ENG. MOHAMED HAMDY

MASTERING EMBEDDED SYSTEM ONLINE DIPLOMA
WWW.LEARN-IN-DEPTH.COM

SECOND TERM (UNIT6LESSON3) – SYSTEM CLOCK CONFIGURATION ON STM32F103C

MY PROFILE:
HTTPS://WWW.LEARN-IN-DEPTH-STORE.COM/PROFILE/MOHAMEDHAMDY-M-H-M-A/PROFILE

GITHUB REPO:
HTTPS://GITHUB.COM/MOHAMED-HAMDY-MA/MASTERING_EMBEDDED_SYSTEM.GIT

Figure 8. Clock tree

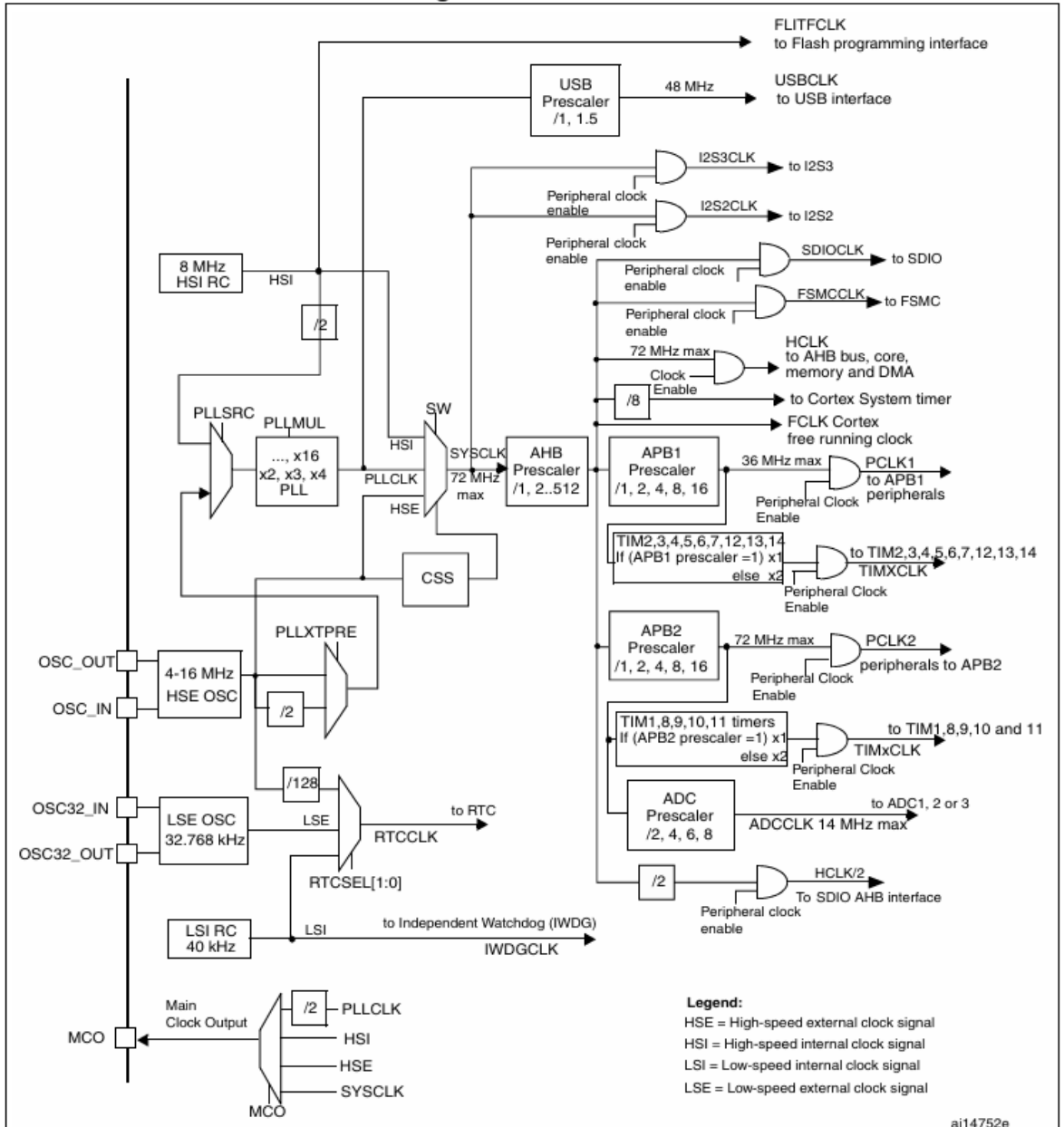


Table 18. RCC register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	RCC_CR	Reserved						PLL RDY	PLL ON	Reserved				CSSON	HSEBYP	HSERDY	HSEON	HSICAL[7:0]								HSITRIM[4:0]					Reserved	HSIRDY	HSION
	Reset value							0	0					0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0		1	1
0x04	RCC_CFGR	Reserved			MCO [2:0]					Reserved	USBPRE	PLLMUL [3:0]				PLLXTPRE	PLLSRC	ADC PRE [1:0]		PPRE2 [2:0]			PPRE1 [2:0]			HPRE[3:0]				SWS [1:0]		SW [1:0]	
	Reset value				0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	RCC_CIR	Reserved								CSSC	Reserved			PLLRDYC	HSERDYC	HSIRDYC	LSERDYC	LSIRDYC	Reserved			PLLRDYIE	HSERDYIE	HSIRDYIE	LSERDYIE	LSIRDYIE	CSSF	Reserved		PLLRDYF	HSERDYF	HSIRDYF	LSERDYF	LSIRDYF
	Reset value									0				0	0	0	0	0				0	0	0	0	0	0			0	0	0	0	0
0x0C	RCC_APB2RSTR	Reserved										TIM11RST	TIM10RST	TIM9RST	Reserved			ADC3RST	USART1RST	TIM8RST	SPI1RST	TIM1RST	ADC2RST	ADC1RST	IOPGRST	IOPFRST	IOPERST	IOPDRST	IOPCRST	IOPBRST	IOPARST	Reserved	AFIORST
