HACETTEPE UNIVERSITY

COMPUTER SCIENCE DEPARTMENT

EMBEDDED SYSTEMS LAB.

LAB. REPORT II

TEAM NAME:

KENDINI MARUL SANAN MARUL

MEMBERS:

KUTAY BARCIN 21526715
DEFNE TUNCER 21627686

MISSION: IMPOSSIBLE

STEP1: I/O

In order to light up the LEDs with given order and delays, **PORTF** initialization is done:

INPUTS: PF0 (SW2) and PF4 (SW1)

OUTPUTS: PF1 (RED), PF2 (BLUE), PF3 (GREEN)

STEP2: LOOPS

While SW1 is not pressed **red, blue, green** loop is active.

RED: 0.50 seconds

BLUE: 1.00 seconds

GREEN: 1.50 seconds

While SW1 is pressed **red, red, red, white, white, white, red, red, red** loop is active.

RED: 0.25 seconds

WHITE: 0.50 seconds

OFF: 0.25 seconds

```
int main(void){
  PortF_Init();
  while(1){
    while(PF4&0x10){
      GPIO_PORTF_DATA_R = 0x02;  // LED is red
      Delay(0.5);
      GPIO_PORTF_DATA_R = 0x04;  // LED is blue
      Delay(1);
      GPIO_PORTF_DATA_R = 0x08;   // LED is green
      Delay(1.5);
    }
    while(!(PF4&0x10)){
      GPIO_PORTF_DATA_R = 0x02; Delay(0.25); GPIO_PORTF_DATA_R = 0x00; Delay(0.25);
      GPIO_PORTF_DATA_R = 0x02; Delay(0.25); GPIO_PORTF_DATA_R = 0x00; Delay(0.25);
      GPIO_PORTF_DATA_R = 0x02; Delay(0.25); GPIO_PORTF_DATA_R = 0x00; Delay(0.25);
      GPIO_PORTF_DATA_R = 0x0E; Delay(0.5); GPIO_PORTF_DATA_R = 0x00; Delay(0.25);
      GPIO_PORTF_DATA_R = 0x0E; Delay(0.5); GPIO_PORTF_DATA_R = 0x00; Delay(0.25);
      GPIO_PORTF_DATA_R = 0x0E; Delay(0.5); GPIO_PORTF_DATA_R = 0x00; Delay(0.25);
      GPIO_PORTF_DATA_R = 0x02; Delay(0.25); GPIO_PORTF_DATA_R = 0x00; Delay(0.25);
      GPIO_PORTF_DATA_R = 0x02; Delay(0.25); GPIO_PORTF_DATA_R = 0x00; Delay(0.25);
      GPIO_PORTF_DATA_R = 0x02; Delay(0.25); GPIO_PORTF_DATA_R = 0x00; Delay(0.25);
    }
  }
}
```

PF4&0x10 → 000**?**0000

All bits but PF4 is cleared of PORTF.

SW1 is pressed → 00000000 which is **False**

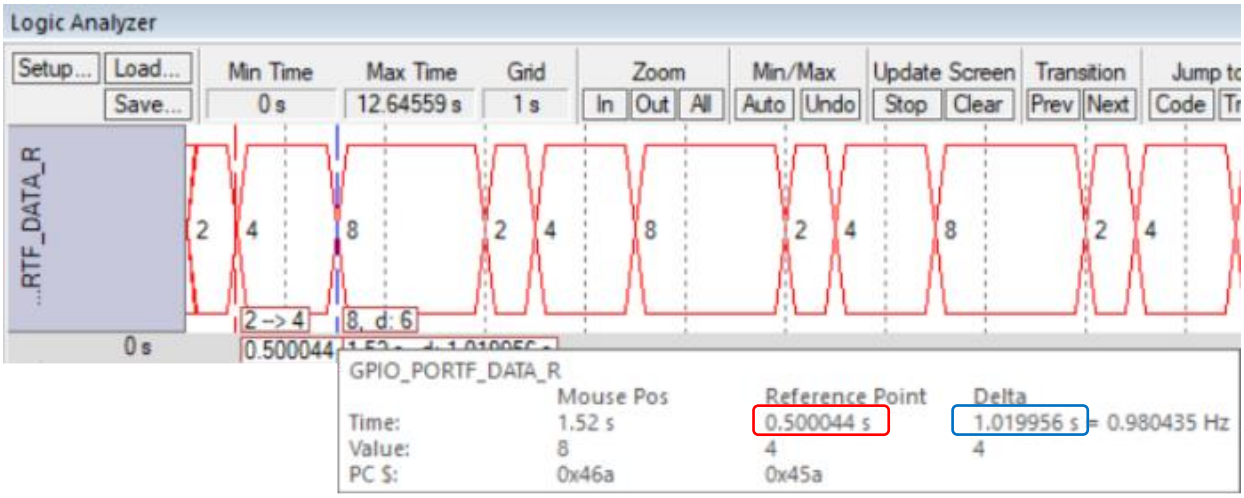
SW1 is not pressed → 00010000 which is **True**

STEP3: DELAY

It takes **x seconds** to return from Delay function.

