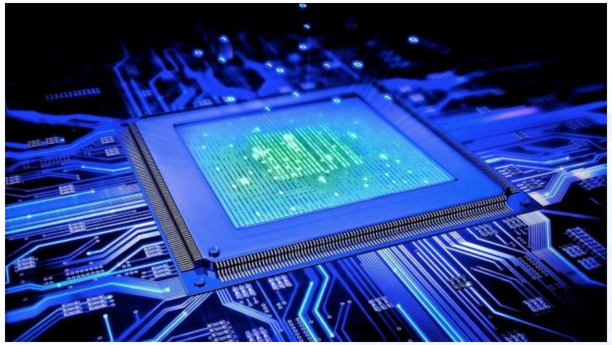
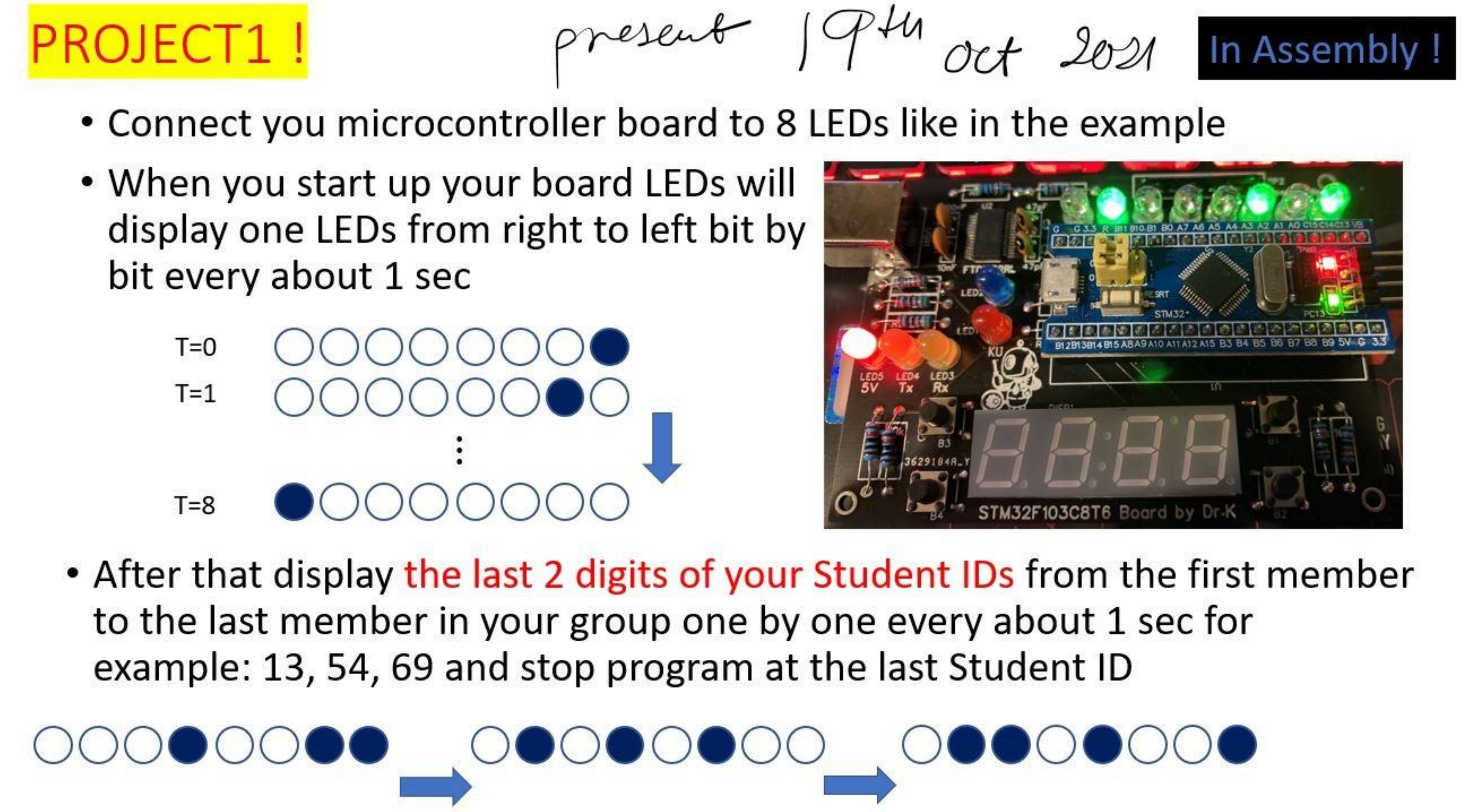
01205311 Microprocessor Project1

Jakkaphob Kongthanarith	6210554725
Areeya Yiampanomkhun	6210554709
Krit Jaithawin	6210554717



What We want



PROJECT1 ! present 19th oct 2021 In Assembly !

