life.augmented

MakeNTU STM32 工作坊

ST MCU Taiwan Team

2022.02.20

議題

1 軟體安裝

2 開發板

3 STM32CubeIDE 簡介

4 Hands-on: GPIO

5 Hands-on: UART

6 Hands-on: Timer

7 Hands-on: TouchGFX

軟體安裝
建立myST帳號

life.augmented

建立mySTM帳號

https://my.st.com/

life.augmented

軟體安裝
下載並安裝STM32CubeIDE

life.augmented

下載STM32CubeIDE
下載頁面

1 https://www.st.com/en/development-tools/stm32cubeide.html

2

下載STM32CubeIDE License Agreement

life.augmented

下載STM32CubeIDE
登入及下載

下載會在完成登入之後開始

安裝STM32CubeIDE – 1

life.augmented

安裝STM32CubeIDE – 2

- 建議使用預設安裝路徑
- 若不使用預設安裝路徑，請確認安裝路徑名稱當中沒有空格

life.augmented

安裝STM32CubeIDE – 3

安裝完成

life.augmented

軟體安裝
STM32CubeF7 MCU Package安裝

life.augmented

開啓STM32CubeIDE

開啓Embedded Software Packages Manager

STM32F746 Discovery開發板

life.augmented

32F746GDISCOVERY

方塊圖

3.3V Power Supply
32KHz Crystal
RTC
4.3-inch WQVGA LCD
LCD
Arduino Uno shield connectors
GPIOs
VCP UART
STM32F746NGH6
ST-LINK/V2-1
SWD
SDRAM
FMC
microSD card connector
SDMMC1
Ethernet PHY and RJ45 connector
RMII
OTG FS
Micro USB connector
OTG HS
HS PHY and Micro USB connector
GPIO
Push-button and Wakeup
SPDIFRX
SPDIF decoder and RCA INput connector
Quad-SPI
Quad-SPI Flash memory
Capacitive Touch Screen Multi points
I2C
I2C extension connector
SAI
Audio DAC and Amplifier
Stereo jack
Line input
Stereo jack
Line Output
DCMI
Camera connector
MSv38837V1

開發板說明

STM32F746G-DISCO
MB1191B
www.st.com/stm32f7-discovery

STM32F746G-DISCO
MB1191B
www.st.com/stm32f7-discovery
Two ST-MEMS microphones
4.3-inch 480x272 color LCD-TFT
with capacitive touch screen

開發板說明

Audio codec
3.5 mm stereo jacks
(IN and OUT)
RJ45 single port
High-Speed USB
PHY-ULPI
Micro USB
Mini USB type B
ST-LINK/V2-1
Power
supply inlet 5V
Power supply selector
Push-buttons (user and RST)
128-Mbit SDRAM 4Mx32
I2C extension connector
STM32F746NGH6
TFBGA216
SPDIF input,
RCA connector
DCMI camera
connector
Headers (Arduino)
128-Mbit Quad-SPI,
Quad/Dual
microSD card connector
Headers (Arduino)

電源供應(1)

Audio codec
3.5 mm stereo jacks
(IN and OUT)
RJ45 single port
High-Speed USB
PHY-ULPI
Micro USB
Mini USB type B
ST-LINK/V2-1
Power
supply inlet 5V
Power supply selector
Push-buttons (user and RST)
128-Mbit SDRAM 4Mx32
I2C extension connector
STM32F746NGH6
TFBGA216
SPDIF input,
RCA connector
DCMI camera
connector
Headers (Arduino)
128-Mbit Quad-SPI,
Quad/Dual
microSD card connector
Headers (Arduino)

5V
ext
5V
link
usb
_fs
usb
_hs
5V
5V
5V
5V
JP1

JP2
5V
GND

電源供應(2)

Audio codec
3.5 mm stereo jacks
(IN and OUT)
RJ45 single port
High-Speed USB
PHY-ULPI
Micro USB
Mini USB type B
ST-LINK/V2-1
Power
supply inlet 5V
Power supply selector
Push-buttons (user and RST)
128-Mbit SDRAM 4Mx32
I2C extension connector
STM32F746NGH6
TFBGA216
SPDIF input,
RCA connector
DCMI camera
connector
Headers (Arduino)
128-Mbit Quad-SPI,
Quad/Dual
microSD card connector
Headers (Arduino)

5V
ext
5V
link
usb
_fs
usb
_hs
5V
5V
5V
5V
JP1

CN6
Vin
GND
GND
5V
3V3
NRST
IOref
nc

電源供應(3)

Audio codec
3.5 mm stereo jacks
(IN and OUT)
RJ45 single port
High-Speed USB
PHY-ULPI
Micro USB
Mini USB type B
ST-LINK/V2-1
Power
supply inlet 5V
Power supply selector
Push-buttons (user and RST)
128-Mbit SDRAM 4Mx32
I2C extension connector
STM32F746NGH6
TFBGA216
SPDIF input,
RCA connector
DCMI camera
connector
Headers (Arduino)
128-Mbit Quad-SPI,
Quad/Dual
microSD card connector
Headers (Arduino)

5V
ext
5V
link
usb
_fs
usb
_hs
5V
5V
5V
5V
JP1

電源供應(4)

Audio codec
3.5 mm stereo jacks
(IN and OUT)
RJ45 single port
High-Speed USB
PHY-ULPI
Micro USB
Mini USB type B
ST-LINK/V2-1
Power
supply inlet 5V
Power supply selector
Push-buttons (user and RST)
128-Mbit SDRAM 4Mx32
I2C extension connector
STM32F746NGH6
TFBGA216
SPDIF input,
RCA connector
DCMI camera
connector
Headers (Arduino)
128-Mbit Quad-SPI,
Quad/Dual
microSD card connector
Headers (Arduino)

5V
ext
5V
link
usb
_fs
usb
_hs
5V
5V
5V
5V
JP1

電源供應(5)

Audio codec
3.5 mm stereo jacks
(IN and OUT)
RJ45 single port
High-Speed USB
PHY-ULPI
Micro USB
Mini USB type B
ST-LINK/V2-1
Power
supply inlet 5V
Power supply selector
Push-buttons (user and RST)
128-Mbit SDRAM 4Mx32
I2C extension connector
STM32F746NGH6
TFBGA216
SPDIF input,
RCA connector
DCMI camera
connector
Headers (Arduino)
128-Mbit Quad-SPI,
Quad/Dual
microSD card connector
Headers (Arduino)

5V
ext
5V
link
usb
_fs
usb
_hs
5V
5V
5V
5V
JP1

Virtual Comport

Arduino Connectors

CN6

Reserved	1	NC
3V3	2	IOREF
NRST	3	RESET
3V3	4	3V3
5V	5	5V
GND	6	GND
GND	7	GND
VIN	8	VIN

PA0	1	A0
PF10	2	A1
PF9	3	A2
PF8	4	A3
PF7	5	A4
PF6	6	A5

CN5

CN7

SCL/D15	10	PB8
SDA/D14	9	PB9
AVDD	8	AVDD
GND	7	GND
SCK/D13	6	PI1 / LD1
MISO/D12	5	PB14
PWM/MOSI/D11	4	PB15
PWM/D10	3	PA8
PWM/D9	2	PA15
D8	1	PI2

D7	8	PI3
PWM/D6	7	PH6
PWM/CS/D5	6	PI0
D4	5	PG7
PWM/D3	4	PB4
D2	3	PG6
TX/D1	2	PC6
RX/D0	1	PC7

CN4

32F746DISCOVERY文件

UM1907
User manual

Discovery kit for STM32F7 Series with STM32F746NG MCU

Introduction

The 32F746GDISCOVERY Discovery board is a complete demonstration and development platform for the STMicroelectronics ARM® Cortex®-M7 core-based STM32F746NGH6 microcontroller. This microcontroller features four I²Cs, six SPIs with three multiplexed simplex I²S, SDMMC, four USARTs, four UARTs, two CANs, three 12-bit ADCs, two 12-bit DACs, two SAIs, 8- to 14-bit digital camera module interface, internal 320+16+4-Kbyte SRAM and 1-Mbyte Flash memory, USB OTG HS and FS, Ethernet MAC, FMC interface, Quad-SPI interface, SWD debugging support. This Discovery board offers everything required for users to get started quickly and develop applications easily.

