

John Duriman

CPE 403

Lab 4

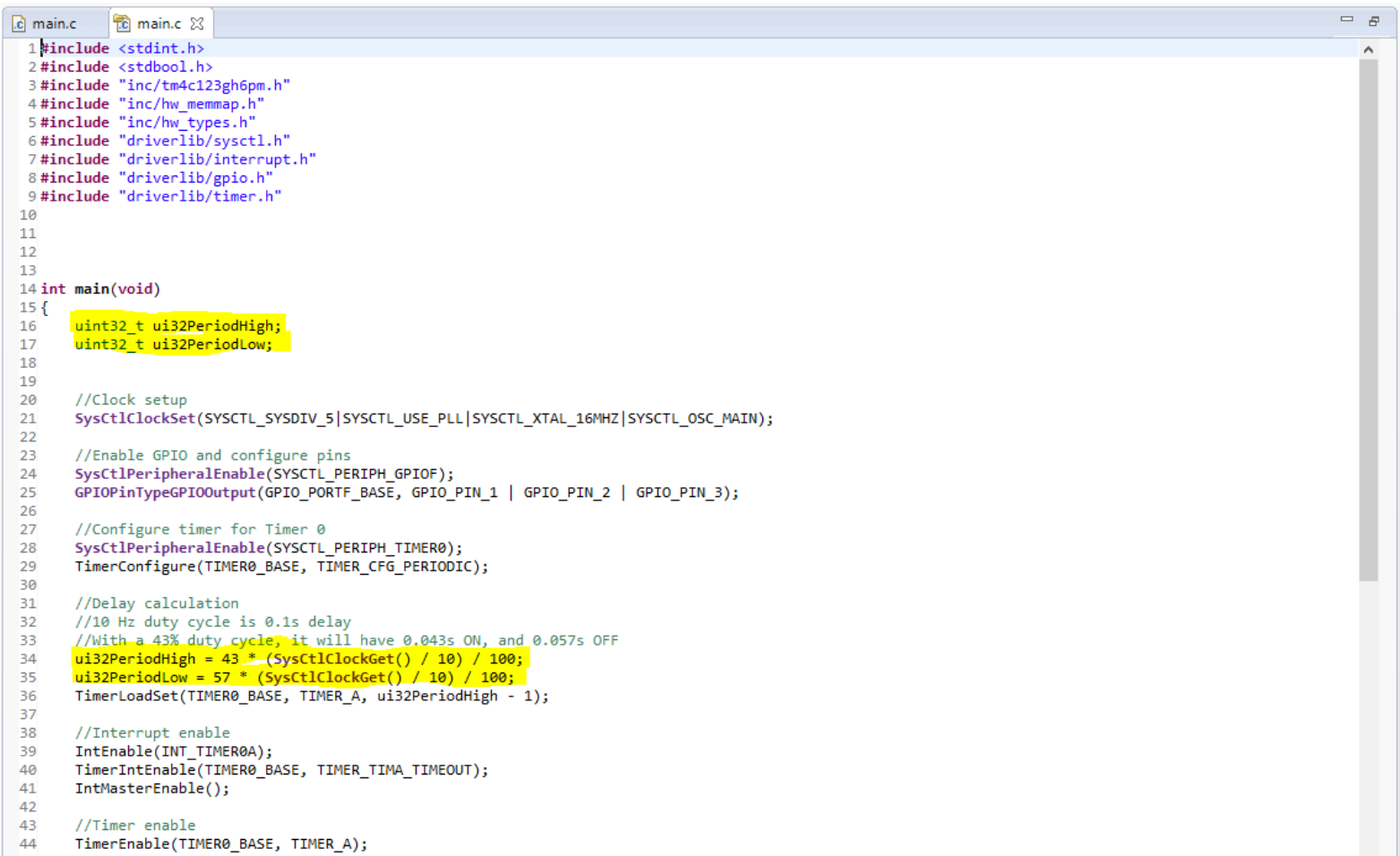
5002373995

Task 00: Execute the provided code, no submission is required.

Task 01: Change the toggle of the GPIO at 10 Hz using Timer0 with 43% duty cycle and verify the waveform generated.

