EE690: Embedded Systems Lab

Lab Assignment 3

Group 10
EE23DP003 Daniel Dsa
EE22DP007 Pradeep Kumar M

GitHub repo: https://github.com/PEG-IITDH/lab-6-blinky-EE23DP003

EK-TM4C123: Implementation of System Timer (Systick)

1 Objective:

To use the system timer (Systick) in polled mode to generate a waveform with f = 1 KHz and d = 20%.

2 Procedure:

1. GPIO port F is initialized by enabling clock to it, unlocking it's commit register and making it configurable. GPIO pins for the onboard LED's (Port F pins 1,2,3) are set as digital output pins.

2. The GPIO pins for the onboard LED's are turned ON, followed by a call to the Systick_Delay function with specific value passed for implementing ON delay of 20us. The onboard LED's are then turned off, followed by a call to the Systick_Delay function again to implement the off delay of 80us.

3. The Systick_Delay function initializes the Systick current register value to 0. The Systick reload value is passed on from the Systick_Delay function call in the main program and the Systick control register is enabled with system clock.

4. The delay function continuously checks the Systick Control register to see if the Systick has counted down to zero and the Systick count flag is set. Once the Systick count flag is set, the system timer Systick is disabled.

3 Documents Referred:

1. TM4C123GH6PM microcontroller datasheet

2. Cortex-M4 Technical Reference Manual

4 System Timer:

The ARM Cortex-M4F integrated system timer, SysTick, provides a 24-bit, clear-on-write, decrementing, wrap-on-zero counter with a flexible control mechanism. Systick has 3 registers, all of which can be accessed from the privileged mode.

1. SysTick Control and Status Register (**STCTRL**); which enables Systick features (tm4c123gh6pm datasheet, Page 138)
2. SysTick Reload Value Register (**STRELOAD**); which specifies the start value to load into the SysTick Current Value (STCURRENT) register when the counter reaches 0 (tm4c123gh6pm datasheet, Page 140).
3. SysTick Current Value Register (**STCURRENT**); which contains the current value of the SysTick counter (tm4c123gh6pm datasheet, Page 141).

5 Code:

```
#include <stdint.h>
#include <stdbool.h>
#include "tm4c123gh6pm.h"


//Definitions for SysTick CSR (Control and Status Register)
#define ENABLE (1 << 0)         // CSR bit 0 to enable the SysTick timer
#define CLK_SRC (1<<2)         // CSR bit 2 to take system clock
#define COUNT_FLAG  (1 << 16) // CSR bit 16 automatically set to 1 when SysTick reaches 0
#define CLOCK_MHZ 16         // Default System Clock

// GPIO PORTF and Systick function declarations
void GPIO_PORTF_Init (void);        //GPIO PORTF Initialization Function
void Systick_Delay (uint32_t us);  //Systick Delay function

/* Main Program */

int main(void)
{
    GPIO_PORTF_Init ();
    while(1)                          //implement continuously
    {
        GPIO_PORTF_DATA_R = 0X0E;    // Red, Blue and Green LED turned on
        Systick_Delay (200);        // Call Delay function
        GPIO_PORTF_DATA_R = 0X00;  // All pins on Port F turned off
        Systick_Delay (800);      // Call Delay function
    }
}

void GPIO_PORTF_Init (void)
{
    SYSCTL_RCGC2_R |= 0x00000020;      // enable clock to GPIOF
    GPIO_PORTF_LOCK_R = 0x4C4F434B;   // unlock GPIO Port F commit register
    GPIO_PORTF_CR_R = 0x1F;          // make PORTF configurable
    GPIO_PORTF_DEN_R = 0x1F;        // enable digital functions on PF 0,1,2,3,4
    GPIO_PORTF_DIR_R = 0x0E;       // set PF1,PF2,PF3 (LEDs) as output
}


/* Delay Function */
void Systick_Delay (uint32_t us)
{
    NVIC_ST_CURRENT_R = 0;                  // SysTick current value register cleared
    NVIC_ST_RELOAD_R = us*CLOCK_MHZ;       // SysTick reload value for 'us' microseconds
    NVIC_ST_CTRL_R |= ( ENABLE | CLK_SRC );  // SysTick Enable, use system clock

    while ((NVIC_ST_CTRL_R & COUNT_FLAG) == 0)  // wait until systick count flag is set
    {
          // do nothing
    }
    NVIC_ST_CTRL_R = 0;                // Disable SysTick when COUNT_FLAG is set
}
```

6 Results:

As seen in Fig. 1 and Fig. 2, a 1KHz signal with 20% duty ratio is generated on pins 1, 2 and 3 of PORTF, which is verified on an oscilloscope.

