Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

**МЕХАНІКО-ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ**

Кафедра програмування та захисту інформації

**Звіт**

з виконаної лабораторної роботи № 5

дисципліни Програмування вбудованих систем”

на тему

«**Мікроконтролери ARM STM32F4xx, використання**

**цифро-аналогового перетворювача напруги — АЦП**

.»

Виконав :

студент академічної групи КІ-15

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

Перевірив :

Викладач

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

Кропивницкий- 2017

ЛАБОЛАТОРНА РОБОТА №5

Тема:

**Мікроконтролери ARM STM32F4xx, використання цифро-аналогового перетворювача напруги — АЦП**

Мета:

1) Створити програмне забезпечення для мікроконтролеру, яке керуватиме світлодіодами на основі виміряної напруги.

2) Час світіння та вимкнення світлодіоду повинне регулюватися з різних каналів АЦП.

**Хід виконання:**

#include <stdbool.h>

#include <stdint.h>

#include "inc/hw\_memmap.h"

#include "driverlib/adc.h"

#include "driverlib/gpio.h"

#include "driverlib/pin\_map.h"

#include "driverlib/sysctl.h"

#include "driverlib/uart.h"

#include "utils/uartstdio.h"

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

#define SEQUENCY\_NUM 0

#define STEP 0

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

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

void SetChannelForADC(uint32\_t ADC\_BASE ,uint32\_t ui32SequenceNum)

{

ADCSequenceEnable(ADC\_BASE, ui32SequenceNum);

ADCSequenceConfigure(ADC\_BASE, STEP, ADC\_TRIGGER\_PROCESSOR, 0);

ADCSequenceStepConfigure(ADC\_BASE, ui32SequenceNum, STEP, ADC\_CTL\_IE |

ADC\_CTL\_END | ADC\_CTL\_CH0);

ADCSequenceEnable(ADC\_BASE, ui32SequenceNum);

ADCProcessorTrigger(ADC\_BASE, ui32SequenceNum);

}

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

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

void GetDataFromADCChannel(uint32\_t ADC\_BASE ,uint32\_t ui32SequenceNum)

{

uint32\_t value;

ADCProcessorTrigger(ADC\_BASE, ui32SequenceNum);

while(!ADCIntStatus(ADC\_BASE, ui32SequenceNum, false)) { }

ADCIntClear(ADC\_BASE, ui32SequenceNum);

ADCSequenceDataGet(ADC\_BASE, ui32SequenceNum, &value);

return value;

}

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

int main(void)

{

uint32\_t ui32SysClock = SysCtlClockFreqSet((SYSCTL\_XTAL\_25MHZ | SYSCTL\_OSC\_MAIN |

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

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_ADC0);

ADCSequenceConfigure(ADC0\_BASE, SEQUENCY\_NUM, ADC\_TRIGGER\_PROCESSOR, 0);

ADCSequenceStepConfigure(ADC0\_BASE, SEQUENCY\_NUM, STEP,ADC\_CTL\_TS|ADC\_CTL\_IE|

ADC\_CTL\_END);

ADCSequenceEnable(ADC0\_BASE, SEQUENCY\_NUM);

ADCIntClear(ADC0\_BASE, SEQUENCY\_NUM);

uint32\_t pui32ADC0Value[1];

while(1)

{

ADCProcessorTrigger(ADC0\_BASE, SEQUENCY\_NUM);

while(!ADCIntStatus(ADC0\_BASE, SEQUENCY\_NUM, false)) { }

ADCIntClear(ADC0\_BASE, SEQUENCY\_NUM);

ADCSequenceDataGet(ADC0\_BASE, SEQUENCY\_NUM, pui32ADC0Value);

//

// TODO: после получения значения обрабатываем

SysCtlDelay(20);

}

}