- Connect you microcontroller board to 8 LEDs like in the example
- When you start up your board LEDs will display one LEDs from right to left bit by bit every about 1 sec

T=0 ○○○○○○○●

T=1 ○○○○○○●○

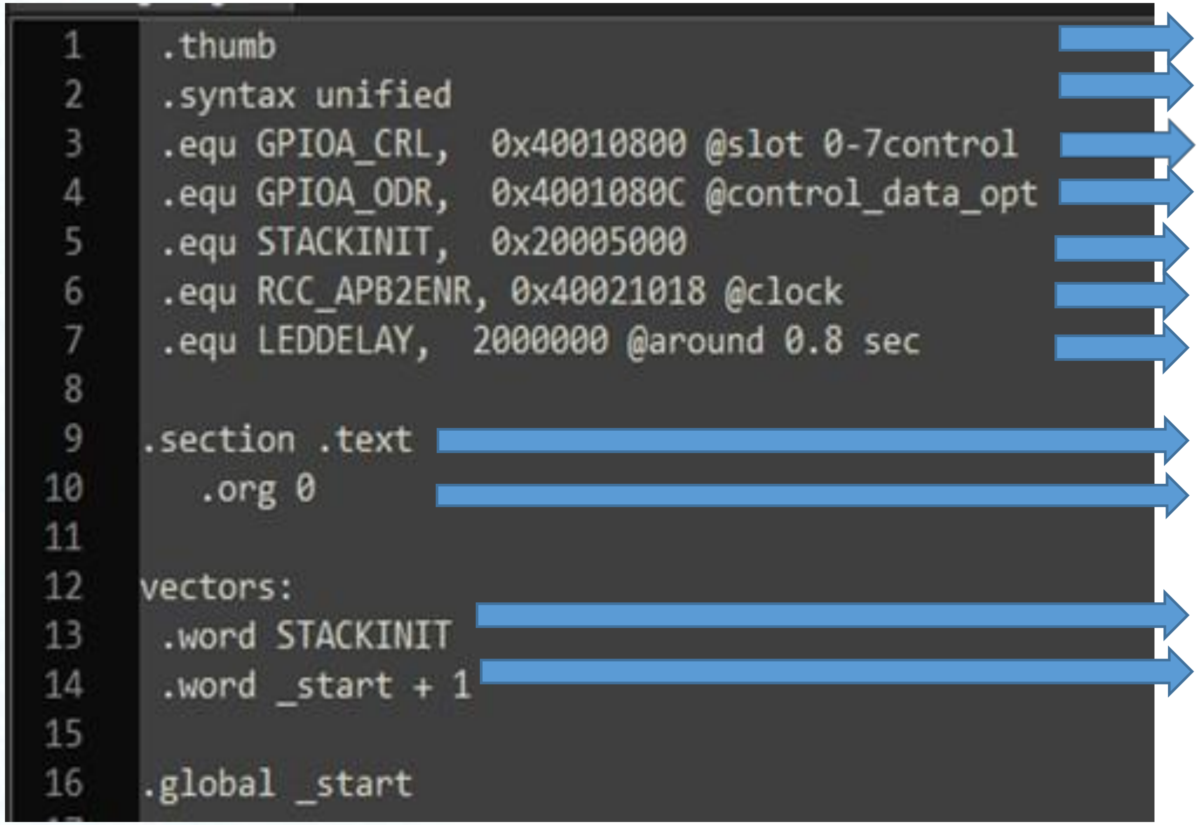
⋮

T=8 ●○○○○○○○

- After that display the last 2 digits of your Student IDs from the first member to the last member in your group one by one every about 1 sec for example: 13, 54, 69 and stop program at the last Student ID

○○○●○○●● → ○●○●○●○○ → ○●●○●○○●

Start coding in Notepad++

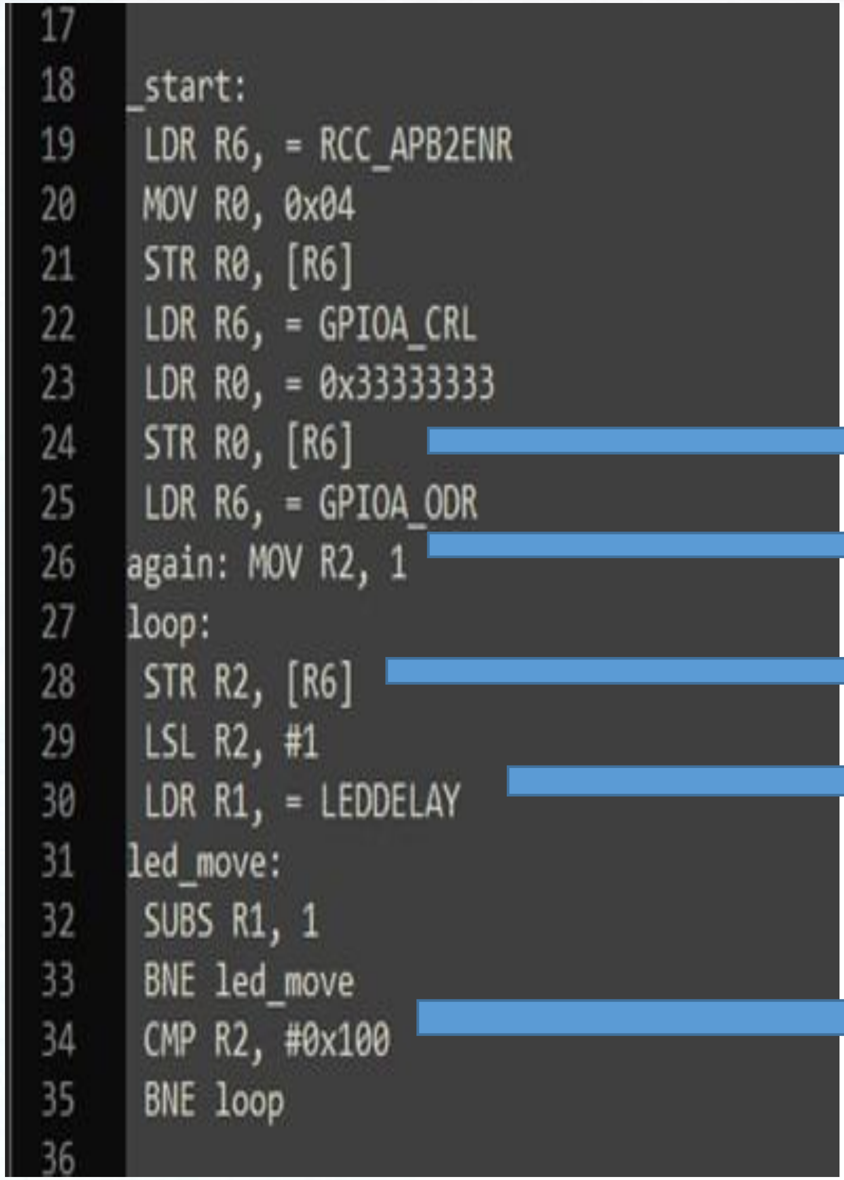


```
 .thumb
 .syntax unified
 .equ GPIOA_CRL,  0x40010800 @slot 0-7control
 .equ GPIOA_ODR,  0x4001080C @control_data_opt
 .equ STACKINIT,  0x20005000
 .equ RCC_APB2ENR, 0x40021018 @clock
 .equ LEDDELAY,  2000000 @around 0.8 sec

.section .text
   .org 0

vectors:
 .word STACKINIT
 .word _start + 1

.global _start
```

