

YILDIZ TECHNICAL UNIVERSITY

DEPARTMENT OF BIOMEDICAL ENGINEERING

BME3321 INTRODUCTION TO MICROCONTROLLER PROGRAMMING LABORATORY

EXPERIMENT SHEETS

Experiment 1: C Programming Language Basics: Data Types, Variables, Arrays, Loops, Conditionals, Functions, Pointers, Structures

Objectives

The objectives of Experiment 1 are

• to learn C programming language basics Data types, Variables, Arrays, Loops, Conditionals, Functions, Pointers, Structures

Apparatus Required:

• Dev C++

Preliminary Work:

• Install required Dev C++ programme (<u>https://sourceforge.net/projects/orwelldevcpp/</u>)



Figure 1

- Study the L03 notes.
- Write the following codes in the experimental work in Dev C++.

Experimental Work:

1. Logical Operator

Write the code that will calculate the truth table of the p'Vq proposition using OR operator and display it on the screen. Don't forget to adjust the spaces to make the output look neat.



Figure 2: Program Output Window

Answer:

```
#include <stdio.h>
int main (void)
{
    printf(" p q ~pVq\n");
    printf("------\n");
    printf("%3d%5d%7d \n", false, false, !false||false);
    printf("%3d%5d%7d \n", false, true, !false||true);
    printf("%3d%5d%7d \n", true, false, !true||false);
    printf("%3d%5d%7d \n", true, true, !true||true);
    return (0);
```

}

2. Loops (for-while) & Conditionals

a. Write the code that produces the following output by using nested loops.



Figure 4: Program Output Window

Answer:

{

}

include <stdio.h>
int main (void)

b. Write the C program that receives the n value from the user, which is a positive integer value, as input and find all prime numbers up to n and display them on the screen.



Figure 3: Program Output Window

Answer:

#include <stdio.h>

int main (void)

{

int n;

int num, p,i;

printf("Enter positive integer: ");

scanf("%d", &n); /*Taking n value*/

printf("\n All prime number between 1-%d: ",n);

printf("\n-----\n");

for(num = 2; num <= n; num = num+1)</pre>

{/*Let's assume that the number is prime: p=1 means the number is prime, p=0 means it is not prime*/

```
p=1; /*Assume the number is prime*/
```

i=2; /*The variable i is used to control which numbers the number entered by the user can be divided by i*/

```
while ((i < num) && p == 1)
```

```
{
```

}

```
if (num%i == 0)
        p=0; /*The number is not prime because it is divisible.*/
        i=i+1;
if (p==1)
printf("%4d" , num);
                      }
return (0);
```

}

3. Functions

Write a code that takes a positive integer value from the user and shows the number of digits of the integer on the screen. While writing the code, use the function named **'basamak_bul ()'**. *basamak_bul ()* function must receive an integer value from where it was called, find the number of digits of the integer and return it to where it was called.





Answer:

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

```
int basamak_bul(int x);
```

```
int main (void)
```

```
{
```

```
int a,t;
```

```
printf("Enter an integer number: ");
```

```
scanf("%d",&a);
```

```
t=basamak_bul(a);
```

```
printf("\nNumber of digits: %d",t);
```

```
return(0);
```

```
}
```

```
int basamak_bul(int x)
```

{

```
int digit = 0;
while (x)
{
  digit = digit+1;
  x=x/10;
}
return(digit); }
```

4. Pointers

a. Write the C program that orders 3 integers received from the user and displays the first input values and sequential values on the screen. Return the sequential numbers as reference parameters. (Use the **pointers** for this.) Use the **'replace** ()' function to sort the 3 numbers.



Figure 6: Program Output Window

Answer:

```
#include <stdio.h>
/* Program that sorts the 3 entered numbers*/
```

```
void replace(int*,int*);
```

```
int main (void)
```

{

```
int x,y,z;
printf("Enter three numbers: ");
scanf("%d%d%d", &x,&y,&z);
printf("First values: %d %d %d\n", x,y,z);
if (x>y) replace(&x,&y);
if (x>z) replace(&x,&z);
if (y>z) replace(&y,&z);
printf("Ordered values: %d %d %d\n", x,y,z);
```

```
}
```

```
/*replace() function that changes the values of two parameters*/
```

```
void replace(int *a, int *b)
```

```
/*a and b are taken as reference parameters. Therefore, changes in this function will be reflected in the sent parameter.*/
```

```
{
```

```
int temporary;
temporary = *a;
*a = *b;
*b = temporary;
```

}

b. Write a function that finds the perimeter and area of a rectangle. In the function, receive the width and length of the rectangle as the value parameter and return the perimeter and area as the reference parameter.

```
Enter the length and width of the rectangle 6
9
perimeter of the rectangle: 30
Area of the rectangle : 54
------
Process exited after 9.533 seconds with return value 0
Press any key to continue . . .
```

Figure 7: Program Output Window

Answer:

/* Program to find the perimeter and area of a rectangle. While the width and height of the rectangle are taken as value parameters, the perimeter and area are returned as reference parameters */

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

```
void perimeter_area(int, int , int*, int*);
```

int main (void)

{

```
int width, length, perimeter, area;
```

printf("Enter the length and width of the rectangle ");

scanf("%d%d",&length,&width);

```
if (length<0 || width<0)/*input control*/</pre>
```

printf("\nYou entered the wrong value");

else {

```
perimeter_area(width, length , &perimeter, &area);
printf("perimeter of the rectangle: %d\n", perimeter);
printf("Area of the rectangle : %d\n", area);
```

```
}
```

}

void perimeter_area(int w, int l, int *e, int *a)

{

```
*e = 2* (w+l); /*Calculate perimeter*/
```

```
*a = w*l; /*Calculate area*/ }
```

5. Arrays

Write a C program in which the user enters integers in a 5-element array and after each integer value entered in the array, it shows whether the entered number is odd or even. This program should consist of two functions. The **'bul ()'** function must receive an integer value from where it is called and show it is odd or even. The **'main ()'** function should receive 5 integer values from the user, store them in an array and show that the array elements are odd or even using the **bul()** function.

Enter an integer values: 5 5 is an odd number
Enter an integer values: 9 9 is an odd number
Enter an integer values: 7 7 is an odd number
Enter an integer values: 8 8 is an even number
Enter an integer values: 12 12 is an even number
Process exited after 13.91 seconds with return value 0 Press any key to continue

Figure 8: Program Output Window

Answer:

```
#include<stdio.h>
void bul (int);
int main (void)
{
    int k[5],i;
    for (i = 0; i<=4; i++)
    {
        printf("Enter an integer values: ");
        scanf("%d",&k[i]);
        bul(k[i]);
    }
    return(0);
}</pre>
```

```
void bul (int a)
{
    if (a%2 == 0)
        printf("%d is an even number\n ", a);
        else
        printf("%d is an odd number\n ", a);
}
```

6. Structures

Write a program that receives the coordinates of two points (x1,y1) and (x2,y2) as input from the user and calculates the distance between them. The x and y coordinates of each point in your program should be kept in a struct. The distance formula is as follows:

distance=
$$\sqrt{(x1 - x2)^2 + (y1 - y2)^2}$$

Seç C:\Users\Supervisor\Desktop\Untitled2.exe
Enter x and y coordinates of point 1 :2 5
Enter x and y coordinates of point 2 :4 8
Distance of the two points : 3.61
Process exited after 8.601 seconds with return value 0
Press any key to continue . . . _



Answer:

```
#include <stdio.h>
#include <math.h>
int main (void)
{
struct point
{
int x,y;
};
struct point p1, p2;
float distance;
/*Read the coordinates of two points*/
printf(" Enter x and y coordinates of point 1 :");
scanf ("%d %d", &p1.x, &p1.y);
printf(" Enter x and y coordinates of point 2 :");
scanf ("%d %d", &p2.x, &p2.y);
/* Calculate the distance between two points*/
distance = sqrt (((p1.x-p2.x)*(p1.x-p2.x))+((p1.y-p2.y)*(p1.y-p2.y)));
printf(" Distance of the two points : %5.2f", distance);
return (0); }
```

EXPERIMENT 2: GENERAL-PURPOSE INPUT/OUTPUT (GPIO)

Objectives

The objectives of Experiment 2 are

- to learn tools/environment for STM32F4 microcontroller programme and architecture
- to use Driving GPIO functions (HAL_GPIO_WritePin, HAL_GPIO_ReadPin, HAL_GPIO_TogglePin) and GPIO Output Data Register (ODR), GPIO Input Data Register (IDR)

Apparatus Required:

- STM32CubeMx
- Keil µVision (MDK ARM)
- STM32 ST-Link Utility
- STM32F4 Microcontroller
- STM32F4 Reference Manual
- STM32F4 User Manual

Preliminary Work:

 Install required programmes (STM32CubeMx, Keil µVision (MDK ARM), STM32 ST-Link Utility) STM32CubeMx----> <u>https://www.st.com/en/development-tools/stm32cubemx.html</u>

ST-Link Utility----> <u>https://www.st.com/en/development-tools/stsw-link004.html</u> Keil µVision----><u>https://www.keil.com/download/product/</u>



Figure 1

2. Study the GPIO (Lecture 4) notes. Write the codes of the experimental work in Keil μ Vision at home.

Experimental Work:

1. First, open the CubeMx program. Select "Manage embedded software packages" from the 'Help' menu (Figure 2).



Figure 2

2. According to the microcontroller you have, the relevant package (STM32F4) is selected and installed (Figure 3).

xisting Projects	🔁 Ersbestiket Software Packages Marager	
Recent Opened Pr	51832Cube MCU Packages and entitledded software packs relisases Ratasien information was last referited 79 days ago	+ - embedded softwar
lab1trialLipc Last modified date: 1	► STAIS2F4	metullise Version Available Version
timer_pwm.ioc Last modified date _1	STM1287? STM12878	hvore packages Mit
USarter, IOC Last modified date : 1	 STM264 STM204 	
exti loc Last modified dale: 2	• STMIZLO	
timer_inputcaptur Last modified date 1	STM32L1 Details	
Other Projects		7 E
	Phier Local	(New System

Figure 3

3. To prevent it from updating in ordinary, settings are made as seen in Figure 4 and Figure 5.



Figure 4



Figure 5

4. Now you can create a new project file to start programming. Select "New Project" from the File menu (Figure 6). The relevant microcontroller is found and selected from the Part Number as in Figure 7. After choosing our STM32F4 discovery card, choose "Start Project" from the top menu (Figure 7).



