

Nirbhay Sharma (B19CSE114)

DSL - Lab - 8

Que-1

```
#include "stm32f4xx.h"
main(void)
{
    int x,y;
    x=0x1B;
    RCC->AHB1ENR |= 0x0E; // Enables Clock

    if (x >= 0 && x < 100) {
        // RCC->AHB1ENR = 0x02;
        GPIOB->MODER = 0X15555;
        GPIOB->ODR = 0X000;
        y = 2 * x + 2;
        GPIOB->ODR = y;
    }

    if (x >= 100 && x < 200){
        // RCC->AHB1ENR = 0x04;
        GPIOC->MODER = 0X55555555;
        GPIOC->ODR = 0X0000;
        y = x * x - 2 * x;
        GPIOC->ODR = y;
    }

    if (x >= 200 && x < 256){
        // RCC->AHB1ENR = 0x08;
        GPIOD->MODER = 0X5555;
        GPIOD->ODR = 0X00;
        y = x - 150;
        GPIOD->ODR = y;
    }

}
```

code-explanation

- first we are enabling the clock
- then we are applying conditions on x and based on the three conditions we are assigning values to ports
- ($0 \leq x < 100$) first we are setting moder value to (0001 0101 0101 0101 0101) which is 0x15555 the last 1 is due to overflow since $2x$ can also be 9 bit number if x is 8 bit number. and then the calculated value is send to ODR

- ($100 \leq x < 200$) here the value is $x^2 - 2x$ which can be maximum 16 bits so all the moders of port B is set to output mode
- ($200 \leq x < 256$) here the value is $x - 150$ which is confined to 8 bits only so to support that only 0x5555 is sufficient and the output is send to odr
- setting the clock to RCC->AHB1ENR=0x02 for port B, 0x04 for port C, 0x08 for port D.

build-output

The screenshot shows the uVision IDE with the file 'toggled.c' open. The code defines a main function that takes an integer 'x' and calculates a value 'y' based on three ranges of 'x'. It then configures the GPIO pins for ports B, C, and D to output the value 'y'. The Build Output window at the bottom shows the compilation process, including the use of the VS.06 compiler and the generation of the 'toggled.elf' file.

```

1 #include "stm32f4xx.h"
2 main(void)
3 {
4     int x,y;
5     x=0x1B;
6     RCC->AHB1ENR |=1; // Enables Clock
7
8     if (x >= 0 && x < 100) {
9         GPIOB->MODER = 0x15555;
10        GPIOB->ODR = 0x000;
11        y = 2 * x + 2;
12        GPIOB->ODR = y;
13    }
14
15    if (x >= 100 && x < 200){
16        GPIOC->MODER = 0x55555555;
17        GPIOC->ODR = 0x0000;
18        y = x * x - 2 * x;
19        GPIOC->ODR = y;
20    }
21
22    if (x >= 200 && x < 256){
23        GPIOD->MODER = 0x5555;
24        GPIOD->ODR = 0x00;
25        v = x - 150;
  
```

Build Output

```

*** Using Compiler 'VS.06 update 6 (build 750)', folder: 'C:\Keil_v5\ARM\ARMCC\Bin'
Build target 'Target 1'
compiling toggleds.c...
toggleds.c(30): warning: #1-D: last line of file ends without a newline
}
toggleds.c: 1 warning, 0 errors
linking...
Program Size: Code=572 RO-data=408 RW-data=0 ZI-data=1632
".\Objects\toggleds.axf" - 0 Error(s), 1 Warning(s).
Build Time Elapsed: 00:00:01
  