The full range of hardware features on the board helps users to evaluate almost all peripherals (USB OTG HS and FS, 10/100-Mbit Ethernet, microSD™ card, USART, SAI Audio DAC stereo with audio jack input and output, ST-MEMS digital microphones, SDRAM, Quad-SPI Flash memory, 4.3-inch color LCD-TFT with a capacitive multi-touch panel, SPDIF RCA input and others) and to develop their applications. Arduino™ Uno V3 connectors make it possible to easily connect extension shields or a daughterboard for users' specific applications. The integrated ST-LINK/V2-1 provides an embedded in-circuit debugger and programmer for the STM32.

The 32F746GDISCOVERY board comes with the STM32 comprehensive software HAL library together with various packaged software examples, as well as a direct access to ARM® mbed Enabled™ online resources at http://mbed.org.

Figure 1. 32F746GDISCOVERY board (top view)

Figure 2. 32F746GDISCOVERY board (bottom view)

1. Pictures are not contractual.

June 2017 DocID027942 Rev 4 1/51

www.st.com

STM32CubeIDE簡介

life.augmented

STM32的開發方式

arm MBED	STM32 Cube	ARDUINO
Online mBed compiler	STM32CubeIDE	Arduino IDE
mBed API	STM32Cube HAL API	Arduino API
printf-based debug	Professional debugger	printf-based debug
Simple editor	Powerful editor	Simple editor

life.augmented

STM32Cube Ecosystem

Background of STM32CubeIDE

One Tool for All Your STM32 Development

Chipset / Board Configuration

Code Development

Validation Debug

life.augmented

Free Multi-Platform Development Tool

eclipse

GCC

Windows

macOS

Eclipse/GCC Based

Free for Commercial Development

Multi-OS Support

life.augmented

Project Management

Importer

TrueSTUDIO Project

SW4STM32 Project

IOC Editor

Code Editor – Navigation

Symbol Hyperlink

Include Browser

Call Hierarchy

Brace Navigation

Macro Expansion Browser

Type Hierarchy

life.augmented

Code Editor – Navigation (2/2)

Outline View

Powerful Search

Open Resource

Task Tags

Bookmark

Minimap

Code Editor – Writing

Highlight Inactive Code

Auto-Complete

Syntax Highlight

File Diff/Compare

Block Select

Code Style

Build Tools

Build Analyzer

Build Analyzer | Static Stack Analyzer | Search

F769-DISCO-ITCM.elf - /F769-DISCO-ITCM/Debug - May 10, 2019 3:28:29 PM

Memory Regions | Memory Details

Region	Start address	End address	Size	Free	Used	Usage (%)
FLASH	0x08000000	0x08200000	2048 KB	2043.22 KB	4.78 KB	0.23%
RAM	0x20000000	0x20080000	512 KB	510.45 KB	1.55 KB	0.30%
ITCMRAM	0x00000000	0x00004000	16 KB	15.48 KB	528 B	3.22%

Build Analyzer | Static Stack Analyzer | Search

F769-DISCO-ITCM.elf - /F769-DISCO-ITCM/Debug - May 10, 2019 3:28:29 PM

Memory Regions | Memory Details

Search

Name	Run address (VMA)	Load address (LMA)	Size
ITCMRAM	0x00000000		16 KB
.itcmram	0x00000000	0x0800110c	528 B
FLASH	0x08000000		2048 KB
.itcmram	0x00000000	0x0800110c	528 B
.isr_vector	0x08000000	0x08000000	60 B
.text	0x0800003c	0x0800003c	4.16 KB
.rodata	0x080010e0	0x080010e0	16 B
.ARM	0x080010f0	0x080010f0	8 B
.preinit_array	0x080010f8	0x080010f8	0 B
.init_array	0x080010f8	0x080010f8	4 B
.fini_array	0x080010fc	0x080010fc	4 B
.data	0x20000000	0x08001100	12 B
RAM	0x20000000		512 KB
.data	0x20000000	0x08001100	12 B
.bss	0x2000000c		32 B
._user_heap_stack	0x2000002c		1.5 KB

Static Stack Analyzer

Build Analyzer | Static Stack Analyzer | Search

F769-DISCO-ITCM.elf - /F769-DISCO-ITCM/Debug - May 10, 2019 3:28:29 PM

List | Call graph

☑ Hide dead code

Function	L...	Type	Location	Info
SystemClock_Config	88	STATIC	main.c:108	
NVIC_EncodePriority	40	STATIC	core_cm7.h:2071	
HAL_RCC_GetSysClockFreq	40	STATIC	stm32f7xx_hal_rcc.c:982	
HAL_NVIC_SetPriority	32	STATIC	stm32f7xx_hal_cortex.c:165	
HAL_RCC_OscConfig	32	STATIC	stm32f7xx_hal_rcc.c:344	
__NVIC_SetPriorityGrouping	24	STATIC	core_cm7.h:1865	
HAL_RCC_ClockConfig	24	STATIC	stm32f7xx_hal_rcc.c:703	
HAL_InitTick	16	STATIC	stm32f7xx_hal.c:231	

Build Analyzer | Static Stack Analyzer | Search

F769-DISCO-ITCM.elf - /F769-DISCO-ITCM/Debug - May 10, 2019 3:28:29 PM

List | Call graph

Function	Depth	Max cost	Local cost	Type	Location
LoopFillZerobss	7	208	0		
SysTick_Handler	0	8	8	STATIC	stm32f7xx_it.c:182
NMI_Handler	0	4	4	STATIC	stm32f7xx_it.c:70
UsageFault_Handler	0	4	4	STATIC	stm32f7xx_it.c:128
PendSV_Handler	0	4	4	STATIC	stm32f7xx_it.c:169
HardFault_Handler	0	4	4	STATIC	stm32f7xx_it.c:83
HAL_IncTick	0	4	4	STATIC	stm32f7xx_hal.c:290
SVC_Handler	0	4	4	STATIC	stm32f7xx_it.c:143
DebugMon_Handler	0	4	4	STATIC	stm32f7xx_it.c:156
MemManage_Handler	0	4	4	STATIC	stm32f7xx_it.c:98
BusFault_Handler	0	4	4	STATIC	stm32f7xx_it.c:113
Reset_Handler	0	0	0		

Headless Build

- **Continuous Integration**
  - Build project without opening IDE

    No GUI shown but build system becomes active

  - Supported for makefile and managed projects

Debug

Debugger

Live expressions

SWV

Integrated UART Terminal

External Loader

Remote Debug

關於STM32CubeIDE

life.augmented

Workspace – Workspace的內容

Workspace的預設目錄為：
C:\Users\XXX huang\STM32CubeIDE\

- Workspace是Eclipse當中用來存放專案目錄或是專案目錄參照的地方
- Workspace可存放多個專案
- Workspace也是用來儲存操作環境設定的地方(.metadata)

Workspace - Workspace與Project

Workspace – Workspace的切换

1

2

3

Perspectives (視角)

C/C++

Debug

Device Configuration Tool (IOC Editor)

