

سوالات تحلیلی:

۱- اگر تنظیمات میکرو خود را در میان کار تغییر دهیم، هنگام تولید مجدد کد در STM32CubeMX چگونه میتوان از از بین رفتن کدهای توسعه داده شده خود جلوگیری کنیم؟

در خود نرم افزار STM32CubeMX در قسمت Project Manager ، در قسمت Code Generator ، در گزینه های مربوط به Generated Files یک گزینه هست به نام Keep User Code When Re-generating. اگر این گزینه را فعال کنیم و در فایل ها نیز در قسمت هایی که با کامنت مشخص شده اند و مربوط به کد های یوزرز هستند (user code) کد های خودمان را بنویسیم، پس از تولید مجدد فایل ها بوسیله STM32CubeMX کد هایی که قبلا خودمان زدیم و در این قسمت ها قرار دادیم، همچنان باقی می ماند و پاک نمی شوند.

me > STM32F401RETx > Timer_Hal.ioc - Project Manager > GENERATE CODE	
Pinout & Configuration	Clock Configuration
Project Manager	Tools
Project	
Code Generator	STM32Cube MCU packages and embedded software packs <input type="radio"/> Copy all used libraries into the project folder <input type="radio"/> Copy only the necessary library files <input checked="" type="radio"/> Add necessary library files as reference in the toolchain project configuration file
Advanced Settings	Generated files <input type="checkbox"/> Generate peripheral initialization as a pair of '.c/.h' files per peripheral <input type="checkbox"/> Backup previously generated files when re-generating <input checked="" type="checkbox"/> Keep User Code when re-generating <input checked="" type="checkbox"/> Delete previously generated files when not re-generated
	HAL Settings <input type="checkbox"/> Set all free pins as analog (to optimize the power consumption) <input type="checkbox"/> Enable Full Assert
	Template Settings Select a template to generate customized code Settings...

۲- واژه کلیدی weak __ در زبان c به چه معنی است و چرا برخی توابع CMSIS و HAL اینگونه تعریف شده اند؟

کلید واژه weak __ اشاره به یک موجودیت ضعیف را دارد. این به معنی آن است که فعلا این تعریف را برای این تابع داشته باشیم که اگر جایی آن را صدا و یا فراخوانی کردیم، بتوانیم به کار ادامه دهیم و به ارور بر نخوریم، از طرفی اگر نیاز بود جای دیگری آن را بدون کلید واژه weak __ تعریف می کنیم که این باعث می شود اولویت ما این تابع تعریف شده جدید قرار بگیرد.

یعنی اگر دو تابع با signature یکسان و اسم های مشابه یکی با کلید واژه weak __ و یکی بدون کلید واژه weak __ تعریف شوند، آن تابعی که کلید واژه weak __ را دارد، در نظر گرفته نمی شود و ایگنور می شود. یعنی با داشتن یک تابع با کلید واژه weak __ می گوییم فعلا این تعریف را داشته باش برای این تابع ولی اگر جای دیگری از پروژه این تابع بازنویسی شد، آن تابع بازنویسی شده را در نظر بگیر و این تابعی که کلید واژه weak __ را دارد بی اثر می شود.

مثلا در CMSIS و HAL برخی از توابع با این کلید واژه تعریف شده اند. این برای این منظور است که اگر یوزر آن را تعریف نکرد به مشکل بر نخوریم و ارور نگیریم و اگر هم یوزر آن ها را تعریف کرد، آن تابعی که یوزر تعریف کرده ملاک در نظر گرفته می شود و تابع با کلید واژه weak __ ایگنور می شود.

۳- جهت بهکارگیری وقفه ها در HAL چه سازوکاری تعبیه شده؟ چگونه باید از آن بهره برد؟ ارتباط آن با سازوکار تعبیه شده در CMSIS چیست ؟

برای بکارگیری وقفه ها در HAL می توان از توابع موجود در HAL استفاده کرد و این وقفه ها را فعال کرد. (مثلا با استفاده از توابعی مانند HAL_NVIC_EnableIRQ() و ...).

در فایل های تولید شده به وسیله نرم افزار stm32CubeMX می توان مشاهده کرد که وقفه هایی را که در نرم افزار stm32CubeMX فعال کرده بودیم، توابعشان (منظور از توابعشان برای مثال برای interrupt line عه یک، تابع void TIM1_IRQHandler(void) که مربوط به کتابخانه CMSIS می باشد و در فایل stm32f4xx_it.c نوشته شده است و در آن تابع void HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_1) فراخوانی شده است که خود این تابع در فایل stm32f4xx_hal_gpio.c تعریف شده است که باز هم ایت تابع نیز در خودش تابع HAL_GPIO_EXTI_Callback(GPIO_Pin) که خود این تابع نیز در همین فایل با کلید واژه weak __ تعریف شده است.)

حال همانطور که در قسمت قبل توضیح داده شد، ما اگر این تابع را بازنویسی نکنیم در فایل main (هر جا باشد ایرادی ندارد ولی ترجیح بر این است که جای مشخصی کنار سایر کد ها باشد، ولی در کل محدودیتی برای جای آن نداریم) از این تابع استفاده میب شود با همین تعریف پیش فرضی که دارد (بدنه این تابع صرفا اعلام می کند که این تابع استفاده نشده است)، جلو می رود. ولی اگر این تابع را خودمان در جایی تعریف کنیم، می توانیم به راحتی از آن استفاده کنیم.

نکته مهم این است که دیگر مثلا برای interrupt line های یک و دو و سه، سه تا تابع جدا نیازی نیست تعریف کنیم، صرفا با بازنویسی همین تابع HAL_GPIO_EXTI_Callback(GPIO_Pin) باید جلو برویم. البته در خود این تابع می توانیم با توجه به ورودی ای که داریم تشخیص دهیم که کدام یک از interrupt line هایمان باعث این وقفه شده است.