	Reset value											0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0		0
0x010	RCC_APB1RSTR	Reserved			DACRST	PWRRST	BKPRST	Reserved	CANRST	Reserved	USBRST	I2C2RST	I2C1RST	UART5RST	UART4RST	USART3RST	USART2RST	Reserved	SPI3RST	SPI2RST	Reserved		WWDGRST	Reserved	TIM14RST	TIM13RST	TIM12RST	TM7RST	TM6RST	TM5RST	TIM4RST	TIM3RST	TIM2RST
	Reset value				0	0	0		0		0	0	0	0	0	0	0		0	0			0		0	0	0	0	0	0	0	0	0
0x14	RCC_AHBENR	Reserved																						SDIOEN	Reserved	FSMCEN	Reserved	CRCEN	Reserved	FLITFEN	Reserved	SRAMEN	DMA2EN	DMA1EN
	Reset value																							0		0		0		1		1	0	0
0x18	RCC_APB2ENR	Reserved										TIM11 EN	TIM10 EN	TIM9 EN	Reserved			ADC3EN	USART1EN	TIM8EN	SPI1EN	TIM1EN	ADC2EN	ADC1EN	IOPGEN	IOPFEN	IOPEEN	IOPDEN	IOPCEN	IOPBEN	IOPAEN	Reserved	AFIOEN
	Reset value											0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0		0
0x1C	RCC_APB1ENR	Reserved			DACEN	PWREN	BKPEN	Reserved	CANEN	Reserved	USBEN	I2C2EN	I2C1EN	UART5EN	UART4EN	USART3EN	USART2EN	Reserved	SPI3EN	SPI2EN	Reserved		WWDGEN	Reserved	TIM14EN	TIM13EN	TIM12EN	TIM7EN	TIM6EN	TIM5EN	TIM4EN	TIM3EN	TIM2EN
	Reset value				0	0	0		0		0	0	0	0	0	0	0		0	0			0		0	0	0	0	0	0	0	0	0	0
0x20	RCC_BDCR	Reserved															BDRST	RTCEN	Reserved					RTC SEL [1:0]		Reserved							LSEBYP	LSERDY	LSEON
	Reset value																0	0						0	0								0	0	0

