Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

**МЕХАНІКО-ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ**

Кафедра кібербезпеки та програмного забезпечення

**Звіт**

з виконаної лабораторної роботи № 3

дисципліни Програмування вбудованих систем”

на тему

«**Мікроконтролери ARM STM32F4xx, використання переривань**

.»

Виконав :

студент академічної групи КІ-15

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

Перевірив :

Викладач

Дрєєв О.М.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кропивницкий- 2017

ЛАБОЛАТОРНА РОБОТА №3

Тема: **Мікроконтролери ARM STM32F4xx, використання переривань**

Мета:

1) Створити програмне забезпечення для мікроконтролеру, яке керуватиме світлодіодами на основі переривань від таймерів.

2) Досягти незалежну зміну яскравості кількох світлодіодів.

**Хід виконання:**

include <stdint.h>

#include <stdbool.h>

#include "driverlib/gpio.h"

#include "driverlib/sysctl.h"

#include "driverlib/systick.h"

#include "inc/hw\_memmap.h"

uint32\_t ligthMaxLevel = 20;

uint32\_t lightLevel = 20;

uint32\_t ligthCounter1 = 0;

uint32\_t ligthCounter2 = 0;

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

void TimerInterruptHnadler(void)

{

if(TimerGetStatus(TIMER0\_BASE, TIMER\_TIMB\_TIMEOUT) == TIMER\_TIMB\_TIMEOUT)

{

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

if(ligthCounter1 < lightLevel)

{

GPIOPinWrite(GPIO\_PORTN\_BASE, GPIO\_PIN\_0, GPIO\_PIN\_0);

}

else

{

GPIOPinWrite(GPIO\_PORTN\_BASE, GPIO\_PIN\_0, 0x0);

}

if(lightCounter1 == ligthMaxLevel)

lightCounter1 = 0;

lightCounter1++;

}

else

if(TimerGetStatus(TIMER1\_BASE, TIMER\_TIMB\_TIMEOUT) == TIMER\_TIMB\_TIMEOUT)

{

TimerIntClear(TIMER1\_BASE, TIMER\_TIMA\_TIMEOUT);

if(ligthCounter2 < lightLevel)

{

GPIOPinWrite(GPIO\_PORTN\_BASE, GPIO\_PIN\_1, GPIO\_PIN\_1);

}

else

{

GPIOPinWrite(GPIO\_PORTN\_BASE, GPIO\_PIN\_1, 0x0);

}

if(ligthCounter2 == ligthMaxLevel)

ligthCounter2 = 0;

ligthCounter2++;

}

}

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

int main(void)

{

uint32\_t g\_ui32SysClock = SysCtlClockFreqSet((SYSCTL\_XTAL\_25MHZ |

SYSCTL\_OSC\_MAIN |

SYSCTL\_USE\_PLL |

SYSCTL\_CFG\_VCO\_480), 120000000);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_TIMER0);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_TIMER1);

TimerLoadSet(TIMER0\_BASE, TIMER\_B, g\_ui32SysClock / 1000);

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

IntEnable(INT\_TIMER0B);

TimerEnable(TIMER0\_BASE, TIMER\_B);

TimerLoadSet(TIMER1\_BASE, TIMER\_B, g\_ui32SysClock / 1000);

IntMasterEnable();

TimerIntEnable(TIMER1\_BASE, TIMER\_TIMB\_TIMEOUT);

IntEnable(INT\_TIMER1B);

TimerEnable(TIMER1\_BASE, TIMER\_B);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPION);

GPIOPinTypeGPIOOutput(GPIO\_PORTN\_BASE, GPIO\_PIN\_0, GPIO\_PIN\_1);

while(1)

{

}

}

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