Perspectives (視角)
Views in C/C++ Perspective

life.augmented

Perspectives (視角)
Views in Debug Perspective

life.augmented

Configuration – Build Configuration

Configuration – Debug Configuration

C/C++ Build Settings

Known Issues and Limitations

- The [Run] button is not yet implemented and is therefore hidden from the toolbar menu and Run menu.
- The creation of static libraries with the option add libraries as reference leads to the unintended generation of the Drivers folder.
- The USE_HAL_DRIVER symbol is not properly removed when switching from HAL to LL driver, which can cause build failure.
- Creating BOARD projects with Code generator options == add necessary library files as reference.. and Initialise all peripherals with default settings can cause build errors if BOARD depends on the USB library.
- The user cannot change [Application Structure] from Basic to Advanced or vice versa without losing user code.
- Importing the ioc file created by stand-alone STM32CubeMX is not fully supported.
- Editor hyperlinks sometimes jump to declaration instead of definition.
- It is not possible to open an SW4STM32 or TrueSTUDIO® workspace with STM32CubeIDE. Refer to Migration guide from TrueSTUDIO® to STM32CubeIDE (UM2578) and Migration guide from System Workbench to STM32CubeIDE (UM2579).
- The [Help]>[Data refresher] can be invoked several times without pop-up dialog.
- Some Linux® installers install a few packages before the license agreement has been accepted.
- The macOS® installer displays incompatible version dialog when installing the stlink-server package. This can safely be ignored.
- Some STM32CubeMX pop-up dialogs are not opened in front of the STM32CubeIDE workbench on all OS.
- On macOS®, the ioc editor does not show the Peripheral categories frame on the left side by default.
- STM32CubeIDE does not support switching from one MCU to another once the project is created.
- Anti-virus tools may be sensitive to STM32CubeIDE and bundled exe files despite the addition of vendor certificates since it is not yet deployed widely.
- STM32CubeIDE SWV selecting large amounts of data to copy to the clipboard may crash STM32CubeIDE.
- Conditional breakpoints do not work with OpenOCD.
- The project importer for SW4STM32 cannot import all settings in projects from very old versions (older than 2.0).
- The Generate code operation re-includes excluded files in folders that are created by STM32CubeMX
- Having a space or non-ascii character in the project/workspace path or installation path is not fully supported
- Updating field [HCLK] in tab Clock Configuration is difficult
- When creating an empty project, the FPU is enabled in the build settings but the project is generated without a SystemInit function to initialize the FPU at run-time.
- In the STM32CubeMX .ioc editor under [Project Manager]>[Code Generator], there is a [settings] button that is not yet implemented.
- Importing a project from an earlier version into the current one will hang STM32CubeIDE when opening the .ioc editor. This only affects macOS®.
- Some radio or check buttons in the debugger tab have unexpected rendering on any Ubuntu® 14.04.

From RN0114 Release Note

參考資料

- **STM32CubeIDE官方文件**
  https://www.st.com/en/development-tools/stm32cubeide.html#documentation

- **STM32CubeIDE basics MOOC（內含Youtube）**
  https://www.st.com/content/st_com/en/support/learning/stm32-education/stm32-moocs/STM32CubeIDE_basics_MOOC.html

life.augmented

Hands-on
01_GPIO_Output

life.augmented

實作目標

每500ms改變LD1閃滅狀態

流程

1. 由STM32CubeIDE產生GPIO專案
2. 設定腳位功能並產生程式碼
3. 加入程式碼並重新編譯
3. 燒錄/除錯

life.augmented

Step 1: 產生專案

life.augmented

Step 2: 腳位設定

workspace_1.2.0_MakeNTU - Device Configuration Tool - STM32CubeIDE
File Edit Navigate Search Project Run Window Help
Quick Access
Project Explorer
F746-DISCO-GPIO
Includes
Core
Drivers
F746-DISCO-GPIO.ioc
STM32F746NGHX_FLASH.ld
STM32F746NGHX_RAM.ld
F746-DISCO-GPIO.ioc
Pinout & Configuration
Clock Configuration
Project Manager
Tools
Additional Software
Pinout
Pinout view
System view
Categories
A->Z
System Core
Analog
Timers
Connectivity
Multimedia
Security
Computing
Middleware
PI1
Reset_State
DCMI_D8
FMC_D25
I2S2_CK
LTDC_G6
SPI2_SCK
TIM8_BKIN2
GPIO_Input
GPIO_Output
GPIO_Analog
EVENTOUT
GPIO_EXTI1
TFBGA216 (Top view)
PI1
Stopped

Step 3: 編譯

workspace_1.2.0_MakeNTU - F746-DISCO-GPIO/Core/Src/main.c - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Project Explorer
F746-DISCO-GPIO
Includes
Core
Inc
Src
main.c
stm32f7xx_hal_msp.c
stm32f7xx_it.c
syscalls.c
sysmem.c
system_stm32f7xx.c
Startup
Drivers
F746-DISCO-GPIO.ioc
STM32F746NGHX_FLASH.ld
STM32F746NGHX_RAM.ld
F746-DISCO-GPIO.ioc
main.c
/* USER CODE BEGIN Header */
/**
* @file : main.c
* @brief : Main program body
* @attention
* <h2> © Copyright (c) 2020 STMicroelectronics.
* All rights reserved. </h2>
* This software component is licensed by ST under BSD 3-Clause license,
* the "License"; You may not use this file except in compliance with the
* License. You may obtain a copy of the License at:
opensource.org/licenses/BSD-3-Clause
*/
/* USER CODE END Header */
/* Includes */
#include "main.h"
/* Private includes */
/* USER CODE BEGIN Includes */
/* USER CODE END Includes */
/* Private typedef */
/* USER CODE BEGIN PTD */
/* USER CODE END PTD */
/* Private define */
/* USER CODE BEGIN PD */
/* USER CODE END PD */
/* Private macro */
/* USER CODE BEGIN PM */
/* USER CODE END PM */
/* Private variables */
/* USER CODE BEGIN PV */
/* USER CODE END PV */
/* Private function prototypes */
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
/* USER CODE BEGIN PFP */
/* USER CODE END PFP */
/* Private user code */
/* USER CODE BEGIN 0 */
/* USER CODE END 0 */
Writable
Smart Insert
1 : 1 : 0
Stopped

workspace_1.2.0_MakeNTU - F746-DISCO-GPIO/Core/Src/main.c - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Project Explorer
F746-DISCO-GPIO
Binaries
Includes
Core
Inc
Src
Startup
Drivers
Debug
F746-DISCO-GPIO.ioc
STM32F746NGHX_FLASH.ld
STM32F746NGHX_RAM.ld
Console
CDT Build Console [F746-DISCO-GPIO]
16:53:04 **** Build of configuration Debug for project F746-DISCO-GPIO ****
make -j8 all
arm-none-eabi-gcc "../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal.c" -mcpu=cortex-m7 -std=gnu1
arm-none-eabi-gcc "../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal_cortex.c" -mcpu=cortex-m7 -s
arm-none-eabi-gcc "../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal_dma.c" -mcpu=cortex-m7 -std=
arm-none-eabi-gcc "../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal_dma_ex.c" -mcpu=cortex-m7 -s
arm-none-eabi-gcc "../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal_exti.c" -mcpu=cortex-m7 -std
arm-none-eabi-gcc "../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal_flash.c" -mcpu=cortex-m7 -st
arm-none-eabi-gcc "../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal_flash_ex.c" -mcpu=cortex-m7
arm-none-eabi-gcc "../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal_gpio.c" -mcpu=cortex-m7 -std
arm-none-eabi-gcc "../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal_i2c.c" -mcpu=cortex-m7 -std=
arm-none-eabi-gcc "../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal_i2c_ex.c" -mcpu=cortex-m7 -s
arm-none-eabi-gcc "../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal_pwr.c" -mcpu=cortex-m7 -std=
arm-none-eabi-gcc "../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal_pwr_ex.c" -mcpu=cortex-m7 -s
arm-none-eabi-gcc "../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal_rcc.c" -mcpu=cortex-m7 -std=
arm-none-eabi-gcc "../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal_rcc_ex.c" -mcpu=cortex-m7 -s
arm-none-eabi-gcc "../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal_tim.c" -mcpu=cortex-m7 -std=
arm-none-eabi-gcc "../Drivers/STM32F7xx_HAL_Driver/Src/stm32f7xx_hal_tim_ex.c" -mcpu=cortex-m7 -s
arm-none-eabi-gcc -mcpu=cortex-m7 -g3 -c -x assembler-with-cpp --specs=nano.specs -mfpu=fpv5-sp-d
arm-none-eabi-gcc "../Core/Src/main.c" -mcpu=cortex-m7 -std=gnu11 -g3 -DUSE_HAL_DRIVER -DSTM32F74
arm-none-eabi-gcc "../Core/Src/stm32f7xx_hal_msp.c" -mcpu=cortex-m7 -std=gnu11 -g3 -DUSE_HAL_DRIV
arm-none-eabi-gcc "../Core/Src/stm32f7xx_it.c" -mcpu=cortex-m7 -std=gnu11 -g3 -DUSE_HAL_DRIVER -D
arm-none-eabi-gcc "../Core/Src/syscalls.c" -mcpu=cortex-m7 -std=gnu11 -g3 -DUSE_HAL_DRIVER -DSTM3
arm-none-eabi-gcc "../Core/Src/sysmem.c" -mcpu=cortex-m7 -std=gnu11 -g3 -DUSE_HAL_DRIVER -DSTM32F
arm-none-eabi-gcc "../Core/Src/system_stm32f7xx.c" -mcpu=cortex-m7 -std=gnu11 -g3 -DUSE_HAL_DRIVE
arm-none-eabi-gcc -o "F746-DISCO-GPIO.elf" @"objects.list" -mcpu=cortex-m7 -T"C:\00_myProjects\
Finished building target: F746-DISCO-GPIO.elf
arm-none-eabi-size F746-DISCO-GPIO.elf
arm-none-eabi-objdump -h -S F746-DISCO-GPIO.elf > "F746-DISCO-GPIO.list"
arm-none-eabi-objcopy -O binary F746-DISCO-GPIO.elf "F746-DISCO-GPIO.bin"
text data bss dec hex filename
8100 20 1572 9692 25dc F746-DISCO-GPIO.elf
Finished building: default.size.stdout
Finished building: F746-DISCO-GPIO.bin
Finished building: F746-DISCO-GPIO.list
16:53:13 Build Finished. 0 errors, 0 warnings. (took 9s.551ms)
Stopped