SCREENSHOTS

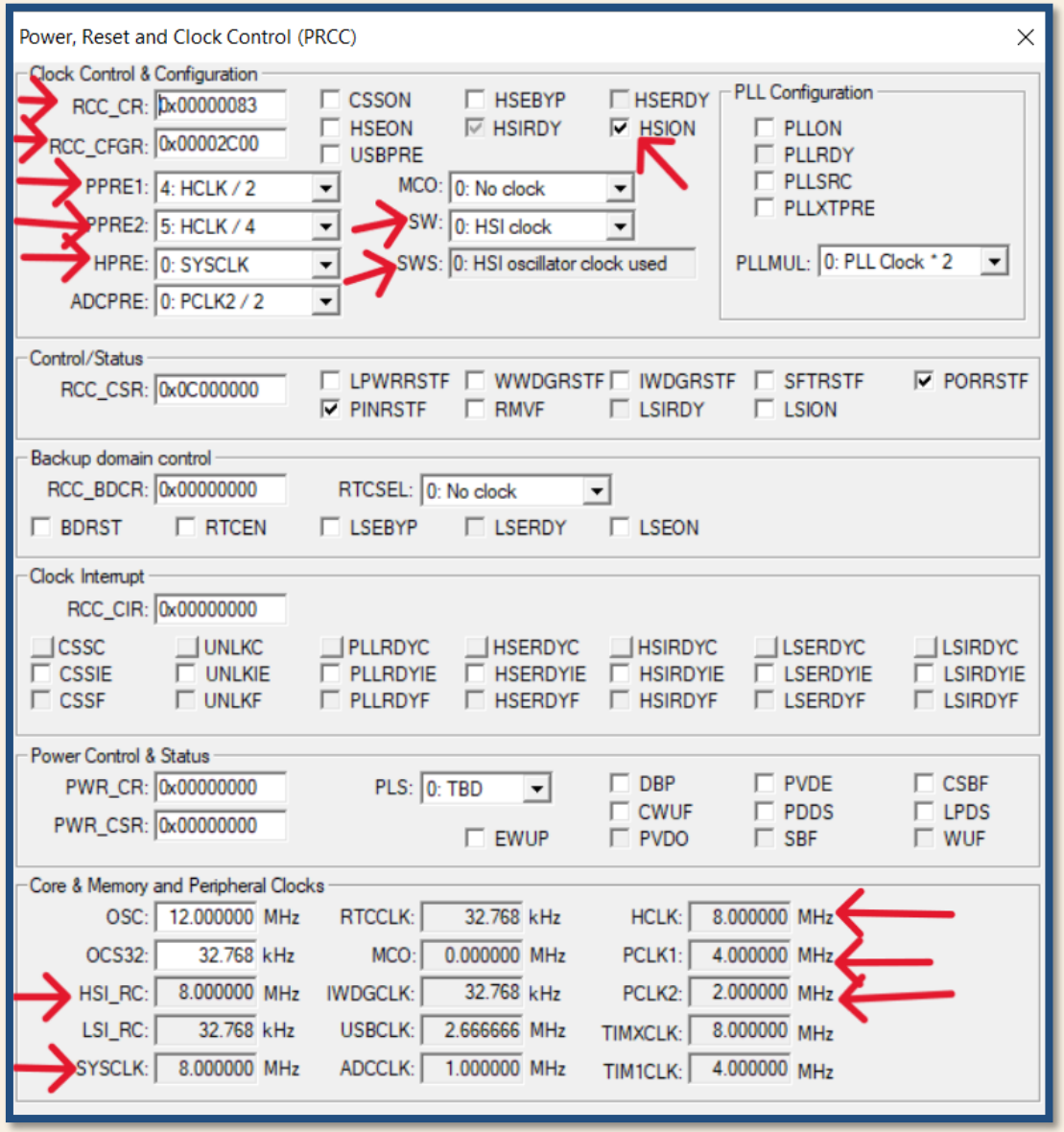
```c
#pragma pack(1)
typedef struct {
    uint32_t HSION:1;
    uint32_t HSIRDY:1;
    uint32_t :1;
    uint32_t HSITRIM:5;
    uint32_t HSICAL:8;
    uint32_t HSEON:1;
    uint32_t HSERDY:1;
    uint32_t HSEBYP:1;
    uint32_t CSSON:1;
    uint32_t :4;
    uint32_t PLLON:1;
    uint32_t PLLRDY:1;
    uint32_t :6;
}SRCC_CR_t;
```

```c
#pragma pack(1)
typedef struct {
    uint32_t SW:2;
    uint32_t SWS:2;
    uint32_t HPRE:4;
    uint32_t PPRE1:3;
    uint32_t PPRE2:3;
    uint32_t ADC:2;
    uint32_t PLLSRC:1;
    uint32_t PLLXTPRE:1;
    uint32_t PLLMUL:4;
    uint32_t USBPRE:1;
    uint32_t :1;
    uint32_t MCO:3;
    uint32_t :5;
}SRCC_CFGR_t;
```

