**Отчет по лабораторной работе**

**«*АЦП прямого счёта*»**

Вариант ХХХ

Исполнитель: Преподаватель:

Нестеров Ю.Г.

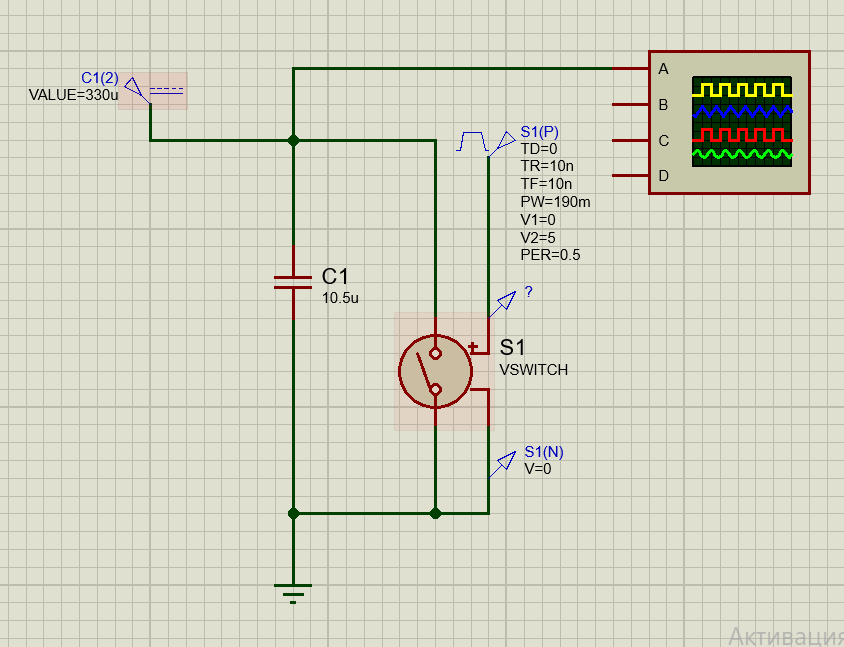
Москва, МГТУ – 2022

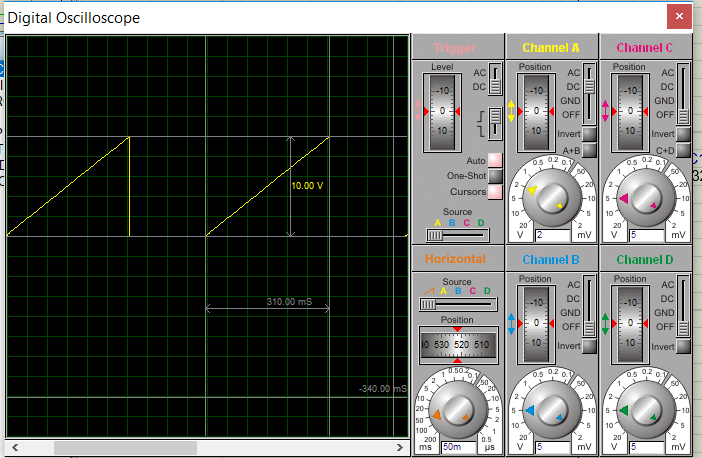
**Вариант Х**

Значения ИД из ЛР 2:

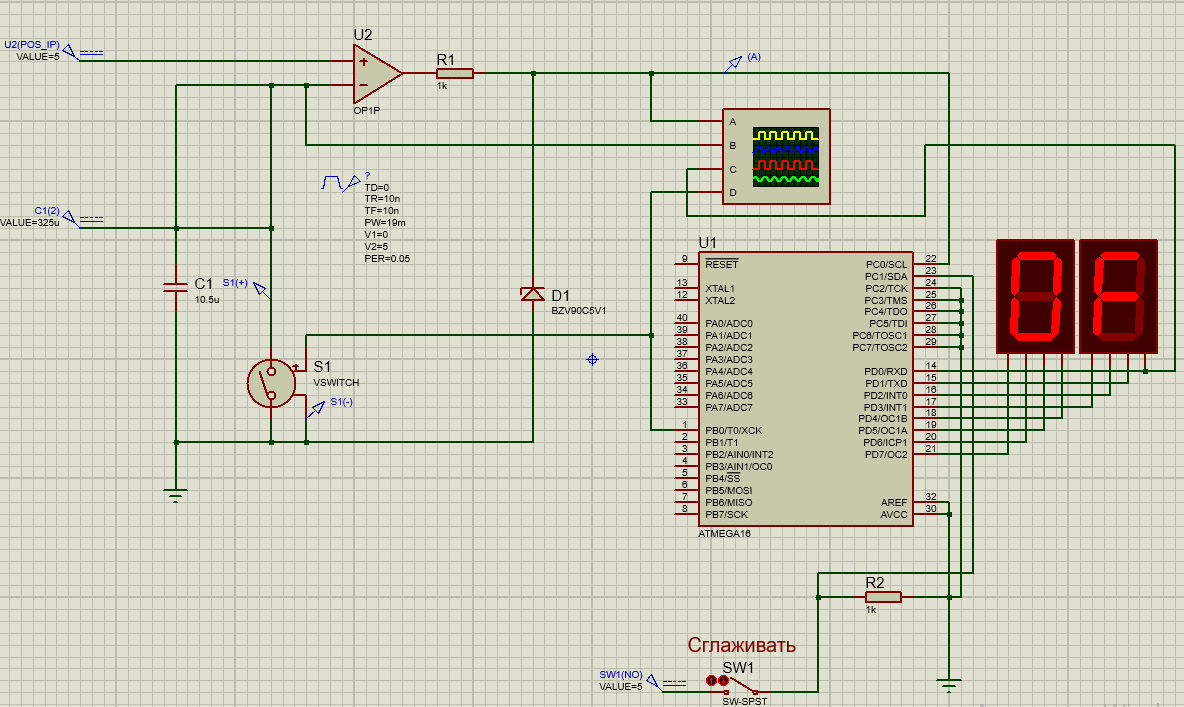
Случайным образом взято значение:

1. **Схема заряда и разряда конденсатора через электронный ключ**

****

****

1. **Полная схема АЦП прямого счёта**

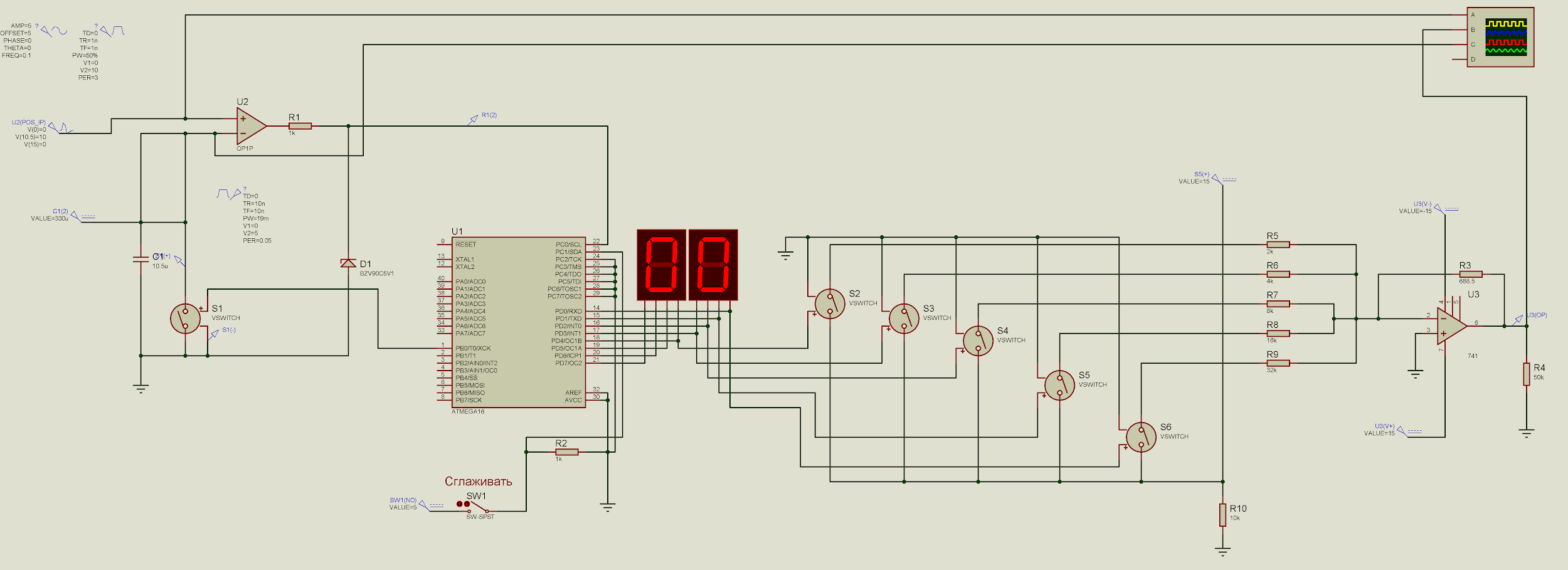
****

1. **Таблица зависимости выходного кода АЦП от уровня постоянного входного напряжения.**