Youtube Link: <https://youtu.be/yasprspzCJ8>

Modified Code:



```
1 #include <stdint.h>
2 #include <stdbool.h>
3 #include "inc/tm4c123gh6pm.h"
4 #include "inc/hw_memmap.h"
5 #include "inc/hw_types.h"
6 #include "driverlib/sysctl.h"
7 #include "driverlib/interrupt.h"
8 #include "driverlib/gpio.h"
9 #include "driverlib/timer.h"
10
11
12
13
14 int main(void)
15 {
16     uint32_t ui32PeriodHigh;
17     uint32_t ui32PeriodLow;
18
19     //Clock setup
20     SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);
21
22     //Enable GPIO and configure pins
23     SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
24     GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3);
25
26     //Configure timer for Timer 0
27     SysCtlPeripheralEnable(SYSCTL_PERIPH_TIMER0);
28     TimerConfigure(TIMER0_BASE, TIMER_CFG_PERIODIC);
29
30     //Delay calculation
31     //10 Hz duty cycle is 0.1s delay
32     //With a 43% duty cycle, it will have 0.043s ON, and 0.057s OFF
33     ui32PeriodHigh = 43 * (SysCtlClockGet() / 10) / 100;
34     ui32PeriodLow = 57 * (SysCtlClockGet() / 10) / 100;
35     TimerLoadSet(TIMER0_BASE, TIMER_A, ui32PeriodHigh - 1);
36
37     //Interrupt enable
38     IntEnable(INT_TIMER0A);
39     TimerIntEnable(TIMER0_BASE, TIMER_TIMA_TIMEOUT);
40     IntMasterEnable();
41
42     //Timer enable
43     TimerEnable(TIMER0_BASE, TIMER_A);
44 }
```

```
46
47 while(1)
48 {
49
50 }
51
52 }
53
54 void Timer0IntHandler(void)
55 {
56     uint32_t ui32PeriodHigh = 43 * (SysCtlClockGet() / 10) / 100;
57     uint32_t ui32PeriodLow = 57 * (SysCtlClockGet() / 10) / 100;
58
59     //Clear timer interrupt
60     TimerIntClear(TIMER0_BASE, TIMER_TIMA_TIMEOUT);
61     // Read the current state of the GPIO pin and
62     // write back the opposite state
63
64     if(GPIOPinRead(GPIO_PORTF_BASE, GPIO_PIN_2))
65     {
66         TimerLoadSet(TIMER0_BASE, TIMER_A, ui32PeriodHigh - 1); //Load low
67         GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, 0);
68     }
69     else
70     {
71         TimerLoadSet(TIMER0_BASE, TIMER_A, ui32PeriodLow - 1); //Load High
72         GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_2, 4);
73     }
74 }
75
```

Writable

Smart Insert

1:1

⋮

⋮

⋮

Task 02: Include a GPIO Interrupt to Task 01 from switch SW2 to turn ON and the LED for 1 sec. Use a Timer1 to calculate the 1 sec delay. The toggle of the GPIO is suspended when executing the interrupt.