First step set Thumb Instruction
Assembler directive
To Control of each slot of LED
To Control the data output
Top of stack
For Feeding the clock
Set the delay

Text Section used for keeping the actual code
Set the assembler location counter

Stack pointer value when stack is empty
Reset vector



```

_start:
 LDR R6, = RCC_APB2ENR
 MOV R0, 0x04
 STR R0, [R6]
 LDR R6, = GPIOA_CRL
 LDR R0, = 0x33333333
 STR R0, [R6]
 LDR R6, = GPIOA_ODR
again: MOV R2, 1
loop:
 STR R2, [R6]
 LSL R2, #1
 LDR R1, = LEDDELAY
led_move:
 SUBS R1, 1
 BNE led_move
 CMP R2, #0x100
 BNE loop

```

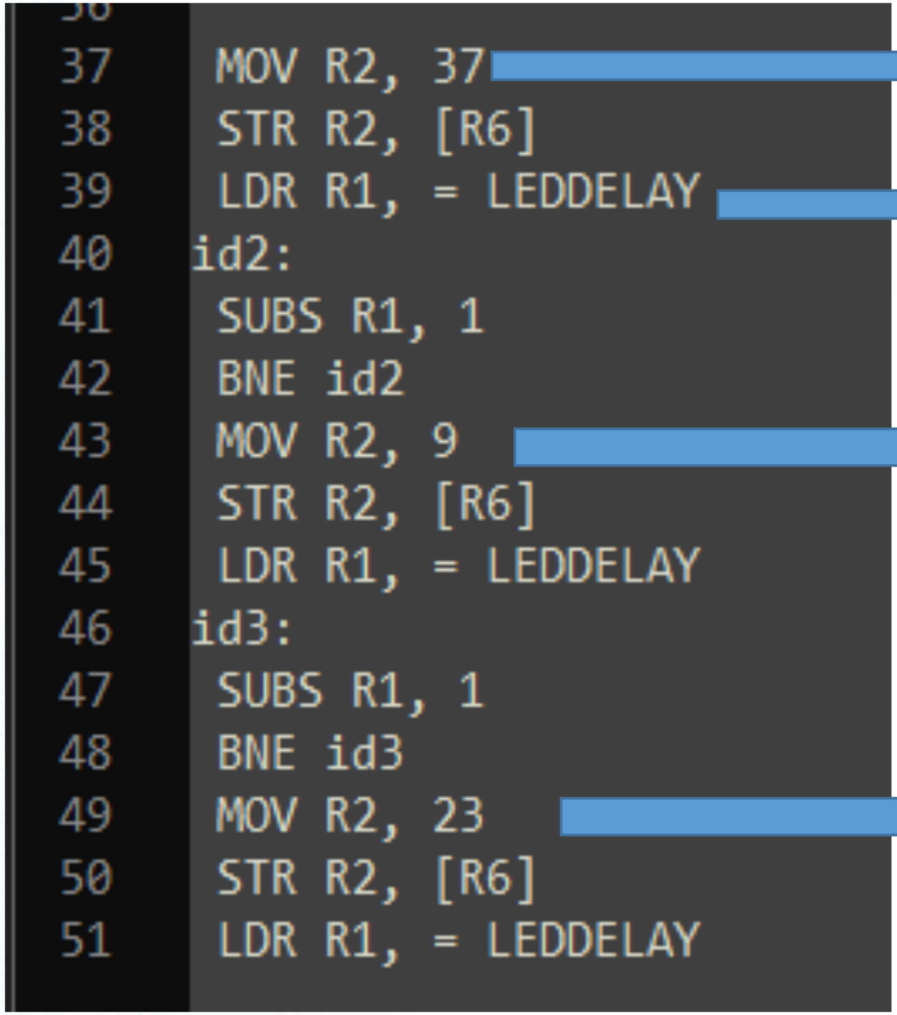
Overwrite new value of GPIOA_CRL is 0x33333333