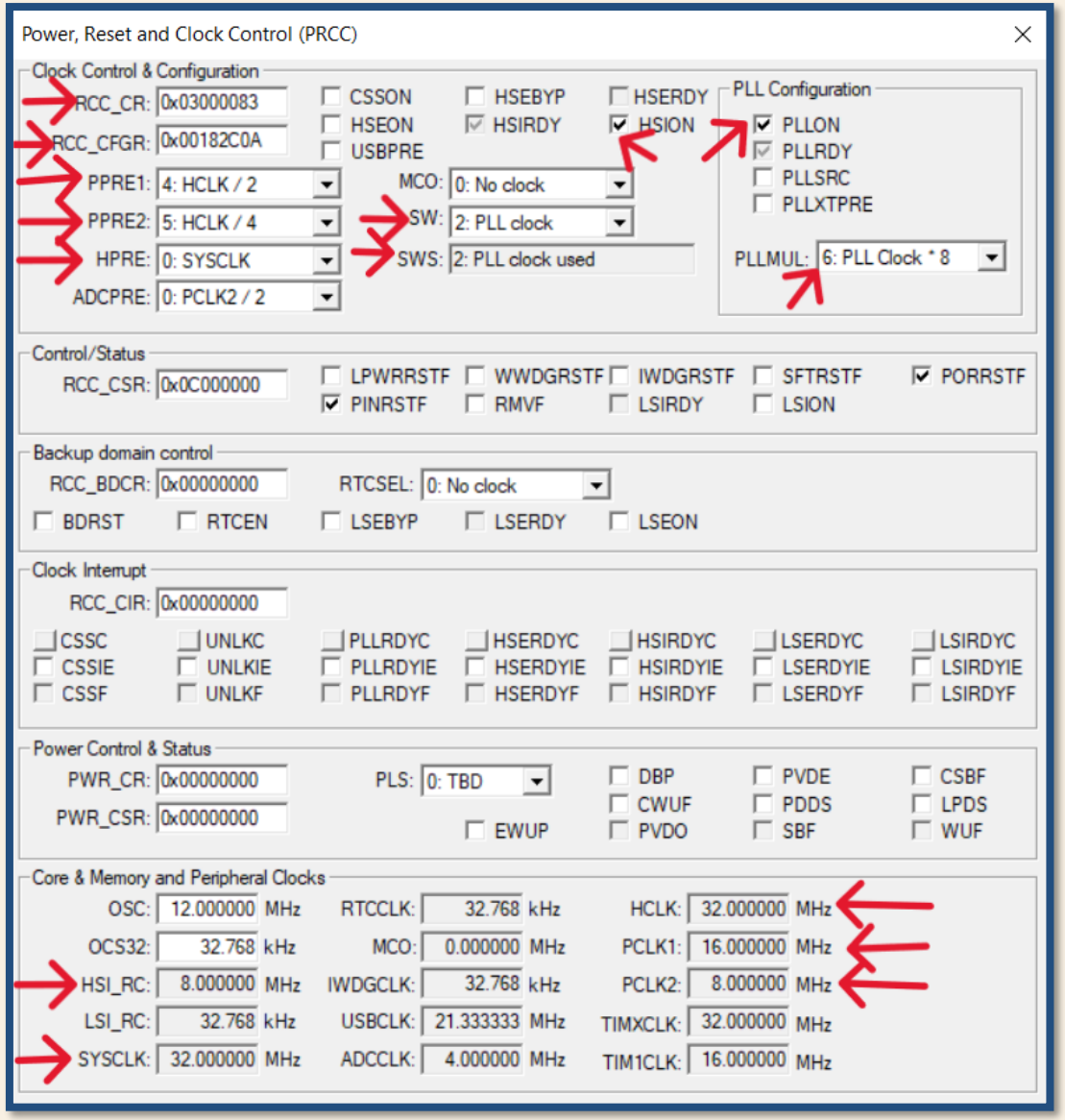
```c
#pragma pack(1)
typedef struct {
    uint32_t :2;
    uint32_t IOPAEN:1;
    uint32_t :29;
}SRCC_APB2ENR_t;

#define RCC_BASE        0x40021000
#define PORTA_BASE      0x40010800

#define GPIOA_CRH       *(volatile uint32_t*)(PORTA_BASE+0x04)
#define GPIOA_ODR       *(volatile uint32_t*)(PORTA_BASE+0x0C)
```

```
147 int main(void)
148 {
149
150     clock_init();
151
152     int i;
153     GPIOA_CRH &=0xFF0FFFFF;
154     GPIOA_CRH |=0x00200000;
155     while (1)
156     {
157         GPIOA_ODR |=1<<13;
158         for (i=0; i<5000; i++);
159         GPIOA_ODR &=~(1<<13);
160         for (i=0; i<5000; i++);
161     }
162     return 0;
163 }
```

