МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»

(БГТУ им. В.Г.Шухова)

**Лабораторная работа №3\_2**

дисциплина «Программирование микроконтроллеров»

тема: «Реализация АЦП последовательного приближения»

Выполнил:

студент группы ВТ-31

Новожен Н.В.

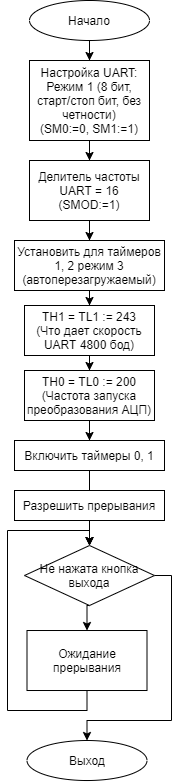
Проверил: Кижук А.С.

Белгород 2020

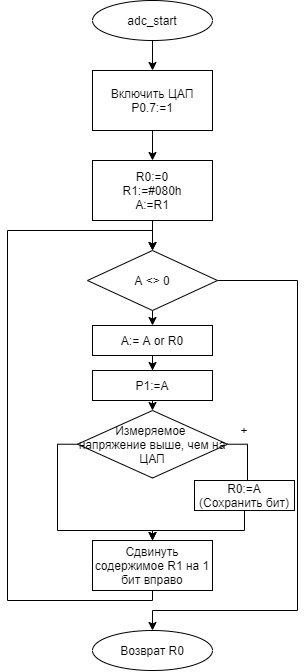
**Цель работы:** изучить принцип работы АЦП последовательного приближения, и реализовать заданный алгоритм на микроконтроллере 8051.

**Задание к работе:** Разработать программу, выполняющую преобразование уровнь напряжений 0-5 вольт в числовые значения. Числовые значения вывести на UART. Программа должна использовать ЦАП и компаратор, входящие в симулятор EdSim51DI.