Figure 6



Figure 7

5. You will see a screen like in Figure 8. You can see the schematic of the microcontroller you will use here. You can use it this way. However, in terms of power consumption and more stable operation, it will be better if you turn off the pins that you will not use in the lab. The green ones show the open pins and the gray ones show the reset state ones. In this lab, you will basically use push pull buttons and leds. So you can turn off the others. Which of these pins are can be understood by looking at the user manual. You should not close the pins that provide the connection between the microcontroller and the programmer. You can check which of these pins are in the user manual (Figure 9). You can close and define tasks of a pin by left-clicking on the pin. You completed the Pinout Configuration.



Figure 8



Figure 9

6. There is nothing we can change in the Clock Configuration section for this experiment. Come to the Project Manager menu and set the fields as in Figure 10 (Give the project name using English characters without spaces. You can create the project in a file on the desktop. Select the MDK-ARM option for Toolchain/IDE. Uncheck 'Use latest available version' to prevent it from updating). Then select the Generate Code and so the template file is out. Select "Open Project" from the opened screen (Figure 11). Keil μVision will be opened.

Du Color	File .	Wadow Hog		
Hame > STMR	F407-0T- STM22F4070-0	DISCI Vintitled - Project Manage	n >	DENERATE CODE
Pinout &	Configuration	Clock Configuration	Project Manager	Tools
free .	Project Settings Project Harms MCU_alst Project Lacation C-subsensprojectal Application Structure		Rouse	
Code Generator	Advanced Technis Finder Location Cruckensproperty/MOU_jal Technian / IDE	Do not generate the ent. Min Variane V5.27 □ □	inematin linder Nadi	
Advanced Setting	Linius Settings Minimum Heap Size Minimum Stack Size	Buggo		
	No. and Fernane Package No. Reference EDMS2F402VGTe			
	Fermine Package Name at STMXXCube PVr_F4 V7.21	nd Verson 1 Use tablet available		
	El Use Default Formane La	nden Philippinese Philippine, Philippine	0.77.3 8000	

Figure 10

THE CARE SOL				10100
STM COMMAN	F 84	Window He	6	🎯 🛐 🖬 🎔 🤆 🖅
Home STM00	F407VGTx - STM32F407G-Di	SC1 MCU_lab1.loc - Project	Manager	GENERATE CODE
Pinout &	Configuration	Clock Configuration	Project Manager	Tools
(Press	Project Settings Project Name MCU set1 Project Location C Icuberrypojects			
Code Generator	Application Structure (Advancent Laplotan Folder Location C. Scolamspagaets/MCU, July Testichan / IDE MOK-ARM	Mie Varoon V VS27 ~	Generaria Under Root	
Advanced Setting	Unker Settings Minimum Heap Size 2 Minimum Stack Size 2	200 The Code is successful (100 The Code is successful Repair Pathon	y generated under C/cubernsprojecte/MCU_Jab1	
	Mcu and Pirmaare Pachage Mcu Rolennce BTM/2F407V0Tx Pirmore Pachage Turns and BTM/2Cube PVP F4 V1 25 0	Version 	is section	

Figure 11

 First, you need to come to the Pack Installer (Figure 12) and install the package of the relevant microcontroller, as you did in CubeMx. The relevant microcontroller is selected from the 'Search' menu and 'Install' is selected for the required package(.....) (Figure 13).



Figure 12

File Packs Window Help				
2 Device: STMicroelectronics	- STMB2F4 Series			
Devices Boards		b 4 Packs Examples		
Search: stm32f4	• × 🖸	Pack	Action	Description
Device /	Summary	B Device Specific	3 Packs	STM32F4 Series selected
All Devices	205 Devices	Clarinox::Wireless	🕸 Install	Clarinox Bluetooth Classic, Bluetooth Low Energ
STMicroelectronics	205 Devices	Keil:STM32F4xx_DFP	😵 Update	STMicroelectronics STM32F4 Series Device Supp
IN THE ETIMONY STATE	205 Devices	ReiESTM32QUCLEO_B	🐵 Install	STMicroelectronics Nucleo Boards Support and
Bullinsand-hannander		😑 Generic	48 Packs	
		 Alibaba:AliOSThings 	🔅 Install	AliOS Things software pack
		Arm-Packs::PKCS11	🔅 Install	OASIS PKCS #11 Cryptographic Token Interface
		+ Arm-Packs::Unity	🗇 Install	Unit Testing for C (especially Embedded Softwa
		-ARM::AMP	🗇 Install	Software components for inter processor comm
		- ARM::CMSIS	🔶 Up to date	CMSIS (Cortex Microcontroller Software Interfa
		ARM::CMSIS-Driver	🤣 Update	CMSIS Drivers for external devices
		ARM::CMSIS-Driver_Va	😔 Install	CMSIS-Driver Validation
		ARM:CMSIS-FreeRTOS	🗇 Install	Bundle of FreeRTOS for Cortex-M and Cortex-A
		ARM::CMSIS-RTOS_Va	🔅 Install	CMSIS-RTOS Validation
		ARM:mbedClient	😔 İnstali	ARM mbed Client for Cortex-M devices
		ARM:mbedCrypto	Install	ARM mbed Cryptographic library
			🔅 Install	ARM mbed Cryptographic and SSL/TLS library
		H-ARM:minar	💩 Install	mbed OS Scheduler for Cortex-M devices
			🔅 Install+	Trusted Firmware-M (TF-M) reference implem
		ASN::Filter_Designer	🔅 Install	Intuitive graphical FIR/IIR digital filter designer
		EmbeddedOffice:Flexi	🗇 İnstall	Flexible Safety RTOS
		Keit:ARM_Compiler	🔶 Up to date	Keil ARM Compiler extensions for ARM Compil
		Keil::MXRT105x_MWP	🗇 İnstall+	NXP i.MX RT 1051/1052 MDK-Middleware exam
		Keik:MXRT1060_MWP	🗇 Install-	NXP i.MX RT 1061/1062 MDK-Middleware exam
		: Keit:iMXRT1064_MWP	😔 Install+	NXP i.MX RT 1054 MDK-Middleware examples
			🗇 Install	Jansson is a C library for encoding, decoding an
		Keil:LPC55S6x_TFM-PF	🔅 İnstall+	NXP LPC5556x MCU Family TF-M Platform Sup
			🕸 Install+	NXP LPC55569 Series LPCXpresso55569 Board 5
		· Keil:MDK-Middleware	😵 Update	Middleware for Keil MDK-Professional and MDI
		Keil:STM32L5xx_TFM	🕸 Install+	STMicroelectronics STM32L5 Series TF-M Platfo
		+ hwIP::hwIP	🕸 Install	I wIP is a light-weight implementation of the TC

Figure 13

8. Select "Options for Target" menu (Figure 14). Come to the "Debug" menu and select "ST Link Debugger" and 'Settings', respectively (Figure 15). Then choose "Reset and Run" from "Flash Download" so that when you run our code, it will reset its previous information. If you do not select it, you have to press the reset button for the code to run (Figure 16). When you go back to the previous menu, you should choose the "C++" menu and select the "Level 0" for Optimization (Figure 17).



Figure 14



Figure 15

- mco_avx	
] main.c*
DK-ARM	72 /* Reset of all peripherals, Initializes the Flash interface and the Systick 73 HAL_Init(): 74 75 /* USER CODE BEGIN Init */
er/Core	Cortex-M Target Driver Setup
	Debug Trace Rash Download Pack
t.c hal_msp.c F4xx_HAL_[Download Function C Frase Full Chip IP Program RAM for Agorithm Long C Frase Full Chip IP Program Stat: Stat
	Programming Agorithm
	Description Device Size Device Type Address Range
	Start. Size
	Add Remove

Figure 16

ain.c*		
/* Reset of all peripher HAL_Init(); /* USER CODE BEGIN Init	als, Initializes the Flash	interface and the S
Options for Target 'MCU_lab1'		
Device Target Output Listing User	C/C++ Asm Linker Debug Utilities	i -
- Pransessar Sumbola		
	V 403	
Define: USE_HAL_DRIVER.STM3.	21-40 /20	
Undefine:		
Language / Code Generation		
F Execute-only Code	Strict ANSI C	Warnings: 🗚 Warnings 💌
Optimization: Level 0 (O0) -	Enum Container always int	Thumb Mode
C Optimize for Time	Plain Char is Signed	No Auto Includes
Split Load and Store Multiple	Read-Only Position Independent	C99 Mode
One ELF Section per Function	Read-Write Position Independent	IT GNU extensions
Include Patha Mac Controls	32F4xx_HAL_Driver/Inc. /Drivers/STM32F4	x_HAL_Driver/Inc/Legacy.
Compiler -c99-c -cpu Cortex-M4.fp. control ./Core/inc -l ./Drivers/STI	p -D_EVAL -D_MICROLIB -g -O3 -apcs +i M32F4xx_HAL_Driver/Inc -I	terwork -splt_sections -l

Figure 17

9. Now you can start to write our code. The "main.c" file on the left is the file created for us in Keil μVision. Here, you can write various codes using the C language.

Follow the steps below for each code you want to run.

- Write the relevant codes in the part reserved for the user in the while loop.
- After writing the codes, click the build button to create the hex file and other files that will be uploaded to the microcontroller.
- Click the 'Load' button to load the codes into the microcontroller.



- Figure 18
- 1. Use the HAL_GPIO_TogglePin function. Write the following code. Then write the same code without using the 'HAL_Delay' function.

//Toggle the LED connected to the D12 pin at half-second intervals. HAL_GPIO_TogglePin(GPIOD,GPIO_PIN_12); // to toggle led which is connected to the D12 pin. HAL_Delay(500); //Wait 500 ms

2. Use the 'HAL_GPIO_WritePin' function. Write the following code.

// Toggle the LED at 2 second intervals HAL_GPIO_WritePin (GPIOD, GPIO_PIN_14, GPIO_PIN_SET); //Write logic 1 to the output data register of the pin HAL_Delay(2000); //Wait 2 s HAL_GPIO_WritePin (GPIOD, GPIO_PIN_14, GPIO_PIN_RESET); //Write logic 0 to the output data register of the pin HAL_Delay(2000); //Wait 2 s 3. Use the Output Data Register (ODR) directly

```
// Use Output Data register directly
```

GPIOD->ODR|=0xF000; // Turn on the leds connected to the D12,D13,D14&D15 pins.