```

Que-2

Task-2

```

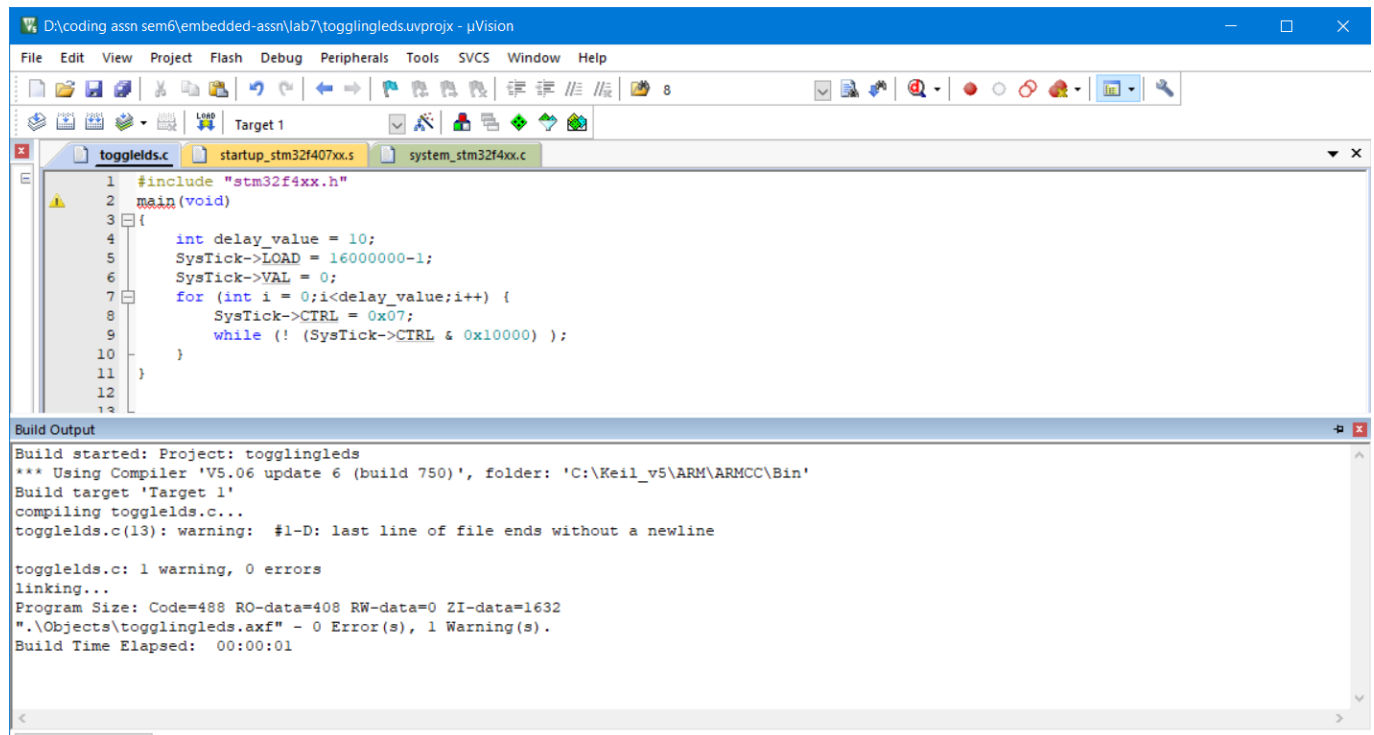
#include "stm32f4xx.h"
main(void)
{
    int delay_value = 10;
    SysTick->LOAD = 16000000-1;
    SysTick->VAL = 0;
    for (int i = 0;i<delay_value;i++) {
        SysTick->CTRL = 0x07;
        while (! (SysTick->CTRL & 0x10000) );
    }
}
  
```

code-explanation

- first find the load value, it will take which is $(\text{Time} * \text{Freq} - 1) = 16 \times 10^6 \times 1 - 1 = 15999999$
- the above value is for 1s delay so to make it 10 seconds we need to run the loop 10 times
- and hence for loop is there

- we are checking the condition for it to generate an interrupt once it completes the counting and once it completes it breaks the while loop and run again.

build-output



The screenshot shows the Keil uVision IDE with the project 'togglingleds' open. The source file 'togglingleds.c' is displayed, showing the main function with a delay loop. The Build Output window at the bottom shows the compilation process, including a warning about a missing newline at the end of the file.

```

1 #include "stm32f4xx.h"
2 main(void)
3 {
4     int delay_value = 10;
5     SysTick->LOAD = 16000000-1;
6     SysTick->VAL = 0;
7     for (int i = 0; i < delay_value; i++) {
8         SysTick->CTRL = 0x07;
9         while (! (SysTick->CTRL & 0x10000) );
10    }
11 }
12
Build Output
Build started: Project: togglingleds
*** Using Compiler 'V5.06 update 6 (build 750)', folder: 'C:\Keil_v5\ARM\ARMCC\Bin'
Build target 'Target 1'
compiling togglingleds.c...
togglingleds.c(13): warning: #1-D: last line of file ends without a newline

togglingleds.c: 1 warning, 0 errors
linking...
Program Size: Code=488 RO-data=408 RW-data=0 ZI-data=1632
".\Objects\togglingleds.axf" - 0 Error(s), 1 Warning(s).
Build Time Elapsed: 00:00:01
  
```

Task-3

```

#include "stm32f4xx.h"
main(void)
{
    RCC->AHB1ENR = 0X01;
    GPIOA->MODER = 0X5000;
    GPIOA->ODR = 0XC0;
    SysTick->LOAD = 16000000-1;
    SysTick->VAL = 0;
    SysTick->CTRL = 0x07;
    while (1){
        if (SysTick->CTRL & 0x10000){
            GPIOA->ODR ^= 0xc0;
        }
    }
}
  