Load constant into R6 by placing address of GPIO_ODR,R2 = 1

For keep value found in R2 to memory address that found in R6

Load constant into R1

That R2 has value more than 0xFF clear R2



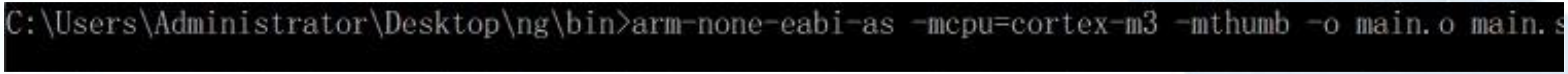
```
 MOV R2, 37
 STR R2, [R6]
 LDR R1, = LEDDELAY
id2:
 SUBS R1, 1
 BNE id2
 MOV R2, 9
 STR R2, [R6]
 LDR R1, = LEDDELAY
id3:
 SUBS R1, 1
 BNE id3
 MOV R2, 23
 STR R2, [R6]
 LDR R1, = LEDDELAY
```

25[0010][0101] = 32+4+1 = 37

When the LED are shown, it will present a delay,R1 value will rise to 2000000

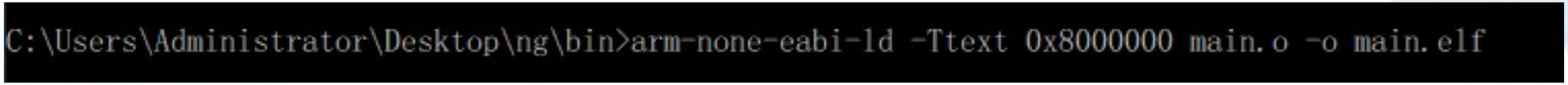
09[0000][1001] = 8+1 =9

17[0001][0111] = 16+4+2+1 = 23



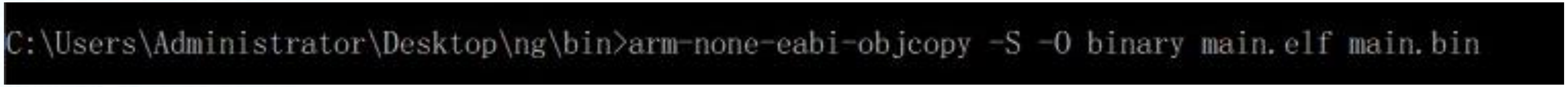
```
C:\Users\Administrator\Desktop\ng\bin>arm-none-eabi-as -mcpu=cortex-m3 -mthumb -o main.o main.s
```

1.Created .O file from .S file



```
C:\Users\Administrator\Desktop\ng\bin>arm-none-eabi-ld -Ttext 0x8000000 main.o -o main.elf
```

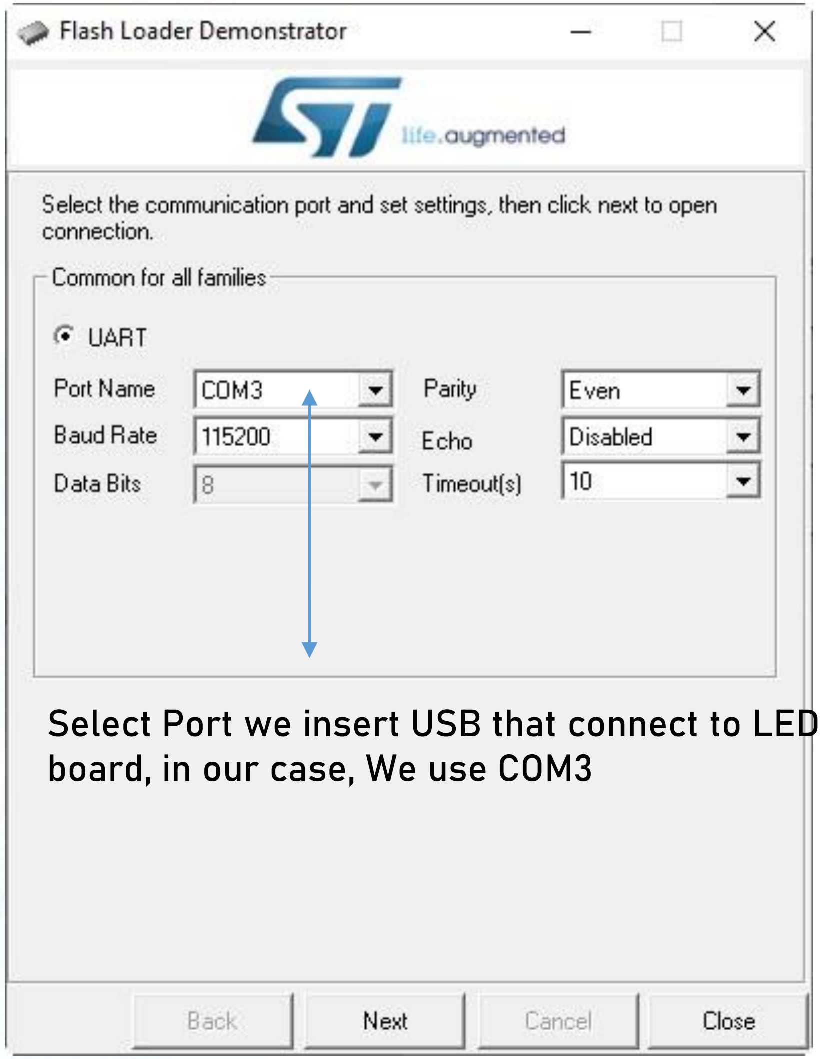
2.Created .elf file .O file



```
C:\Users\Administrator\Desktop\ng\bin>arm-none-eabi-objcopy -S -O binary main.elf main.bin
```

3.Created .bin file form .elf file

Import information to LED board



Flash Loader Demonstrator

Select the communication port and set settings, then click next to open connection.

