**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ **«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. В. Г. ШУХОВА»**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**ЛАБОРАТОРНАЯ РАБОТА №3.2**

**Дисциплина: Программирование микроконтроллеров**

**Тема: АЦП последовательного приближения**

Выполнил: ст. группы ВТ-31

Подкопаев Антон Валерьевич

Проверил: ст. пр. кафедры ТК

Гольцов Юрий Александрович

**Белгород 2020**

**Цель работы:** изучить принцип работы АЦП последовательного приближения и написать программу для его тестирования.

Задание к работе:

Создать программу для генерации сигнала на осциллографе через АЦП последовательного приближения.

**Код программы**

; This program reads the analogue input

; voltage on the ADC and displays it on

; the scope via the DAC.

; A sample is taken from the ADC every 50 us

; This is achieved by setting timer 0

; to interrupt the main program every 50 us.

; The timer 0 ISR then initiates an ADC

; conversion.

; When the conversion is complete the

; ADC interrupt line goes low. This line

; is interfaced with the 8051 external 0

; interrupt line. The external 0 ISR

; therefore takes the reading from the ADC

; on P2 and passes it to the DAC on P1.

; Therefore, while the program is running,

; the scope voltage level should be the

; same as the ADC input voltage.

; However, when a change is made to the

; ADC input voltage it will take some time

; for the scope to update (ie; until the

; next timer 0 interrupt).

; Note: when running this program make sure

; the ADC is enabled (not the comparator).

ORG 0 ; reset vector

JMP main ; jump to the main program

ORG 3 ; external 0 interrupt vector

JMP ext0ISR ; jump to the external 0 ISR

ORG 0BH ; timer 0 interrupt vector

JMP timer0ISR ; jump to timer 0 ISR

ORG 30H ; main program starts here

main:

SETB IT0 ; set external 0 interrupt as edge-activated

SETB EX0 ; enable external 0 interrupt

CLR P0.7 ; enable DAC WR line

MOV TMOD, #2 ; set timer 0 as 8-bit auto-reload interval timer

MOV TH0, #-50 ; | put -50 into timer 0 high-byte - this reload value,

; | with system clock of 12 MHz, will result in a timer 0 overflow every 50 us

MOV TL0, #-50 ; | put the same value in the low byte to ensure the timer starts counting from

; | 236 (256 - 50) rather than 0

SETB TR0 ; start timer 0

SETB ET0 ; enable timer 0 interrupt

SETB EA ; set the global interrupt enable bit

JMP $ ; jump back to the same line (ie; do nothing)

; end of main program

; timer 0 ISR - simply starts an ADC conversion

timer0ISR:

CLR P3.6 ; clear ADC WR line

SETB P3.6 ; then set it - this results in the required positive edge to start a conversion

RETI ; return from interrupt

; external 0 ISR - responds to the ADC conversion complete interrupt

ext0ISR:

CLR P3.7 ; clear the ADC RD line - this enables the data lines

MOV P1, P2 ; take the data from the ADC on P2 and send it to the DAC data lines on P1

SETB P3.7 ; disable the ADC data lines by setting RD

RETI ; return from interrupt

**Результаты работы программы на симуляторе EdSim51DI**

Изображение выглядит как снимок экрана, внутренний, монитор, компьютер

Автоматически созданное описание