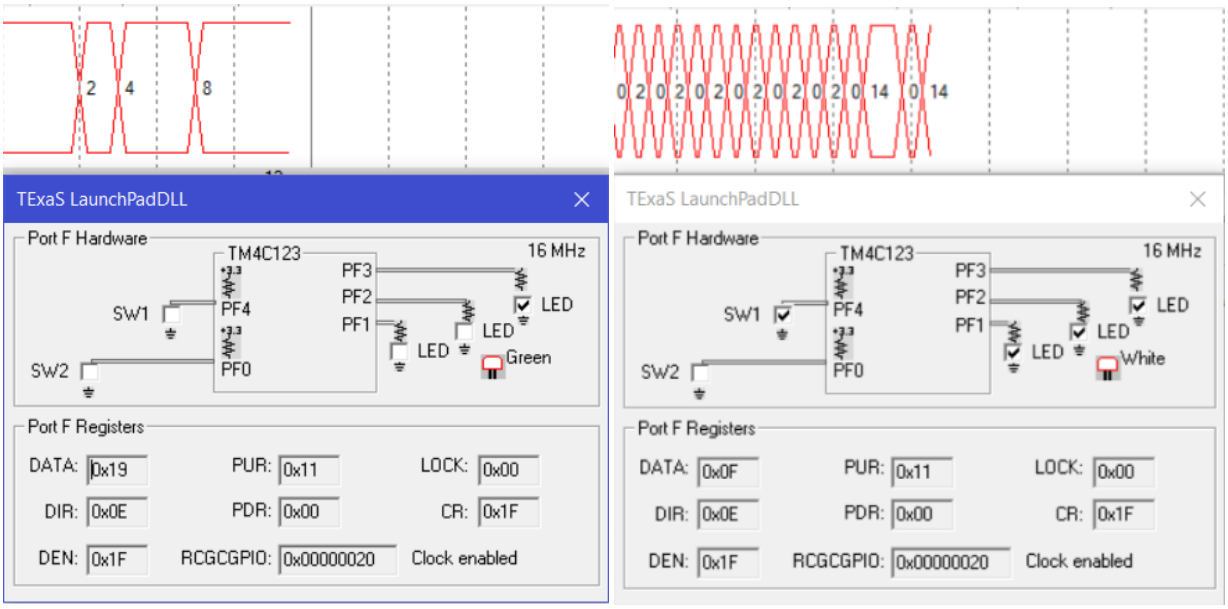
```
void Delay(double x){unsigned long volatile time;
  time = 1454480*x;  // x sec
  while(time){
    time--;
  }
}
```

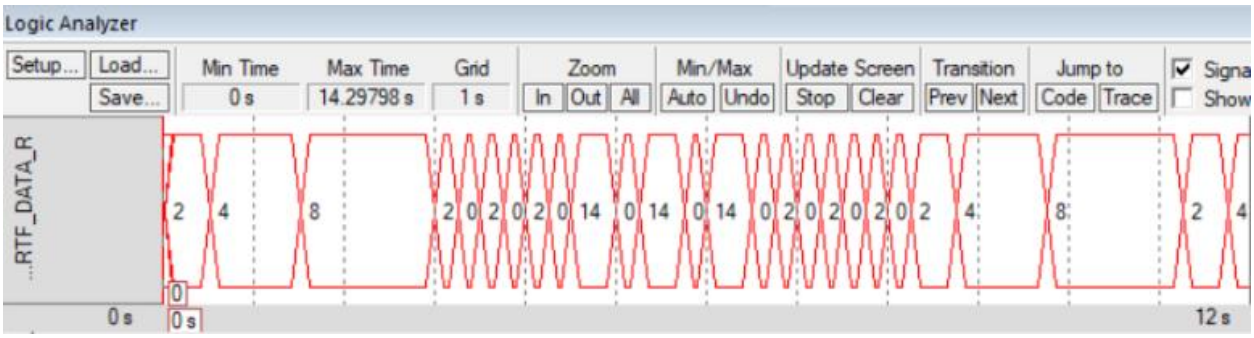
To check whether decrementing from **1454480** really is taking **1 second**, we controlled loops using **Logic Analyzer.**