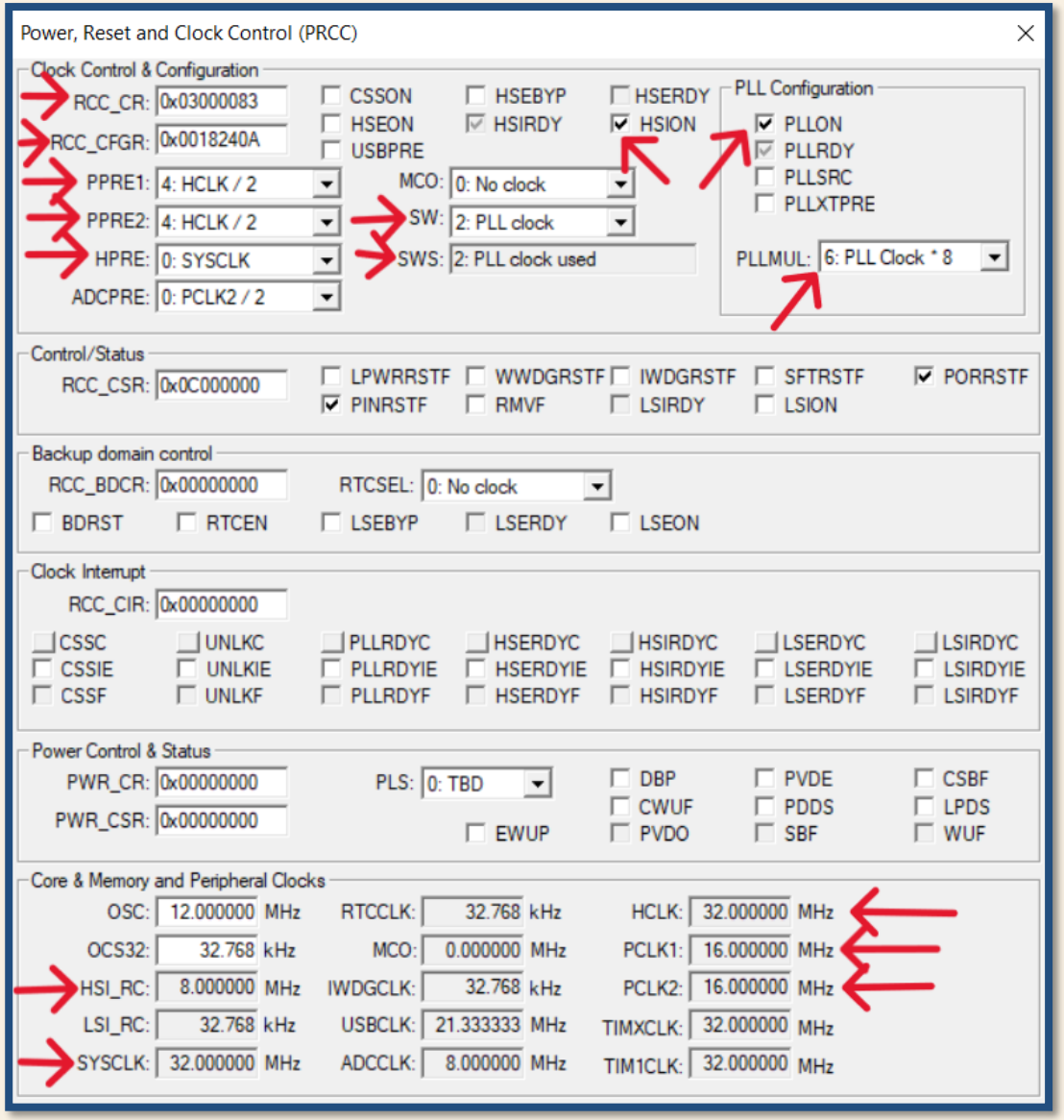
```c
void clock_init(){
    volatile SRCC_CR_t* const RCC_CR = (SRCC_CR_t*)(RCC_BASE+0x00);
    volatile SRCC_CFGR_t* const RCC_CFGR = (SRCC_CFGR_t*)(RCC_BASE+0x04);
    volatile SRCC_APB2ENR_t* const RCC_APB2ENR = (SRCC_APB2ENR_t*)(RCC_BASE+0x18);

    //  APB1 Bus frequency 4MHZ
    //  APB2 Bus frequency 2MHZ
    //  AHB frequency 8 MHZ
    //  SysClk 8 MHZ
    //  Use only internal HSI_RC

    RCC_CFGR->PPRE1 = 0b100; //100: HCLK divided by 2
    RCC_CFGR->PPRE2 = 0b101; //101: HCLK divided by 4
    RCC_APB2ENR->IOPAEN = 0b1; //Enable PORTA Clock
}
```