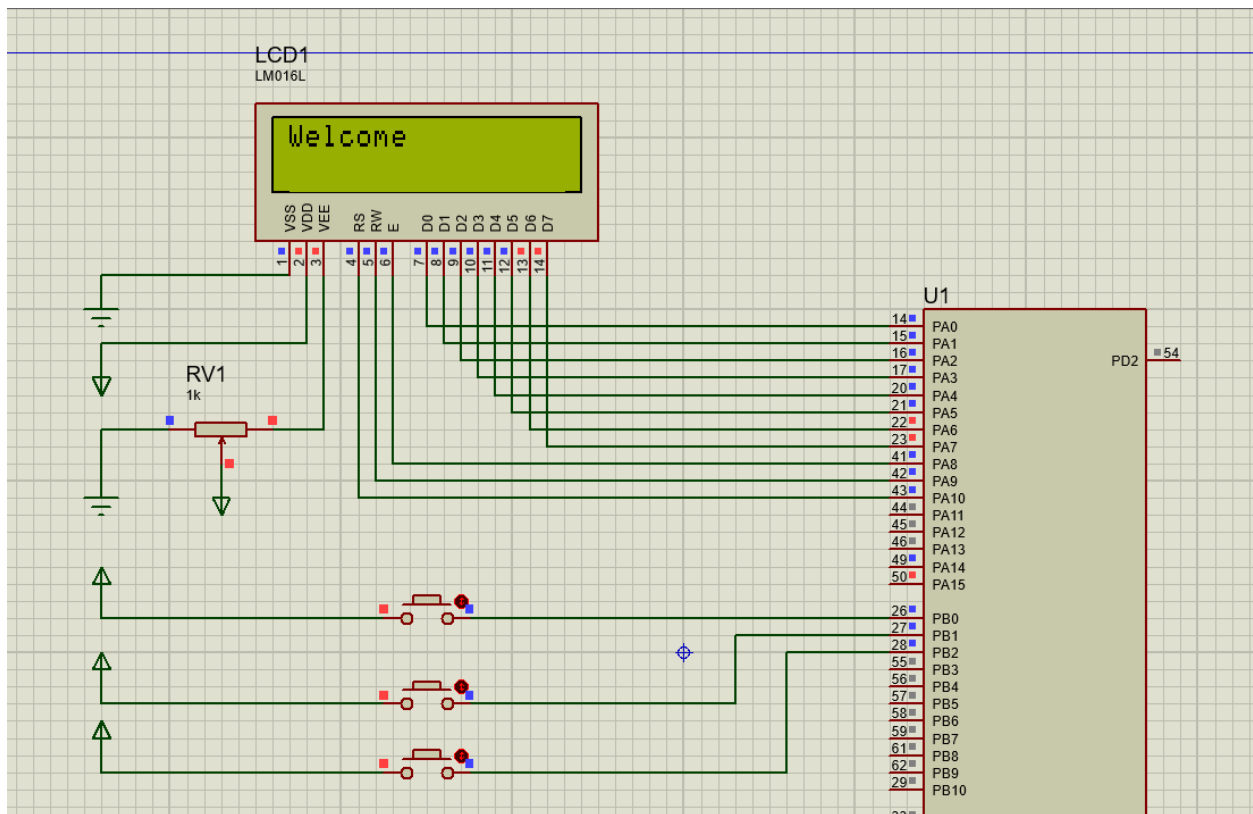
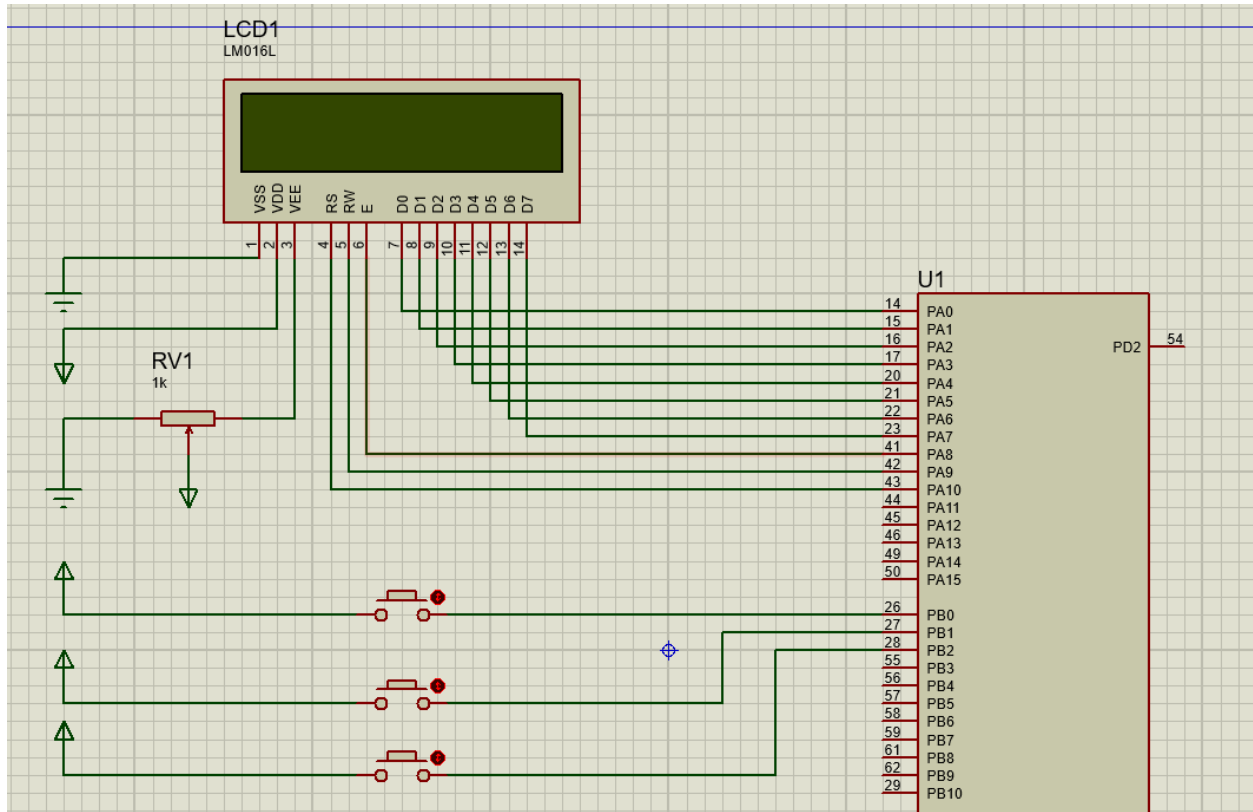
ارتباط بین این توابع از کتابخانه HAL و CMSIS نیز در ابتدای همین سوال (اسم توابع و به همراه فایل هایی که در آن ها هستند)، گفته شد.