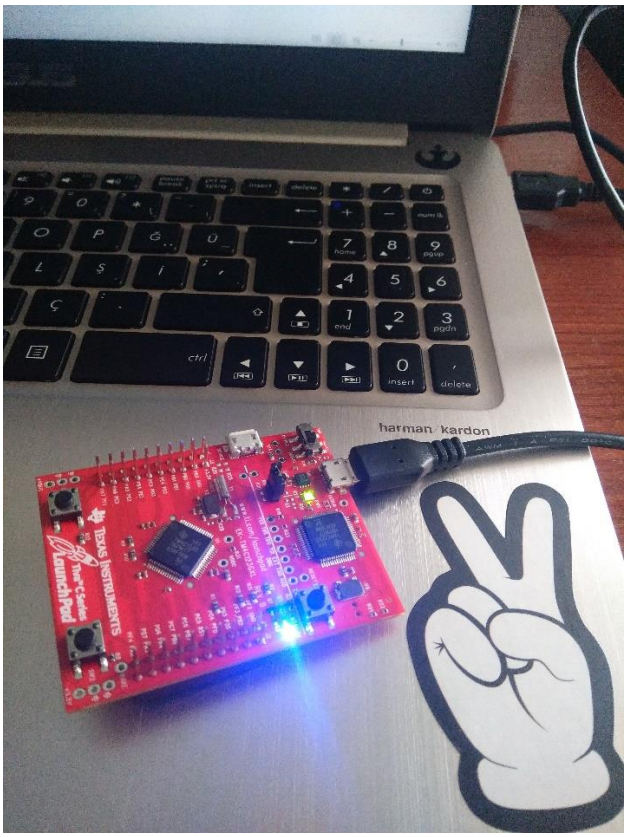
From the screenshot of Logic Analyzer we can observe that, **red (2)** light up 0.5 seconds and **blue (4)** light up 1.0 seconds.

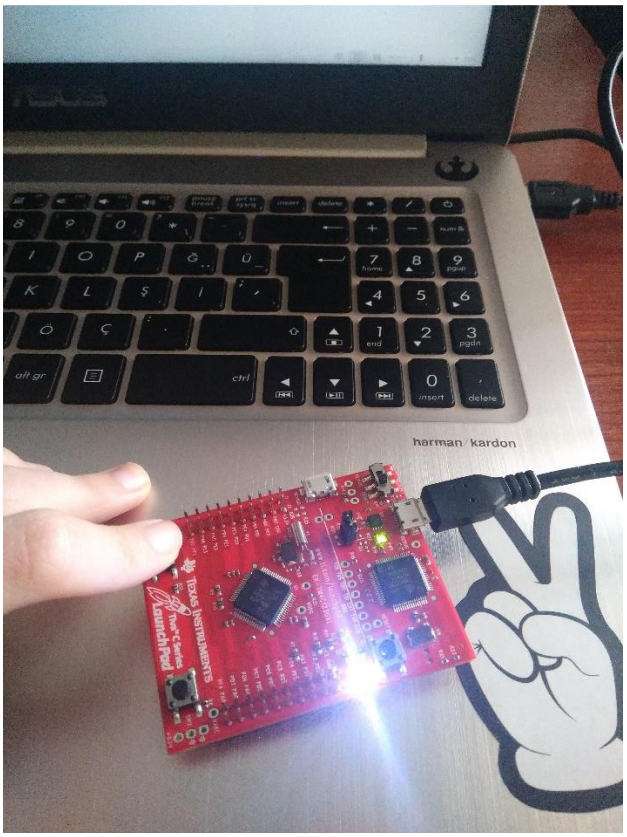
STEP4: SIMULATOR





STEP5: LAUNCHPAD





1. USE STELLARIS ICDI
2. BUILD TARGET
3. DOWNLOAD TO FLASH
4. START DEBUG SESSION
5. RUN

STEP6: EXPERIMENTS

1. **In this lab, you are detecting the pressing of a button via polling. Do you think if it is a good practice? Are there any disadvantages?**
   When we detect the pressing of a button via polling, it is hard to react to pressing immediately. But if we were to use interrupts it would immediately trigger that event we want to accomplish.
2. **In this lab, you introduced delay via looping. Do you think if it is a good practice? Are there any disadvantages?**
   Delays via looping is a good practice for flashing lights that are not meant for situations which require precise time since the time passes depends on the current power of hardware. If it was a very serious situation such as a pacemaker or a rocket launcher, introducing delay via looping would have been a very risky movement.
3. **How long does it take for a single iteration of the 1-second delay loop on the development board?**
   1/1454480 seconds = 0.0000006875 seconds = 687.5 ns