

Разработка схемы компьютерного моделирования и отладка сквозного тракта передачи сигнала от входа до выхода

В рамках данного этапа нужно реализовать сквозную передачу сигнала. Это значит, что нужно подать сигнал на вход АЦП, провести его через МК и затем вывести из ЦАП. На рисунке 1 схема соединённых вместе устройств.

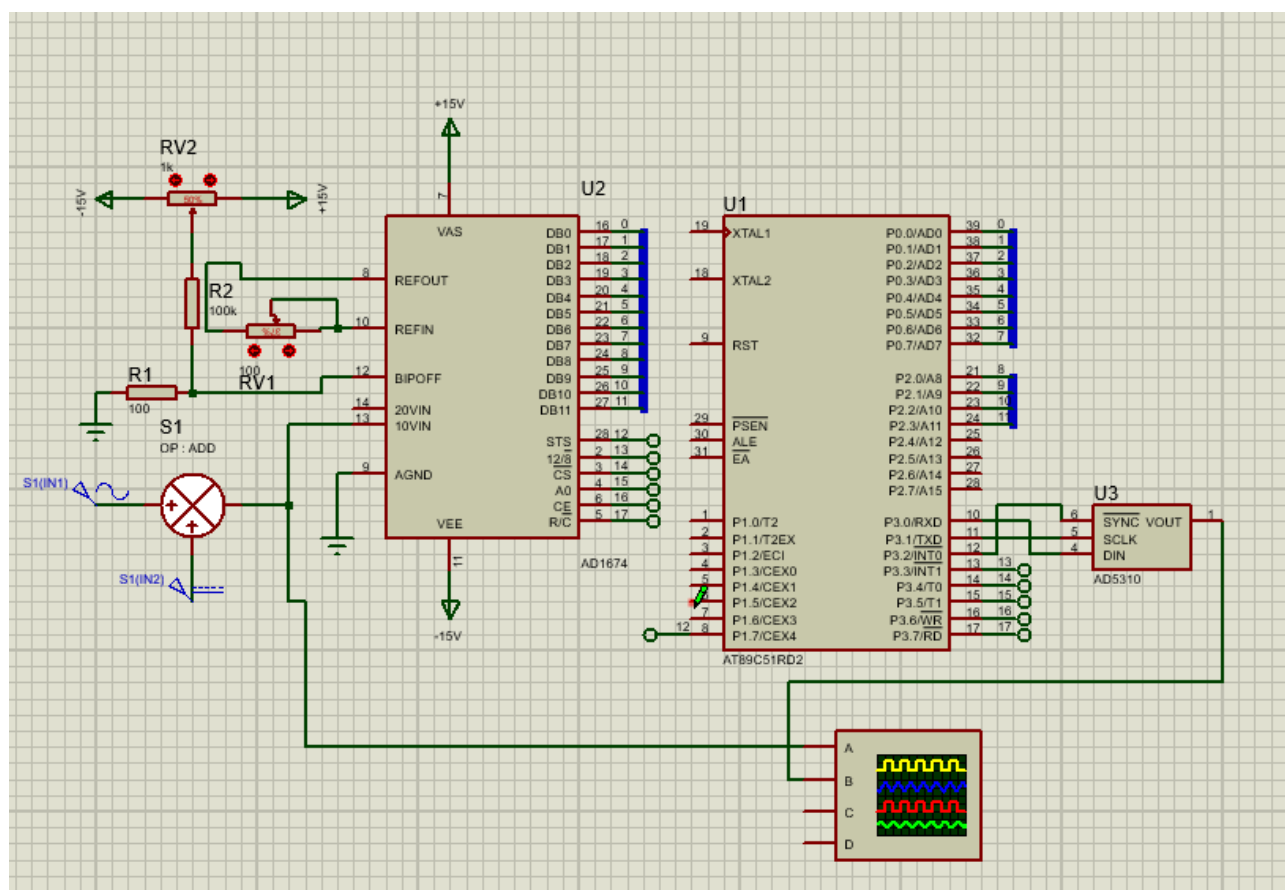


Рисунок 1 – Схема сквозного тракта АЦП – ЦАП

Код составлен из частей 8-ого и 9-ого этапов, а значит мы не будем заострять на нём внимание. Его можно посмотреть в приложении 1.