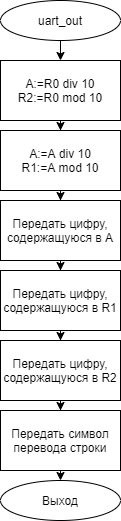
**Блок-схема разработанной программы**

****

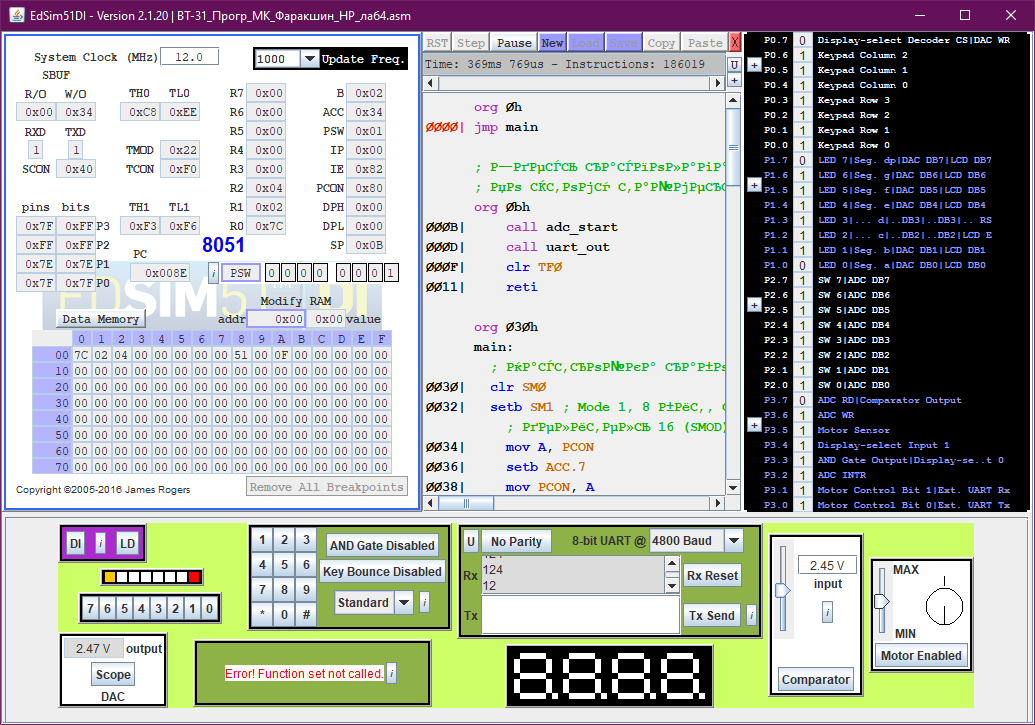
**Алгоритм работы АЦП**

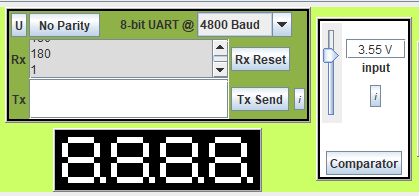
****

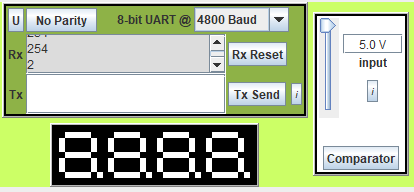
**Функция вывода значения на экран**

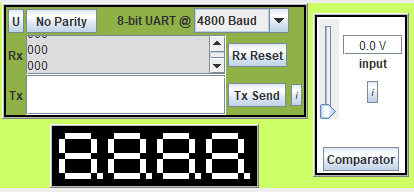


**Результаты работы программы на симуляторе EdSim51DI:**









**Код программы**

org 0h

jmp main

; Здесь располагается прерывание от таймера 0

; По этому таймеру запускается АЦП и по завершении его работы результат выводится в UART

org 0bh

call adc\_start

call uart\_out

clr TF0

reti

org 030h

main:

; Настройка работы UART

clr SM0

setb SM1 ; Mode 1, 8 бит, стоп/старт

; делитель 16 (SMOD)

mov A, PCON

setb ACC.7

mov PCON, A

; Настраиваем таймер

; Режим работы таймера: M1 = 1, M0 = 0 (3) для таймера 0, 1

; Автоперезагружаемый 8 бит таймер

mov TMOD, #22h

; Настройка таймера

mov TH0, #200

mov TL0, #200

; Чтобы установить 4800 бод, в таймер надо записать 243

; TH1 = 256 - ((system frequency / (12 \* 16)) / baud)

mov TH1, #243

mov TL1, #243

; Настройка прерываний

setb ET0 ; Прерывание от таймера 0

setb TR0 ; Включить таймер 0, 1

setb TR1

setb EA ; разрешить все прерывания

sjmp $

adc\_start:

; Функция преобразования аналогового сигнала в цифровой (АЦП)

; Используется метод последовательного приближения

; Включаем ЦАП

clr P0.7

; Начальное возможное напряжение

mov R0, #0h ; 0000 0000

mov R1, #080h ; 1000 0000

mov A, R1

loop\_adc:

orl A, R0 ; Примеряем текущий вариант + очередной бит

mov P1, A

jnb P3.7, continue ; Если компаратор выдал 1 (входное напряжение выше чем на ЦАП), сохраняем бит

mov R0, A

continue:

mov A, R1 ; Передвигаем проверяемый бит

rr A

clr ACC.0 ; Если сдвинут A = 1, то крайний бит перенесется влево. Избавляемся от этого эффекта

mov R1, A

jnz loop\_adc

ret

uart\_out:

; Функция служит для вывода на UART содержимого регистра R0

mov A, R0

mov B, #10

div AB

mov R2, B ; разряд 3

mov B, #10

div AB

mov R1, B ; разряд 2

add A, #'0'

mov SBUF, A ; разряд 1 передан

jnb TI, $

clr TI

mov A, R1

add A, #'0'

mov SBUF, A ; разряд 2 передан

jnb TI, $

clr TI

mov A, R2

add A, #'0' ; разряд 3 передан

mov SBUF, A

jnb TI, $ ; ожидание пока установится бит TI

clr TI

mov SBUF, #'\r'

jnb TI, $

clr TI

mov SBUF, #'\n'

jnb TI, $

clr TI

ret