Step 4: 燒錄/除錯

每500ms改變LD1閃滅狀態

CUBE HAL使用建議

- **永遠使用STM32CubeIDE IOC Editor來新增刪減週邊裝置和變更週邊設定**

  如此可以確保**永遠保持程式碼和**IOC檔的設定是一致的

- **不要修改變更週邊的驅動程式stm32f7xx_hal_xxx.c**

  所有在週邊驅動程式所做的修改會在用STM32CUBEMX重新產生程式碼之後被移除

- **將自己的程式碼置於USER CODE區塊當中**

  若將自己的程式碼置於USER CODE區塊之外，則自己的程式碼會在用STM32CUBEMX重新產生程式碼之後被移除

CUBE HAL使用建議

- **永遠使用STM32CubeIDE IOC Editor來新增刪減週邊裝置和變更週邊設定**

  如此可以確保**永遠保持程式碼和IOC檔的設定是一致的**

- **不要修改變更週邊的驅動程式stm32f7xx_hal_xxx.c**

  所有在週邊驅動程式所做的修改會在用

- **將自己的程式碼置於USER CODE區塊當中**

  若將自己的程式碼置於USER CODE區塊之
  新產生程式碼之後被移除

CUBE HAL使用建議

- **永遠使用STM32CubeIDE IOC Editor來新增刪減週邊裝置和變更週邊設定**

  如此可以確保永遠保持程式碼和IOC檔的設定是一致的

- **不要修改變更週邊的驅動程式stm32f7xx_hal_xxx.c**

  所有在週邊驅動程式所做的修改會在用STM32CubeIDE重新產生程式碼之後被移除

- **將自己的程式碼置於USER CODE區塊當中**

  若將自己的程式碼置於USER CODE區塊之外，則自己的程式碼會在用STM32CUBEMX重新產生程式碼之後被移除

CUBE HAL使用建議

- **永遠使用STM32CubeIDE IOC Editor來新增刪減週邊裝置和變更週邊設定**

  如此可以確保永遠保持程式碼和IOC檔的設定是一致的

- **不要修改變更週邊的驅動程式stm32f7xx_hal_xxx.c**

  所有在週邊驅動程式所做的修改會在用STM32CUBEMX重新產生程式碼之後被移除

- **將自己的程式碼置於USER CODE區塊當中**

  若將自己的程式碼置於USER CODE區塊之外，則自己的程式碼會在用STM32CUBEMX重新產生程式碼之後被移除

```
int main(void)
{
  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */
  …

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */
  …
  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */
  …
  /* USER CODE BEGIN 2 */

  /* USER CODE END 2 */
  …
  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    /* USER CODE BEGIN 3 */
      HAL_Delay(500);
      HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
  }
  /* USER CODE END 3 */
}
```

life.augmented

補充：如何查找HAL API

搜尋Doxygen文件 (以搜尋GPIO HAL API為例)

Doxygen文件置於目錄: STM32F7 MCU Package Repository\Drivers\STM32F7xx_HAL_Driver

life.augmented

補充：如何查找HAL API

從STM32CubeIDE的Outline視窗 (以搜尋GPIO HAL API為例)

STM32CubeIDE基本功能介紹

life.augmented

功能體驗 – Auto-Complete (1)

欲輸入HAL_Delay()

功能體驗 – Auto-Complete (2)
組合鍵查詢

功能體驗 – Outline View

workshop - 01_F746-DISCO/Src/main.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

- 01_F746-DISCO
  - Binaries
  - Includes
  - Drivers
  - Src
    - main.c
    - stm32f7xx_hal_msp.c
    - stm32f7xx_it.c
    - syscalls.c
    - sysmem.c
    - system_stm32f7xx.c
  - Startup
  - Debug
  - semihost
  - Inc
  - 01_F746-DISCO_JLINK.elf.launch
  - 01_F746-DISCO_STLINK-OPENOCD.launch
  - 01_F746-DISCO.elf.cfg
  - 01_F746-DISCO.elf.launch
  - 01_F746-DISCO.ioc
  - STM32F746NGHX_FLASH.ld
  - STM32F746NGHX_RAM.ld

main.c | startup_stm32f746nghx.s

```
702   GPIO_InitStruct.Pull = GPIO_NOPULL;
703   GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
704   GPIO_InitStruct.Alternate = GPIO_AF5_SPI2;
705   HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
706
707   /* EXTI interrupt init*/
708   HAL_NVIC_SetPriority(EXTI15_10_IRQn, 0, 0);
709   HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);
710
711 }
712
713 /* USER CODE BEGIN 4 */
714 void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
715 {
716   static uint8_t testState = 0x00;
717   testState ^= 0x01;
718 }
719
720 #ifdef __GNUC__
721 #define PUTCHAR_PROTOTYPE int __io_putchar(int ch)
722 #else
723   #define PUTCHAR_PROTOTYPE int fputc(int ch, FILE *f)
724 #endif /* __GNUC__ */
725 PUTCHAR_PROTOTYPE
726 {
727   //HAL_UART_Transmit(&huart1, (uint8_t *)&ch, 1, 0xFFFF);
728   ITM_SendChar((uint32_t)ch);
729   return ch;
730 }
731 /* USER CODE END 4 */
732
733 /**
734   * @brief  This function is executed in case of error occurrence.
735   * @retval None
736   */
737 void Error_Handler(void)
738 {
739   /* USER CODE BEGIN Error_Handler_Debug */
740   /* User can add his own implementation to report the HAL error return state */
741
742   /* USER CODE END Error_Handler_Debug */
743 }
744
745 #ifdef  USE_FULL_ASSERT
746 /**
747   * @brief  Reports the name of the source file and the source line number
748   *         where the assert_param error has occurred.
749   * @param  file: pointer to the source file name
750   * @param  line: assert_param error line source number
751   * @retval None
752   */
753 void assert_failed(uint8_t *file, uint32_t line)
754 {
755   /* USER CODE BEGIN 6 */
756   /* User can add his own implementation to report the file name and line number,
757      tex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
758   /* USER CODE END 6 */
759 }
760 #endif /* USE_FULL_ASSERT */
```