Чтобы протестировать схему были поданы синусоидальный и импульсный сигналы, объединённые с постоянным сигналом, чтобы приподнять их над нулём(наш АЦП работает в униполярном режиме). Синусоидальный сигнал имеет период 300 Гц и изменяется в диапазоне от 0 до 10В. Импульсный сигнал имеет те же характеристики. На выходе получается слегка сдвинутый сигнал той же частоты изменяющийся в диапазоне от 0 до 5В.

Разница обусловлена диапазонами работы АЦП и ЦАП. Результат работы схемы с синусоидальным сигналом показан на рисунке 1, а на рисунке 2 – с импульсным.

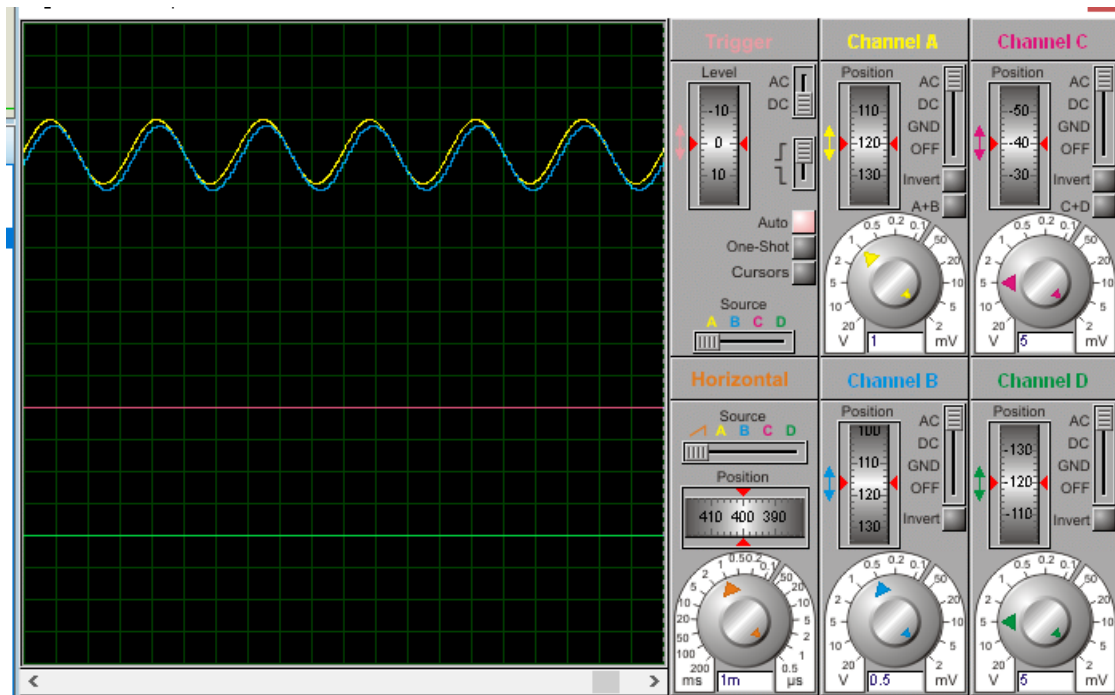


Рисунок 2 – Результат работы схемы с синусоидальным сигналом.
Жёлтый – входной, синий – выходной.