```

code-explanation

- here we need to toggle PA6 and PA7 so again we are setting RCC->AHB1ENR to 1 and setting moder = 0x5000 and odr = 0xc0 for the purpose of PA6 and PA7
- and setting the load according to 16MHz freq and 1s delay

- and in while loop we are waiting systick->ctrl register to have countflag 1 and once it is 1 we are toggling the led

build-output

The screenshot shows the Keil uVision IDE with the project 'togglingleds' open. The main file 'togglingleds.c' is displayed, showing the initialization of the RCC, GPIOA, and SysTick registers, followed by a while loop that toggles the LED state when the SysTick counter reaches 0x10000.

```

1 #include "stm32f4xx.h"
2 main(void)
3 {
4     RCC->AHB1ENR = 0x01;
5     GPIOA->MODER = 0x5000;
6     GPIOA->ODR = 0xc0;
7     SysTick->LOAD = 16000000-1;
8     SysTick->VAL = 0;
9     SysTick->CTRL = 0x07;
10    while (1){
11        if (SysTick->CTRL & 0x10000){
12            GPIOA->ODR ^= 0xc0;
13        }
14    }
15 }
16
17
18

```

The Build Output window shows the following information:

```

Build started: Project: togglingleds
*** Using Compiler 'V5.06 update 6 (build 750)', folder: 'C:\Keil_v5\ARM\ARMCC\Bin'
Build target 'Target 1'
compiling togglingleds.c...
togglingleds.c(18): warning: #1-D: last line of file ends without a newline

togglingleds.c: 1 warning, 0 errors
linking...
Program Size: Code=508 RO-data=408 RW-data=0 ZI-data=1632
".\Objects\togglingleds.axf" - 0 Error(s), 1 Warning(s).
Build Time Elapsed: 00:00:01

```

Que-3

Task-4

```

#include "stm32f4xx.h"
main(void)
{
    RCC->AHB1ENR = 0x01;
    GPIOA->MODER = 0x04;
    GPIOA->ODR = 0xc0;

    RCC->APB1ENR = 0x01;
    TIM2->PSC = 1599;
    TIM2->ARR = 999;
    TIM2->CNT = 0;
    TIM2->CR1 = 0x01;

    while (1) {
        while (!(TIM2->SR & 1)) {
            TIM2->SR = 0;
            GPIOA->ODR ^= 0xc0;
        }
    }
}

```

```
}

```

code-explanation

- first enable the GPIOA clock and then set GPIOA->MODER in output mode for pinA1 and glow the led
- then enable tim2 using RCC->APB1ENR and set prescaler and arr value and set count register to 0 and enable the counter using TIM2->CR1
- and now until a request is pending toggle the led and it will generate a square wave since it will be 1 for sometime and 0 for other time

build-output

The screenshot shows the Keil uVision IDE with the project 'togglingleds' open. The main window displays the code for 'togglingleds.c', which includes the 'stm32f4xx.h' header and implements a main function. The code enables the GPIOA clock (RCC->AHB1ENR), sets the pin mode (GPIOA->MODER), and enables the TIM2 clock (RCC->APB1ENR). It then sets the TIM2 prescaler (TIM2->PSC), arr (TIM2->ARR), and count register (TIM2->CNT). A while loop toggles the LED (GPIOA->ODR) based on the TIM2 status register (TIM2->SR).

The Build Output window shows the following messages:

```
Build started: Project: togglingleds
*** Using Compiler 'V5.06 update 6 (build 750)', folder: 'C:\Keil_v5\ARM\ARMCC\Bin'
Build target 'Target 1'
compiling togglingleds.c...
togglingleds.c(24): warning: #1-D: last line of file ends without a newline

togglingleds.c: 1 warning, 0 errors
linking...
Program Size: Code=528 RO-data=408 RW-data=0 ZI-data=1632
".\Objects\togglingleds.axf" - 0 Error(s), 1 Warning(s).
Build Time Elapsed: 00:00:04
```