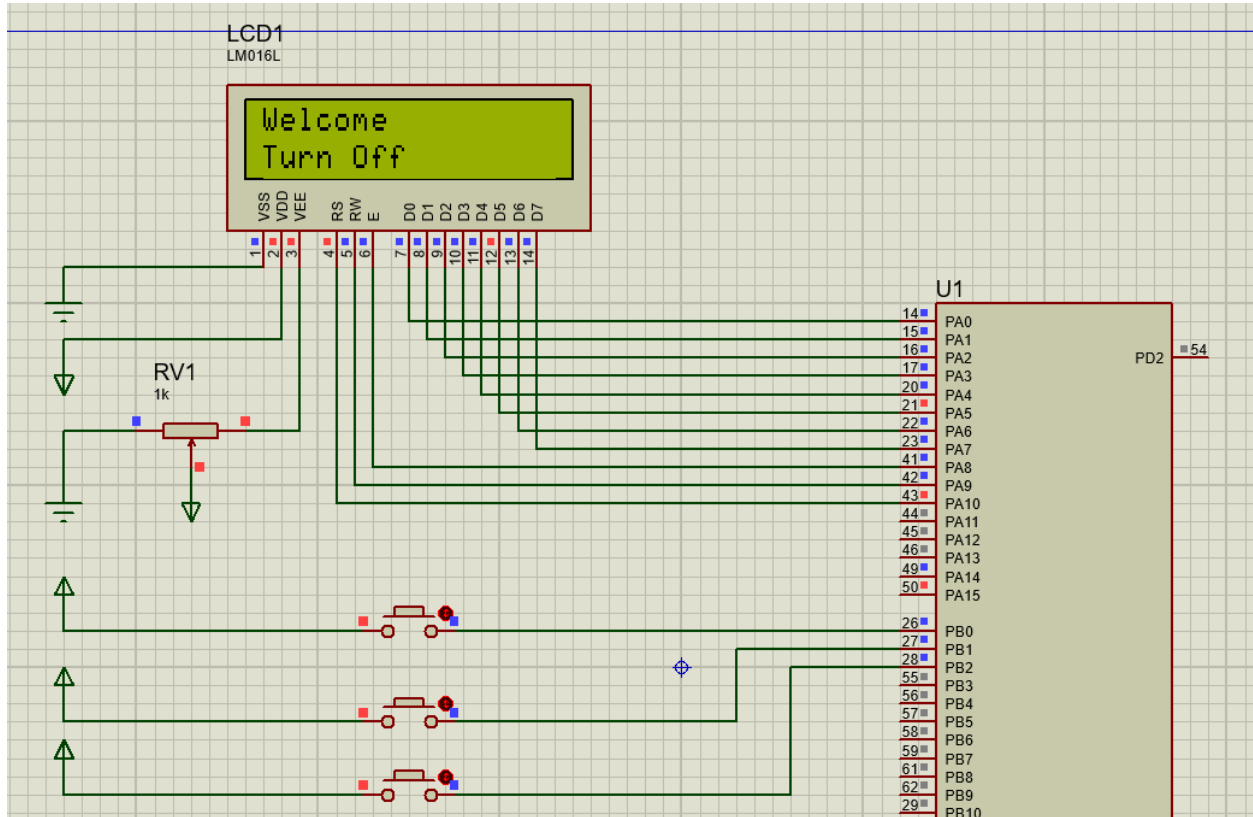
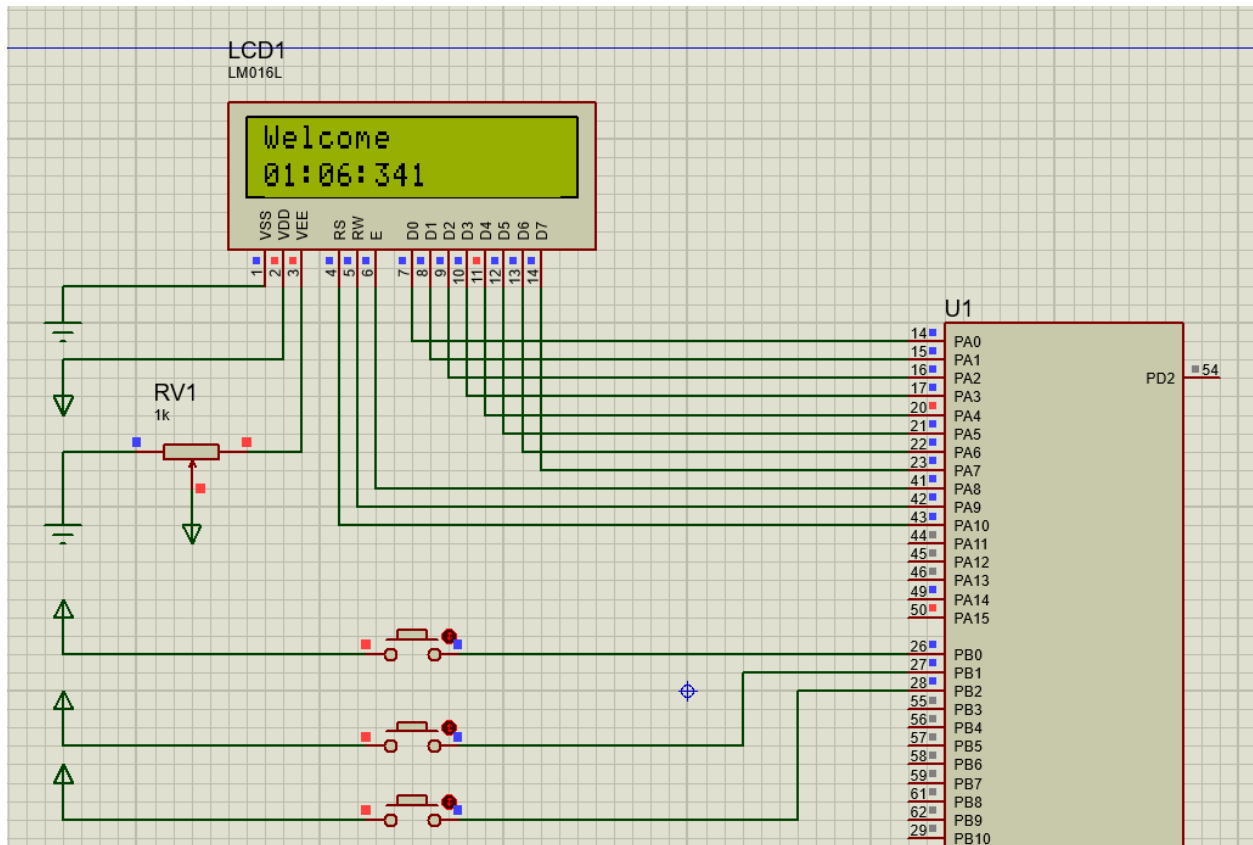
سوالات کدی:

جدول مربوط به پین های ورودی و خروجی های میکروکنترلر می باشد:

پین	ورودی / خروجی میکرو	کاربرد	مقدار MODER هر پین	مقدار PUPDR
PA0	خروجی	LCD در D0	01	-
PA1	خروجی	LCD در D1	01	-
PA2	خروجی	LCD در D2	01	-
PA3	خروجی	LCD در D3	01	-
PA4	خروجی	LCD در D4	01	-
PA5	خروجی	LCD در D5	01	-
PA6	خروجی	LCD در D6	01	-
PA7	خروجی	LCD در D7	01	-
PA8	خروجی	LCD در E	01	-
PA9	خروجی	LCD در RW	01	-
PA10	خروجی	LCD در RS	01	-
PB0	ورودی	Push button شماره یک	00	10
PB1	ورودی	Push button شماره سه	00	10
PB2	ورودی	Push button شماره دو	00	10

در ادامه عکس هایی را از شماتیک و خروجی آزمایش در محیط پروتئوس، قرار داده ایم:





حال به سراغ فیلدهایی که در محیط stm32CubeMX آن‌ها را ست کردیم می‌رویم و سپس کدهای زده شده در نرم افزار keil را توضیح می‌دهیم:

: STM32CubeMX

قسمت‌های مختلفی که آن‌ها را در این نرم افزار ست کردیم عبارتند از:

• Pinout & configuration

در اینجا در قسمت GPIO system core های مختلف را همانگونه که در جدول ابتدای گزارش نشان داده شده، ست کردیم. در ادامه عکس مربوط به آن‌ها را قرار می‌دهیم:

Pin Name	Signal on ...	GPIO outp...	GPIO mode	GPIO Pull...	Maximum...	User Label	Modified
PA0-WKUP	n/a	Low	Output Pu...	Pull-down	Low		✓
PA1	n/a	Low	Output Pu...	Pull-down	Low		✓
PA2	n/a	Low	Output Pu...	Pull-down	Low		✓
PA3	n/a	Low	Output Pu...	Pull-down	Low		✓
PA4	n/a	Low	Output Pu...	Pull-down	Low		✓
PA5	n/a	Low	Output Pu...	Pull-down	Low		✓
PA6	n/a	Low	Output Pu...	Pull-down	Low		✓
PA7	n/a	Low	Output Pu...	Pull-down	Low		✓
PA8	n/a	Low	Output Pu...	Pull-down	Low		✓
PA9	n/a	Low	Output Pu...	Pull-down	Low		✓
PA10	n/a	Low	Output Pu...	Pull-down	Low		✓

Pin Name	Signal on ...	GPIO outp...	GPIO mode	GPIO Pull...	Maximum...	User Label	Modified
PA3	n/a	Low	Output Pu...	Pull-down	Low		✓
PA4	n/a	Low	Output Pu...	Pull-down	Low		✓
PA5	n/a	Low	Output Pu...	Pull-down	Low		✓
PA6	n/a	Low	Output Pu...	Pull-down	Low		✓
PA7	n/a	Low	Output Pu...	Pull-down	Low		✓
PA8	n/a	Low	Output Pu...	Pull-down	Low		✓
PA9	n/a	Low	Output Pu...	Pull-down	Low		✓
PA10	n/a	Low	Output Pu...	Pull-down	Low		✓
PB0	n/a	n/a	External In...	Pull-down	n/a		✓
PB1	n/a	n/a	External In...	Pull-down	n/a		✓
PB2	n/a	n/a	External In...	Pull-down	n/a		✓

سپس در قسمت (همچنان در قسمت GPIO ها) NVIC، EXTI Line های صفر و یک و دو را فعال می‌کنیم و قسمت enable را فعال می‌کنیم.