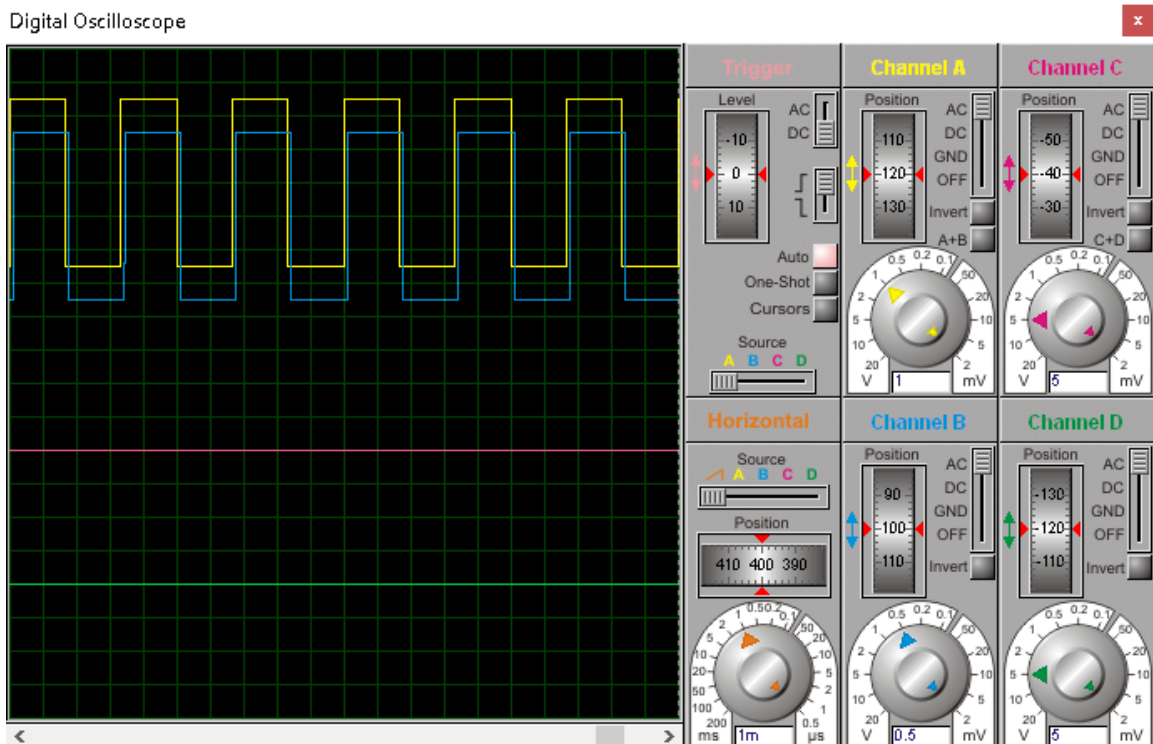


Рисунок 3 – Результат работы схемы с импульсным сигналом. Жёлтый – входной, синий – выходной.

ПРИЛОЖЕНИЕ 1

```
#include <at89c51xd2.h>

#define bit_12_8 P3_3
#define STS      P1_7
#define CS       P3_4
#define A0       P3_5
#define CE       P3_6
#define R_C      P3_7
#define sync     P3_2

#define extract_first_byte(value) (value & 0xF) << 4
#define extract_second_byte(value) value >> 4

unsigned int reverse(unsigned int value)
{
    unsigned int result = 0;
    if ( value & 0x200 ) result |= 0x001;
    if ( value & 0x100 ) result |= 0x002;
    if ( value & 0x080 ) result |= 0x004;
    if ( value & 0x040 ) result |= 0x008;
    if ( value & 0x020 ) result |= 0x010;
    if ( value & 0x010 ) result |= 0x020;
    if ( value & 0x008 ) result |= 0x040;
    if ( value & 0x004 ) result |= 0x080;
    if ( value & 0x002 ) result |= 0x100;
    if ( value & 0x001 ) result |= 0x200;
    return result;
}

void shift_transmit(unsigned int data_to_transmit){
    char first_byte;
    char second_byte;

    data_to_transmit = reverse(data_to_transmit);

    first_byte = extract_first_byte(data_to_transmit);
    second_byte = extract_second_byte(data_to_transmit);

    sync = 0;
    SBUF = first_byte;
    while(TI==0);
    TI = 0; TI = 0;
```

```

        SBUF = second_byte;
        while(TI==0);
        TI = 0; TI = 0;
        sync = 1;
    }

void Delay(int nCount){
    while(nCount--);
}

unsigned int ad1674_read(void){
    R_C = 0;
    while(STS==1);
    R_C = 1; R_C = 1;
    return P0 | ((P2 & 0xF) << 8);
}

void main (void)
{
    unsigned int adc_data;

    bit_12_8 = 1; CS = 0; A0 = 0; CE = 1;

    while(1){
        adc_data = ad1674_read() >> 2;
        shift_transmit(adc_data);
    }
}

```