4. Toggle the LEDs using 'ODR'.

// Use Output Data register directly to do toggle leds GPIOD->ODR|=0xF000; // Turn on the leds connected to the D12,D13,D14&D15 pins. HAL_Delay(2000); //Wait 2 s GPIOD->ODR&=0x0000; // Turn off the leds connected to the D12,D13,D14&D15 pins. HAL_Delay(2000); //Wait 2 s

5. Floating light respectively using 'ODR'

```
//bitwise shifting
GPIOD->ODR|=0xF000; //All of the leds on
HAL_Delay(500); //Wait 500 ms
//Shift the bits right
for (i=1;i<5;i++)
{
GPIOD->ODR>>=1; //Shift the bits right
HAL_Delay(500); //Wait 500 ms
```

}

6. Floating light respectively using 'ODR'.

```
//bitwise shifting
GPIOD->ODR=0x0F00; //All of the leds off (assign bit values)
HAL_Delay(500); //Wait 500 ms
//Shift the bits left
for (i=1;i<5;i++)
{
GPIOD->ODR<<=1; //Shift the bits left
HAL_Delay(500); //Wait 500 ms
}</pre>
```

7. The program that toggles the LEDs when we push the button using IDR (Input Data Register).

```
if (GPIOA->IDR&0x0001) //Checking if the button is pushed
{
    HAL_GPIO_TogglePin(GPIOD,GPIO_PIN_12); // to toggle led which is connected to
the D12 pin.
    HAL_Delay(200); //Wait 200 ms
}
```

EXPERIMENT 3: GENERAL-PURPOSE INPUT/OUTPUT (GPIO)

Objectives

The objectives of Experiment 3 are

- to learn how to use
- ✔ GPIO Output Data Register (ODR),
- ✔ Reading Button Value using Input Data Register (IDR),
- ✔ Debugger,
- ✓ Bit Set Reset Register (BSRR),
- ✔ GPIO_ReadPin function

Apparatus Required:

- STM32CubeMx
- Keil µVision (MDK ARM)
- STM32 ST-Link Utility
- STM32F4 Microcontroller
- STM32F4 Reference Manual
- STM32F4 User Manual

Preliminary Work:

- 1. Study the GPIO (lecture 4) notes.
- 2. Write the codes of the experimental work in Keil μ Vision.

Experimental Work:

 Reading Button Value (Button debouncing). You can understand whether your code is running and control the changes of the variable (Figure 1->Start/Stop Debug Session) using the debugger. Come to the i variable and right click. We select "Add i to" and "Watch 1" (Figure 2). Then click to "Run" and follow the changes of variable i (Figure 1). Right-click on the i which is in the Watch 1 window to convert the i displayed as hexadecimal to decimal. Here you can reset the i value by pushing the reset button on STM32F4G-DISC card and doing a hardware reset.

// Program that increase the value of the variable i by one each time the button is pushed

if (GPIOA->IDR&0x0001) //Checking if the button is pushed using IDR

i=i+1; // Increase the value of the variable i by one each time the button is pushed }







Figure 2

2. Reading Button Value (Prevent button debouncing using HAL_Delay)

// Program that increments the value of the variable i by one each time the button is pushed

```
if (GPIOA->IDR&0x0001) //Checking if the button is pushed using IDR
```

```
{
i=i+1;
HAL_Delay(200); //Wait 200 ms
}
```

3. The program that turns on the LEDs when we push the button.

```
if (GPIOA->IDR&0x0001) //Check if the button is pushed
{
i=i+1;
HAL_Delay(200); //Wait 200 ms
GPIOD->ODR=0xF000; //Assign 1 to the PD12, PD13, PD14 & PD15 pins
}
```

4. The program that turns the LEDs on when we push the button otherwise turns the LEDs off (use BSRR to assign logic 0 to the relevant pins)

```
GPIOD->BSRR=0xFFFF0000; //Reset ODR pins of the D port using BSRR
if (GPIOA->IDR&0x0001) //Check if the button is pushed using IDR
{
i=i+1;
GPIOD->ODR=0xF000; //Assign 1 to the PD12, PD13, PD14 & PD15 pins and
HAL_Delay(2000); //Wait 2 second
}
```

5. The program that turns the LEDs on when we push the button otherwise turns the LEDs off (use ODR to assign logic 0 to the relevant pins)

```
GPIOD->ODR=0x0000; //Assign logic 0 to the pins at the D port
if (GPIOA->IDR&0x0001) //Checking if the button is pushed using IDR
{
i=i+1;
HAL_Delay(200); //Wait 0.2 second
GPIOD->ODR=0xF000; //Assign 1 to the PD12, PD13, PD14 & PD15 pins,
HAL_Delay(2000); //Wait 2 second
}
```

6. The program that turns the LEDs on when we push the button otherwise turns the LEDs off (use ODR to assign logic 0 to the relevant pin and use ReadPin function to read the button).

```
GPIOD->ODR=0x0000; //Assign logic 0 to the pins at the D port
```

if(HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_0)) //Check if the button is pushed using ReadPin function

```
{
  i=i+1;
  HAL_Delay(200); //Wait 0.2 second
  GPIOD->ODR=0xF000; //Assign 1 to the PD12, PD13, PD14 & PD15 pins, assign
  HAL_Delay(2000); //Wait 2 second
}
```

7. A program that increases the value of i by one each time a button is pushed, at the same time, if i is an even number, toggles the related LED(which is connected to the PD12 pin), otherwise it turns off the LED (Control the i value using debugger). Use debug to see how to change the i variable.

```
while (1)
Ł
if(GPIOA->IDR&0x0001) //Check if the button is pushed using IDR
ł
i=i+1; // Increase the value of i by one each time a button is pushed.
HAL Delay(200);
}
// If i is an even number, let the led toggle otherwise led is off
if(i%2==0) //Check for an even number
{
HAL GPIO TogglePin(GPIOD,GPIO PIN 12); //Toggle 12<sup>th</sup> pin of the D port.
HAL Delay(200); //Wait 0.2 second
}
else
{
GPIOD->BSRR=0xFFFF0000; //Reset ODR's pins of the D port using BSRR
}
}
```

8. Create a function named button that checks if the value of the variable i is an even number or not. If the value of the variable i is an even number, this function will turn on the LEDs connected to the 12th and 14th pins of the D port, otherwise LEDs which are connected to 13th and 15th pins of the D port. Then, write another code in the while loop that checks if the button is pushed and increments the value of the variable i by one each time the button is pushed. Call the button function here. Use the debugger to monitor what the value of the variable i is each time you push the button, and also observe which led is lit based on that value.

(Create the button function at the part of the Private function prototypes (PFP) in the main.c file.

Write a code in while loop to check if button is pushed and call button function here)

//If it is an even number, the function that turn on the LEDs connected to the 12^{th} . and 14^{th} pins of the D port, otherwise the function that turns on the LEDs connected to the 13^{th} . and 15^{th} pins of the D port.

```
void button (int a) // Creating a function named button
if (a\%2==0) //Check if a is an even number
 GPIOD->ODR=0x5000; //Turn on the LEDs which are connected to 12<sup>th</sup>. and 14<sup>th</sup>
pins of the D port
}
else
ł
  GPIOD->ODR=0xA000; //Turn on the LEDs which are connected to 13<sup>th</sup>. and 15<sup>th</sup>
pins of the D port
}
while (1)
//Checking if the button is pushed and calling the button function
if( GPIOA->IDR&0x0001) // Check if the button is pushed
i=i+1;
HAL Delay(200);
button(i); //Call the button function
```

}

EXPERIMENT 4: INTERRUPTS

Objectives

The objectives of Experiment 4 are

• to learn how to use interrupt peripherals

Apparatus Required:

- STM32CubeMx
- Keil µVision (MDK ARM)
- STM32 ST-Link Utility
- STM32F4 Microcontroller
- A Jumper Cable (female-female)

Preliminary Work:

- 1. Study the Interrupt (L05) notes
- 2. Write the codes of the experimental work in Keil μ Vision.

Experimental Work:

- 1. Create a new project in CubeMx. Select STMF407VGTx and then STM32F407G-DISC1. First adjust the Pinout&Configuration settings. Close the unnecessary pins. Select the PA0 pin as GPIO_EXTI0 and PA1 pin as GPIO_EXTI1. Select the PD 12-13-14-15 pins as GPIO_Output.
- Come to the System Core menu. You can change the pin configurations by selecting related pins from here (Figure 1). Select the pull down for PA0&PA1 pins. Select "Output Push Pull" for the GPIO Mode, "Low" for the GPIO output level & Maximum output speed for related pins (PD12- PD13- PD14- PD15).

						SH UP S		_	-	0.11000				
Pinout	& Conti	Gruppon			Clock Con	liguration		Set Elizabeth	Project Mr	inager		To	ols	-
a	90				OFIC Mode :	and Confidential	ion .	- Internet			C Finant viv	w 11 Suster	u dener	
Situation As		-			Card	question				-				
 Bystem Core	14	Group I	ly Pauphin	h.					1					
		0.000	. • Sr	igie Mapped	Signala 🛛 🛛 A	icc 💿 sr	5 0	MC						
 · Conor		Seath	Gaula									99		1
MPDG APAC		Death	(Col=2)					🗋 Shise only h	foldfied Piris			PEL O		
· 200		Pre Na .	- Secol	e . (G710 a	u jano na	NORO PAR	Martin	n Une Land	Monited.			75+		
WWDG		PAS-WK	UP n/a n/a	11/2	Esternal I.	Pul-down Pul-down	11.0		8			708		
		PD12	104	1.000	Output Ptu	his par-op	Low	LDF (Gree	72			TAR		
Analog	2	PD13	1/2	Low	Output Pu Output Pu	tio pull-up	Low	LD6 [Red				PE14.		
Timera	- X.	2016	- 10	Low	Ovtput Pu	tio pull-up.	Law	(D6)Blue	10			NIN.		
Contractivity	- X											104		
Materiada											PHD-000_3H			J
	10										Automotion (MALE .		ľ
Decumy	-											PAE		4
Campiang	2													
Multimare	×	P012 0	infigration											s
			/									VISA		
		eros	dist level			Law			्रम्			Contra Co		
	/	OPO #	-			Output Plush	N. I.		1		4940_81745			
-							18				SPID_E1715			
4		GPIC P	ve-spPul-	COVER 1		to pull-up and	t no pail-i	Cowin .				2 5 2	1 2 2 3	l
		Maxim	in sulpit o	peed.	1	Lew.			<u></u>			1961 21 2	41 41 41 4	1
		User La	Del		- Di	LD4 (Green La	-							
										0 F		17.2	-	Į.
										2 1	j Q	-		8

3. Come to the NVIC menu. Firstly set the enable mode for the EXTI line0 & EXTI line1(Figure 2, 1 and 2 steps). Then, identify the priority levels of the interrupts. Select the Priority Group as 2 bits (which indicate how many bits are needed to identify the priority level) (Figure 2, 3. step). Then, select preemption priority as 1 for the EXTI0 and as 2 for the EXTI1 (Figure 2, 4. step).