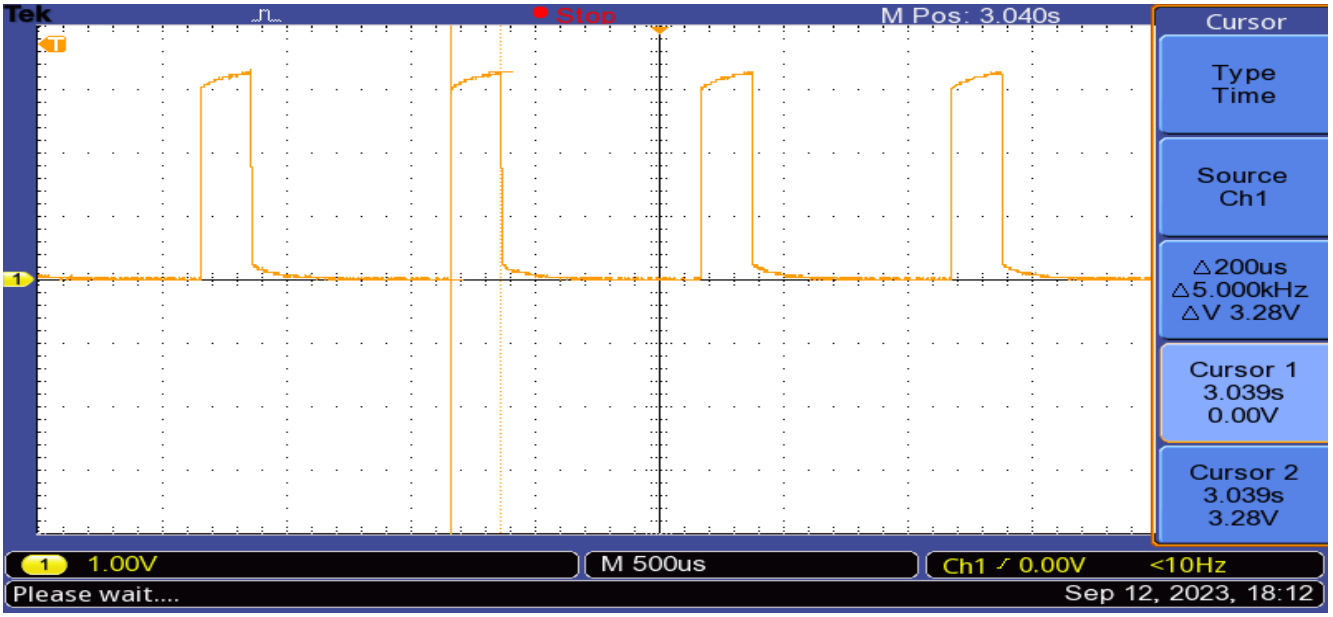

Figure 1: Oscilloscope waveform for generated pulses measured at PF1 (on time =200us)

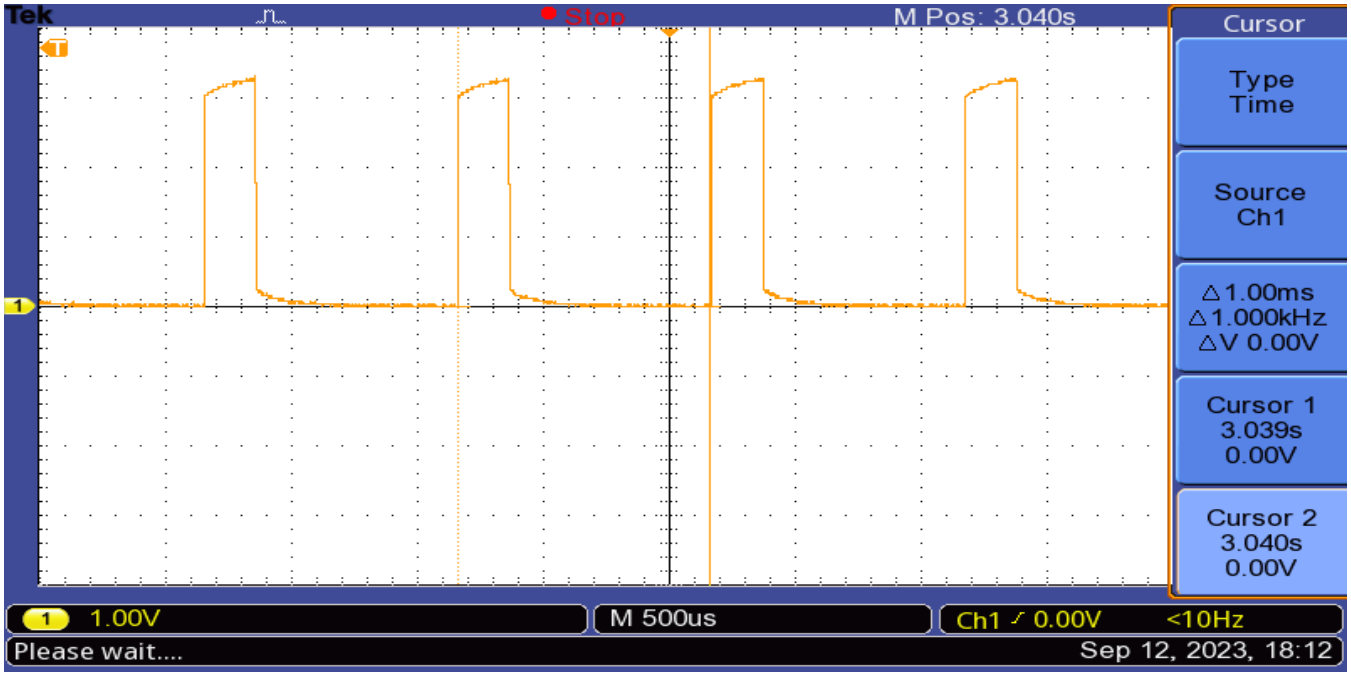

Figure 2: Oscilloscope waveform for generated pulse measured at PF1 (1KHz period)

7 Conclusion

The system timer ‘SysTick’ can be used for generating delays, and it is better than using a while() or for() loop to generate delay as it frees the processor to perform other activities while the Systick independently counts down to produce the desired value of delay.