Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

**МЕХАНІКО-ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ**

Кафедра програмування та захисту інформації

**Звіт**

з виконаної лабораторної роботи № 4

дисципліни Програмування вбудованих систем”

на тему

«**Мікроконтролери ARM STM32F4xx, апаратне генерування ШІМ**

.»

Виконав :

студент академічної групи КІ-15

Аннаєв А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Перевірив :

Викладач

Савеленко О.К.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кропивницкий- 2017

**Мета:** Навчитися виконувати апаратне генерування ШІМ в мікроконтролерах ARM STM32F4xx.

**Завдання:**

1) Створити програмне забезпечення для мікроконтролеру, яке керуватиме світлодіодами на основі апаратного генерування ШІМ.

2) Досягти незалежну зміну яскравості кількох світлодіодів.

#include "driverlib/pin\_map.h"

#include <stdint.h>

#include <stdbool.h>

#include "inc/hw\_gpio.h"

#include "inc/hw\_types.h"

#include "inc/hw\_memmap.h"

#include "driverlib/sysctl.h"

#include "driverlib/pin\_map.h"

#include "driverlib/gpio.h"

#include "driverlib/pwm.h"

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#define PWM\_PERIOD 60000

#define PWM\_HALF\_PERIOD 30000

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int main(void)

{

ui32SysClock = SysCtlClockFreqSet((SYSCTL\_XTAL\_25MHZ | SYSCTL\_OSC\_MAIN |

SYSCTL\_USE\_OSC), 25000000);

SysCtlPWMClockSet(SYSCTL\_PWMDIV\_1);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOF);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_PWM0);

GPIOPinConfigure(GPIO\_PF0\_M0PWM0);

GPIOPinTypePWM(GPIO\_PORTF\_BASE, GPIO\_PIN\_0);

PWMGenConfigure(PWM0\_BASE, PWM\_GEN\_0, PWM\_GEN\_MODE\_DOWN | PWM\_GEN\_MODE\_NO\_SYNC);

PWMGenPeriodSet(PWM0\_BASE, PWM\_GEN\_0, PWM\_PERIOD);

PWMPulseWidthSet(PWM0\_BASE, PWM\_OUT\_0,PWM\_PERIOD);

PWMGenEnable(PWM0\_BASE, PWM\_GEN\_0);

PWMOutputState(PWM0\_BASE, PWM\_OUT\_0\_BIT , true);

bool fadeUp = true;

uint32\_t increment = 200, pwmNow = 0;

while(1) {

SysCtlDelay (2);

if (fadeUp) {

pwmNow += increment;

if (pwmNow >= PWM\_PERIOD)

fadeUp = false;

} else {

pwmNow -= increment;

if (pwmNow <= 0)

fadeUp = true;

}

PWMPulseWidthSet(PWM0\_BASE, PWM\_OUT\_0,pwmNow);

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#define PWM\_PERIOD 60000

#define PWM\_HALF\_PERIOD 30000

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

uint32\_t ui32SysClock;

bool fadeUp = true;

uint32\_t increment = 200, pwmNow = 0;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Timer0InterruptHandler(void)

{

TimerIntClear(TIMER0\_BASE, TIMER\_TIMB\_TIMEOUT);

if (fadeUp) {

pwmNow += increment;

if (pwmNow >= PWM\_PERIOD)

fadeUp = false;

} else {

pwmNow -= increment;

if (pwmNow <= 0)

fadeUp = true;

}

PWMPulseWidthSet(PWM0\_BASE, PWM\_OUT\_0,pwmNow);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int main(void)

{

ui32SysClock = SysCtlClockFreqSet((SYSCTL\_XTAL\_25MHZ |

SYSCTL\_OSC\_MAIN |

SYSCTL\_USE\_OSC), 25000000);

SysCtlPWMClockSet(SYSCTL\_PWMDIV\_1);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOF);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_PWM0);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_TIMER0);

TimerConfigure(TIMER0\_BASE, TIMER\_CFG\_SPLIT\_PAIR | TIMER\_CFG\_B\_PERIODIC);

TimerLoadSet(TIMER0\_BASE, TIMER\_B, ui32SysClock / 1000);

IntMasterEnable();

TimerIntEnable(TIMER0\_BASE, TIMER\_TIMB\_TIMEOUT);

IntEnable(INT\_TIMER0B);

TimerEnable(TIMER0\_BASE, TIMER\_B);

GPIOPinConfigure(GPIO\_PF0\_M0PWM0);

GPIOPinTypePWM(GPIO\_PORTF\_BASE, GPIO\_PIN\_0);

PWMGenConfigure(PWM0\_BASE, PWM\_GEN\_0, PWM\_GEN\_MODE\_DOWN | PWM\_GEN\_MODE\_NO\_SYNC);

PWMGenPeriodSet(PWM0\_BASE, PWM\_GEN\_0, PWM\_PERIOD);

PWMPulseWidthSet(PWM0\_BASE, PWM\_OUT\_0,PWM\_PERIOD);

PWMGenEnable(PWM0\_BASE, PWM\_GEN\_0);

PWMOutputState(PWM0\_BASE, PWM\_OUT\_0\_BIT , true);

while(1)

SysCtlDelay (2);

}