Pinout & Configuration	Clock Configuratio	n	Project Manager		Tools
	 Software Packs 	 Pinut 			
9	NVIC Mode and Configu	ration		Pinout view	PR System view
Contraction (Marcal	Configuration				
System Core	ode generation				
. Privaty Group 8	tits for pre	Priority and Sub Priority	4		
DMA		ander H end	and a second		2 2 3 8 8 8 8
Geno Search (E		interrupts 🖬 Force D	MA channels Inter	lare .	2 2 2 C E E E E E
IWOG	10/10 Internation Table	Enabled Press	alos Petersy Tists I	100	
V Hotel				124	
A DYD Hard facil internet	outs.		X	1922	
WWDG Memory managem	ant fault	0 15	0	100	
Pro-letch fault, me	mory access fault	89 O	0	VEAT	
Undefined instruction	on or illegal state	23 O	0	POTE	
Analog System service ca	# sis SVA instruction	85 0	0	PCH	
Timers Debug monitor		E 0	0	PC10.	
Pendable request	br system senice	E 0	0	1/20	
Connectivity > Time back: System	n lick timer	B 0	<u> </u>	VIE	
PVD interrupt theo	gn EXII ine 16	0.0	2	PHGIOSE_JH	
Multimedia > Press goods internet			0	PHILOSO OUT	
EXT line0 externut			0	No. 27	
Security 1 - EXTI line1 interrupt		2 . 0 0	0	P05	
Computer PPU global interror	1	0	0	P01	
	Þ			PCI .	
Mddexare >				9123	<pre></pre>
				100	
				VERA	
				UNER	
				VODA	4
				BRID_EXTO	4
				BPID_EX711	
				[AN]	and the local section and have
					2 1 8 7 9 8 8
					CALCULATION AND AND

Figure 2

4. Come to the Clock Configuration menu and control the settings as in Figure 3.



Figure 3

5. Come to the Project Manager and adjust necessary settings as in Figure 4. Then click the Generate Code (Figure 4). Keil μVision programme will be opened. You can see the settings which were already done in CubeMx in the main.c file in Keil μVision (Figure 5). You can change the adjustments from here without going back to the CubeMx. Build the codes in the main.c file. Double click the interrupt file (stm32f4xx.c). You can see the interrupts functions here (Figure 6). You can write codes in the functions. If you want to understand what the function does, you can right click on the function and select the 'Go to Definition...' shown as in Figure 7.

STM32	File	Window Help		🐵 🖪 ല 🏏
Home > STM32F4	107VGTx - STM92F407G-D	ISC1 VINtitled - Project Manager		GENERADE CODE
Pinout & C	Configuration	Clock Configuration	Project Manager	Tools
Project	Project Sattings Project Name MCU_lab3 Project Location C:\cubernxprojects\		Brawse	
	Application Structure Advanced	✓ Do not generate the mail	n()	
Code Generator	C \cubemxprojects\MCU_tat Toolchain / IDE MDK-ARM	31 Min Version ↓ V5.27 →	inis Under Aber	
Advanced Settings	Linker Settings Minimum Heap Size Minimum Stack Size	3x200		
	Mcu and Firmware Package Mcu Reference STM32F407VGTx			
	Firmware Package Name an STM32Cube FV_F4 V125	d Version	on	
	Use Default Firmware Lo	cation STV:32CuberReposition/ISTV00Cube_PW_P4_V1	25 II) Browie	

Figure 4



Figure 5

🗈 🖽 🥔 • 📖 🛱 🛛 MCU_IA63	▲ 本 * * * ● ● ●
4 🔛	main.c stm3204xx, tt.c
Project: MCU_lab3	189
MCU_leb3	190 /* USER CODE END SysTick_IRQn 1 */
E Carlor Application/MDK-ARM	191)
CMSIS	192
> Application //hen/Cate	193 /
	194 / STHIJPARK Peripheral Incerrupt Handlers
a main.c	195 /* And here the interrupt managers for the used peripherals/
(i) stm3264xx_it.c	197 / roles of walland priphting interve and the start near (197)
III	198 /
# 🤄 Drivers/STM32F4xx_HAL_Driv	199
🗉 🚰 Drivers/CMSIS	200 = /**
	201 * Bbrief This function handles EXTI lineD interrupt.
	202 - +/
	203 void EXTID_IRQHandler(void)
	204 - 1
	205 /* USEN CODE SECON EXILO INCH O */
	207 24 DAFE CODE FUD EXTERIDE 4 4
	208 Hal OPIO EXTL INCHANGER (OPIO PIN Cor
	209 /* USER CODE REGIN EXTIC INCH 1 *7
	210
	211 /* USER CODE END EXTIG_IRQM 1 */
	212
	218 T
	214 []/** *
	The sector fals function manages that line interrupt.
	217 vaid EXTII IRORandler(vaid)
	218 - 1
	219 /* USER CODE BEGIN EXTIL IRON 0 */
	220
	221 /* USER CODE END EXTILINGS 0 */
	222 HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_1);
	223 /* USER CODE BEGIN EXTLLINGN 1 */
	224
	225 /* USER CODE END EXTILINGE 1 */
	44 7 1 227
	228 /* UNER CODE REGIN 1 */
	229
	230 /* USER CODE END 1 */
	231 /********************************** (C) COPYRIGHT STRicroelectronics *****END OF FILE****/
	232

Figure 6



Figure 7

6. **a.** Write an interrupt handler function that increases by 1 the value of the variable i if an interrupt is generated from the PA0 pin. Write the codes in EXTI0_IRQHandler function in stm32f4xx it.c file.

b. Write an interrupt handler function that increases by 1 the value of the variable a if an interrupt is generated from the PA1 pin. Write the codes in EXTI1_IRQHandler function in stm32f4xx_it.c file.

Don't forget to identify variable a and i variables in the 'private variables' part of the stm32f4xx_it.c file.

Observe the change of the i variable when you push the button using debugger.



Figure 8

- 7. Follow the instructions given in a, b, c in order. The relevant codes are given below.
 - **a.** When there is no interrupt, the LED connected to the 12th pin lights up continuously. (Write the relevant code inside the while loop in main.c).

```
while (1)
{
   /* USER CODE BEGIN 3 */
   //Light the 12th pin when the interrupt handler is not working
   HAL_GPIO_TogglePin(GPIOD,GPIO_PIN_12); // Toggle the PD12 LED
   HAL_Delay(100); //Wait 100 ms
}
/* USER CODE END 3 */
```

b. When the interrupt is received from the PA0 pin, the value of the i variable increases by 1. Reset all pins connected to port D using BSRR. After the LED is connected to the PD13 pin lights for 5 seconds, all the pins connected to the D port are reset again. Write the relevant code inside the EXTI0_IRQHandler function.

```
void EXTI0_IRQHandler(void)
{
    HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_0);
    /* USER CODE BEGIN EXTI0_IRQn 1 */
i=i+1; //increase i value by 1
    GPIOD->BSRR=0xFFFF0000; //Reset the PD pins
    GPIOD->BSRR=0xFFFF2000; // Set 1 PD13
    HAL_Delay(5000); //Wait 5 s
    GPIOD->BSRR=0xFFFF0000; // Reset the PD pins
    /* USER CODE END EXTI0_IRQn 1 */
}
```

c. When the interrupt is received from the PA1 pin, the value of the 'a' variable increases by 1. Reset all pins connected to port D using BSRR. After the LED is connected to the PD15 pin lights for 5 seconds, all the pins connected to the D port are reset again. Write the relevant code inside the EXTI1_IRQHandler function.

```
void EXTI1_IRQHandler(void)
{
    HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_1);
    /* USER CODE BEGIN EXTI1_IRQn 1 */
    a=a+1; //increase a value by 1
    GPIOD->BSRR=0xFFFF0000; //Reset the PD pins
    GPIOD->BSRR=0xFFFF8000; // Set 1 PD15
    HAL_Delay(5000); //Wait 5 s
    GPIOD->BSRR=0xFFFF0000; // Reset the PD pins
    /* USER CODE END EXTI1_IRQn 1 */
}
```

- **d.** Compile the codes and upload them to the microcontroller. Observe the change of i and a variable using debugger. Use the button on the microcontroller to send an interrupt from the PA0 pin. Use the 5V on the microcontroller discovery card to send the interrupt from the PA1 pin (You can connect 5V to the PA1 pin with the help of a jumper).
- 8. Use priorities of the interrupts (Go back to the 3 to remember the priorities of the interrupts). Use the same codes as in 7.

a. After giving an interrupt from PA0 pin, give another interrupt from PA1 pin before the interrupt handler is completed. Observe the changes of i, variables and LEDs. Observe the 'Tail Chaining Scenario'.

b. After giving an interrupt from PA1 pin, give another interrupt from PA0 pin before the interrupt handler is completed (**Late Arrival Scenario**). Observe the changes of i, variables and LEDs.

9. Learn how to use the interrupt mask register (Examine the properties of the register from Reference Manual). Write a code inside the EXTI1_IRQHandler function. When a value is greater than 5, mask pin 1 using the Interrupt Mask Register. Use EXTI->IMR statement to reach the interrupt mask register and assign a hexadecimal number to this register that will set the corresponding pin value to zero. Build and Load the code. Use a debugger to observe the chaining of a value. Write down your observations about what the result was.

void EXTI1_IRQHandler(void)

{

```
HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_1);
```

```
/* USER CODE BEGIN EXTI1_IRQn 1 */
```

a=a+1;

```
GPIOD->BSRR=0xFFFF0000;//Reset the PD pins
```

```
GPIOD->BSRR=0xFFFF8000;// Set 1 PD14
```

```
HAL_Delay(5000); //Wait 5 s
```

```
GPIOD->BSRR=0xFFFF0000;// Reset the PD pins
```

//Since a>5, interrupts from line 1 are not detected

```
if (a>5)
{
EXTI->IMR=0x7FFFD; //Masked the 1. pin
}
/* USER CODE END EXTI1_IRQn 1 */
}
```

EXPERIMENT 5: TIMERS

Objectives

The objectives of Experiment 5 is

• to learn how to use Timer peripherals

Apparatus Required:

- STM32CubeMx
- Keil µVision (MDK ARM)
- STM32 ST-Link Utility
- STM32F4 Microcontroller
- A Female-Female Jumper Cable

Preliminary Work:

- 1. Study the Timer (lecture 6,7) notes
- 2. Write the codes of the experimental work in Keil µVision.