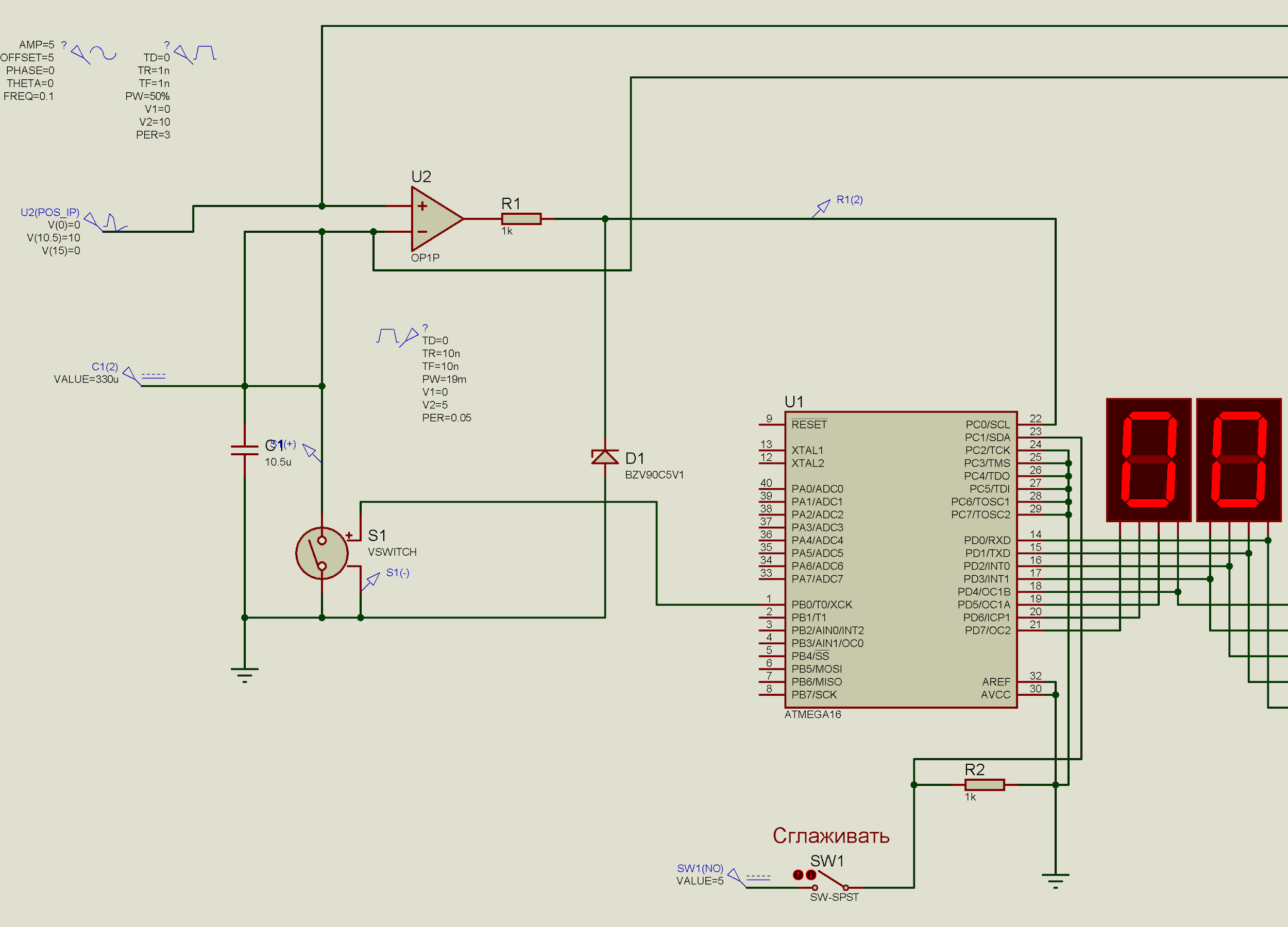
|  |  |
| --- | --- |
| Uвх, В | X |
| 0 | 0 |
| 0,3225 | 1 |
| 0,645 | 2 |
| 0,9675 | 3 |
| 1,29 | 4 |
| 1,6125 | 5 |
| 1,935 | 6 |
| 2,2575 | 7 |
| 2,58 | 8 |
| 2,9025 | 9 |
| 3,225 | 10 |
| 3,5475 | 11 |
| 3,87 | 12 |
| 4,1925 | 13 |
| 4,515 | 14 |
| 4,8375 | 15 |
| 5,16 | 16 |
| 5,4825 | 17 |
| 5,805 | 18 |
| 6,1275 | 19 |
| 6,45 | 20 |
| 6,7725 | 21 |
| 7,095 | 22 |
| 7,4175 | 23 |
| 7,74 | 24 |
| 8,0625 | 25 |
| 8,385 | 26 |
| 8,7075 | 27 |
| 9,03 | 28 |
| 9,3525 | 29 |
| 9,675 | 30 |
| 10 | 31 |