سپس در قسمت EXTI Line NVIC های صفر و یک و دو و همچنین TIM های ۲ و ۳ را فعال می‌کنیم و priority آن‌ها را برابر با یک قرار می‌دهیم. (به این دلیل برابر با یک قرار دادیم که اگر priority آن‌ها را برابر با صفر

می گذاشتیم، در استفاده از تایمر سیستم (تایمر اصلی که آن را خود Sys Tick قرار داده ایم) ، به مشکل بر می خوردیم و نمی توانستیم از تایمر اصلی استفاده کنیم.) در ادامه عکس مربوط به این قسمت قرار داده شده است:

✓ NVIC
✓ Code generation

Priority Group ..
☐ Sort by Preemption Priority and Sub Priority
☐ Sort by interrupts names

Search 🔍
Show available interrupts
☒ Force DMA channels Interrupt

NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
Non maskable interrupt	<input checked="" type="checkbox"/>	0	0
Hard fault interrupt	<input checked="" type="checkbox"/>	0	0
Memory management fault	<input checked="" type="checkbox"/>	0	0
Pre-fetch fault, memory access fault	<input checked="" type="checkbox"/>	0	0
Undefined instruction or illegal state	<input checked="" type="checkbox"/>	0	0
System service call via SWI instruction	<input checked="" type="checkbox"/>	0	0
Debug monitor	<input checked="" type="checkbox"/>	0	0
Pendable request for system service	<input checked="" type="checkbox"/>	0	0
Time base: System tick timer	<input checked="" type="checkbox"/>	0	0
PVD interrupt through EXTI line 16	<input type="checkbox"/>	0	0
Flash global interrupt	<input type="checkbox"/>	0	0
RCC global interrupt	<input checked="" type="checkbox"/>	0	0
EXTI line0 interrupt	<input checked="" type="checkbox"/>	1	0
EXTI line1 interrupt	<input checked="" type="checkbox"/>	1	0
EXTI line2 interrupt	<input checked="" type="checkbox"/>	1	0
TIM2 global interrupt	<input checked="" type="checkbox"/>	1	0
TIM3 global interrupt	<input checked="" type="checkbox"/>	1	0
FPU global interrupt	<input type="checkbox"/>	0	0

Configuration

✓ NVIC
✓ Code generation

Enabled interrupt table	<input type="checkbox"/> Sel...	<input checked="" type="checkbox"/> Generate IRQ handler	Call HAL hand...
Non maskable interrupt	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Hard fault interrupt	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Memory management fault	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Pre-fetch fault, memory access fault	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Undefined instruction or illegal state	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
System service call via SWI instruction	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Debug monitor	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Pendable request for system service	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Time base: System tick timer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
RCC global interrupt	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
EXTI line0 interrupt	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
EXTI line1 interrupt	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
EXTI line2 interrupt	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
TIM2 global interrupt	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Interrupt unmasking ordering table (interrupt init code is moved after all the peripheral init code)

Rank	Interrupt name
------	----------------

سپس در قسمت RCC ، در قسمت NVIC Setting ، گزینه RCC Global Interrupt را فعال می کنیم. عکس از اسن قسمت هم در ادامه قرار داده شده است:

Configuration

Reset Configuration

Parameter Settings User Constants **NVIC Settings**

NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
RCC global interrupt	<input checked="" type="checkbox"/>	0	0

سپس در قسمت SYS ، Timebase Source را برابر با Sys Tick قرار می دهیم. عکس مربوط به این قسمت نیز در ادامه آورده شده است:

SYS Mode and Configuration

Mode

Debug

☐ System Wake-Up

Timebase Source

سپس به سراغ قسمت Timer می رویم. در اینجا TIM2 و TIM3 را فعال می کنیم و تنظیماتشان را انجام می دهیم. عکس از تنظیمات این دو تایمر در ادامه آمده است:

۱- TIM2 :

Counter Settings

Prescaler (PSC - 16 bits value) 15999

Counter Mode Up

Counter Period (AutoReload Register -... 1

Internal Clock Division (CKD) No Division

auto-reload preload Disable

Trigger Output (TRGO) Parameters

Master/Slave Mode (MSM bit) Disable (Trigger input effect not delayed)

Trigger Event Selection Reset (UG bit from TIMx EGR)

Reset Configuration

Parameter Settings User Constants **NVIC Settings** **DMA Settings**

NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
TIM2 global interrupt	<input checked="" type="checkbox"/>	1	0

Reset Configuration

✓ Parameter Settings

✓ User Constants

✓ NVIC Settings

✓ DMA Settings

Configure the below parameters :

Search (Ctrl+F)

⏪

⏩

i

▼ Counter Settings

Prescaler (PSC - 16 bits value)

1599

Counter Mode

Up

Counter Period (AutoReload Register - ...)