Experimental Work:

1. Create a new project in CubeMx (Figure 1). Select STMF407VGTx, then STMF407G-DISC1 and finally Start Project (Figure 2). First adjust the Pinout&Configuration settings. Close the unnecessary pins (Figure 3).

STM32CubeMX Untitled				
STM32	File	2	Window Help	9
Home >	New Roject	CHI-N CHI-N		
Existing Projects	Save Project Save Project As Obse Project	CH-S CH-A	New Project	Manage software installa
Recent Opened Projects	Recent Projects	CILK .	Terretory	Check for STM32Cubel
lab5_trial.ioc Last modified date : 26/12/202	Exit	Сних	Start My project from MCU	GEOR
MCU_lab4.ioc Last modified date : 22/12/202	0 13 30 07	MX	ACCESS TO MCU SELECTOR	Install or remove embed
lab_trial4.ioc Last modified date : 19/12/202	0 16 46 55	MX	Start My project from ST Board	
MCU_lab3.ioc Last modified date _ 15/12/202	0 12 35 31	МХ	Start My project from Example	
lab2trial.ioc		МХ		

Figure 1



STM32F407VGTx LQFP100 MAL MARKA



2. We use General Purpose Timers (TIM2 to TIM5) mostly, so look at the reference manual to get information about these timers. Find out which bus is connected to TIM2 from the reference manual. Then come back to CubeMx and check the speed of this bus in the Clock Configuration tab (Figure 4). Then come back to Pinout&Configuration tab and choose which timer you will use (we will use TIM2) and make the necessary settings for this timer (select clock source as internal clock) (Figure 5). Adjust these settings as the prescaler value is 41999, counter mode is up, counter period is 1999. Think about what the meaning of these adjustments are. Go to the NVIC tab and enable the interrupt (Figure 6). Also, go to NVIC in the System Core tab and adjust Preemption Priority value as 1 for the TIM2 (Figure 7).

Pinout & Configuration

Configuration Project Manager

Tools



Figure 4



Figure 5

TIM14		Paset Configuration	Configuration		
Connectivity	×	Parameter Settings Ø User Constant	s 🙆 NMC Settings	OMA Settings	
		TVVIC interrupt Table	Enabled	Preemption Priority	Sub Priority
Multimedia	>:	TIM2 global interrupt	2	0	1
Security	>				
Computing	2				

Figure 6

Select Priority Group as 1 bits NVIC Mode and Configuratio Asz 0 R System Core Priority Group 1 bits for pre-emption priority 3. - Sort by Premption Priority and Sub Priority DMA GPIO IWDG Non maskable interrupt Hard fault interrupt WWDG Memory management fault Pre-fetch fault, memory access fault Undefined instruction or illegal state Analog > System service call via SWI instruction Debug monitor 5 Timers Pendable request for system service Time base: System tick timer > Connectivity Time base. System tick timer PVD interrupt through EXTI line 15 Flash global interrupt RCC global interrupt TIM2 global interrupt Multimedia > > Security FPU global interrupt 5 Computing Select Premption Priority as 1 for TIM2. 2 Middleware

Figure 7

3. Come to the Project Manager tab and set necessary configurations here. Then you can continue with the Keil μ Vision. Don't close the CubeMx, because you will change something from CubeMx.

Project Name Project Name Project Name Project Name Project Name Project Name Project Name Project Name Project Name Project Name Project Name Project Name Project Name Project Name Project Name Project Name Project Name Project Name Application Transferrationary Convidences of the many. De not persente the many. Totcham / DOP Min Version Money Head State Vision Money Head State Vision Money Head State Name Money Head State Name	Instal Project Name Project Name th=""><th>Minist Project Name Project Name <td< th=""><th>Pinout & Configuration</th><th>Clock Configuration</th><th>Project Manager</th><th>Tools</th></td<></th></td<>	Minist Project Name Project Name th=""><th>Pinout & Configuration</th><th>Clock Configuration</th><th>Project Manager</th><th>Tools</th></td<>	Pinout & Configuration	Clock Configuration	Project Manager	Tools
Project Location Project Location Application Structure Application Project Location Project Location Reveal Project Location Reveal Project Location Advanced	Project Locatie Project Locatie Project Locatie Project Locatie Application Structure Application Structure Application Structure Application Structure Advanced Image: Dot not generate the man() Toolstand TOC Min Version Toolstand TOC Min Version Moment Stack Stars Model Model of Horizont Model Stars	Project Netlings Project Name MCU_Land				
Pathener Pathener Image: Comparison of the many image	Advanced v) Do not generate the man() Solution Field Location Toolshan Field Location Politikan Field Location Min Vension Min ARAB Min Vension	Advanced v Cb not generate the man() Torbuhan Fulled Logitier Torbuhan Fulled Logitier Forgenitarity (CP) Pagenitarity (CP) Pagenitarity (CP) Label (CP) (CP) (CP) (CP) (CP) (CP) (CP) (CP)	Project Location weiBME1321 Milloodeneticysc Programiamaya_Gen/V Application Structure	Moncontroller 1.464.464.46. Programlan		
NDGK-ABBI vis.27 Vis.27 III Times are State Plant Lodor Settings Monimum State Plant Monimum State Plant Monimum Research Setting ModBin Monimum State Plant	MCK-AFRA VS.27 Immuno State Final Momine State State Booton Momine State State Booton Moti and Fermoure Package Moti and Fermoure Package	MCK-AFRA VE.27 Common Union Float Model Contract Hog State Model Model Controw Model Model Co	Genoteine Toshtan Fulde Loatin ustura EMETUT Marolenelleyici Programianaja Toshtan / IDE Mer Venan	Clino Ad generate the man)		
	Moral Betrings	Ad Centings Adva and Ferman Package Adva and Ferman Package Adva and Ferman Package Information Package In	MDH-ARM V [VE 27	Gamers that Red		



4. Build the main.c file.

Childelen, VTU_UME, Lectures/BME	3321 Mikodevelleyko, Programiamaya, Gras Mikorcontoster LabiLabiLab Programian/MCU LabiMMOK-ARM/MCU LabiLusproja - "Wisko	- d X
File Edit View Project Flash De	bug Periphexals Tools SVCS Window Help	
1 2 4 4 2 4 3 9	○ • ● 在在在 年出市出 通 □	4
🕸 🔛 🖽 🥔 • 🔠 🕅 🛛 менд	ah4 🛛 🐼 🛔 🗣 🕈 🃾	
Prusit (1) Build (77)	1 mains	* X
 Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters Battlangetters<!--</td--><td>19. /* URER CODE END Incides */ 0 /* Includes 21. * Frivate includes 22. /* Frivate includes 23. /* Frivate includes 24. /* URER CODE ENDI Includes */ 25. /* URER CODE ENDI Includes */ 26. /* URER CODE ENDI Includes */ 27. /* Private spreiff 28. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDIE #D */</td><td></td>	19. /* URER CODE END Incides */ 0 /* Includes 21. * Frivate includes 22. /* Frivate includes 23. /* Frivate includes 24. /* URER CODE ENDI Includes */ 25. /* URER CODE ENDI Includes */ 26. /* URER CODE ENDI Includes */ 27. /* Private spreiff 28. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDI Includes */ 29. /* URER CODE ENDIE #D */	

Fig	ure	9
~		

5. Start the TIM2 in interrupt mode using the HAL_TIM_Base_Start_IT function under /*USER CODE BEGIN 2*/ comment line in main.c file. Then go to the interrupt file (stm32f4xx_it.c) to write the interrupt.



Figure 10

6. Come to the 'TIM2_IRQHandler' function in the stm32f4xx_it.c file. When the timer completes each period value, write the code that comes into the interrupt request and increases the i variable by 1. Don't forget to define 'int i'. Build the file and load the codes to the microcontroller. Observe the change of i variable using debugger. Think about what's going on inside the TIM2_Handler function and how.

```
202 - */
203 void TIM2 IRQHandler (void)
204 - {
     /* USER CODE BEGIN TIM2 IRQn 0 */
205
206
207
     /* USER CODE END TIM2 IRQn 0 */
208
    HAL TIM IRQHandler(Shtim2);
     /* USER CODE BEGIN TIM2 IRQn 1 */
209
   i=i+1;
210
211
     /* USER CODE END TIM2 IRQn 1 */
   3
212
213
214 /* USER CODE BEGIN 1 */
215
216 /* USER CODE END 1 */
```

Figure 11

7. Make a clock application using the TIM2 timer. Define three variables: second, minute, hour. Increment the value of the second variable each time an interrupt request is generated. When the value of the variable i is equal to 60, reset it again and assign a value to the minute variable. When the value of the minute variable is equal to 60, the value of the minute is reset and the value of the hour variable is increased by one. When the value of the hour variable is equal to 12, the value of the hour variable is also reset. Let the cycle continue like this. Observe the change of variables using debug.(Figure 12.1)

Change the htim2.Init.Period value as 19 in main.c file (Figure 12.2). Observe the change of variables using debug. Explain how there has been a change in the operation of the code.

```
void TIM2 IRQHandler (void)
] {
  /* USER CODE BEGIN TIM2 IRQn 0 */
  /* USER CODE END TIM2 IRQn 0 */
  HAL TIM IRQHandler(&htim2);
  /* USER CODE BEGIN TIM2 IRQn 1 */
  second=second+1;
  if (second==60)
1
 -
    second=0;
    minute=minute+1;
  1
  if (minute==60)
1
 {
    minute=0;
    hour=hour+1;
  1
  if (hour==12)
} {
    hour=0:
  1
  /* USER CODE END TIM2_IRQn 1 */
3
```



Figure 12

8. Close the Keil μ Vision and go back to CubeMx to generate a PWM signal. Use another TIM (TIM4) so disable the clock source for TIM2 (Figure13). Find out which bus TIM4 is connected from the reference manual. Then come back to CubeMx and check the speed of this bus in the Clock Configuration tab. Then come back to Pinout&Configuration tab and choose which timer you will use (we will use TIM4) and make the necessary settings for this timer (select clock source as internal clock and PWM Generation CH4)(Figure 14). Adjust these settings as prescaler value is 41999, counter mode is up, counter period is 1999. Explain what the meaning of these adjustments are. Come to the Project Manager tab and set necessary configurations here. Then you can continue with the same Keil μ Vision file (Figure 15).