****

1. **Схема АЦП с ЦАП, подключенным к выходу**

****

**4б. Только АЦП, крупнее**

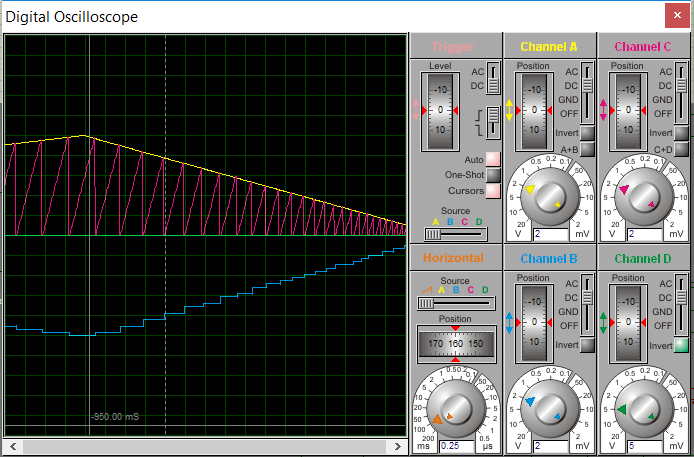
****

1. **Вид сигнала**

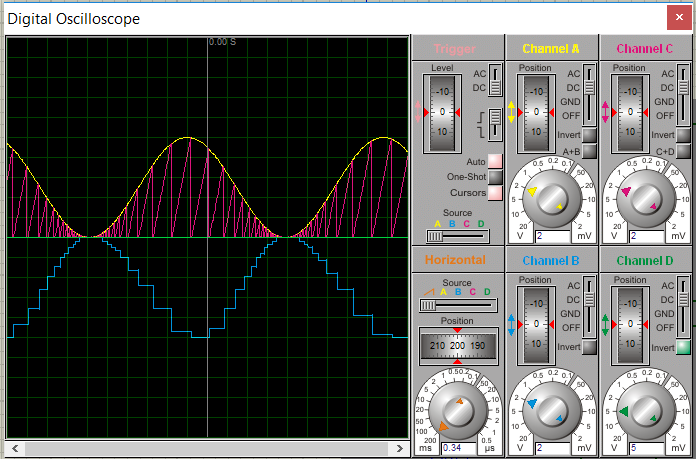
Жёлтый - генератор линейного напряжения,

красный - работа АЦП,

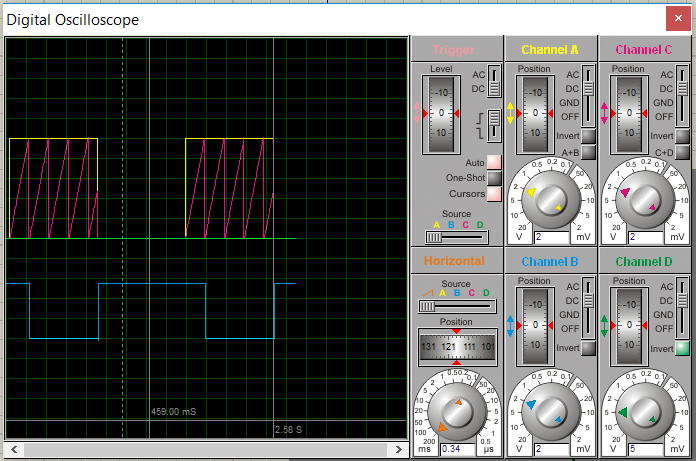
голубой - результат работы ЦАП.

