Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

**МЕХАНІКО-ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ**

Кафедра програмування та захисту інформації

**Звіт**

з виконаної лабораторної роботи № 2

дисципліни Програмування вбудованих систем”

на тему

«**Мікроконтролери ARM STM32F4xx, керування напругою**

**на виходах мікроконтролеру, вимірювання часу**»

Виконав :

студент академічної групи КІ-15

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

Перевірив :

Викладач

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

Кропивницкий- 2017

ЛАБОЛАТОРНА РОБОТА №2

Тема: **Мікроконтролери ARM STM32F4xx, керування напругою на виходах мікроконтролеру, вимірювання часу**

Мета:

1) Створити програмне забезпечення для мікроконтролеру, яке керуватиме світлодіодами.

2) Організувати задані інтервали часу за допомогою таймерів.

3) Досягти плавну зміну яскравості світлодіоду.

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

#include <stdint.h>

#include <stdbool.h>

#include "inc/hw\_ints.h"

#include "inc/hw\_memmap.h"

#include "inc/hw\_types.h"

#include "driverlib/interrupt.h"

#include "driverlib/pin\_map.h"

#include "driverlib/sysctl.h"

#include "driverlib/timer.h"

bool pause = true;

void Timer0IntHandler(void)

{

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

pause = false;

}

void Delay(volatile uint32\_t count)

{

TimerLoadSet(TIMER0\_BASE, TIMER\_A, count/10);

while(pause) { }

}

int main(void)

{

// Конфигурация тактирования

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

SYSCTL\_USE\_PLL| SYSCTL\_CFG\_VCO\_480), 120000000);

//

// активация модуля Gpio порта N

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPION);

//

// конфигурирование пина на выход

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

//

// активация таймера 0

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_TIMER0);

//

// конфигурирование таймера на режим периодического срабатывания

TimerConfigure(TIMER0\_BASE, TIMER\_CFG\_PERIODIC);

//

// загрузка начального значения для таймера

TimerLoadSet(TIMER0\_BASE, TIMER\_A, g\_ui32SysClock);

//

// активация прерывания

IntEnable(INT\_TIMER0A);

//

//прерывания таймера разрешено

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

// включение таймера

TimerEnable(TIMER0\_BASE, TIMER\_A);

// Бесконечный цикл

//

uint32\_t r = 1000;

uint32\_t counter = r;

while(1)

{

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

Delay(counter);

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

Delay(r- counter);

counter-=100;

if(counter == 0)

counter = r;

}

}