Que-4

```
#include "stm32f4xx.h"
main(void)
{
    RCC->APB2ENR |= 0X100;
    RCC->AHB1ENR |= 0X01;

    ADC1->CR2 |= 0X0402; // ENABLING 10TH AND 2ND BIT OF ADC1 REGISTER
    GPIOA->MODER |= 0X03; // ADC MODE

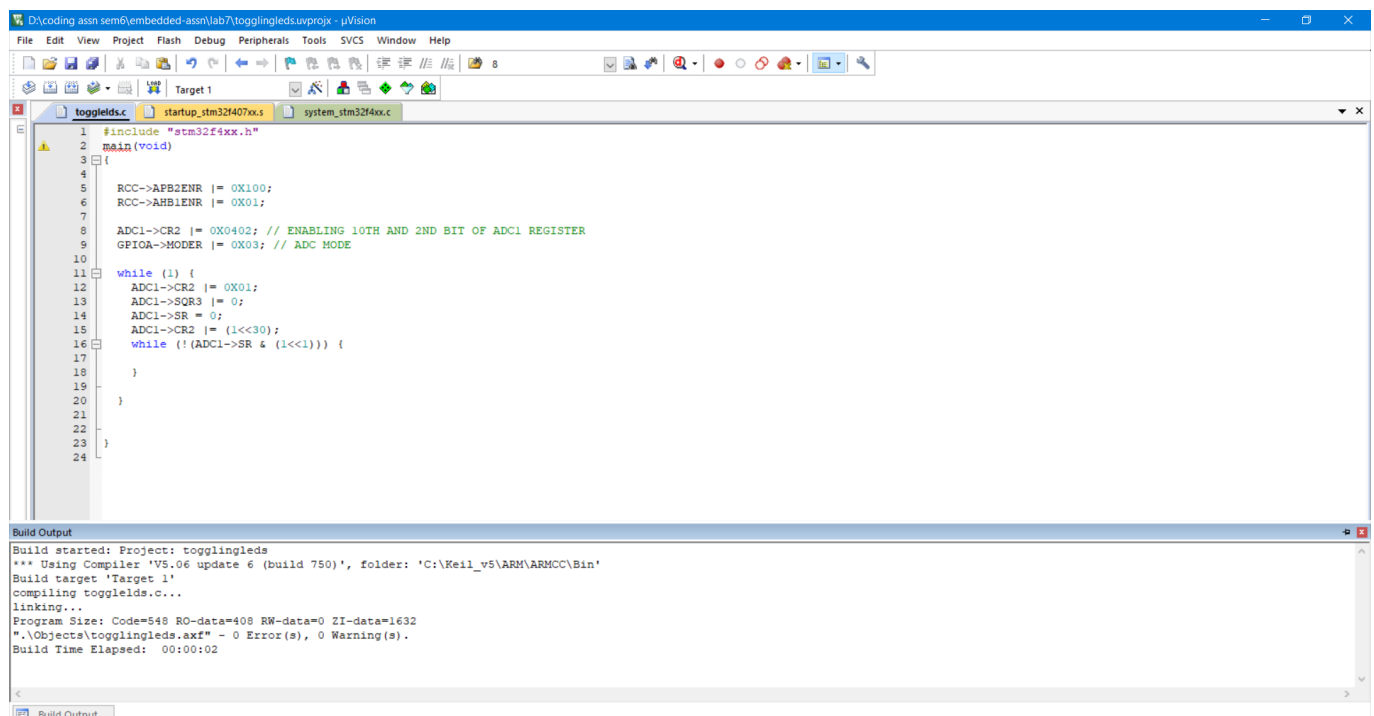
```

```
while (1) {  
    ADC1->CR2 |= 0X01;  
    ADC1->SQR3 |= 0;  
    ADC1->SR = 0;  
    ADC1->CR2 |= (1<<30);  
    while (!(ADC1->SR & (1<<1)));  
  
}
```

```
}
```

code-explanation

build-output



The screenshot shows a Visual Studio IDE window titled "D:\coding\asn\sem6\embedded-asm\lab7\togglingleds.uvprojx - μVision". The main editor displays the file "togglingleds.c" with the following code:

```
1 #include "stm32f4xx.h"  
2 main(void)  
3 {  
4  
5     RCC->APB2ENR |= 0X100;  
6     RCC->AHB1ENR |= 0X01;  
7  
8     ADC1->CR2 |= 0X0402; // ENABLING 10TH AND 2ND BIT OF ADC1 REGISTER  
9     GPIOA->MODER |= 0X03; // ADC MODE  
10  
11     while (1) {  
12         ADC1->CR2 |= 0X01;  
13         ADC1->SQR3 |= 0;  
14         ADC1->SR = 0;  
15         ADC1->CR2 |= (1<<30);  
16         while (!(ADC1->SR & (1<<1))) {  
17  
18         }  
19     }  
20 }  
21  
22 }  
23  
24 }
```

The "Build Output" window at the bottom shows the following text:

```
Build started: Project: togglingleds  
*** Using Compiler 'VS.06 update 6 (build 750)', folder: 'C:\Keil_v5\ARM\ARMCC\Bin'  
Build target 'Target 1'  
compiling togglingleds.c...  
linking...  
Program Size: Code=548 RO-data=408 RW-data=0 ZI-data=1632  
".\Objects\togglingleds.axf" - 0 Error(s), 0 Warning(s).  
Build Time Elapsed: 00:00:02
```