Figure 13



Figure 14

STM32	File	Window Help		🎯 🖪 🗖 🎽 🔆 🏹
Hame STM32F	407/GTx - STM32F407G-D Configuration	ISC1 XICU_lab4.loc - Project M: Clock Configuration	anager > Project Manager	GENERATE CODE Tools
Project	Project Settings			
Code Generator	Advanced Toolchain Folder Location C 'scuberoxprojects'/MCU List Toolchain / IDE MDK-ARM	Min Version	main()	
Advanced Settings	Linkar Sattingi Minimum Heap Size Minimum Stack Size	0200 Copying libraries files	4	
	Mcu and Firmware Package Mcu Reference STM32F407VGT#			
	Firmware Package Name an STM32Cube FW_F4 V1.25.1	d Version Use latest available	version	
	E.OverdilMAL CARTORN	THE COMPANY AND COMPANY FR	VT ISS Brown	

Figure 15

9. You can see the functions for PWM in the HAL Library as shown in Figure 16. We use the HAL_TIM_PWM_Start function. This PWM signal is on in 500 of the Counter Period and off in the remaining 1500. Use __HAL_TIM_SetCompare macron (Figure 17). Observe the condition of the LED connected to the PD15 pin.



Figure 16



Figure 17

10. Use CCR (Capture Compare Register) directly to generate PWM signals. This PWM signal is ON mode in 100 of the Counter Period and OFF mode in the remaining 1900. Change the value of ON and OFF mode as 1500, 1000 respectively (Changing duty cycle). Observe the conditions of the LED connected to the PD15 pin.



11. Now, we continue with input capture mode. First, you close the Keil μVision and go back to the CubeMx and change some configurations. Use TIM2 in input capture mode and Channel 1 (which is connected to PA0 pin) of the TIM2 is used in input capture direct mode. So, use a cable to connect PA0 with PD15 pins. Set prescaler value is 41999, counter mode is up, counter period is 1999 for TIM2 (Figure 19). Go to NVIC settings and enable the interrupt for TIM2 (Figure 20). Go to System Core and set the Preemption Priority as 1 for TIM2 (Figure 21). Then click Generate Code and Open Project in Keil μVision.

STM32CubeMX MCU	lab4.inc*: 5	TM32F407VGTx STM32F407G	DISC1								-	15
MIS2 ubeMX		File	Window	()	Help				6	f 🚥	¥ ×	5
ome 🗲 STM3	2F407VG1	Tx - STM32F407G-DIS	C1 > MCU_	lab4.ioc - Pinou	rt & Configurat	ion >	8		GEN	ERATE CODE		
Pinout 8	Config	uration	Clock Co	onfiguration			Project N	lanager		То	ols	
			V Software P	acks	🛩 Pio	out						
	- G	TIM2 Mod	le and Configuration	1			_	D Pinout view	System v	iew.		
Magonias A-SZ			Mode			1				-		-
System Core		Stave Mode Disable		~		PEA						
	1.	Trigger Source Disabl	le	v		PE0						
Analog		Clock Source Interr	tal Clock	¥		URAT						
Timers		Channel1 input Captu	ne direct mode	~		PC13.						
+	_	Channel2 Disable		~		PE14.						
RTC		Channel3 Disable		~		122						
TIM1		Channel4 Disable		~		VOD						
TIMB		Combined Channels	Disable		PHD-OSC_IR	mo l						
TIM		I Lite ETR as Cas	ing Source		PHI-OSC_OUT	Pret.						
TIM6		XOR activition				MRST.						
TIM7		One Pulse Mode				PC0						
1008		57.000 COVER				P02						
TB4110						PCI		eT	MODEAL	VOT		
TIM11 TIM12						000		51	11132541	, vGI)	6	
TIM13			and the second se			VISA			I OFP	100		
TIM14			oringurations			UREF			LORIF	100		
		Reset Configuration			T1W2_CH1							
Connectivity	2	DMA Settings	O GPIO	Settings		PAT						
No. 12	1	User Constants	O NVIC	Settings		PAT	The		-			
Muttimedia	30	Continue the below name	natare				3 50 7	88333	8 5 8 4	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	1114	100
Security	2	Q Compt Colors	0 0	0		1	- all	and an all all a			-Later al	-Lap
		~ Counter Settings	2 21	1								
Computing	<u>.</u>	Prescaler (PS	C 41999									
Middleware	5	Counter Mode	Up									
		Internal Clock	D No Dission	•								
		auto-reload pry	al Disable									
		 Trigger Output (TRGO 	19	H								
		Master/Stave I	M. Disable (Trigger	input affec								
		 Input Capture Channe 	a Preset (UG bit b { 1	um (max	0	6.7	0	05	1.11	-		
		Polarity Select	tion Rising Edge		Q	23	a	1	100			v

Figure 19

TIM13 TIM14				Configuration		
Connectivity	2	Reset Configuration	User Constants	NVIC Settings	DMA Settings	GPI0 Settings
Multimedia	5	NVIC Inter TIM2 global interrupt	upt Table	Enabled	Preemption Priority	Sub Priority
Security	>					