Outline | Build Targets

- main.h
- stdio.h
- _newlib_version.h
- huart1 : UART_HandleTypeDef
- SystemClock_Config(void) : void
- MX_GPIO_Init(void) : void
- MX_USART1_UART_Init(void) : void
- initialise_monitor_handles(void) : void
- main(void) : int
- SystemClock_Config(void) : void
- MX_USART1_UART_Init(void) : void
- MX_GPIO_Init(void) : void
- HAL_GPIO_EXTI_Callback(uint16_t) : void
- PUTCHAR_PROTOTYPE
- PUTCHAR_PROTOTYPE
- __io_putchar(int) : int
- Error_Handler(void) : void
- assert_failed(uint8_t*, uint32_t) : void

life.augmented

功能體驗 – 巨集的展開 (1)

workshop - 01_F746-DISCO/Src/main.c - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Project Explorer
01_F746-DISCO
Binaries
Includes
Drivers
Src
main.c
stm32f7xx_hal_msp.c
stm32f7xx_it.c
syscalls.c
sysmem.c
system_stm32f7xx.c
Startup
Debug
semihost
Inc
01_F746-DISCO_JLINK.elf.launch
01_F746-DISCO_STLINK-OPENOCD
01_F746-DISCO.elf.cfg
01_F746-DISCO.elf.launch
01_F746-DISCO.ioc
STM32F746NGHX_FLASH.ld
STM32F746NGHX_RAM.ld
main.c
/* USER CODE END 2 */
/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
HAL_Delay(500);
HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
printf("LD1: %d\n\
Macro Expansion
((GPIO_TypeDef *) ((0x40000000UL + 0x00020000UL) + 0x2000UL))
Press 'F2' for macro expansion step
void SystemClock_Config(void)
RCC_OscInitTypeDef RCC_OscInitStruct = {0};
RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
RCC_PeriphCLKInitTypeDef PeriphClkInitStruct = {0};
__HAL_RCC_PWR_CLK_ENABLE();
__HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE2);
RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
RCC_OscInitStruct.HSIState = RCC_HSI_ON;
RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
RCC_OscInitStruct.PLL.PLLM = 10;
RCC_OscInitStruct.PLL.PLLN = 210;
RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
RCC_OscInitStruct.PLL.PLLQ = 2;
if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
Error_Handler();
RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV4;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV2;
if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_5) != HAL_OK)
Error_Handler();
Writable
Smart Insert
112 : 28
1 將滑鼠游標停在欲查詢處
2 一秒後出現巨集展開的結果

功能體驗 – 巨集的展開 (2)

workshop - 01_F746-DISCO/Src/main.c - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Project Explorer
main.c
選取欲查詢的巨集
1
2
3
Undo Ctrl+Z
Revert File
Save Ctrl+S
Open Declaration F3
Open Type Hierarchy F4
Open Call Hierarchy Ctrl+Alt+H
Quick Outline Ctrl+O
Quick Type Hierarchy Ctrl+T
Explore Macro Expansion Ctrl+=
Toggle Source/Header Ctrl+Tab
Open With
Show In Alt+Shift+W
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Quick Fix Ctrl+1
Source Alt+Shift+S
Surround With Alt+Shift+Z
Refactor
Declarations
References
Search Text
Resource Configurations
Run As
Debug As
Profile As
Team
Compare With
Replace With
Preferences...
Writable
Smart Insert
114 : 29
Explore Macro Expansion - 4 step(s)
#define GPIOI ((GPIO_TypeDef *) GPIOI_BASE)
Original
GPIOI
Fully Expanded
((GPIO_TypeDef *) ((0x40000000UL + 0x00020000UL) + 0x2000UL))

功能體驗 – 巨集的展開 (3)

life.augmented

功能體驗 – Symbol Hyperlink

基本功能 – 編譯專案（Build Project）的方式

Option 1

Option 2

Option 3

基本功能 – 開啟編譯設定視窗的方式

Option 1

Option 2

life.augmented

基本功能 – 最佳化設定

1 開啟編譯設定視窗

基本功能 – 產生Hex或BIN檔

1 開啟編譯設定視窗

基本功能 – 執行專案的方式

Option 1

Option 2

Option 3

基本功能 – 設置中斷點

workshop - 01_F746-DISCO/Src/main.c - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Debug
Project E...
01_F746-DISCO.elf [STM32 MCU De
01_F746-DISCO.elf [cores: 0]
Thread #1 [main] 1 [core: 0] (
HAL_GetTick() at stm32f7x
HAL_Delay() at stm32f7xx_
main() at main.c:102 0x80(
C:/ST/STM32CubeIDE_1.0.2/STM
ST-LINK (ST-LINK GDB server)
01_F746-DISCO.ioc
main.c
x2
/* Configure the system clock */
SystemClock_Config();
/* USER CODE BEGIN SysInit */
/* USER CODE END SysInit */
/* Initialize all configured peripherals */
MX_GPIO_Init();
/* USER CODE BEGIN 2 */
/* USER CODE END 2 */
/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
/* USER CODE END WHILE */
/* USER CODE BEGIN 3 */
HAL_Delay(500);
HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
/* USER CODE END 3 */
}
/**
* @brief System Clock Configuration
* @retval None
*/
void SystemClock_Config(void)
{
RCC_OscInitTypeDef RCC_OscInitStruct = {0};
RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
/** Configure the main internal regulator output voltage
*/
__HAL_RCC_PWR_CLK_ENABLE();
__HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE2);
/** Initializes the CPU, AHB and APB busses clocks
*/
Console
Registers
Problems
Executables
Debugger Console
Memory
01_F746-DISCO.elf [STM32 MCU Debugging] ST-LINK (ST-LINK GDB server)
Erasing internal memory sector 0
Download in Progress:
File download complete
Time elapsed during download operation: 00:00:00.683
Writable
Smart Insert
76 : 1

基本功能 – Live Expressions

基本功能 – Registers

基本功能 – SFR

workshop - 01_F746-DISCO/Src/main.c - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
New Window
Editor
Appearance
Show View
Perspective
Navigation
Preferences
SWV
Breakpoints Alt+Shift+Q, B
Build Analyzer
Console Alt+Shift+Q, C
Debug
Debugger Console
Disassembly
Error Log Alt+Shift+Q, L
Executables
Expressions
Fault Analyzer
Live Expressions
Memory
Memory Browser
Modules
Outline Alt+Shift+Q, O
Problems Alt+Shift+Q, X
Progress
Project Explorer
Registers
SFRs
Signals
Static Stack Analyzer
Templates
Trace Control
Variables Alt+Shift+Q, V
Other... Alt+Shift+Q, Q
1
2
Registers
SFRs
Memory
Register
Address
Value
RNG
HASH
CRYP
DCMI
FMC
DBG
DMA2
DMA1
RCC
GPIOD
GPIOC
GPIOK
GPIOJ
GPIOI
GPIOH
GPIOG
GPIOF
GPIOE
GPIOB
GPIOA
SYSCFG
WWDG
C_ADC
TIM1
TIM8
TIM2
藉由SFRs視窗可以
觀察週邊暫存器的內容
NOTE:
SFR = Special unction Registers

基本功能 – Memory

workshop - 01_F746-DISCO/Src/main.c - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
New Window
Editor
Appearance
Show View
Perspective
Navigation
Preferences
SWV
Breakpoints Alt+Shift+Q, B
Build Analyzer
Console Alt+Shift+Q, C
Debug
Debugger Console
Disassembly
Error Log Alt+Shift+Q, L
Executables
Expressions
Fault Analyzer
Live Expressions
Memory
Memory Browser
Modules
Outline Alt+Shift+Q, O
Problems Alt+Shift+Q, X
Progress
Project Explorer
Registers
SFRs
Signals
Static Stack Analyzer
Templates
Trace Control
Variables Alt+Shift+Q, V
Other... Alt+Shift+Q, Q
1
2
藉由Memory視窗可以觀察
指定的記憶體位址HEX內容