Common for all families

● UART

Port Name: COM3

Baud Rate: 115200

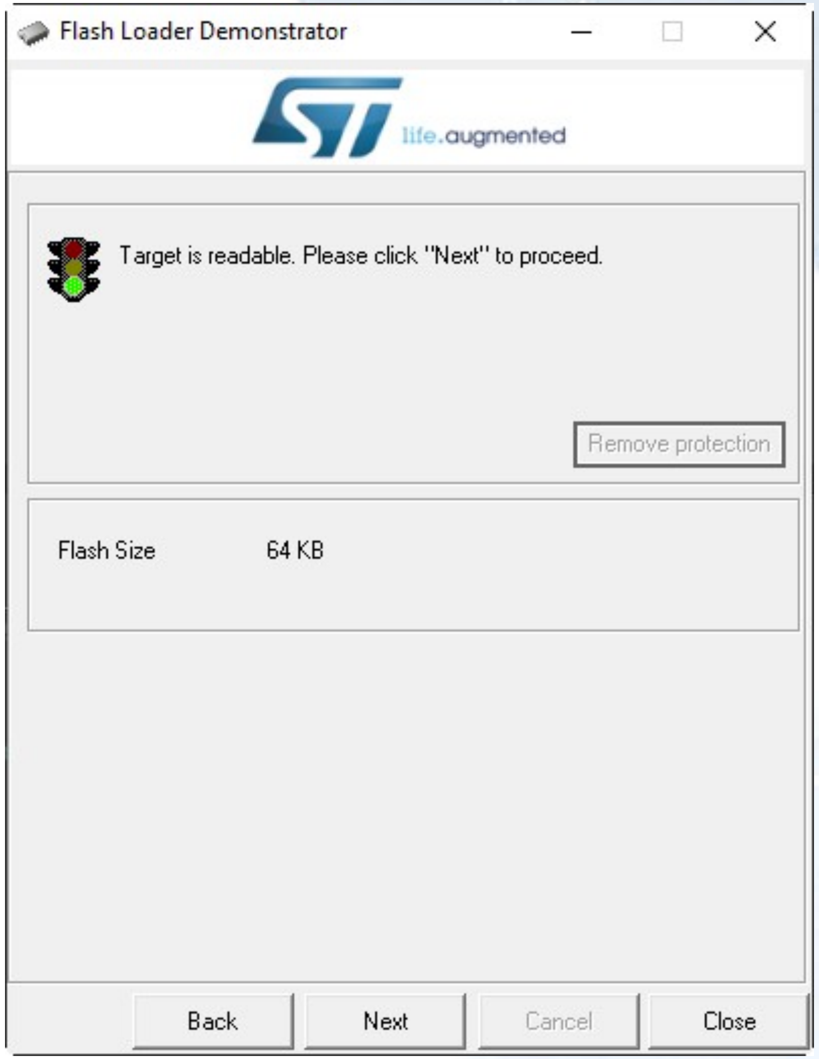
Data Bits: 8

Parity: Even

Echo: Disabled

Timeout(s): 10

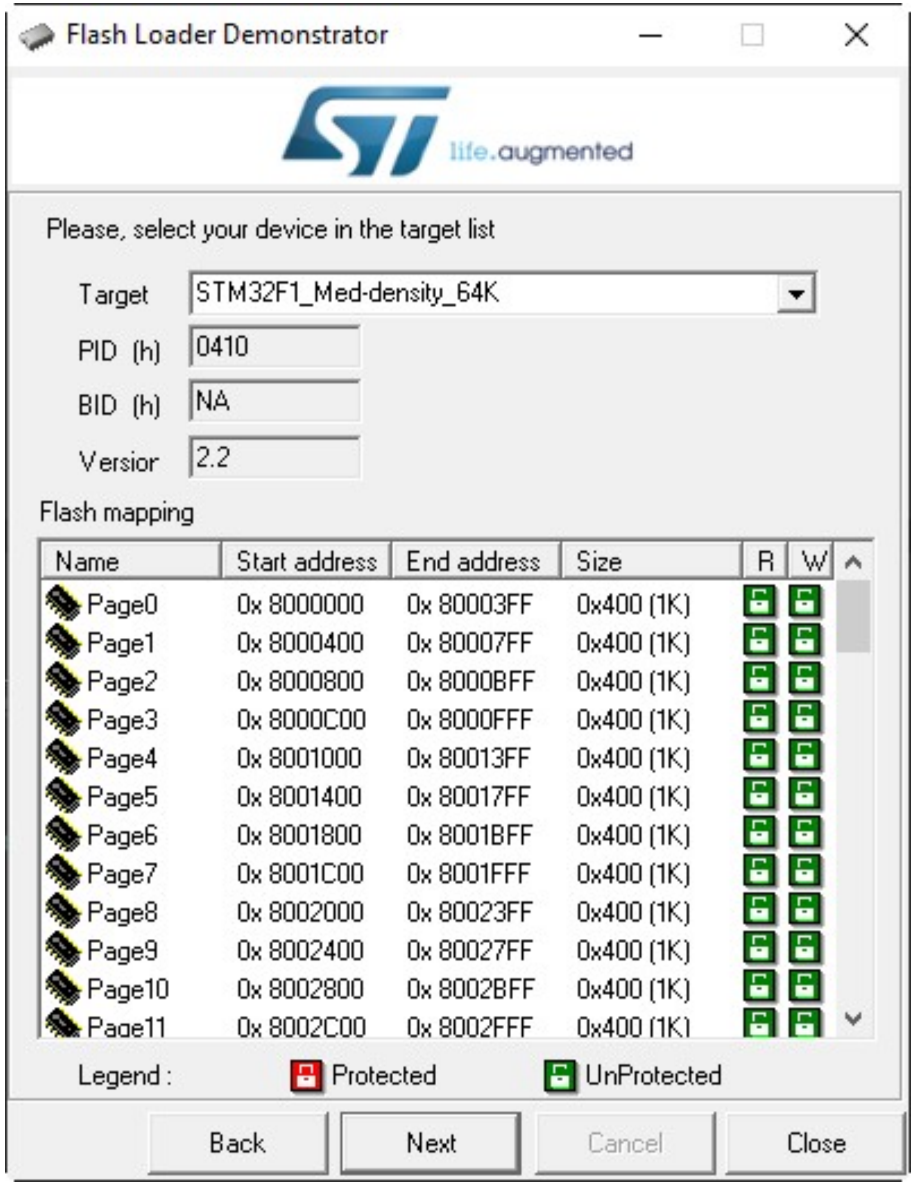
Select Port we insert USB that connect to LED board, in our case, We use COM3