500

Internal Clock Division (CKD)

No Division

auto-reload preload

Disable

▼ Trigger Output (TRGO) Parameters

Master/Slave Mode (MSM bit)

Disable (Trigger input effect not delayed)

Triquer Event Selection

Reset (UG bit from TIMx EGR)

Reset Configuration

✓ Parameter Settings

✓ User Constants

✓ NVIC Settings

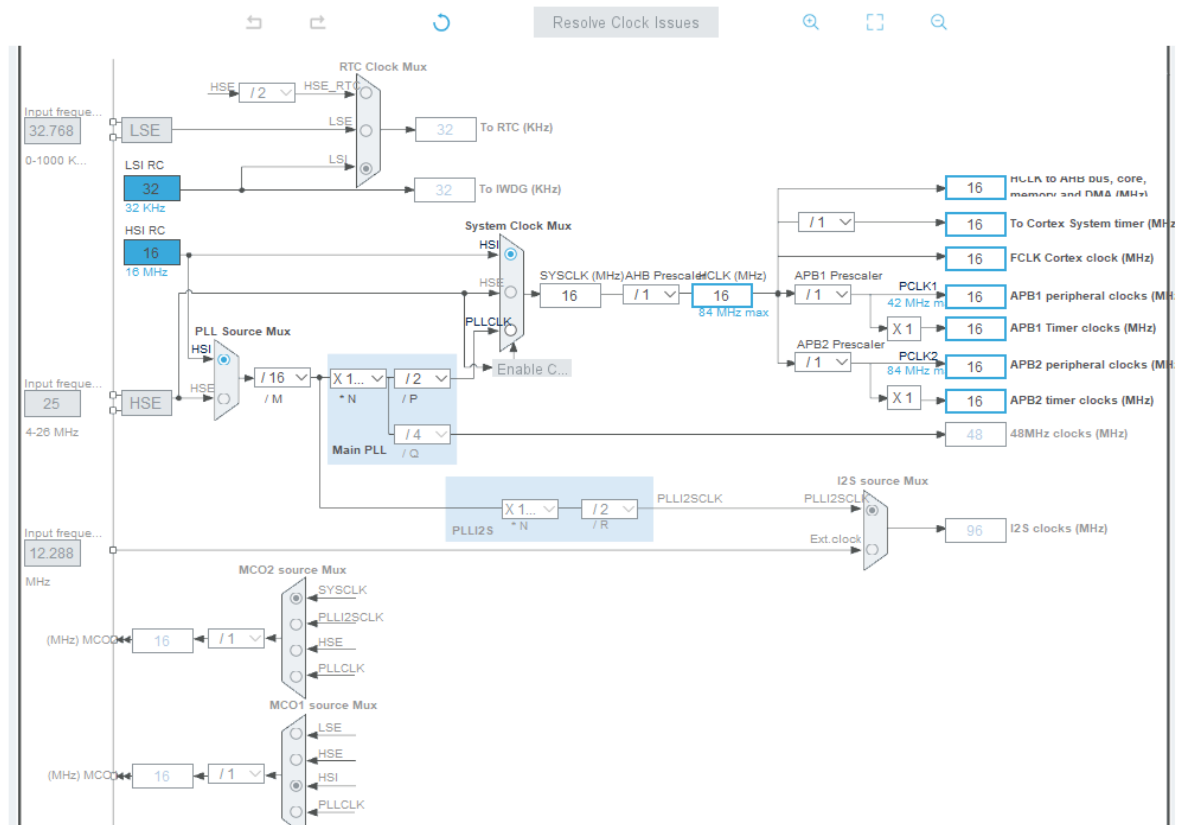
✓ DMA Settings

NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
TIM3 global interrupt	<input checked="" type="checkbox"/>	1	0

** اعداد قرار داده شده برای قسمت های Counter Setting، با آزمون و خطاهای بسیار زیاد بدست آمده اند و با این اعداد، خروجی کار به حالت واقعی نزدیک تر می شد.

• : Clock Configuration

در این قسمت تغییرات آنچنانی ای اعمال نکردیم. شکل مربوط به تنظیمات این قسمت در ادامه آمده است:



• : Project Manager

در این قسمت، در بخش Project قسمت Toolchain / IDE گزینه MDK-ARM را انتخاب می کنیم تا کدهای مورد نیاز و مورد استفاده در نرم افزار Keil برایمان تولید شود. عکس تنظیمات این بخش در ادامه آمده است:

Project Settings

Project Name

Timer_Hal

Project Location

C:\Users\parsa\Desktop\Riz\6th-Assignment\STM32Cube-MX

Browse

Application Structure

Advanced

☐ Do not generate the main()

Toolchain Folder Location

C:\Users\parsa\Desktop\Riz\6th-Assignment\STM32Cube-MX\Timer_Hal\

Toolchain / IDE

MDK-ARM

Min Version

V5.32

☐ Generate Under Root

Linker Settings

Minimum Heap Size

0x200

Minimum Stack Size

0x400

Thread-safe Settings

Cortex-M4NS

☐ Enable multi-threaded support

Thread-safe Locking Strategy

Default – Mapping suitable strategy depending on RTOS selection.

Mcu and Firmware Package

Mcu Reference

STM32F401RETx

Firmware Package Name and Version

STM32Cube FW_F4 V1.26.2

☒ Use Default Firmware Location

Firmware Relative Path

C:\Users\parsa\STM32Cube\Repository\STM32Cube_FW_F4_V1.26.2

Browse

سپس در بخش Code Generator گزینه های Keep user code when re-generating را فعال می کنیم. عکس از تنظیمات این بخش نیز در ادامه آمده است:

STM32Cube MCU packages and embedded software packs

☐ Copy all used libraries into the project folder
☐ Copy only the necessary library files
☒ Add necessary library files as reference in the toolchain project configuration file

Generated files

☐ Generate peripheral initialization as a pair of '.c/.h' files per peripheral
☐ Backup previously generated files when re-generating
☒ Keep User Code when re-generating
☒ Delete previously generated files when not re-generated

HAL Settings

☐ Set all free pins as analog (to optimize the power consumption)
☐ Enable Full Assert

Template Settings

Select a template to generate customized code

Settings...