Figure 20

STM32F407VGTX - STM32F402tG-OISC1 MCU_Lab4/sec - Pinout & Configuration Pinout & Configuration Clock Configuration System Core NVIC Mode and Configuration System Core Configuration Privaty Group Extra for pre-singtion p. Soft by Premption Priority and Sub Priority Search Configuration WVOG Privaty Group Extra for pre-singtion p. Soft by Premption Priority and Sub Priority Search Central Clift/File O Show only enabled interrupts Force DMA channels NVOG Privaty Group Extra for pre-singtion p. Softwares of pre-singtion p. Show only enabled interrupts Force DMA channels NVOG Privaty Group Extra for pre-singtion p. Softwares of pre-singtion p. Softwares of pre-singtion p. NVOG Privaty Group Extra for pre-singtion p. Softwares of pre-singtion p. Software pre-singtion p. NAGE Privaty Group Extra for pre-singtion p. O O NAGE Privaty Group Extra for pre-singtion p. Software pre-singtion p. Privaty Present pre-singtion p. NAGE Privaty Group Extra for pre-singtion p. Software pre-singtion p. Privaty Pre-singtion p. Name Privaty Group Extra for pre-singtion p. Software p. S	MX	File	Window	Help				
Pinout & Configuration Clock Configuration Imagenes Av2 Imagenes NVIC Moder and Configuration System Core Imagenes Configuration Imagenes Av2 Imagenes Configuration System Core Imagenes Configuration Imagenes Configuration Imagenes Imagenes Imagenes Imagenes Configuration Imagenes Imagenes <th>e 🔪 STM32F407VGTx - ST</th> <th>M32F407G-DISC</th> <th>MCU_Lab4.io</th> <th>c - Pinout & Configura</th> <th>tion ></th> <th></th> <th></th> <th></th>	e 🔪 STM32F407VGTx - ST	M32F407G-DISC	MCU_Lab4.io	c - Pinout & Configura	tion >			
Volt Software V P Multiplier Av2 NVIC Mode and Configuration Configuration System Core Configuration Configuration Configuration DMA Configuration Configuration Configuration WWOC Configuration Configuration Configuration WWOC Non maskable interrupt Configuration Configuration WWOC Non maskable interrupt Configuration Configuration Marcia Configuration Configuration Configuration Configuration WWOC Non maskable interrupt Configuration Configuration	Pinout & Co	nfiguration		Cloc	k Configuration	n		
Iterations AVIC Mode and Configuration System Core Configuration DMA Priority Group 1 bits for pre-emption p Soft by Premption Priority and Sub Priority DMA Priority Group 1 bits for pre-emption p Soft by Premption Priority and Sub Priority DMA Priority Group 1 bits for pre-emption p Soft by Premption Priority and Sub Priority DMA Priority Group 1 bits for pre-emption p Soft by Premption Priority and Sub Priority Priority Group 1 bits for pre-emption p Soft by Premption Priority and Sub Priority Priority Group 1 bits for pre-emption p Soft by Premption Priority and Sub Priority Priority Group 1 bits for pre-emption p Soft by Premption Priority and Sub Priority Non massable interrupt 0 0 Pre-start bala, memory access fault 0 0 Product excepts for system service 0 0 Priority Group 1 bits for pre-emption p 0 0 Product excepts for system service 0 0 Product excepts for system service 0 0 Product excepts for system service 0 0 TMA Product ex					✓ Software Pa			
Avz Configuration System Core MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVCC MVC MVCC	~	0		NV/C Mod	ie and Configuration			
System Core	points A->Z			C	orfiguration			
Priority Group 1 hts for pre-emption p Soft by Premption Pliority and Sub Priority DMA DMA Group MACC WWDDG Presente full interrupt Image I	stem Core		O NMC O Code gene	ration	10922-001-00			
BMA WWOG Search Search Image: A (C)(1177) Image: A (C)(11			Priority Group 1 bits for pr	e-emption p 🗌 Sor	t by Premption Priori	ty and	Sub Priority	
Search Search <td>DMA</td> <td></td> <td>Count In country</td> <td></td> <td></td> <td></td> <td>-</td> <td>the last start at</td>	DMA		Count In country				-	the last start at
INVOC INVOC NAC String String String WMODG Image: String Marking Image: String Analog Image: String String Image: String Image: String Image:	GPIC		bearch [Dawrith (Ch	UT U U D Sho	w only enabled inten	wpts	Force UMA channe	is interrup
Non maskable interrupt 0 0 WWODG 0 0 0 Analog 0 0 0 0 Analog 0 0 0 0 0 Analog 0 0 0 0 0 0 Analog 0<	INVIC		NVI	Interrupt Table	Enable	5	Preemption Priority	Sub Print
WWOG Imade in interrupt Imade interrupt Imade in interrupt Imade in interrupt Imade in interrupt Imade in interrupt Imade in interrupt Imade in interrupt Imade in interrupt Imade in interrupt Imade in interrupt Imade in interrupt Imade in interrupt Imade in interrupt Imade in interrupt Imade interrupt Imade interrupt Imade interrupt Imade interrupt Imade interrupt Imade interrupt Imade interrupt Imade interrupt Imade interrupt Imade interrupt Imade interrupt Imade interrupt Imade interrupt Imade interrupt Imade interrupt Imade interrupt	RCC		Non maskable interrupt		10	0.0	0	
Memory management turk Image of the point state Image of the point state <thimage of="" point="" state<="" th="" the=""> <thimage of="" td="" th<=""><td>SYS MANDE</td><td></td><td>Hard fault interrupt</td><td></td><td>15</td><td>0</td><td>0</td><td></td></thimage></thimage>	SYS MANDE		Hard fault interrupt		15	0	0	
Analog Pre-Start hult, memory access fact Pre-Start hult, memory access fact Pre-Star	www.cd	1	Memory management fault		5	0	0	
ballog Condition of instruction		2 C	Pre-fetch fault, memory acc	ess fault	12	0	0	
System service all val SVM instruction B 0 0 Timeta Debug monther B 0 0 0 Timeta Debug monther B 0 0 0 0 Pendadle request for system service B 0 0 0 0 0 TiMA PVD interrupt through EXTI line 16 0 0 0 0 0 TiMA PCC global interrupt 0 <t< td=""><td>aloo</td><td></td><td>Undefined instruction or illeg</td><td>pal state</td><td>5</td><td>0</td><td>0</td><td></td></t<>	aloo		Undefined instruction or illeg	pal state	5	0	0	
Immers Datage monotor B 0 0 * Pendade request for system service B 0 0 0 Tree base. System tick timer B 0 0 0 0 0 TRC PVD idencyt tick timer B 0 </td <td></td> <td></td> <td>System service call via SWI</td> <td>enstruction</td> <td>5</td> <td>0</td> <td>0</td> <td></td>			System service call via SWI	enstruction	5	0	0	
Permanan registrin ter syntam service Ba O O RTC Time base Syntam tick timer Bi O O O PVD interrupt through KXT line 16 O O O O O TM4 PVD interrupt through KXT line 16 O O O O TM4 RCC global interrupt O O O O TM4 RCC global interrupt O O O O TM45 TM4 global interrupt O O O O TM6 FPU global interrupt O O O O TM47 TM48 TM49 O O O O TM41 TM413 TM414 O O O O O	ners	~	Disbug monitor		EX	0	0	
RTC PVU interrupt through EXTLine 16 0 0 0 TIM1 Flash global interrupt 0 0 0 TIM2 RCC global interrupt 0 0 0 TIM3 RCC global interrupt 0 0 0 TIM3 RCC global interrupt 0 0 0 TIM4 PU global interrupt 0 0 0 TIM5 TIM4 global interrupt 0 0 0 TIM6 FPU global interrupt 0 0 0 TIM6 TIM4 global interrupt 0 0 0 TIM5 TIM6 interrupt 0 0 0 TIM6 FPU global interrupt 0 0 0 TIM7 TIM8 TIM8 1 0 0 TIM11 TIM12 TIM13 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 <td></td> <td></td> <td>Time have. Sustant for system</td> <td>n service</td> <td>83</td> <td>0</td> <td>0</td> <td></td>			Time have. Sustant for system	n service	83	0	0	
TM41 Plash global interrupt 0 0 TM23 RCC global interrupt 0 0 TM24 0 0 0 TM25 TM24 global interrupt 2 3 0 TM45 TM45 global interrupt 0 0 0 TM45 FPU global interrupt 0 0 0 TM46 TM45 FPU global interrupt 0 0 0 TM46 TM47 TM48 TM49 0 0 0 TM411 TM413 TM413 TM414 0 0 0	RTC		PVD internet through EXT	ine 16		0	ő	
TMA: RCC global interrupt 0 0 TMA: TMA global interrupt 25 1 0 TMA: TMA global interrupt 0 0 0 TMA: TMA: TMA: 0 0 0 TMA: TMA: TMA: 0 0 0 TMA: TMA: TMA: 0 0 0	TIM1		Flash dinhal interrupt	and to		0	0	
TIM3 TM2 global interrupt CI 1 0 <td>1942</td> <td></td> <td>RCC global interrupt</td> <td></td> <td></td> <td>0</td> <td>ő</td> <td></td>	1942		RCC global interrupt			0	ő	
TRAS TRAS TRAS Control Interrupt 0 0 0 TRAS FPU global interrupt 0 0 TRAS TRAS TRAS TRAS TRAS TRAS TRAS TRAS	TIM3		TIM2 global interrupt		10	1	0	
INNS TNAS FPU global interrupt 0 0 TMA TMA 0 0 TMAS TMAS 0 0	TIMAK		FIM4 global interrupt			0	0	
TM7 TM8 TM9 TM10 TM11 TM12 TM13 TM13 TM13	TIME	8	FPU global interrupt			0	0	
	TIM7 TIM9 TIM10 TIM10 TIM11 TIM12 TIM13 TIM14							
Jonnectwity 3	nnectivity	(s)						



12. Start the input capture mode for TIM2 using HAL_TIM_IC_Start_IT function with PWM signal like in Figure 22. So, you can capture the generated PWM signal frequency. You will use the TIM2 interrupt mode. Go to the stm32fxx.c file and write the codes like in Figure23 in this file. Build and load the code. Observe a and b value using debug. Find 'Capture value' (CNT1-CNT2) and calculate applied signal period according to this value multiplying period value for each value (Figure 24).







Figure 23

Input capture mode

 $\begin{aligned} Period &= Capture. \left(\frac{TIMx_CLK}{(Prescaler + 1)(CH_{Prescaler})(Polarity_{Index})} \right)^{-1} \\ Capture &= CNT_1 - CNT_0 \ if \ CNT_1 > CNT_0 \\ Capture &= TIMx_{Period} - CNT_0 + CNT_1 \ if \ CNT_1 < CNT_0 \end{aligned}$

Figure 24

EXPERIMENT 6: USART PERIPHERALS

Objectives

The objectives of Experiment 6 is

• to learn how to use Universal Synchrous / Asynchrous Serial Communications peripherals

Apparatus Required:

- STM32CubeMx
- Keil µVision (MDK ARM)
- STM32 ST-Link Utility
- STM32F4 Microcontroller
- 2 Jumper Cables (female-female)

Preliminary Work:

- 1. Study the USART (L08) notes.
- 2. Write the codes of the experimental work in Keil μ Vision.

Experimental Work:

1. Create a new project in CubeMx (Figure 1). Select STMF407VGTx, then STMF407G-DISC1 and finally Start Project (Figure 2). First adjust the Pinout&Configuration settings. Close the unnecessary pins (Figure 3).

STMS2CubeMX Untitled				
STM32	File	١	Window Help	9
Home Keisting Projects	New Roject Load Project Import Project Save Project Save Project Obser Project Observation Statement	CH4.N CH44 CH44 CH44 CH44 CH44	New Project	Manage software installa
Recent Opened Projects	Recent Projects	,	I need to :	Check for STM32Cubel
lab5_trial.ioc Last modified date : 26/12/202	Exit 0 15:24:59	CHIX MX	Start My project from MCU	GH€CK P
MCU_lab4.ioc Last modified date = 22/12/202	0 13 30 07	МΧ	ACCESS TO MOU SELECTOR	Install or remove embed
lab_trial4.ioc Last modified date : 19/12/202	0 16 46 55	MIX	Start My project from ST Board	
MCU_lab3.ioc Last modified date 15/12/202	0 12 35 31	MX	Start My project from Example	
lab2trial ioc		MX		

Figure 1

	0	Features Large Picture	Occs & Resources	et Citesy Ethonogene
Part Number	C1 4	* STM32F4 Series		
UstM32F407G Dis	<u>cı</u>	STM32F407G-DISC1	STMicroelectronics STM32F407G Dise Examples	covery Kit Board Support and
Туре	>	ACTIVE Active	Part Number: STM32F4DISCOVERY	Unit Price (USS): 19.89
MCUMPU Series	3.	Product is in mass production	Commercial Part Number : STM32F407G-DISC1	Mounted Device : STM32F407VGTx
			digital microphone, one audio DAC with in	degrated class D speaker driver LEDs.
			Push-buttons and an USB OTG micro-AB To expand the functionality of the STM32 connectivity, LCD display and more, visit webpage. With the latest board enhancement, the n replaced the old reference STM32F4Disc Features	connector #4DISCOVERY MIt with ethernet me www.st.comstm02f4dis-expansion new order code STM32F407G-DISC1 has COVERY.

Figure 2



Figure 3

2. We use USART1 and USART2 peripherals in the lecture. Set the configuration for USART1 (Figure 4) and USART2 (Figure 5). Use USARTs in Asynchronous Mode. BoudRate is 9600 Bits/s, Word Length is 8 Bits, Parity is None, Stop Bit is 1, Data Direction is Receive and Transmit, Over Sampling is 16 Samples for both of the USARTs. Select the PA0 pin as GPIO_EXTI0. Go to GPIO settings and set the GPIO configurations for the PA0 pin as in Figure 6. Set the NVIC configurations as in Figure 7.