Flash Loader Demonstrator

Target is readable. Please click "Next" to proceed.

Flash Size 64 KB



Please, select your device in the target list

Target: STM32F1_Med-density_64K

PID (h): 0410

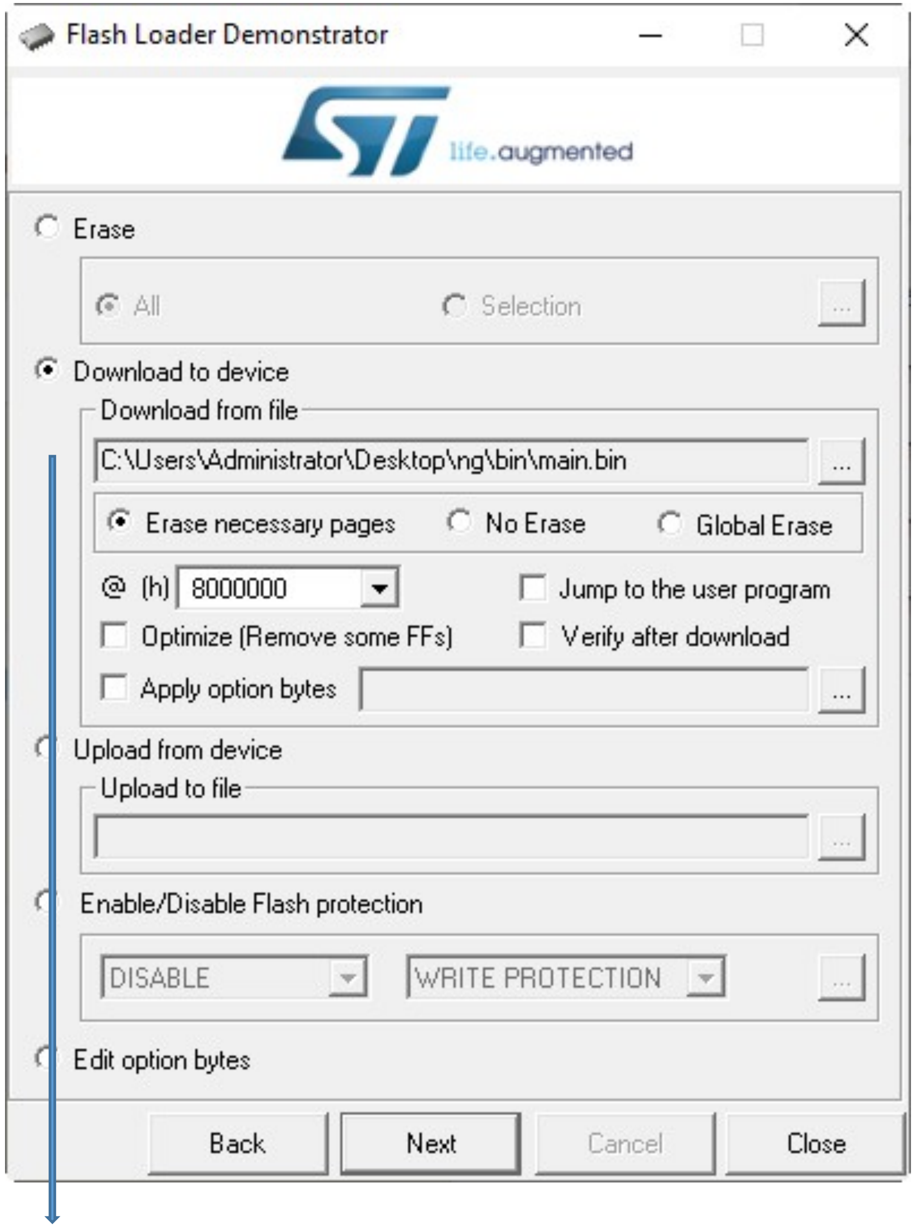
BID (h): NA

Version: 2.2

Flash mapping

Name	Start address	End address	Size	R	W
Page0	0x 8000000	0x 80003FF	0x400 (1K)	🟩	🟩
Page1	0x 8000400	0x 80007FF	0x400 (1K)	🟩	🟩
Page2	0x 8000800	0x 8000BFF	0x400 (1K)	🟩	🟩
Page3	0x 8000C00	0x 8000FFF	0x400 (1K)	🟩	🟩
Page4	0x 8001000	0x 80013FF	0x400 (1K)	🟩	🟩
Page5	0x 8001400	0x 80017FF	0x400 (1K)	🟩	🟩
Page6	0x 8001800	0x 8001BFF	0x400 (1K)	🟩	🟩
Page7	0x 8001C00	0x 8001FFF	0x400 (1K)	🟩	🟩
Page8	0x 8002000	0x 80023FF	0x400 (1K)	🟩	🟩
Page9	0x 8002400	0x 80027FF	0x400 (1K)	🟩	🟩
Page10	0x 8002800	0x 8002BFF	0x400 (1K)	🟩	🟩
Page11	0x 8002C00	0x 8002FFF	0x400 (1K)	🟩	🟩