سپس در انتها نیز تنظیمات advanced setting را به صورت زیر ست می کنیم:

Driver Selector

Search (Ctrl+F)

⌕

⌕

RCC

GPIO

▼ TIM

TIM2

TIM3

HAL

HAL

HAL

HAL

HAL

Generated Function Calls

⌕

⌕

⌕

Generate Code	Rank	Function Name	Peripheral Instance Name	<input type="checkbox"/> Do Not Generate Function Call	<input type="checkbox"/> Visibility (Static)
<input checked="" type="checkbox"/>	1	SystemClock_Config	RCC	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	2	MX_GPIO_Init	GPIO	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	3	MX_TIM2_Init	TIM2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	4	MX_TIM3_Init	TIM3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

• Tools :

در این بخش هیچ گونه تغییری ایجاد نکردیم.

حال به سراغ کدهای زده شده در نرم افزار Keil می رویم.

: Keil

در اینجا روال کاری که انجام می دهیم یعنی همان روش کارمان، کاملاً مشابه با آزمایش سری قبل می باشد. بنابر این برای توضیحات تکمیلی به آزمایش قبل رجوع می کنیم و خلاصه ای از آن را در اینجا ذکر می کنیم:

- یک آرایه در نظر می گیریم که اعداد در هر لحظه در آن نمایش داده می شوند.
- یک متغیر با نام status در نظر می گیریم که مقدار آن نشان دهنده حالتی است که در آن می باشیم که این حالات در ادامه کد به صورت کامنت توضیح داده شده اند.
- در ادامه یک سری تعریف ها برای مشخص کردن نوع ورودی و خروجی تابع ها آماده است.

در ابتدا تعریف تابعه HAL_Delay را داریم. چون به صورت weak __ تعریف شده بود و به دلیل وجود یک بلاک if در آن، این تایمر در هنگام استفاده کند تر از حد انتظار عمل می کرد، بنابر این آن را دوباره تعریف کردیم و بدنه آن را دقیقاً کپی کردیم و فقط آن بلاک عه if را کامنت کردیم.

سپس تابع MyInit را داریم که در این تابع خیلی از کد ها را حذف کردیم چون این کد ها بوسیله نرم افزار STM32CubeMX ساخته شدند.

** در این آزمایش، دلیلی ها را به جای لحاظ کردن بعد از هر سری فراخوانی تابع LCD_put_char ، آن دلیلی را در خود این تابع LCD_put_char قرار دادیم. (برخلاف آزمایش قبل که آن ها را پس از هر سری فراخوانی این تابع قرار داده بودیم.)

سپس تابع incMiliSecond را داریم که مشابه با همان تابع آزمایش قبلی است.

** این تابع به عنوان اولین ورودی، ایندکس آن عددی را که باید عدد موجود در ورودی دو را به آن اضافه کند را می گیرد و سپس به عنوان ورودی دوم، عددی که باید به عدد موجود در ایندکس آرایه اضافه کند را میگیرد. یعنی برای مثال اگر فراخوانی این تابع به صورت زیر باشد، incMiliSecond(8, 23)، یعنی به از ایندکس ۸ عه آرایه شروع کن و به عدد موجود در این ایندکس، ۲۳ واحد اضافه کن. (پایاده سازی این تابع بصورت بازگشتی است.)

به صورت کلی این تابع عدد ورودی دو را با ایندکس ۸ صدا می کنیم (ایندکس ۸ ایندکس آخرین عضو آرایه می باشد) و سپس به صورت بازگشتی هربار این مقدار به آن ایندکس آرایه اضافه می شود و اگر تمام شد و دیگر نیازی به اضافه کردن نبود، کار تمام می شود در غیر این صورت این تابع را با ورودی منهای عددی که می توانست اضافه کند، با یک ایندکس کمتر فراخوانی می کند.

سپس در ادامه تابع های هندلر برای دکمه ها و تایمر ها تعریف کردیم تا بتوانیم از آنها در تابع های مربوط به قسمتی که در آن اینتراپت ها هندل می شوند، استفاده کنیم.

با توجه به اینکه تابع HAL_GPIO_EXTI_Callback به صورت weak __ تعریف شده است (در قسمت سوالات تحلیلی توضیح دادیم) آنرا تعریف و پیاده سازی می کنیم.

در سازوکار در نظر گرفته شده برای هندل کردن اینترپت ها در کتابخانه HAL به ازای هر اینترپت لاین، ما یک تابع را به صورت جدا فراخوانی نمیکنیم، بلکه کلا یک تابع را فراخوانی می کنیم و از ورودی آن متوجه می شویم که این تابع به خاطر بوجود آمدن اینترپت از کدام خط صدا شده است، و در یک سوئیچ کیس و یا یک ایف و الس مناسب، عملیات های مرتبط با هر کدام را به طور جداگانه انجام می دهیم.

دقیقا توضیحات بالا برای اینترپت های ناشی از تایمر ها هم صدق می کند فقط نام تابعی که باید آن را پیاده سازی کنیم، برای تایمر ها HAL_TIM_PeriodElapsedCallback می باشد.

در تابع main تولید شده بوسیله نرم افزار STM32CubeMX ما دقیقا قبل از ورود به حلقه تابع MyInitialize را فراخوانی میکنیم تا کار های مورد نیاز (مانند تنظیمات اولیه LCD و آرایه) را انجام بدهیم.