基本功能 – Memory Browser

功能體驗 – 使用Dynamic Printf

!? 補充 – Tool Settings消失了？！

life.augmented

Hands-on
02_GPIO_Interrupt

life.augmented

實作目標

- 按下藍色按鈕後改變LD1閃滅狀態
- 同時在Debug Console看到狀態訊息

流程

1. 設定腳位功能並產生程式碼
2. 新增程式碼
3. 重新編譯
4. 燒錄/除錯
5. 使用除錯功能

life.augmented

Step 4: 燒錄/除錯

中斷處理說明

- Blue Button = PI11
- LD1 = PI1
- Blue button pressed

→ PI11 goes from HIGH to LOW,
→ PI11 rising edge detected
→ EXTI11 fires

→ EXTI15_10_IRQHandler(){…} //in stm32f7xx_it.c

→ HAL_GPIO_EXTI_IRQHandler(KEY_BUTTON_PIN) //in stm32f7xx_hal_gpio.c

→ HAL_GPIO_EXTI_Callback(){…} //in main.c
→ HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1); //in main.c

life.augmented

功能體驗 – 使用Dynamic Printf

Hands-on
03_UART_printf

life.augmented

實作目標

每500ms改變LD1閃滅狀態，
並透過終端機顯示LD1狀態

流程

1. 由STM32CubeIDE產生UART專案
2. 開啟UART1
3. 加入程式碼/重新編譯/燒錄
4. 開啟終端機檢視UART輸出

Step 1: 另存CUBEMX專案

由HandsOn 2另存CUBEMX Project

Step 2: 開啟UART1

於Pinout & Configuration分頁中的Connectivity項目開啟USART1

Step 3: 加入程式碼/重新編譯/燒錄

於main.c加入程式碼

STM32_workspace_9.3 - C/C++ - HandsOn3/Application/User/main.c - Atollic TrueSTUDIO for STM32

```
699   GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
700   GPIO_InitStruct.Pull = GPIO_NOPULL;
701   GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
702   GPIO_InitStruct.Alternate = GPIO_AF5_SPI2;
703   HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
704
705   /* EXTI interrupt init*/
706   HAL_NVIC_SetPriority(EXTI15_10_IRQn, 0, 0);
707   HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);
708
709 }
710
711 /* USER CODE BEGIN 4 */
712 int __io_putchar(int ch)
713 {
714   HAL_UART_Transmit(&huart1, (uint8_t *)&ch, 1, 0xFFFF);
715   return ch;
716 }
717
718 /* USER CODE END 4 */
```

1

```
int __io_putchar(int ch)
{
  HAL_UART_Transmit(&huart1, (uint8_t *)&ch, 1, 0xFFFF);
  return ch;
}
```

```
730 }
731
732 #ifdef  USE_FULL_ASSERT
733 /**
734   * @brief  Reports the name of the source file and the source line number
735   *         where the assert_param error has occurred.
736   * @param  file: pointer to the source file name
```

STM32_workspace_9.3 - C/C++ - HandsOn3/Application/User/main.c - Atollic TrueSTUDIO for STM32

```
10   * All rights reserved.</center></h2>
11   *
12   * This software component is licensed by ST under BSD 3-Clause license,
13   * the "License"; You may not use this file except in compliance with the
14   * License. You may obtain a copy of the License at:
15   *                        opensource.org/licenses/BSD-3-Clause
16   *
17   ******************************************************************************
18   */
19 /* USER CODE END Header */
20
21 /* Includes ------------------------------------------------------------------*/
22 #include "main.h"
23
24 /* Private includes ----------------------------------------------------------*/
25 /* USER CODE BEGIN In
26 #include <stdio.h>
27 /* USER CODE END Incl
```

2

```
#include <stdio.h>
```

```
92   MX_GPIO_Init();
93   MX_USART1_UART_Init();
94   /* USER CODE BEGIN 2 */
95
96   /* USER CODE END 2 */
```

```
HAL_Delay(500);
HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
printf("LD1 : %d\n\r", HAL_GPIO_ReadPin(GPIOI, GPIO_PIN_1));
```

3

```
104     /* USER CODE BEGIN 3 */
105     HAL_Delay(500);
106     HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
107     printf("LD1 : %d\n\r", HAL_GPIO_ReadPin(GPIOI, GPIO_PIN_1));
108   }
109   /* USER CODE END 3 */
110 }
```

life.augmented

Step 3: 加入程式碼/重新編譯/燒錄

於main.c加入程式碼

STM32_workspace_9.3 - C/C++ - HandsOn3/Application/User/main.c - Atollic TrueSTUDIO for STM32

File Edit Source Refactor View Navigate Search Project Run Window Help

Project Explorer

- HandsOn1
- HandsOn2
- HandsOn3
  - Binaries
  - Includes
  - Application
    - TrueSTUDIO
    - User
      - main.c
      - stm32f7xx_hal_msp.c
      - stm32f7xx_it.c
  - Debug
  - Drivers
  - HandsOn3.elf.launch
  - STM32F746NG_FLASH.ld

1

main.c

```c
  GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  GPIO_InitStruct.Alternate = GPIO_AF5_SPI2;
  HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

  /* EXTI interrupt init*/
  HAL_NVIC_SetPriority(EXTI15_10_IRQn, 0, 0);
  HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);

}

/* USER CODE BEGIN 4 */
int __io_putchar(int ch)
{
  HAL_UART_Transmit(&huart1, (uint8_t *)&ch, 1, 0xFFFF);
  return ch;
}

/* USER CODE END 4 */
```

2

```c
int __io_putchar(int ch)
{
  HAL_UART_Transmit(&huart1, (uint8_t *)&ch, 1, 0xFFFF);
  return ch;
}
```

```c
}

#ifdef USE_FULL_ASSERT
/**
  * @brief  Reports the name of the source file and the source line number
  *         where the assert_param error has occurred.
  * @param  file: pointer to the source file name
```

STM32_workspace_9.3 - C/C++ - HandsOn3/Application/User/main.c - Atollic TrueSTUDIO for STM32

File Edit Source Refactor View Navigate Search Project Run Window Help

Project Explorer

- HandsOn1
- HandsOn2
- HandsOn3
  - Binaries
  - Includes
  - Application
    - TrueSTUDIO
    - User
  - Debug
  - Drivers
    - CMSIS
    - STM32F7xx_HAL_Driver
  - HandsOn3.elf.launch
  - STM32F746NG_FLASH.ld

3

main.c

```c
  * All rights reserved.</center></h2>
  *
  * This software component is licensed by ST under BSD 3-Clause license,
  * the "License"; You may not use this file except in compliance with the
  * License. You may obtain a copy of the License at:
  *                        opensource.org/licenses/BSD-3-Clause
  *
  ******************************************************************************
  */
/* USER CODE END Header */

/* Includes ------------------------------------------------------------------*/
#include "main.h"

/* Private includes ----------------------------------------------------------*/
/* USER CODE BEGIN Inc
#include <stdio.h>
/* USER CODE END Incl
```

```c
#include <stdio.h>
```

```c
  MX_GPIO_Init();
  MX_USART1_UART_Init();
  /* USER CODE BEGIN 2 */

  /* USER CODE END 2 */
```

```c
HAL_Delay(500);
HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
printf("LD1 : %d\n\r", HAL_GPIO_ReadPin(GPIOI, GPIO_PIN_1));
```

4

```c
    /* USER CODE BEGIN 3 */
    HAL_Delay(500);
    HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
    printf("LD1 : %d\n\r", HAL_GPIO_ReadPin(GPIOI, GPIO_PIN_1));
  }
  /* USER CODE END 3 */
}
```

life.augmented

Step 3: 加入程式碼/重新編譯/燒錄

重新編譯/燒錄

Step 4: 開啟終端機檢視UART輸出

以TeraTerm為例

本次工作坊範例使用的終端機為Tera-Term， 可由以下網址下載：
https://ttssh2.osdn.jp/index.html.en

Hands-on
04_UART_Receiving_Polling

life.augmented

流程

1. 加入程式碼
2. 重新編譯/燒錄
3. 開啟終端機檢視執行結果

life.augmented

Step 3: 開啟終端機檢視執行結果

在終端機下打字， 並觀察uart1_rx_buffer

Step 3: 開啟終端機檢視執行結果

在終端機下打字， 並觀察uart1_rx_buffer

life.augmented

Hands-on
05_UART_Receiving_IT

life.augmented

流程

1. 開啟USART1中斷
2. 加入程式碼/重新編譯/燒錄
3. 開啟終端機檢視執行結果

life.augmented

Step 1: 開啟USART1中斷

開啟USART1中斷

Step 3: 開啟終端機檢視執行結果

在終端機下打字， 並觀察uart1_rx_buffer

Hands-on
06_UART_Receiving_DMA

life.augmented

流程

1. 開啟USART1 DMA
2. 加入程式碼/重新編譯/燒錄
3. 開啟終端機檢視執行結果

life.augmented

Step 1: 開啟USART1 DMA

開啟USART1 DMA

Step 3: 開啟終端機檢視執行結果

在終端機下打字， 並觀察uart1_rx_buffer

Hands-on
07_TIMER

life.augmented

程式碼實作目標

每500ms變LD1閃滅狀態

流程

1. 由STM32CubeIDE產生TIMER專案
2. 開啟GPIO和TIM6
3. 加入程式碼/重新編譯/燒錄
4. 檢視執行結果

Step 2: 開啟GPIO和TIM6

PI1 = GPIO-Output

TIM6

Timer功能方塊示意圖

以Basic Timer (TIM6)為例

Notes:

Reg — Preload registers transferred to active registers on U event according to control bit

Event

Interrupt & DMA output

$$UI\ Frequency\ (Hz) = \frac{TIMxCLK\ (Hz)}{(ARR+1)\times(PSC+1)}$$

MS33142V1

TIMxCLK = ?