Legend : 🟥 Protected 🟩 UnProtected



○ Erase

● All ○ Selection

● Download to device

Download from file: C:\Users\Administrator\Desktop\ng\bin\main.bin

● Erase necessary pages ○ No Erase ○ Global Erase

@ (h) 8000000 □ Jump to the user program

□ Optimize (Remove some FFs) □ Verify after download

□ Apply option bytes

○ Upload from device

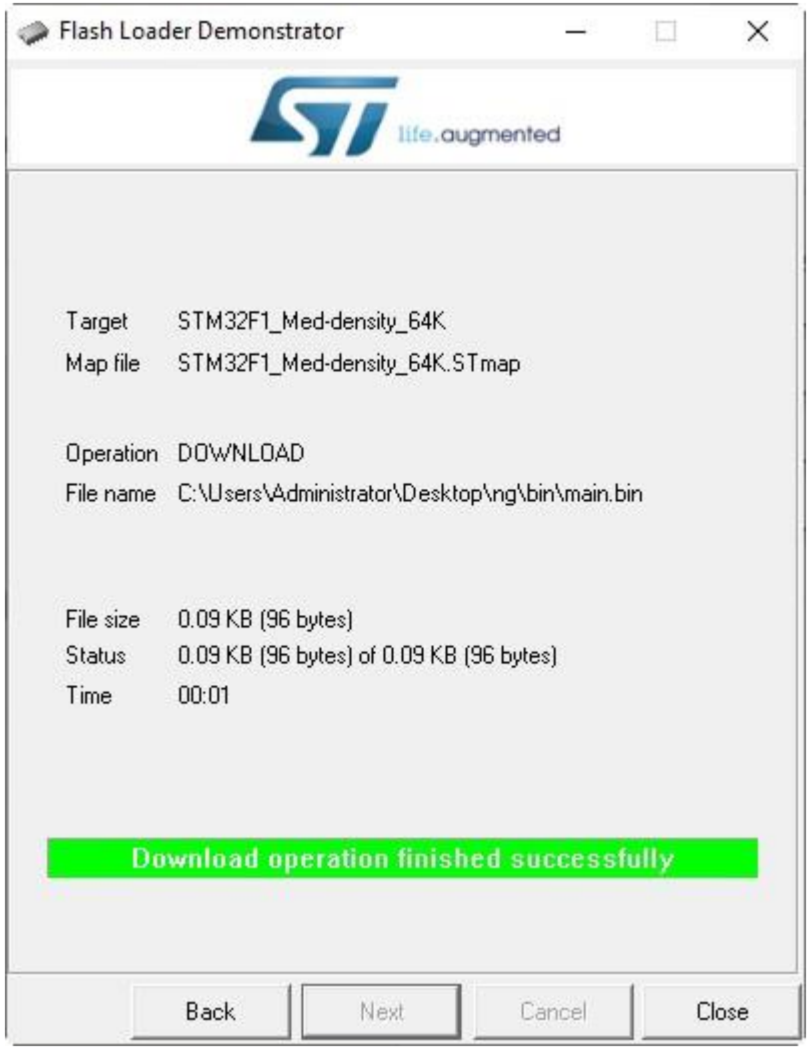
Upload to file

○ Enable/Disable Flash protection

DISABLE WRITE PROTECTION

○ Edit option bytes

Select file .bin that have been created from .S file



Target STM32F1_Med-density_64K

Map file STM32F1_Med-density_64K.STmap

Operation DOWNLOAD

File name C:\Users\Administrator\Desktop\ng\bin\main.bin

File size 0.09 KB (96 bytes)

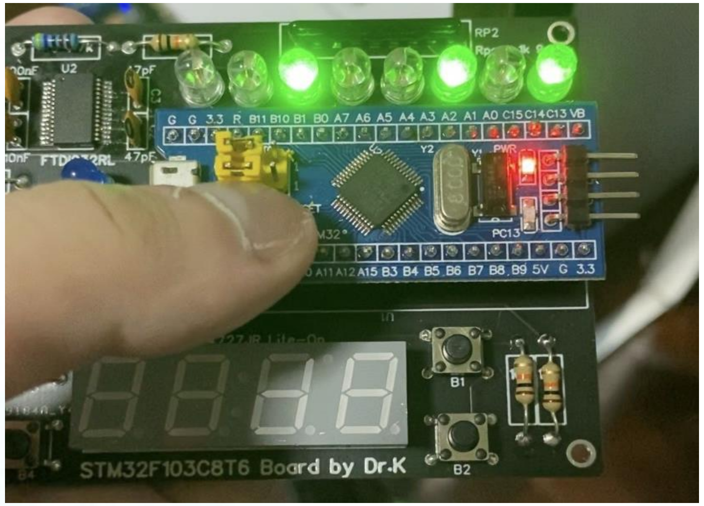
Status 0.09 KB (96 bytes) of 0.09 KB (96 bytes)