STM:12 CubeMX	File	Window	Help	0
Home > STM32F407VC	STx - STM32F407G-DISC1	Untitled - Pino	ut & Configuration	\rangle
Pinout & Confi	guration	Clock Configur	ation	Project Manager
	*	Software Packs	🛩 Pin	out
Q @	USAR	C1 Mode and Configuration	99)	Pinout view
2	Mode Asunchananus			
System Core 7	Hardware Flow Control (RS	5232) Disable		PE4
Analog >				P5
Timers >				PER
Connectivity ~				PC13
PBNC ICC1 ICC2 ICC3 SDIO SPR1 SPR2 SPR2 UARTA UARTA UARTS IEADET2	Reset Configuration MMC Settings Parameter Settin Configure the below parameter Settin Scott? Sascia Scott? Sascia Scott?	DMA Settings	• GPIC Settings User Constants	typedef struct {
USARTJ USARTJ USB_OTG_FS USB_OTG_HS	Baud Rate Word Length Parky Stop Bhs -> Advanced Parameters Data Direction	9600 Bits/s 8 Bits (Include None 1 Receive and T	ng Parityj 3	uint32_t BaudRate; uint32_t WordLength; uint32_t StopBits; uint32_t Parity;
Multimedia >	Over Sampling	16 Samples		uint32_t Mode;
Security >				uint32_t HWFlowCtl;
Computing >				} UART_InitTypeDef;
The Archest Control of the State				

Figure 4



Figure 5



Figure 6

Cube/XX		File	Window	1	Help				
Home > STM32F40	7VGTx - ST	M32F407G-DIS	C1 Untitled	- Pinout & Co	onfiguration	\rangle			
Pir	nout & Co	nfiguration			Clo	ck Conf	gurat	ion	
						🛩 Sof	tware	Packs	Y Pi
۹ 🔄 👻	0			NVIC	Mode and Conf	guration			
Categories A->Z		-		-	Configuration				
System Core	~2		Code generation						
	-	Priority Group	1 bits for pre-emption	priority 3 v	Sort by Pre	mption Pric	rity and	Sub Priority	
DMA		Search	Searoli (Gd/+F)	00	Show only	enabled inte	erupta	Force DMA channels	Interrupti
IWDG									
NVIC	1	_	NVIC Intern	upt Table		Enable	II P	teemption Priority Sal	a Priority
V RCC		Non maskable in	nterrupt			82	0	0	
A 313		Hard fault intern	apt			82	0	. (P	
AAAADG		Memory manage	ement fault			52	0	0	
		Pre-fetch fault, r	nemory access fault			12	0	0	
		Undefined instru	ction or illegal state			82	0	0	
Analog	2	System service	call via SWI instructio	0		55	0	0	
		Debug monitor				10	0	0	
Timers	×.	Pandable remie	at for system service			10	0	0	
		Time have Sur	tion tick times			80	0	0	
Connectivity	~	PVD interned th	rough EVTI line 15					0	
		Foot alabeliate	roogn Extraine to			1		0	
CANA		Plate global inte	rerupt			1	0	0	
CAND	25	RUC global intel	rupt				.0	0	-
A ETH	3	EXTI line0 intern	upt				3	0	-
ESMC		USART1 global	interrupt				1	1	
IPC1	4	USART2 diobal	interrunt.			82	1	2	
12C2		FPU global inter	rupt				0	0	
SDIO SPI1 SPI2 SPI3 UART4 UART5 VUART5 VUART5 VUART5 VUART5 VUART5 VUART5 VUART5 VUART5 VUART5 VUART5 VUART5 VUART5 VUA VUART5 VUA VUA									

Figure 7

3. There is no hardware connection between the USART. Use the cables to connect the USART peripherals to each other. (Connect PA2 with PA10; Connect PA3 with PA9 using cables). Now, you can set the project manager configurations as in Figure 8 and go to the Keil μVision by selecting Generate Code.

TM 12 CoderAll	File.	Window Help		🎯 🖪 🚥 🈏 🔆 🕰
tatta > STAROF	407V01+ - STM33F4070-D	SCI 🔪 Untitled - Project Manager 🔪		GENERATE CODE
Pinout & (Configuration	Clock Configuration	Project Manager	Tools
	Project Datings Project Ibarris MC12, 1465			
	Project Canadian C'Inviterençingingin		Brank	
	Application Diractions Advanced	- El Du not generale tre many		
	Textchain Partier Location El scalamegingens MCLL M			
	Teletan / EE (VDCARB	Mex Varning	Concesse:	
	Loine Sellege Meanum Heap Sale Meanum Deale Sale	utili		
	Mcc and Extreme Package Mcc Reference IED/COF40TV0Ts			

Figure 8

4. Go to main.c file and build the codes. Examine the main.c file to observe the configurations which are already done in CubeMx. You should write the codes in interrupt mode. Open the stm32fxx_it.c interrupt file.

100000000000000000000000000000000000000		
State and a state of the second	PA A = + 20	
and O Re	(1) mad	
TT Design Advit tabl		
a project on a set	40 /* UMER CODE END PN */	
= #2 Mr.U_MED	14 Jan Factorian and Aller	
III 🚘 Application/MDK-ARM	14 / FIGTE Participants (houses)	
OMS5	14 UAST BandleTypeDef huszl2:	
E 2 Application/User/Core	4	
men.c	46 /- UMER CODE MEDIN BV +/	
- D. sterilities it a	47	
a Charlen bei will ann a	(E /- USER CODE END FV -/	
a a successive interest	49	
In the Deventry Institution Peak Dev	50 /* Trivate function prototypes*/	
E Dovers/CMSD	<pre>si void systemcloce config(void)/</pre>	
	54 PLATE TOLO AN UPID_AILS (VILA):	
	The static valid MC Party (Mari (Mari (Mari))	
	14	
	ST /- UMER CODE END PEP -/	
	58	
	15 //* Trivate user code*/	
	40 /* USER CODE BEGIN S */	
	41	
	62 /* UMER CODE END 0 */	
	63	
	66 - Dervel ine apparentien mary posts.	
	67/	
	68 Int main (void)	
	69 日1	
	70 /* UMER CODE BEAZH 1 */	
	71	
	72 /* UMER CODE END 1 */	
	78	
	74 /* HCU Configuration	
	76 In Process of all annihilations. This allows the Winds increasing and the Branch of	
	(%) With the set of the peripetricity, continuents for finite interiment and out systems. "	
	78	
	TW /* UNER CODE MADIE Taxt */	
	=0	
	01 /* UMER CODE END Init */	
	62	
1 1 1	83 /* Configure the system clock */	
	Kell Russemüssik Coefinis	

Figure 9

5. When the button is pushed, transmit data from UART2 and write it into the transmit buffer. Then go to UART2 and receive data from UART1. Write the data into the receive buffer.

You should write the codes in interrupt mode. Open the stm32fxx_it.c interrupt file and write the codes in that file (Figure 12). First, define the transmit and receive buffer which will keep the transmitted and received data (Figure 10) . Use HAL_UART_Transmit_IT function in EXTI0_IRQHandler function and HAL_UART_Receive_IT function in USART2_IRQHandler function which is given in Figure 10. Finally, build and Load the codes. Follow the change of variables using the Debug menu (Figure 13).



Figure 11



Figure 12

A. Citosbimopreparter.MCU.	taty MDR-ABLF MCU (add. argunge - a Water (Nam-Commercial Use License)					1459 40 ME			
Fie Est Vew Print	Flash Debug Perjaherals Tosts SVCS Window Help								
DUNNA	3 2 - + - P 3 3 5 F # # # # 9 veter	23 A 1027 4	04-10-4						
215 DI B B C	IN A LTHE AMOUNT OF THE R. M. M.								
HEF I LEF WITH LINE U	THE PART AND A DESCRIPTION OF A DESCRIPR								
tepter • M	Disarrowy	al section data and the							
Regatar Va	Categolice 2204 HOVS 22, KONDE	STRUCTINGSTAR) (1							
10 0	0x080011CA 4903 LDR r1, [pc, #12] : 00x080	001108				0			
(1)	Controlling Figuretter & W Hat Harrison TT	LDM D0.[pc.#ll] / #0x0000110C				11			
10 D.	0#16001102 0000 DCM 0#0000								
- 14 - 6-									
10 0.	amania dentaritata Aa atariag statutata a					- ×			
-R7 DL	209 /* USER CODE END EXTID_INON 0 */								
- R8 0x.	210 HAL GFID EXTI INCHANNILES (SPID FIN 9) /								
-10 Dr.	212 HAL UART Transmit IT (shuart2, (wint8 t*)transmit	t. sizeofitzatemitii)							
-R11 Dk.	215 /* DIER CODE END EXTIG INCH 1 */								
R12 Gr.	215								
1111 (SPS) 04-	216 (3/14								
111170 6	217 · Burief This function handles USARTI global :	Ltherruge.							
* 015 D.	215 word USART1 IBGRandler (void)								
R Sutare	220日1								
8 Internal	221 /* USER CODE BEGIN UNARTLINGS 0 */								
Mode Th.	225 /* USER CODE END USARTS TRUN D */								
- Sack NSF	224 MAL_UABT_IBQMandler(shuartl);								
Sales 1875	225 /* THEN CODE BELIN UNANTL HQB 1 */								
340 0.0. # FPG	227 /* UND CODE END UNARTS INQS 5 */					14			
COMPA	228 1								
	230								
E Priset I Regitters	4					*			
Densate		Watch 3							
Running with Code Si	se limit: 31%	* Nume	Value	Type		Waters 1			
Load "NCU_Labs/ NCU_	1929.821"	In the Instantial	(su200000000 transmit)	sched)		Name	Value	Туре	
		* 回	0.68 W	ucher			0.601	uchar	
		e (11)	1045 V	uchar		• 111	0.60 7	uchar	
		• III	BARC T	ucher		(A)	0+6F 'e'	uchar	
		- • (3)	INC T	sicher		· 151	0.00	ocher	
		9 [8]	DMF W	uthar		= " receive	0x2000006 receive[] *-	uchar(6)	
		DI B	UADD	UCTUR		2 (0)	Cu68 W	uchar	
		A MI	articonne second)	ocharge)		• 01	0.65 %	othar	
		0 (11	8,00	luther	-	· 171	0.6C T	orhar	
		· · · · (2)	0,00	uther	-	• m	0.66 7	other	
		· 01	0.00	activer .		¥ (41	DATE W	in the lat	
		* (A)	0.00	uther.		. (51	0.00 h	terihar	
\$		· · · · · · · · · · · · · · · · · · ·	0.00	sicher		a Foter emperiment	IS	5211D-	
2		- fate comments	- [19]	1000		Cutter erbrittenter.		-	
ADDING STREEDINGSIE	ScenkEnable Breakfill Breaklist Breaklet BreakAccess	\$2710 Arth - Tology MAN	h1						
		ST-Livia Debug	ger (3; 5,00015	730 MPI 12342 C-4	T DIP NU	M SCRE GYR R/W			

Figure 13

6. You will make the changes within the EXTI0_IRQHandler function. Define tansmit1, transmit2, receive and i variables. Assign "hello" to transmit1 and "world" to transmit2 in char type (Figure 14). Each time the button is pushed, the value of the i variable increases by 1. If it is an even number, transmit1 variable, if odd, transmit 2 variable from USART2 (Figure 15). Let USART2 also receive this data from USART1. Follow the change of variables using the Debug menu (Figure 16).

```
42
43 /* Private variables ------
44 /* USER CODE BEGIN PV */
45 int i;
46 char transmit1[]="hello";
47 char transmit2[]="world";
48 char receive [6];
49 /* USER CODE END PV */
50
```







Figure 16