****یک مشکلی وجود داشت!** با ست کردن تمام فیلد های گفته شده در نرم افزار STM32CubeMX و آزمون و خطا های بسیار، متوجه شدیم که برای فعال کردن وقفه های تایمری در تابع های تولید شده برایشان توسط نرم افزار HAL ، باید دو خط را اضافه کنیم، وگرنه در غیر این صورت این وقفه های تایمری فعال نمی شوند!

برای فعال کردن وقفه های تایمری خط های زیر را برای هر کدام از آنها اضافه کردیم:

۱- برای TIM2 در تابع پیاده سازی شده توسط نرم افزار (MX_TIM2_Init)، در انتهای تابع در قسمت User Code عه آخر، این دو خط را اضافه کردیم:

```
htim2.Instance->CR1 = 1; // active  
htim2.Instance->DIER |= 1;
```

۲- برای TIM3 در تابع پیاده سازی شده توسط نرم افزار (MX_TIM3_Init)، در انتهای تابع در قسمت User Code عه آخر، این دو خط را اضافه کردیم:

```
htim3.Instance->CR1 = 1; // active  
htim3.Instance->DIER |= 1;
```

با اضافه کردن این دو خط در انتهای هر کدام از این توابع (با توجه به ویژگی های کتابخانه HAL) وقفه های تایمریمان هم فعال می شدند و مشکلمان حل می شد.

- کلاس درس
- اسلاید ها
- Reference manual
- Datasheet
- STM32Cube-MX-HAL-MOOC
- https://www.st.com/resource/en/application_note/dm00236305-generalpurpose-timer-cookbook-for-stm32-microcontrollers-stmicroelectronics.pdf
- <https://stm32f4-discovery.net/2015/07/hal-library-07-usart-for-stm32fxxx/>
- https://www.st.com/resource/en/user_manual/um1940-description-of-stm32f2-hal-and-lowlayer-drivers-stmicroelectronics.pdf
- <https://hal.archives-ouvertes.fr/hal-01318566>
- https://www.disca.upv.es/aperles/arm_cortex_m3/livre/st/STM32F439xx_User_Manual/stm32f4xx_hal_dma_8c_source.html
- https://www.st.com/resource/en/user_manual/dm00154093-description-of-stm32f1-hal-and-lowlayer-drivers-stmicroelectronics.pdf
- <https://letanphuc.net/2015/06/stm32f0-timer-tutorial-and-counter-tutorial/>
- <https://hasanyavuz.ozderya.net/?p=437>
- <https://riptutorial.com/stm32/example/25059/first-time-setup-with-blink-led-example-using-sw4stm32-and-hal-library>
- <https://microcontrollerslab.com/stm32-blue-pill-timer-interrupt-stm32cube-ide-hal-libraries/>
- https://www.st.com/resource/en/user_manual/dm00105879-description-of-stm32f4-hal-and-ll-drivers-stmicroelectronics.pdf
- <https://visualgdb.com/tutorials/arm/stm32/timers/hal/>
- <https://deepbluembedded.com/stm32-timer-interrupt-hal-example-timer-mode-lab/>
- <https://stackoverflow.com/questions/40323461/stm32-hal-timer-interrupt-isnt-triggered>
- <https://community.st.com/s/question/0D53W00000wV41eSAC/error-unknown-type-name-timhandletypedef>

- <https://community.st.com/s/question/0D50X00009XkfsRSAR/hal-timer-interrupt>
- https://www.st.com/resource/en/user_manual/dm00104712-stm32cubemx-for-stm32-configuration-and-initialization-c-code-generation-stmicroelectronics.pdf
- <https://www.openstm32.org/forumthread2145#threadId2146>
- <https://stackoverflow.com/questions/53899882/hal-delay-stuck-in-a-infinite-loop>
- <https://community.st.com/s/question/0D50X00009XkhFxSAJ/cubemx-cannot-disable-systick-int>
- <https://community.st.com/s/question/0D50X00009XkibgSAB/cubemx-and-systick>
- <https://stm32f4-discovery.net/2015/07/hal-library-3-delay-for-stm32fxxx/>
- <https://stackoverflow.com/questions/46062122/delay-in-hal-library-hal-delay>
- <https://community.st.com/s/question/0D50X00009nKalVSAS/stm32cubemx-cortex-m3-generated-code-hangs-in-hal-delay>
- https://cpp.hotexamples.com/examples/-/-/HAL_GPIO_WritePin/cpp-hal_gpio_writepin-function-examples.html
- https://www.disca.upv.es/aperles/arm_cortex_m3/livre/st/STM32F439xx_User_Manual/group_gpio_exported_functions_group2.html
- https://www.st.com/content/st_com/en/products/development-tools/software-development-tools/stm32-software-development-tools/stm32-utilities/stsw-link009.html?dl=AfD5UwXzmWltbAbLE76mVA%3D%3D%2CnGLuNW7qysUz pav1m%2BPwCWIXh%2F14uvqH9rpYREFTmH42UrCJ%2Fg%2FyoiLlommtpia AcRkmpDKudHj3VWAORFYmXCysoj1WM9oo%2BQ65jX7p3TyIRlc5ERanhK7 MJclN%2Bw1c8eiQ3kuzxj74q3xQJFD8hwLL8Pi3vRMY6ZnuGuB0OptWpyRfo fmiJslYdvDc7Fn8poD5JPh6DIBlqwqdJamt0ugBwsjmplUyHcL0MTTVVLeN09c euc14IX8JHmp55yxjm2ykp8C3MBFO1rxmcgn5ZIMGMK1XFLoSjWnFCQJAm 3hgTXD29CBs9W2k3K0BeM1H
- <https://community.st.com/s/question/0D50X0000AlcoWmSQJ/change-default-value-of-pins-gpio-in-cubemx>