

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

**ІКНІ**  
**Кафедра ПЗ**

**ЗВІТ**

до лабораторної роботи № 6  
з дисципліни: *“Основи програмування вбудованих систем”*  
на тему: *“Робота з АЦП ”*

**Лектор:**  
доц. каф. ПЗ  
Марусенкова Т. А.

**Виконав:**  
ст. гр. ПЗ-32  
Бурець В.В.

**Прийняв:**  
доц. каф. ПЗ  
Крук О.Г.

« \_\_\_\_ » \_\_\_\_\_ 2021 р.

$\Sigma$  = \_\_\_\_ .....

**Тема роботи:** Робота з АЦП.

**Мета роботи:** Ознайомитися на практиці з параметрами АЦП у середовищі Keil uVision.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

1. У чому полягає дискретизація за часом та квантування за рівнем?

Дискретизація за часом це зчитування даних у певні часові періоди. Тобто ми отримуємо послідовність значень через кратні інтервали часу отриманих із аналогового сигналу.

Квантування за рівнем це заміна величини дискретизованого сигналу найближчим значенням з набору фіксованих величин

## ЗАВДАННЯ

Створити у Keil uVision проект, в якому при температурі 15°C і вище засвічується червоний світлодіод, інакше – синій. Створити різні температурні умови і перевірити роботу проекту.

## ХІД РОБОТИ

Для реалізації завдання я виконав наступні дії:

1. Сконфігурував порт GPIO аналогічно до попередньої роботи: порт D – для роботи зі світлодіодами. Відповідні налаштування реалізував функцією GPIOInit().
2. Сконфігурував порт ADC1 для роботи з АЦП. Відповідні налаштування реалізував функцією adc\_Init().

## КОД ПРОГРАМИ

```
void adc_init() {  
    ADC_InitTypeDef ADC_InitStructure;  
    ADC_CommonInitTypeDef adc_init;  
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_ADC1, ENABLE);  
    ADC_DeInit();  
    ADC_StructInit(&ADC_InitStructure);  
    adc_init.ADC_Mode = ADC_Mode_Independent;  
    //частота АЦП  
    adc_init.ADC_Prescaler = ADC_Prescaler_Div2;
```

```

//багатоканальний чи одноканальний режим вимірювання(одноканальний)
    ADC_InitStructure.ADC_ScanConvMode = DISABLE;
//Вимірювання проводити багаторазово чи одноразово(одноразово)
    ADC_InitStructure.ADC_ContinuousConvMode = DISABLE;
    ADC_InitStructure.ADC_ExternalTrigConv = ADC_ExternalTrigConvEdge_None;
//Вимірювання даних АЦП
    ADC_InitStructure.ADC_DataAlign = ADC_DataAlign_Right;
//розмір
    ADC_InitStructure.ADC_Resolution = ADC_Resolution_12b;
    ADC_CommonInit(&adc_init);
    ADC_Init(ADC1, &ADC_InitStructure);
    ADC_Cmd(ADC1, ENABLE);
}
u16 readADC1(u8 channel) {
//налаштовує вибраний канал ADC
ADC-RegularChannelConfig(ADC1, channel, 1, ADC_SampleTime_3Cycles);
//почати передачу даних
ADC_SoftwareStartConv(ADC1);
// Очікування на зчитування даних поки статус лінії ADC1 не стане RESET
while (ADC_GetFlagStatus(ADC1, ADC_FLAG_EOC) == RESET);
    return ADC_GetConversionValue(ADC1);
}
int main(void){
    GPIOInit();
    adc_init();
    while(1){
        unsigned int CODEres = readADC1(ADC_Channel_1);
        double Vref = 5.0;
        double CODEmax = 0xFFFF,
        //напруга живлення.повернутий код. Максимальний допустимий код
        double Vres = Vref*CODEres/CODEmax;
        double temperature = (Vres - 0.76)/0.0025 + 25
    }
}

```

```

        if(temperature > 15)
        {
            GPIO_ToggleBits(GPIOD, GPIO_Pin_14); //red
        }else{
            GPIO_ToggleBits(GPIOD, GPIO_Pin_15); //blue
        }
    }
}
}

```

## РЕЗУЛЬТАТИ

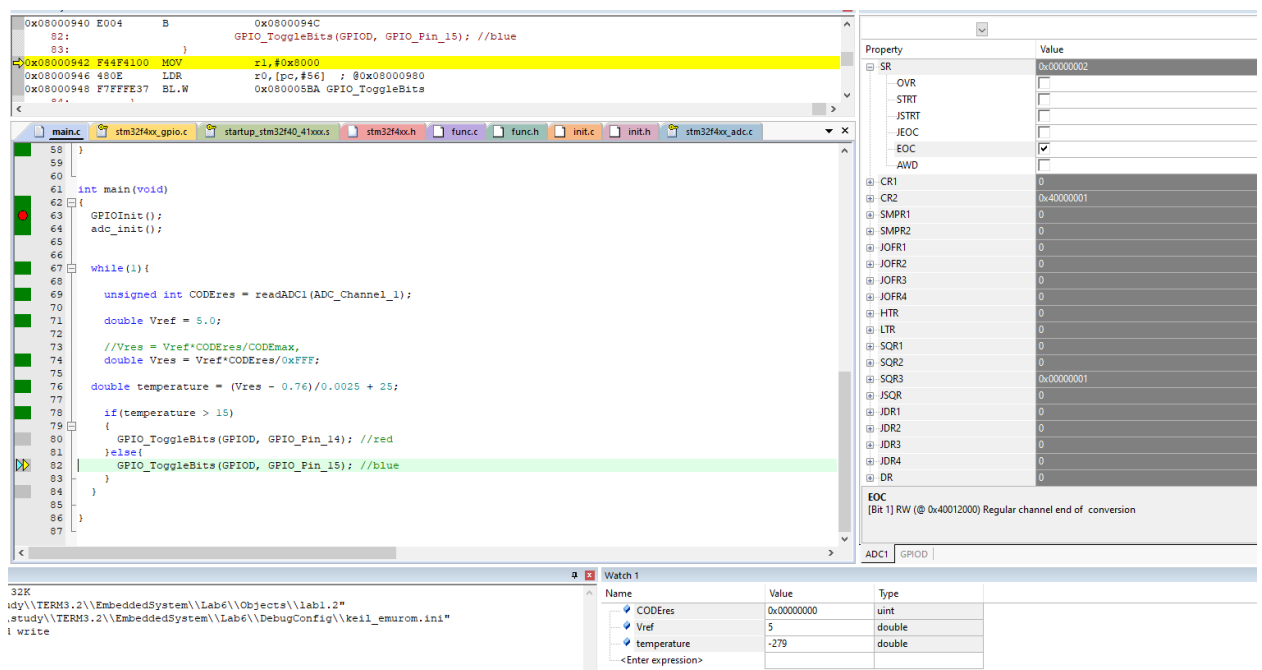


Рис.1. Відлагодження роботи програми

## ВИСНОВКИ

Виконуючи цю лабораторну роботу, я ознайомився та навчився керувати АЦП, ознайомився із параметрами АЦП, навчився ініціалізувати АЦП на платі STM32F4DISCOVERY. Написав програму для роботи із АЦП, температурним датчиком який присутній на платі STM32F4DISCOVERY та світлодіодами. Програма призначена для зчитування даних із температурного датчика, та подальшої роботи з ними.