 ARM CoreSight JTAG-DP 4 0x06418041 Unknown JTAG device 5	Up
Device Family: Cortex-M	TDI	Down
Firmware Version: V1.37	Automatic Detection ID CODE:	
🗆 SWJ Port: JTAG 💌	C Manual Configuration Device Name:	
Max Clock: 1MHz 🗨	Add Delete Update IR len:	
Debug Connect & Reset Options	Cache Options Download Options	
Connect: Normal 💌 Reset	t: Autodetect 💽 🔽 Cache Code 🗌 🔽 Verify Code Down	load
✓ <u>R</u> eset after Connect	Cache Memory Download to Flash	h
	OK Cancel	{elp

3) Check that whether ULINK2 can detect the development board or not, optional. Click the Settings button in the Debug TAB, if the red marked part appears, it proves that



ULINK2 has detected the development board.

Cortex-I Target Driver Set	1 <u>p</u>	×
Debug Trace Flash Download		
ULINK USB - JTAG/SW Adapter Serial No: V004353E ULINK Version: ULINK2 Device Family: Cortex-M	TDO 0x3BA00477 ARM CoreSight JTAG-DP 4 Up 0x06418041 Unknown JTAG device 5 Down	
Max Clock: 1MHz	Automatic Detection ID CODE: Manual Configuration Device Name: Add Delete Update IR Ien:	
Debug Connect & Reset Options Connect: Normal Reset: Reset:	Autodetect	
[OK Cancel Help	

4) Set the Flash Programmer, configure the Utilities TAB.

Options for Target 'Embest_STE3210C'
Device Target Output Listing User C/C++ Asm Linker Debug Utilities
Configure Flash Menu Command
Use Target Driver for Flash Programming
ULINK Cortex Debugger 💽 Settings 🔽 Update Target before Debugging
Init File:Edit
O Use External Tool for Flash Programming
Command:
Arguments:
E Run Independent
OK Cancel Defaults Help

Then click Settings button, it appears:



Cortex-I Target Driver Setup	X
Debug Trace Flash Download	
Download Function Image: Erase Full Chip: Image: Program Image: Erase Sectors Image: Program Image: Start: I	
Programming Algorithm	
Description Device Type Device Size Address Bange STM32F10x Connectivity Lin On-chip Flash 256k 08000000H - 0803FFFFH	
Start: Size:	
Add Remove	
OK Cancel Help	

If nothing is in the Programming Algorithm, then you should add the corresponding Flash programming algorithm by clicking the Add button, as follows:

A	dd Flash Programming	g Algorithm		×
	Description	Device Type	Device Size	^
	LPC17xx IAP 128kB Flash	On-chip Flash	128k	
	LPC17xx IAP 256kB Flash	On-chip Flash	256k	
	LPC17xx IAP 32kB Flash	On-chip Flash	32k	
	LPC17xx IAP 512kB Flash	On-chip Flash	512k	
	LPC17xx IAP 64kB Flash	On-chip Flash	64k	
	RC28F640J3x Dual Flash	Ext. Flash 32-bit	16M	
	S29JL032H_BOT Flash	Ext. Flash 16-bit	4M	_
	S29JL032H_TOP Flash	Ext. Flash 16-bit	4M	
	STM32F10x Med-density Flash	On-chip Flash	128k	
	STM32F10x Low-density Flash	On-chip Flash	16k	
	STM32F10x High-density Flash	On-chip Flash	512k	Ξ
	STM32F10x Connectivity Lin	On-chip Flash	256k	
	STM32F10x M25P64 SPI Fla	Ext. Flash SPI	8M	
	STM32F10x Flash Options	On-chip Flash	16	
	TMPM330FWx 128kB Flash	On-chip Flash	128k	
	TMPM330FYx 256kB Flash	On-chip Flash	256k	×
	Add	Cancel		

After you choose a Flash Programming Algorithm, then Click Add button.

5) Start to Debug the routine by clicking shortcut button ⁴⁴ or clicking Debug->Start/Stop Debug Session, the status of the debug as follows:



🔣 Enbest_STE32	10C - μVi	ision4	- ×
<u>F</u> ile <u>E</u> dit ⊻iew P	roject Fl <u>a</u> sh <u>D</u>	<u>Debug Peripherals T</u> ools <u>S</u> VCS <u>W</u> indow <u>H</u> elp	
🗋 🚅 🛃 🥥 🔅	6 🗈 🖻 🤊	◎ ← → № & & & 準 準 准 版 20	
: 👫 🔜 📀 🔂	ᠿ ᠿ () ↔		
Registers	→ ‡ ×	K Disassembly	• 4 ×
Register Va Core R0 0s R1 0s R3 0s R3 0s R4 0s R4 0s R5 0s	alue x08000804 x20000450 x00000000 x080007b9 x080009b0 x080009b0 x080009b0	<pre>40: { 41: int i; 42: /* System Clocks Configuration */ 42: /* System Clocks Configuration */ 42: /* Configuration(); 44: 45: /* Configuration(); 44: 45: /* Configure the GPIO ports */ 0x06000806 F7FFPB3 BL.W RCC Configuration (0x08000378)</pre>	
R5 03 R7 03 R8 03 R9 05 R10 05 R11 05	x2c909645 x8c915201 xcad60a94 x20000160 x585e0636 xc8a64da3	46: GPIO_Configuration(); 47: 47: 48: /# Configure HRAPT #2 #/	>
	x0000000f	main.c	▼ X
R13 GST 05 R14 (IR) 05 R15 (PC) 05 ⊕	x20000450 x08000194 x0800080c x81000000 hread rivileged SP ▼	0.33 int main(void) (0.33 int i; (0.41 int i; (0.42 /* System Clocks Configuration */ (0.43 RCC_Configuration(); (0.44	•
Command			• 4 ×
Load "F:\\03-s	oftware\\Ex:	<pre>kamples\\01-8leds\\project\\obj\\Embe E main 0 </pre>	
ASSIGN BreakDis	sable BreakH	KEnable BreakKill BreakList BreakSet 🛛 🖗 Call Stack 💭 Locals 💭 Watch 1 🔲 Memory 1 💽 Symbols	
		ULINK Cortex Debugger t1: 0.00017590 sec L:40 C:1 CAPI NUM SCRL O	VR R/W

- 2. Download the routine using ULINK2
- 1) Check the Flash Programmer's setting

Options for Target 'Embest_STE3210C'
Device Target Output Listing User C/C++ Asm Linker Debug Utilities
Configure Flash Menu Command
 Use Target Driver for Flash Programming
ULINK Cortex Debugger 🚽 Settings 🔽 Update Target before Debugging
Init File: Edit
C Use External Tool for Flash Programming
Command
Arguments:
E Bun Independent
OK Cancel Defaults Help



Cortex-I Target Driver Setup	
Debug Trace Flash Download	
Download Function C Erase Full Child ✓ Program Image: Start Start Start C Do not Erase ✓ Verify Image: Start S	3
Programming Algorithm	
Description Device Type Device Size Address Bange	
STM32F10x Connectivity Lin On-chip Flash 256k 08000000H - 0803FFFFH	
Start: Size:	
Add Remove	
OK Cancel	Help

2) Start to Download by click Flash->Download or the shortcut button as follows:

🔣 Embest_STE3210C – µ Vision	4			
Eile Edit View Project Flash Debug Image: Im	Peripherals Tools SVCS Window Help			
Embest_STM3210C CMSIS CMSIS CMSIS StdPeriph_Driver User Main.c Read_me Proj & Books {} Fun 0, Tem	038 */ 039 int main(void) 040 { 041 int i; 042 /* System Clocks Configuration */ 043 RCC_Configuration(); 044 045 /* Configure the GPIO ports */ 046 GPIO_Configuration(); 047 048 /* Configure USART #2 */ 049 USART_Configuration();			
Build Output				
Load "F:\\03-software\\Examples\\01-8leds\\project\\obj\\Embest_STM3210C.AXF" Erase Done. Programming Done. Verify OK.				
Ruild Output Eind in Files				
Download code to flash memory	ULINK Cortex Debugger			

5.3.2 Download and Debug the routine using Emlink JTAG Emulator

- 1. Debug the routine using Emlink JTAG Emulator
- 1) Install the Emlink for MDK driver.
- 2) Choose the Emulator.

www.embedinfo.com/en

1	
Embes t	EMBEST CO., LIMITED

Options for Target 'Embest_SIM3210C'	X
Device Target Output Listing User C/C++ 1	Asm Linker Debug Utilities
C Use Simulator Settings □ Limit Speed to Real-Time ✓ Load Application at Startup ✓ Initialization File:	 ✓ Lise: Emlink Debugger For ARM ✓ Emlink Debugger For ARM ✓ ULINK ARM Debugger ✓ ULINK Cortex Debugger NDI Interface Driver Initialization Altera Blaster Cortex Debugger Signum Systems JTAGjet
Restore Debug Session Settings Breakpoints I Toolbox Watchpoints & PA Memory Display	Restore Debug Session Settings Breakpoints Watchpoints Memory Display
CPU DLL: Parameter: SARMCM3.DLL	Driver DLL: Parameter: SARMCM3.DLL
Dialog DLL: Parameter: DARMSTM.DLL -pSTM32F107VC	Dialog DLL: Parameter: TARMSTM.DLL -pSTM32F107VC
OK Car	ncel Defaults Help

3) Click the Settings button to configure Emlink, as follows:

Colink Driver Setup 🛛 🔀
Colink Setup
JTAG Speed: 666K 💌 Hz
Reset Option: SYSRESETREQ 💌
Cache Options
Cancel

4) Set the Flash Programmer.