Youtube Link: <https://youtu.be/yasprspzCJ8>

Modified Code:

```
Resource Explorer | tm4c123gh6pm_startup_ccs.c | main.c | GPIOPinWrite(unsigned int, unsigned char, unsigned char) at gpio.c:1,147 0xbea
1 #include <stdint.h>
2 #include <stdbool.h>
3 #include "inc/tm4c123gh6pm.h"
4 #include "inc/hw_memmap.h"
5 #include "inc/hw_types.h"
6 #include "driverlib/sysctl.h"
7 #include "driverlib/interrupt.h"
8 #include "driverlib/gpio.h"
9 #include "driverlib/timer.h"
10 #include "inc/hw_gpio.h"
11 #include "driverlib/pin_map.h"
12 #include "driverlib/rom_map.h"
13
14 int main(void)
15 {
16     uint32_t ui32PeriodHigh;
17     uint32_t ui32PeriodLow;
18     uint32_t ui32Period;
19
20     //Clock setup
21     SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);
22
23     //Enable GPIO and configure pins as outputs
24     SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
25     GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3);
26
27     //Unlock Pin PF0
28     HWREG(GPIO_PORTF_BASE + GPIO_O_LOCK) = GPIO_LOCK_KEY;
29     HWREG(GPIO_PORTF_BASE + GPIO_O_CR) = 0x1;
30     HWREG(GPIO_PORTF_BASE + GPIO_O_LOCK) = 0;
31
32     //Switch Interrupt
33     //Enable GPIO peripheral and configure pins as inputs
34     GPIOPinTypeGPIOInput(GPIO_PORTF_BASE, GPIO_PIN_0);
35     GPIOPadConfigSet(GPIO_PORTF_BASE, GPIO_PIN_0, GPIO_STRENGTH_2MA, GPIO_PIN_TYPE_STD_WPU);
36     GPIOIntEnable(GPIO_PORTF_BASE, GPIO_INT_PIN_0);
37     GPIOIntTypeSet(GPIO_PORTF_BASE, GPIO_INT_PIN_0, GPIO_RISING_EDGE);
38     IntEnable(INT_GPIOF);
39
40     //Configure Timer 0 and Timer 1
41     SysCtlPeripheralEnable(SYSCTL_PERIPH_TIMER0);
42     TimerConfigure(TIMER0_BASE, TIMER_CFG_PERIODIC);
43     SysCtlPeripheralEnable(SYSCTL_PERIPH_TIMER1);
44     TimerConfigure(TIMER1_BASE, TIMER_CFG_PERIODIC);
45 }
```

```

46 //Delay calculation
47 //10 Hz duty cycle is 0.1s delay
48 //With a 43% duty cycle, it will have 0.043s ON, and 0.057s OFF
49 ui32PeriodHigh = 43 * (SysCtlClockGet() / 10) / 100;
50 ui32PeriodLow = 57 * (SysCtlClockGet() / 10) / 100;
51 TimerLoadSet(TIMER0_BASE, TIMER_A, ui32PeriodHigh - 1);
52
53 //Interrupt enable
54 IntEnable(INT_TIMER0A);
55 TimerIntEnable(TIMER0_BASE, TIMER_TIMA_TIMEOUT);
56 IntMasterEnable();
57
58 //Timer enable
59 TimerEnable(TIMER0_BASE, TIMER_A);
60
61 while(1){    }
62 }
63
64 void Timer0IntHandler(void)
65 {
66     uint32_t ui32PeriodHigh = 43 * (SysCtlClockGet() / 10) / 100;
67     uint32_t ui32PeriodLow = 57 * (SysCtlClockGet() / 10) / 100;
68
69     //Clear timer interrupt
70     TimerIntClear(TIMER0_BASE, TIMER_TIMA_TIMEOUT);
71     // Read the current state of the GPIO pin and
72     // write back the opposite state
73     if(GPIOPinRead(GPIO_PORTF_BASE, GPIO_PIN_2))
74     {
75         TimerLoadSet(TIMER0_BASE, TIMER_A, ui32PeriodHigh - 1);          //Load low
76         GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, 0);
77     }
78     else
79     {
80         TimerLoadSet(TIMER0_BASE, TIMER_A, ui32PeriodLow - 1);          //Load High
81         GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_2, 4);
82     }
83 }
84 }
85
86 void PortFPin0IntHandler(void)
87 {
88     //Clear GPIO interrupt
89     GPIOIntClear(GPIO_PORTF_BASE, GPIO_INT_PIN_0);
90
91     //Configure delay and enable timer 1
92     TimerLoadSet(TIMER1_BASE, TIMER_A, SysCtlClockGet()-1);
93     TimerEnable(TIMER1_BASE, TIMER_A);
94
95     //Make all Pins low
96     GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, 0);
97
98     //Set Pin 3 to high
99     GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_3, 8);
100
101     //TIMER 1 Count reached
102     TimerIntEnable(TIMER1_BASE, TIMER_TIMA_TIMEOUT);
103     while(1)
104     {
105         if(TimerIntStatus(TIMER1_BASE, true) & TIMER_TIMA_TIMEOUT == TIMER_TIMA_TIMEOUT)
106         {
107             TimerIntClear(TIMER1_BASE, TIMER_TIMA_TIMEOUT);
108             break;
109         }
110     }
111
112     TimerDisable(TIMER1_BASE, TIMER_A);
113 }
114

```