查閱RM0385

RM0385

Table 1. STM32F75xxx and STM32F74xxx register boundary addresses (continued)

Boundary address	Peripheral	Bus	Register map
0x4000 7C00 - 0x4000 7FFF	UART8		Section 31.8.12: USART register map on page 1048
0x4000 7800 - 0x4000 7BFF	UART7		
0x4000 7400 - 0x4000 77FF	DAC		Section 16.5.15: DAC register map on page 479
0x4000 7000 - 0x4000 73FF	PWR		Section 4.4.4: PWR power control register 2 (PWR_CSR2) on page 124
0x4000 6C00 - 0x4000 6FFF	HDMI-CEC		Section 39.7.7: HDMI-CEC register map on page 1620
0x4000 6800 - 0x4000 6BFF	CAN2		Section 36.9.5: bxCAN register map on page 1292
0x4000 6400 - 0x4000 67FF	CAN1		
0x4000 6000 - 0x4000 63FF	I2C4		Section 30.7.12: I2C register map on page 984
0x4000 5C00 - 0x4000 5FFF	I2C3		Section 30.7.12: I2C register map on page 984
0x4000 5800 - 0x4000 5BFF	I2C2		
0x4000 5400 - 0x4000 57FF	I2C1		
0x4000 5000 - 0x4000 53FF	UART5		Section 31.8.12: USART register map on page 1048
0x4000 4C00 - 0x4000 4FFF	UART4		
0x4000 4800 - 0x4000 4BFF	USART3		
0x4000 4400 - 0x4000 47FF	USART2		
0x4000 4000 - 0x4000 43FF	SPDIFRX	APB1	1 Section [illegible] SPDIFRX interface register map on page [illegible]
0x4000 3C00 - 0x4000 3FFF	SPI3 / I2S3		Section 32.9.10: SPI/I2S register map on page 1109
0x4000 3800 - 0x4000 3BFF	SPI2 / I2S2		
0x4000 3000 - 0x4000 33FF	IWDG		
0x4000 2C00 - 0x4000 2FFF	WWDG		
0x4000 2800 - 0x4000 2BFF	RTC & BKP Registers		
0x4000 2400 - 0x4000 27FF	LPTIM1		
0x4000 2000 - 0x4000 23FF	TIM14		Section 24.5.12: TIM10/TIM11/TIM13/TIM14 register map on page 820
0x4000 1C00 - 0x4000 1FFF	TIM13		
0x4000 1800 - 0x4000 1BFF	TIM12		Section 24.4.13: TIM9/TIM12 register map on page 810
0x4000 1400 - 0x4000 17FF	TIM7		Section 25.4.9: TIM6/TIM7 register map on page 835
0x4000 1000 - 0x4000 13FF	TIM6		
0x4000 0C00 - 0x4000 0FFF	TIM5		Section 23.4.21: TIMx register map on page 771
0x4000 0800 - 0x4000 0BFF	TIM4		
0x4000 0400 - 0x4000 07FF	TIM3		
0x4000 0000 - 0x4000 03FF	TIM2		

查閱RM得到
TIM6為APB1-Bus

Prescaler = ?
Counter Period = ?

$$UI\ Frequency\ (Hz) = \frac{TIM6CLK\ (Hz)}{(ARR+1)\times(PSC+1)}$$

$$2\ Hz = \frac{84\ MHz}{(ARR+1)\times(PSC+1)}$$

$$\therefore (ARR+1)\times(PSC+1) = \frac{84MHz}{2Hz} = 42000000$$

取 (ARR+1) = 10000，　則(PSC+1) = 4200

$$\therefore ARR = 9999, \qquad PSC = 4199$$

加入程式碼

```
int main(void)
{
 /* USER CODE BEGIN 2 */
  HAL_TIM_Base_Start_IT(&htim6);
  /* USER CODE END 2 */
 …
  while(1){}
}

/* USER CODE BEGIN 4 */
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef*htim)
{
  HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
}
/* USER CODE END 4 */
```

```
void TIM6_DAC_IRQHandler(void)
{
  HAL_TIM_IRQHandler(&htim6);
}
```

```
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef*htim)
{
  HAL_GPIO_TogglePin(GPIOI, GPIO_PIN_1);
}
```

life.augmented

Hands-on
08_TIMER-PWM

life.augmented

程式碼實作目標

每500ms變LD1閃滅狀態

流程

1. 設定GPIO、關閉TIM6並開啟TIM4
2. 加入程式碼/重新編譯/燒錄
3. 檢視執行結果

life.augmented

Step 2: 設定PI1和關閉TIM6

PI1 = GPIO-Input

TIM6

Step 2: 設定TIM4

PB8 = TIM4_CH3 (PWM Output)

TIM4-CH3

此除頻設定會讓TIM4產生週期為1Hz且duty cycle為50%的PWM訊號由PB8輸出

life.augmented

PI1 = GPIO-Input
PB8 = TIM4_CH3 (PWM Output)

Timer-PWM功能方塊示意圖

以General Purpose Timer (TIM4)為例

$$PWM\ Frequency\ (Hz) = \frac{TIMxCLK\ (Hz)}{(ARR + 1) \times (PSC + 1)}$$

$$PWM\ Duty\ Cycle\ (\%) = \frac{(Pulse + 1)}{(ARR + 1)}$$

TIMxCLK = ?

查閱RM0385

RM0385

Table 1. STM32F75xxx and STM32F74xxx register boundary addresses (continued)

Boundary address	Peripheral	Bus	Register map
0x4000 7C00 - 0x4000 7FFF	UART8		Section 31.8.12: USART register map on page 1048
0x4000 7800 - 0x4000 7BFF	UART7		
0x4000 7400 - 0x4000 77FF	DAC		Section 16.5.15: DAC register map on page 479
0x4000 7000 - 0x4000 73FF	PWR		Section 4.4.4: PWR power control register 2 (PWR_CSR2) on page 124
0x4000 6C00 - 0x4000 6FFF	HDMI-CEC		Section 39.7.7: HDMI-CEC register map on page 1620
0x4000 6800 - 0x4000 6BFF	CAN2		Section 36.9.5: bxCAN register map on page 1292
0x4000 6400 - 0x4000 67FF	CAN1		
0x4000 6000 - 0x4000 63FF	I2C4		Section 30.7.12: I2C register map on page 984
0x4000 5C00 - 0x4000 5FFF	I2C3		Section 30.7.12: I2C register map on page 984
0x4000 5800 - 0x4000 5BFF	I2C2		
0x4000 5400 - 0x4000 57FF	I2C1		
0x4000 5000 - 0x4000 53FF	UART5		Section 31.8.12: USART register map on page 1048
0x4000 4C00 - 0x4000 4FFF	UART4		
0x4000 4800 - 0x4000 4BFF	USART3		
0x4000 4400 - 0x4000 47FF	USART2		
0x4000 4000 - 0x4000 43FF	SPDIFRX	APB1	Section [illegible] SPDIFRX interface register map on page [illegible]
0x4000 3C00 - 0x4000 3FFF	SPI3 / I2S3		Section 32.9.10: SPI/I2S register map on page 1109
0x4000 3800 - 0x4000 3BFF	SPI2 / I2S2		
0x4000 3000 - 0x4000 33FF	IWDG		
0x4000 2C00 - 0x4000 2FFF	WWDG		
0x4000 2800 - 0x4000 2BFF	RTC & BKP Registers		
0x4000 2400 - 0x4000 27FF	LPTIM1		
0x4000 2000 - 0x4000 23FF	TIM14		Section 24.5.12: TIM10/TIM11/TIM13/TIM14 register map on page 820
0x4000 1C00 - 0x4000 1FFF	TIM13		
0x4000 1800 - 0x4000 1BFF	TIM12		Section 24.4.13: TIM9/TIM12 register map on page 810
0x4000 1400 - 0x4000 17FF	TIM7		Section 25.4.9: TIM6/TIM7 register map on page 835
0x4000 1000 - 0x4000 13FF	TIM6		
0x4000 0C00 - 0x4000 0FFF	TIM5		Section 23.4.21: TIMx register map on page 771
0x4000 0800 - 0x4000 0BFF	TIM4		
0x4000 0400 - 0x4000 07FF	TIM3		
0x4000 0000 - 0x4000 03FF	TIM2		

1

查閱RM得到
TIM6為APB1-Bus

PSC / ARR / Pulse = ?