Configure the Utilities TAB, as follows:

Options for Target 'Embest_STM3210C'	×
Device Target Output Listing User C/C++ Asm Linker Debug Utilities	_
Configure Flash Menu Command	
Use Target Driver for Flash Programming	
Emlink Debugger For ARM 💽 Settings 🔽 Update Target before Debugging	
Init File: ULINK ARM Debugger	
O Use Extern Simure TAGet	
Command:	
Arguments:	
T Run Independent	
OK Cancel Defaults Help	

Then click the Settings buttons, it will show the next Dialog.

Flash Download Setu	р			×
Download Function	ip O Erase Sectors	C Dor	not Erase	
Programming Algorithm				
Description	Device Type	Device Size	Address Range	
STM32F10x Connectivity L.	On-chip Flash Memory	256KB	08000000H0803FFFFH	
	Add	Remove		
	ОК	Cancel		

If nothing is in the Programming Algorithm, then you should add the corresponding Flash programming algorithm by clicking the Add button, as follows:

1	dd Programming Algor	ithm		×
	Description	Device Type	Device Size	^
	LPC17xx IAP 256kB Flash	On-chip Flash Memory	256KB	
	LPC17xx IAP 32kB Flash	On-chip Flash Memory	32KB	
	LPC17xx IAP 512kB Flash	On-chip Flash Memory	512KB	
	LPC17xx IAP 64kB Flash	On-chip Flash Memory	64KB	
	RC28F640J3x Dual Flash	External Flash Device 32-bit	16MB	
	S29JL032H_BOT Flash	External Flash Device 16-bit	4MB	
	S29JL032H_TOP Flash	External Flash Device 16-bit	4MB	
	STM32F10x Med-density Flash	On-chip Flash Memory	128KB	
	STM32F10x Low-density Flash	On-chip Flash Memory	16KB	
	STM32E10x High-density Elash	On-chip Elash Memory	512KB	
	STM32F10x Connectivity Lin	On-chip Flash Memory	256KB	
	STM32F10x M25P64 SPI Flash	External Flash Device SPI	8MB	
	STM32F10x Flash Options	On-chip Flash Memory	16B	
	TMPM330FW× 128kB Flash	On-chip Flash Memory	128KB	<u>×</u>
		OK Cancel		

5) Start to debug the routine by clicking the shortcut button er or clicking Debug->Start/Stop Debug Session, the status of the debug is as follows:

🖫 Embest_STII3210C – μVision4				
Eile Edit Yiew Project Flash Debug Peripherals Iools SVCS Window Help				
🗋 💕 🛃 🗿 X 🖻 🛍 🤊 🗠 🔶 → 隆 🎘 🎘 🖏 導 導 進 版 💆	💌 🔜 🦚 🔍 🔺 🔍 👻			
: X 🗉 🛇 (건) (건 (건 (건 (전	■ - 🋠 - 🔜 -			
Registers 💌 🗭 X Disassembly	→ ‡ ×			
Register Value 90: LED_CHIP_SEL(LED_ONE) Core 0x00000031 NOVS r0,#00 Ri 0x00000080 91: SEVEN_LED_DISPLAY R2 0x00000080 91: SEVEN_LED_DISPLAY R3 0x10000080 92: Delay(1); 0x08000836 F7FFF51 BL.W LED_CHIP_SEVEN R4 0x00000080 92: Delay(1); 0x08000836 F7FFF51 BL.W SEVEN R5 0x08000836 F7FFF51 BL.W SEVEN R5 0x08000836 F7FFF52 BL.W Delay(1); 0x08000836 F7FFF52 BL.W Delay(1); 0x08000836 F7FFF52 BL.W Delay(1); 0x08000836 F7FFF52 BL.W Delay(1); 0x08000756 R15 0x08000766 R6 R14 0x08000766 R6 F0r (1 = 200; i > 0; i) 0x13 0x2000048 R14 0x0800076 R6 0x14 0x2000048 R14 0x0800076 R6 0x15	<pre>; x00 HIP_SEL (0x080002E8) '(ONE); x61 [_LED_DISPLAY (0x0800045C) x01 * (0x080001A4)</pre>			
>				
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet	Call Stack ALCCals All Memory 1 Symbols			
	ULINK Cortex Debugger 11: 0.06326810 sec 1:90 C:1 CAP NUM SCRU OVR R/W			

- 2. Download the routine using Emlink
- 1) Check the Flash Programmer's setting.