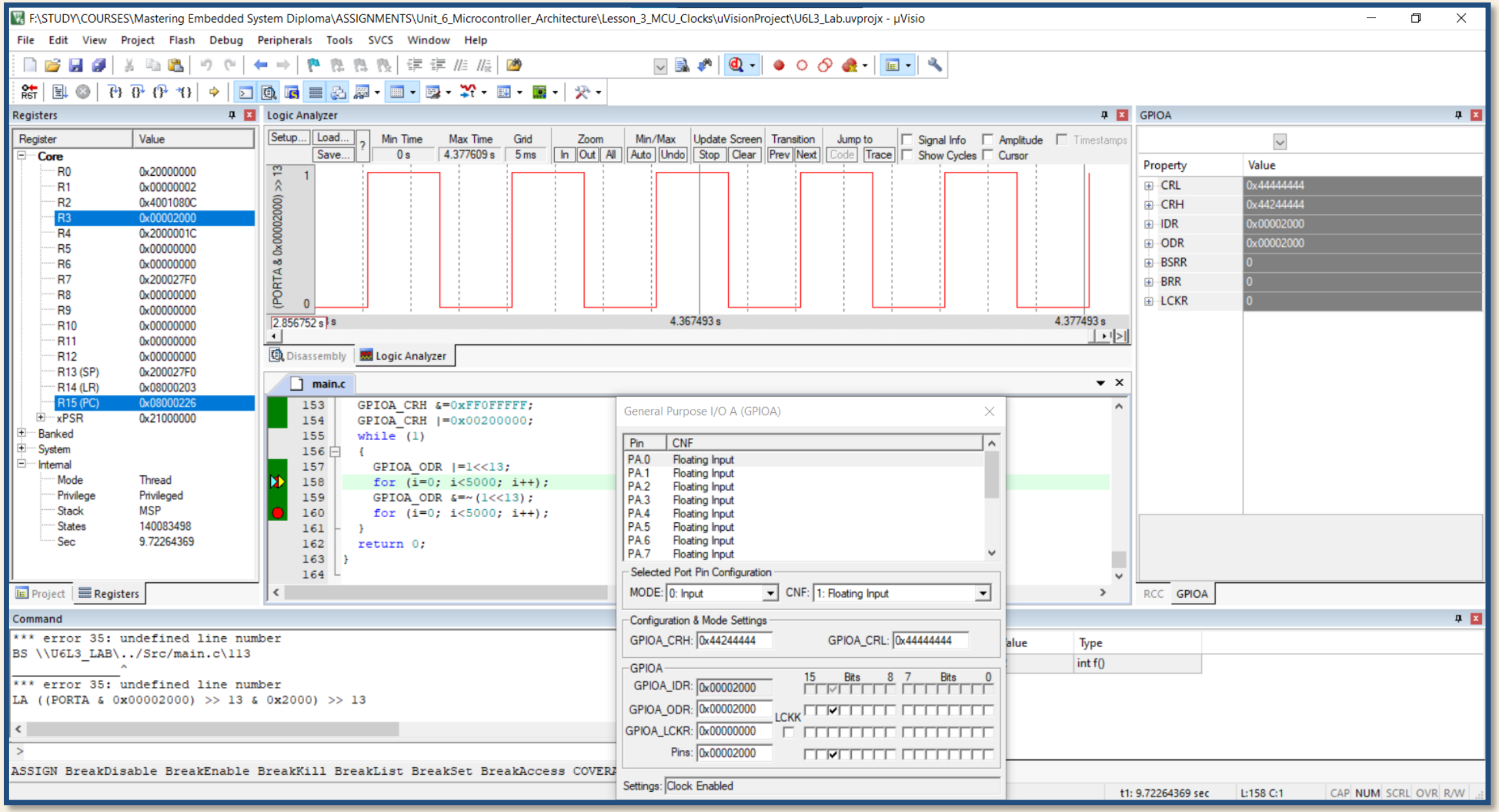


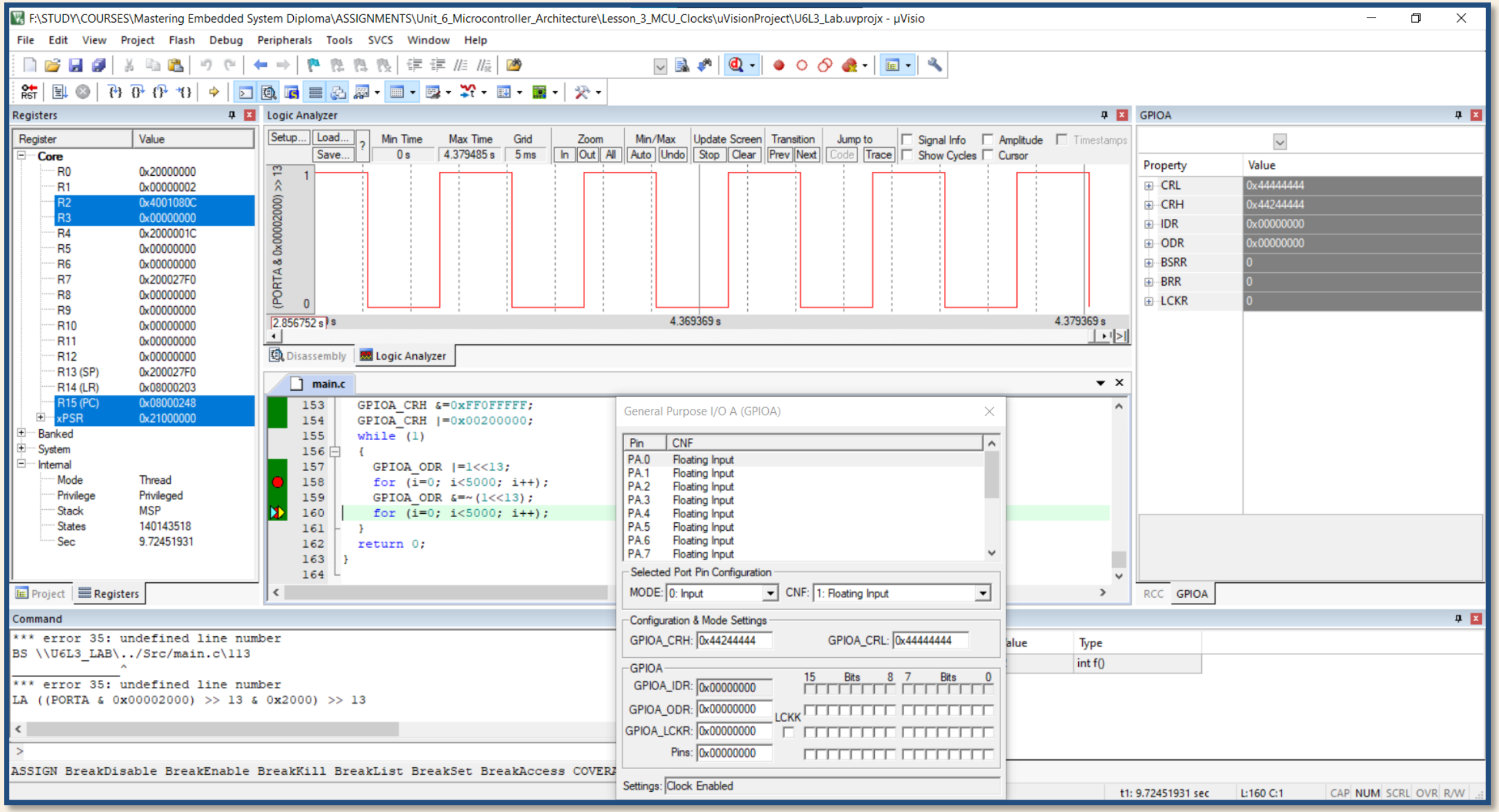
```c
void clock_init(){
    volatile SRCC_CR_t* const RCC_CR = (SRCC_CR_t*)(RCC_BASE+0x00);
    volatile SRCC_CFGR_t* const RCC_CFGR = (SRCC_CFGR_t*)(RCC_BASE+0x04);
    volatile SRCC_APB2ENR_t* const RCC_APB2ENR = (SRCC_APB2ENR_t*)(RCC_BASE+0x18);

    //  APB1 Bus frequency 16MHZ
    //  APB2 Bus frequency 8MHZ
    //  AHB frequency 32 MHZ
    //  SysClk 32 MHZ
    //  Use only internal HSI_RC

    // PLLSRC & PLLMUL bits can be written only when PLL is disabled.
    RCC_CFGR->PLLSRC = 0b0; //0: HSI oscillator clock / 2 selected as PLL input clock
    RCC_CFGR->PLLMUL = 0b0110; //0110: PLL input clock x 8

    RCC_CR->PLLON = 0b1; //1: PLL ON

    while(!RCC_CR->PLLRDY){
    }

    RCC_CFGR->PPRE1 = 0b100; //100: HCLK divided by 2
    RCC_CFGR->PPRE2 = 0b101; //101: HCLK divided by 4

    RCC_CFGR->SW = 0b10; //10: PLL selected as system clock