****

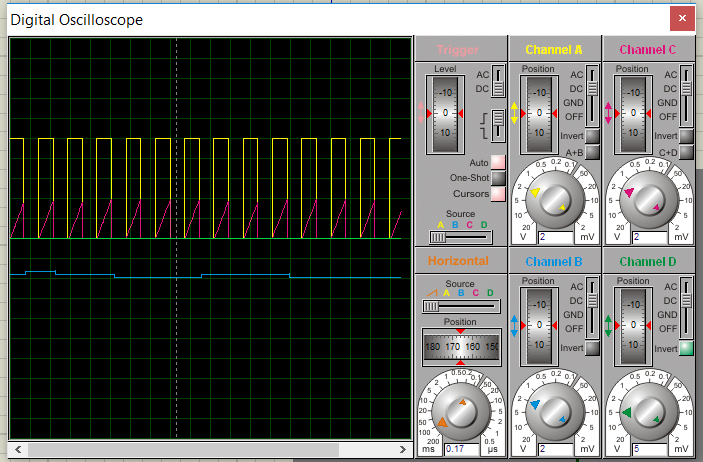
Аналогично для синусоидального генератора:



Меандр:



А вот для меандра высокой частоты АЦП ведёт себя неправильно:



Это объясняется тем, что “красный” сигнал не успевает дойти до уровня сигнала, и АЦП показывает неправильные данные. Таким образом, АЦП прямого счёта очень точный, но его лучше не использовать там, где нужно работать на высоких частотах.

1. **Исходный код программы микроконтроллера:**

**/\***

**\* Lab5.c**

**\***

**\* Created: 16.11.2019 21:10:09**

**\* Author : Igor Vodka**

**\*/**

**#include <avr/io.h>**

**#define bool int**

**#define true 1**

**#define false 0**

**void launchCapacitor() {**

**PORTB = 1;**

**}**

**void stopCapacitor() {**

**PORTB = 0;**

**}**

**bool isComparatorOn() {**

**return PINC & 1;**

**}**

**bool isSmoothingOn() {**

**return PINC & 2;**

**}**

**void output(int value) {**

**/\***

**Не подходит для схемы АЦП+ЦАП**

**int tens = value / 10;**

**int ones = value - (tens) \* 10;**

**PORTD = tens \* 16 + ones;**

**\*/**

**PORTD = value;**

**}**

**int abs(int a) {**

**return a > 0 ? a : -a;**

**}**

**// сглаживание скачущих значений**

**int previousValues[] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };**

**int totalPreviousValues = 10;**

**int currentValueIndex = 0;**

**int rememberAndGetAverage(int newValue) {**

**previousValues[currentValueIndex++] = newValue;**

**if (currentValueIndex == totalPreviousValues) {**

**currentValueIndex = 0;**

**}**

**int sum = 0;**

**for (int i = 0; i < totalPreviousValues; i++) {**

**sum += previousValues[i];**

**}**

**int averageValue = sum / totalPreviousValues;**

**return averageValue;**

**}**

**int main(void)**

**{**

**int time = 0;**

**int result = 0;**

**int megaticks = 0;**

**bool isInTimerLoop = false;**

**// Используем порт 1 как выход на VSWITCH, остальное - на вход**

**DDRB = 0x01;**

**// А все порты D используем как выходы для дисплея**

**DDRD = 0xFF;**

**// Предделение**

**TCCR0 = 0b011;**

**// Калибровка таймера**

**// OSCCAL = 5;**

**while (1)**

**{**

**if ((!isInTimerLoop && TCNT0 == 160) || (isInTimerLoop && TCNT0 == 5)) {**

**megaticks++;**

**TCNT0 = 0;**

**}**

**if (!isInTimerLoop) {**

**if (isComparatorOn()) {**

**result = megaticks;**

**} else {**

**time = 40;**

**launchCapacitor();**

**isInTimerLoop = true;**

**}**

**}**

**if (isInTimerLoop) {**

**if (megaticks == time) {**

**megaticks = 0;**

**stopCapacitor();**

**if (isSmoothingOn()) {**

**output(rememberAndGetAverage(result));**

**} else {**

**output(result);**

**}**

**isInTimerLoop = false;**

**}**

**}**

**}**

**}**