- $PWM\ Frequency\ (Hz) = \frac{TIMxCLK\ (Hz)}{(ARR+1)\times(PSC+1)}$

$$1\ Hz = \frac{84000000}{(ARR+1)\times(PSC+1)}$$

$$\therefore (ARR+1)\times(PSC+1) = 84000000$$

取 (ARR+1) = 10000， 則(PSC+1) = 8400

$$\therefore ARR = 9999, \qquad PSC = 8399$$

- $PWM\ Duty\ Cycle\ (\%) = \frac{(Pulse+1)}{(ARR+1)}$

$$50\% = \frac{(Pulse+1)}{10000} \quad \therefore Pulse = 4999$$

加入程式碼

```
/* USER CODE BEGIN 2 */
HAL_TIM_PWM_Start(&htim4, TIM_CHANNEL_3);
/* USER CODE END 2 */
```

life.augmented

軟體安裝
X-CUBE-TOUCHGFX的安裝

life.augmented

X-CUBE-TouchGFX的安裝 – 1

https://my.st.com/content/my_st_com/en/products/embedded-software/mcu-mpu-embedded-software/stm32-embedded-software/stm32cube-expansion-packages/x-cube-touchgfx.license=1582225224468.product=X-CUBE-TOUCHGFX.version=4.13.0.html#get-software

life.augmented

X-CUBE-TouchGFX的安裝 – 2

X-CUBE-TouchGFX的安裝 – 3

STM32CubeIDE Repository的位置

STM32CubeIDE-Reposiboty\Packs\STMicroelectronics\X-CUBE-TOUCHGFX\4.13.0\Utilities\PC_Software\TouchGFXDesigner

Windows

macOS

X-CUBE-TouchGFX的安裝 – 4

STM32CubeIDE Repository的位置

life.augmented

Hands-on
09_TouchGFX

life.augmented

實作目標

流程說明

使用STM32CubeIDE產生專案

- **開啟X-CUBE-TouchGFX Plugin**
- 開啟Cache
- 關閉ETH/關閉USB_OTG_FS/FATFS
- 調整LTDC /FreeRTOS設定
- **設定X-CUBE-TouchGFX Plugin**
- 產生專案
- **執行TouchGFX Designer**

STM32CubeIDE/IOC-Editor

使用TouchGFX Designer設計GUI

- 新增專案所需圖片
- 新增Box元件
- 新增Image元件
- 新增Toggle Button元件
- 對Toggle Button元件新增Interaction
- 產生TouchGFX程式碼

TouchGFX Designer

修改TouchGFX程式碼並用模擬器驗證

- 修改Screen1Presenter.hpp
- 修改Screen1Presenter.cpp
- 修改Screen1View.hpp
- 修改Screen1View.cpp
- 使用模擬器驗證互動行為

STM32CubeIDE/Code-Editor及TouchGFX Designer模擬器

新增BSP以啟用觸控功能

- 在專案目錄下新增BSP目錄及所需檔案
- 為BSP相關程式碼作編譯相關的設定
- 修改程式碼以啟用觸控和顯示功能 (main.c和STM32F7TouchController.cpp)

STM32CubeIDE

編譯並執行

- 編譯
- 燒錄
- 執行

STM32CubeIDE

TouchGFX Designer – 1

TouchGFX Designer – 2

MVP (Model – View - Presenter) 模型

程式碼說明 (1)

```cpp
Screen1ViewBase::Screen1ViewBase() :
    buttonCallback(this, &Screen1ViewBase::buttonCallbackHandler)
{
    background.setPosition(0, 0, 480, 272);
    background.setColor(touchgfx::Color::getColorFrom24BitRGB(5, 250, 163));

    bulb.setXY(134, 0);
    bulb.setBitmap(Bitmap(BITMAP_BULB_OFF_ID));

    power_button.setXY(176, 234);
    power_button.setBitmaps(Bitmap(BITMAP_BLUE_TOGGLEBARS_TOGGLE_ROUND_LARGE_BUTTON_OFF_ID), Bitmap(BITMAP_BLUE_TOGGLEBARS_TOGGLE_ROUND_LARGE_BUTTON_ON_ID));
    power_button.setAction(buttonCallback);

    add(background);
    add(bulb);
    add(power_button);
}

void Screen1ViewBase::setupScreen()
{

}

void Screen1ViewBase::buttonCallbackHandler(const touchgfx::AbstractButton& src)
{
    if (&src == &power_button)
    {
        //Interaction1
        //When power_button clicked execute C++ code
        //Execute C++ code
        presenter->toggleLight();
    }
}
```

```cpp
#include <gui/screen1_screen/Screen1View.hpp>
#include <gui/screen1_screen/Screen1Presenter.hpp>

Screen1Presenter::Screen1Presenter(Screen1View& v)
    : view(v)
{
}

void Screen1Presenter::activate()
{
  lightState = false;
}

void Screen1Presenter::deactivate()
{

}

void Screen1Presenter::toggleLight()
{
    if (lightState == false)
    {
      lightState = true;
      view.switchBulbOn();
    }
    else
    {
      lightState = false;
      view.switchBulbOff();
    }
}
```

```cpp
#include <gui/screen1_screen/Screen1View.hpp>
#include "BitmapDatabase.hpp"

Screen1View::Screen1View()
{

}

void Screen1View::setupScreen()
{
    Screen1ViewBase::setupScreen();
}

void Screen1View::tearDownScreen()
{
    Screen1ViewBase::tearDownScreen();
}

void Screen1View::switchBulbOn()
{
  bulb.setBitmap(Bitmap(BITMAP_BULB_ON_ID));
  bulb.invalidate();
}

void Screen1View::switchBulbOff()
{
  bulb.setBitmap(Bitmap(BITMAP_BULB_OFF_ID));
  bulb.invalidate();
}
```

程式碼說明 (2)

```
void Screen1Presenter::toggleLight()
{
    if (lightState == false)
    {
      lightState = true;
      view.switchBulbOn();
    }
    else
    {
      lightState = false;
      view.switchBulbOff();
    }
}
```

GUI操作邏輯

更新

狀態改變

使用者事件

更新

硬體介面

GUI外觀

```
void Screen1View::switchBulbOn()
{
  bulb.setBitmap(Bitmap(BITMAP_BULB_ON_ID));
  bulb.invalidate();
}

void Screen1View::switchBulbOff()
{
  bulb.setBitmap(Bitmap(BITMAP_BULB_OFF_ID));
  bulb.invalidate();
}
```

參考資料

- **TouchGFX 官方文件**
  https://support.touchgfx.com/docs/introduction/welcome

- **MOOC - STM32 Graphics Workshop （TouchGFX課程Youtube播放清單）**
  https://www.youtube.com/watch?v=4KNVNqOZHG0&list=PLnMKNibPkDnGNY9Ny4FWpM5bgC3b4XdbN

Thank you

© STMicroelectronics - All rights reserved.
ST logo is a trademark or a registered trademark of STMicroelectronics International NV or its affiliates in the EU and/or other countries.
For additional information about ST trademarks, please refer to www.st.com/trademarks.
All other product or service names are the property of their respective owners.

life.augmented