Options for Target 'Embest_S	TE3210C"	X
Device Target Output Listing User	r C/C++ Asm Linker	Debug Utilities
Configure Elash Menu Command		
 Use Target Driver for Flash Programmi 	ng	
Emlink Debugger For ARM	✓ Settings I	Update Target before Debugging
Init File: Emlink Debugger For ARM		Edit
ULINK Cortex Debugger RDI Interface Driver		
Signum Systems JTAGjet		
Arguments:		
Run Independent		
OK	Cancel I	lefaults Help
OK	Cancel I	lefaults Help
OK Flash Download Setup	Cancel I	lefaults Help
OK Flash Download Setup Download Function © Erase Full Chip	Cancel I	Do not Erase
	Cancel I	Defaults Help
OK Flash Download Setup Download Function © Erase Full Chip Programming Algorithm Description	Cancel I CErase Sectors	Do not Erase
OK Flash Download Setup Download Function © Erase Full Chip Programming Algorithm Description Device 1 STM32F10x Connectivity L On-chip	Cancel I CErase Sectors C Type Device C Flash Memory 256KB	Do not Erase
OK Flash Download Setup Download Function © Erase Full Chip Programming Algorithm Description Description STM32F10x Connectivity L	C Erase Sectors C Type Device O Flash Memory 256KB	Do not Erase
OK Flash Download Setup Download Function © Erase Full Chip Programming Algorithm Description Description STM32F10x Connectivity L	Cancel I CErase Sectors C Type Device C Flash Memory 256KB	Do not Erase
OK Flash Download Setup Download Function © Erase Full Chip Programming Algorithm Description Description STM32F10x Connectivity L	Cancel I CErase Sectors C Cype Device C Flash Memory 256KB	Do not Erase
OK Flash Download Setup Download Function Erase Full Chip Programming Algorithm Description Description Description STM32F10x Connectivity L 	Cancel I CErase Sectors	lefaults Help
OK Flash Download Setup Download Function © Erase Full Chip Programming Algorithm Description Device 1 STM32F10x Connectivity L On-chip	Cancel I CErase Sectors C Fype Device S Flash Memory 256KB	lefaults Help
OK Flash Download Setup Download Function © Erase Full Chip Programming Algorithm Description STM32F10x Connectivity L On-chip	Cancel I CErase Sectors	Do not Erase
OK Flash Download Setup Download Function © Erase Full Chip Programming Algorithm Description STM32F10x Connectivity L On-chip	Cancel I Cancel I CErase Sectors Cype Device Flash Memory 256KB dd Remo	Ve

2) Start to download by clicking Flash->Download or the shortcut button, as follows:



🔣 Embest_STN3210C – µVisio	n4			
Eile Edit View Project Fl <u>a</u> sh <u>D</u> ebug	; Pe <u>ripherals T</u> ools <u>S</u> VCS <u>Wi</u> ndow <u>H</u> elp			
i 🗋 💕 🛃 🗿 🛛 🔏 📴 🙀 Download	■ 🐘 🐘 🛊 🎼 /// /// 🖄	🖃 🔜 🥐 🔍 🕒		
i 🕸 🕮 🧼 🗮 🕼 📴 Erase	l 📥 🕾			
Project <u>C</u> onfigure	Flash Tools	▼ ×		
Embest_STM3210C Startups CMSIS CMSIS CMSIS StdPeriph_Driver User User Iser Read_me Proj	103 104 for (i = 200; i > 0; i) 105 (106 LED_CHIP_SEL(LED_ONE); 107 SEVEN_LED_DISPLAY(FIVE); 108 Delay(1); 109 LED_CHIP_SEL(LED_TWO); 110 SEVEN_LED_DISPLAY(SIX); 111 Delay(1); 112 LED_CHIP_SEL(LED_THREE); 113 SEVEN_LED_DISPLAY(SEVEN):	- - -		
Build Output		→ ‡ ×		
Build target 'Embest_STM3210C' linking Program Size: Code=2032 RO-data=448 RW-data=76 ZI-data=1028 FromELF: creating hex file User command #1: E:\MDK\Keil\ARM\BIN40\fromelf.exebin -o ./obj/Embest_STM3210C.bin ./obj/ ".\obj\Embest_STM3210C.axf" - 0 Error(s), 0 Warning(s).				
<				
🖻 Build Output 🛛 🙀 Find in Files				
Download code to flash memory	ULINK Corte	ex Debugger		



Appendix A: After-sale Service

Embest is at your service, and we have special Technical Support Engineers to provide support and consultation of our products and services in forms of telephone, E-mail, Fax and website.

- TEL: +86-755-25621715
- FAX: +86-755-25616057
- E-mail for Technical Support: support.en@embedinfo.com

Website: you can leave your questions at <u>http://www.embedinfo.com/en/Request.asp</u>, and then our technical support engineers will get back to you as soon as possible.