    RCC_APB2ENR->IOPAEN = 0b1; //Enable PORTA Clock
}
```



```c
void clock_init(){
    volatile SRCC_CR_t* const RCC_CR = (SRCC_CR_t*)(RCC_BASE+0x00);
    volatile SRCC_CFGR_t* const RCC_CFGR = (SRCC_CFGR_t*)(RCC_BASE+0x04);
    volatile SRCC_APB2ENR_t* const RCC_APB2ENR = (SRCC_APB2ENR_t*)(RCC_BASE+0x18);

    //  APB1 Bus frequency 16MHZ
    //  APB2 Bus frequency 16MHZ
    //  AHB frequency 32 MHZ
    //  SysClk 32 MHZ
    //  Use only internal HSI_RC

    // PLLSRC & PLLMUL bits can be written only when PLL is disabled.
    RCC_CFGR->PLLSRC = 0b0; //0: HSI oscillator clock / 2 selected as PLL input clock
    RCC_CFGR->PLLMUL = 0b0110; //0110: PLL input clock x 8

    RCC_CR->PLLON = 0b1; //1: PLL ON

    while(!RCC_CR->PLLRDY){
    }

    RCC_CFGR->PPRE1 = 0b100; //100: HCLK divided by 2
    RCC_CFGR->PPRE2 = 0b100; //100: HCLK divided by 4

    RCC_CFGR->SW = 0b10; //10: PLL selected as system clock

    RCC_APB2ENR->IOPAEN = 0b1; //Enable PORTA Clock
}
```




