

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМ. ІГОРЯ
СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
КАФЕДРА АВТОМАТИЗОВАНОГО УПРАВЛІННЯ ТЕХНІЧНИМИ
СИСТЕМАМИ

Лабораторна робота № 4

Варіант 1

По дисципліні «Програмування мікроконтролерних систем»

Тема: «Вимірювання ширині імпульсів по збільшенню.»

Виконали:

студенти групи ІТ-51

Бессмертний Р.С.

Цитовцева А.С.

(підпис, дата)

Перевірив:

ст. викладач кафедри АУТС

Катін П. Ю.

(підпис, дата)

Мета: Користуючись результатами лабораторної роботи 2, написати програму, що здійснює вимірювання ширини імпульсів по попередньому фронту за допомогою TIM2.

Хід роботи:

Варіант 1

Користуючись результатами лабораторної роботи 2, побудувати програму для приймача і передавача, що використовує ПДП для *USART1*. Програма має передавати блоки даних розміром у 16 байт, після успішного прийому відображати інформацію про прийом на цифровому індикаторі. Початок передачі визначається натисканням кнопки на передавачі.

Написати і налагодити програми. Протестувати програми на реальному обладнанні, зробити тестування. При необхідності зробити корекцію у програмних і апаратних налаштуваннях.

```
#include "stm32f10x.h"
#include "stm32f10x_gpio.h"
#include "stm32f10x_rcc.h"
#include "stm32f10x_tim.h"
#include "timer_lib.h"
#include "tm1637.h"

#define STATE_GET_TIMER_INPUT 1
#define STATE_IDLE 0

#define TIMER_NO_INPUT 410
#define TIMER_INPUT_IS_ZERO 404

#define PERIOD 2000 // probably milliseconds

volatile uint16_t PPMBuffer = 0x0000;
volatile uint16_t PPMValue_Prev, PPMValue;

volatile uint16_t finite_state = STATE_IDLE;

void TIM2_IRQHandler(void){
    volatile uint16_t PPM;

    if (TIM_GetITStatus(TIM2, TIM_IT_CC2) != RESET)
    {
        /* Reset flag */
        TIM_ClearITPendingBit(TIM2, TIM_IT_CC2);

        PPMValue_Prev = PPMValue;
        PPMValue = TIM_GetCapture2(TIM2);
        PPM = (PPMValue >= PPMValue_Prev) ? (PPMValue - PPMValue_Prev) :
(UINT16_MAX - PPMValue_Prev + PPMValue);
```

```

    if (PPM < 10000) { // Pause
        PPMBuffer = PPM;
    }

    else PPMBuffer = 9999;

    /* over-capture */
    if (TIM_GetFlagStatus(TIM2, TIM_FLAG_CC2OF) != RESET)
    {
        TIM_ClearFlag(TIM2, TIM_FLAG_CC2OF);
        // ...
    }
}

int main(void)
{
    SetSysClockTo72();
    TIMER2_input_init_A2(7200);
    TIMER4_PWM_init_B8(72, PERIOD);

    double pulse_width = 0.01;
    TIM4->CCR3 = PERIOD * pulse_width;

    TM1637_init();
    TM1637_brightness(BRIGHTTEST);
    TM1637_display_all(TIMER_NO_INPUT);

    while(1)
    {
        if (PPMBuffer != 0)
        {
            TM1637_display_all(PPMBuffer);
        }
        else
        {
            TM1637_display_all(TIMER_INPUT_IS_ZERO);
        }
    }
}

```

Висновки: В даній лабораторній роботі був написаний і перевірений практично код, що здійснює вимірювання ширини імпульсів по попередньому фронту за допомогою TIM2.

https://github.com/roman-bessmertnyi/Seventh-semester/tree/master/Subjects/%D0%9F%D0%9C%D0%A1/Timer_Capture