Time 00:01

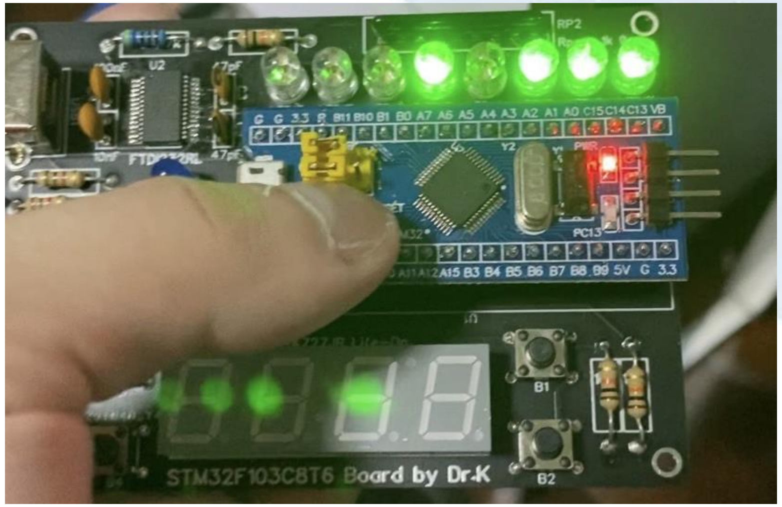
Download operation finished successfully

If the LED board are connected, Boot 0 is switched to 1 and press reset button on LED board, everything should proceed smoothly as picture.

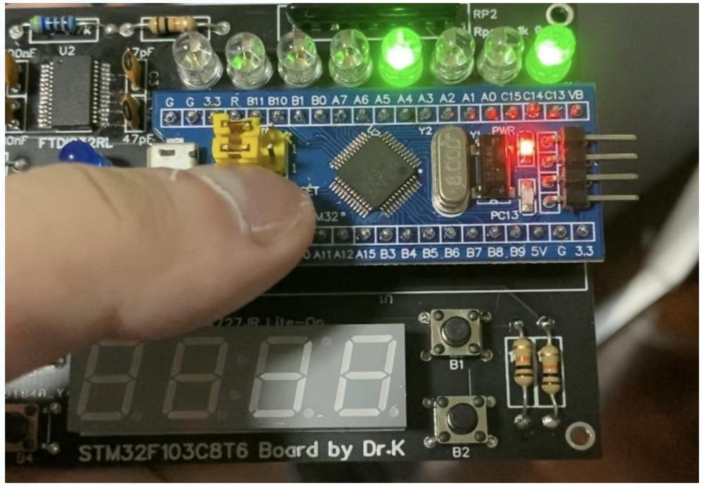
Result



25[0010][0101]= 32+4+1 = 37

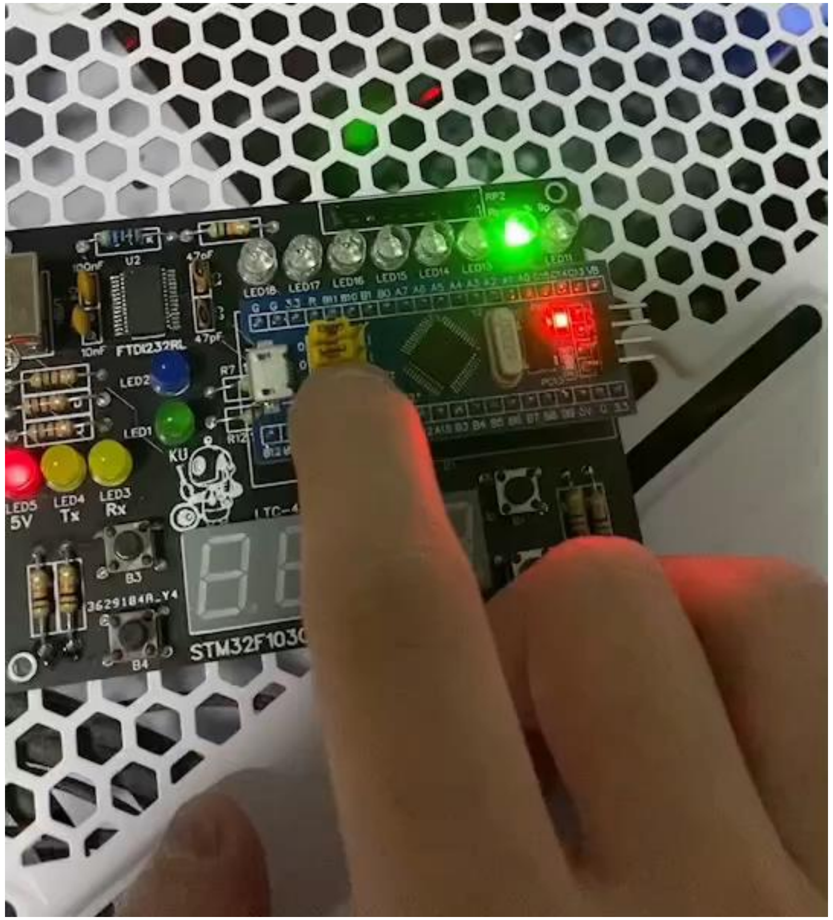


17[0001][0111] = 16+4+2+1 =23



09[0000][1001]= 8+1 = 9

Video


LED18
LED17
LED16
LED15
LED14
LED13
LED11
FTDI232RL
LED2
LED1
LED5
5V
LED4
Tx
LED3
Rx
B3
B4
3629184A_Y4
STM32F103C

Thank you all for coming here today.