F:\STUDY\COURSES\Mastering Embedded System Diploma\ASSIGNMENTS\Unit_6_Microcontroller_Architecture\Lesson_3_MCU_Clocks\uVisionProject\U6L3_Lab.uvprojx - µVisio
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Register Value
Core
R0 0x20000000
R1 0x00000002
R2 0x4001080C
R3 0x00002000
R4 0x2000001C
R5 0x00000000
R6 0x00000000
R7 0x200027F0
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x200027F0
R14 (LR) 0x08000203
R15 (PC) 0x08000226
xPSR 0x21000000
Banked
System
Internal
Mode Thread
Privilege Privileged
Stack MSP
States 140083498
Sec 9.72264369
Logic Analyzer
Setup... Load... Save...
Min Time 0 s
Max Time 4.377609 s
Grid 5 ms
Zoom In Out All
Min/Max Auto Undo
Update Screen Stop Clear
Transition Prev Next
Jump to Code Trace
Signal Info Amplitude Timestamps
Show Cycles Cursor
(PORTA & 0x00002000) >> 13
2.856752 s
4.367493 s
4.377493 s
Disassembly Logic Analyzer
main.c
153 GPIOA_CRH &=0xFF0FFFFF;
154 GPIOA_CRH |=0x00200000;
155 while (1)
156 {
157 GPIOA_ODR |=1<<13;
158 for (i=0; i<5000; i++);
159 GPIOA_ODR &=~(1<<13);
160 for (i=0; i<5000; i++);
161 }
162 return 0;
163 }
164
General Purpose I/O A (GPIOA)
Pin CNF
PA.0 Floating Input
PA.1 Floating Input
PA.2 Floating Input
PA.3 Floating Input
PA.4 Floating Input
PA.5 Floating Input
PA.6 Floating Input
PA.7 Floating Input
Selected Port Pin Configuration
MODE: 0: Input
CNF: 1: Floating Input
Configuration & Mode Settings
GPIOA_CRH: 0x44244444
GPIOA_CRL: 0x44444444
GPIOA
GPIOA_IDR: 0x00002000
GPIOA_ODR: 0x00002000
GPIOA_LCKR: 0x00000000
Pins: 0x00002000
LCKK
15 Bits 8 7 Bits 0
Settings: Clock Enabled
GPIOA
Property Value
CRL 0x44444444
CRH 0x44244444
IDR 0x00002000
ODR 0x00002000
BSRR 0
BRR 0
LCKR 0
RCC GPIOA
Command
*** error 35: undefined line number
BS \\U6L3_LAB\../Src/main.c\113
*** error 35: undefined line number
LA ((PORTA & 0x00002000) >> 13 & 0x2000) >> 13
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERA
Value Type
int f()
Project Registers
t1: 9.72264369 sec
L:158 C:1
CAP NUM SCRL OVR R/W


F:\STUDY\COURSES\Mastering Embedded System Diploma\ASSIGNMENTS\Unit_6_Microcontroller_Architecture\Lesson_3_MCU_Clocks\uVisionProject\U6L3_Lab.uvprojx - µVisio
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Register Value
Core
R0 0x20000000
R1 0x00000002
R2 0x4001080C
R3 0x00000000
R4 0x2000001C
R5 0x00000000
R6 0x00000000
R7 0x200027F0
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x200027F0
R14 (LR) 0x08000203
R15 (PC) 0x08000248
xPSR 0x21000000
Banked
System
Internal
Mode Thread
Privilege Privileged
Stack MSP
States 140143518
Sec 9.72451931
Logic Analyzer
Setup... Load... Save...
Min Time 0 s
Max Time 4.379485 s
Grid 5 ms
Zoom In Out All
Min/Max Auto Undo
Update Screen Stop Clear
Transition Prev Next
Jump to Code Trace
Signal Info
Show Cycles
Amplitude
Cursor
Timestamps
(PORTA & 0x00002000) >> 13
2.856752 s
4.369369 s
4.379369 s
Disassembly
Logic Analyzer
main.c
153 GPIOA_CRH &=0xFF0FFFFF;
154 GPIOA_CRH |=0x00200000;
155 while (1)
156 {
157 GPIOA_ODR |=1<<13;
158 for (i=0; i<5000; i++);
159 GPIOA_ODR &=~(1<<13);
160 for (i=0; i<5000; i++);
161 }
162 return 0;
163 }
164
General Purpose I/O A (GPIOA)
Pin CNF
PA.0 Floating Input
PA.1 Floating Input
PA.2 Floating Input
PA.3 Floating Input
PA.4 Floating Input
PA.5 Floating Input
PA.6 Floating Input
PA.7 Floating Input
Selected Port Pin Configuration
MODE: 0: Input
CNF: 1: Floating Input
Configuration & Mode Settings
GPIOA_CRH: 0x44244444
GPIOA_CRL: 0x44444444
GPIOA
GPIOA_IDR: 0x00000000
GPIOA_ODR: 0x00000000
GPIOA_LCKR: 0x00000000
Pins: 0x00000000
LCKK
15 Bits 8 7 Bits 0
Settings: Clock Enabled
GPIOA
Property Value
CRL 0x44444444
CRH 0x44244444
IDR 0x00000000
ODR 0x00000000
BSRR 0
BRR 0
LCKR 0
RCC GPIOA
Command
*** error 35: undefined line number
BS \\U6L3_LAB\../Src/main.c\113
*** error 35: undefined line number
LA ((PORTA & 0x00002000) >> 13 & 0x2000) >> 13
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERA
Type
int f()
Project Registers
t1: 9.72451931 sec
L:160 C:1
CAP NUM SCRL OVR R/W

THANK YOU

Mohamed Hamdy

Mohamedhamdy.m.h.m.a@gmail.com