

1. Цель работы

Освоить основные приемы настройки аппаратной части микроконтроллера и разработки подпрограмм обработки прерываний от внешних событий.

1. Исходные данные

Таблица 1. Входные данные.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Индикация | АЦП, бит | Кнопка 1 | Кнопка 2 |
| 17 | 16x | 6 | Y = X and (-Y) | Ввод Х |

Три семисегментных индикатора → P0, P1, P2 соответственно;

Кнопки и выходные линии АЦП → P3;

Линии P3.2 и P3.3 выполняют функции вызова внешнего прерывания INT0 и INT1 соответственно.

1. Ход работы

На рисунке 3.1. представлен листинг кода программы:

|  |
| --- |
| yl equ 30h  yh equ 31h  xl equ 32h  xh equ 33h  org 0  jmp main  ;область векторов прерываний  org 03h  jmp form ;обработчик сигнала INT0  org 13h  jmp input ;обработчик сигнала INT1  org 30h  main:  mov yl, #low(179) ;yl -> B3  mov yh, #high(179) ;yh -> 00  mov sp,#120  ;активация кнопок  mov p0, #0  mov p1, #0  mov p2, #0  mov p3, #0ffh  ;активация прерываний  mov tcon, #5  mov ie,#85h ;разрешение прерываний  ;основной цикл  ;=================================  loop:  mov a, xl  jz L1  jmp le  L1:  call ferrisWheel  le: jmp loop  ;=================================  ferrisWheel:  mov dptr, #tab  mov a, p3  anl a, #243  loopTr16: ;цикл перевода в 16-ную  mov b, #16  div ab  mov r1, a ;второй разряд  mov r0, b ;первый разряд  cjne r0, #16, d16  d16: jc write  jmp loopTr16  write:  swap a  orl a, r0  mov r2, a  mov a, r0 ;вывод первого разряда (единицы)  movc a, @a+dptr ;  mov p2, a ;  mov a, r1 ;вывод второго разряда (десятки)  movc a, @a+dptr ;  mov p1, a ;  mov p0, #63 ;вывод третьего разряда (сотни)  ret  ;обработчик прерывания INT0  form:  loopForm:  jb p3.2, loopForm  clr c  mov a, #0 ;вычитание из нуля  subb a, yl ;  mov r3, a ;  mov a, #0 ;  subb a, yh ;  mov r4, a ;  mov a, r3 ;операция "И" и запись  anl a, xl ;  mov yl, a ;  mov a, r4 ;  anl a, xh ;  mov yh, a ;  mov a, yl ;разделение переменных для их последующего вывода  anl a, #240 ;  swap a ;  mov r6, a ;  mov a, yl ;  anl a, #15 ;  mov r5, a ;  loopView:  mov dptr, #tab  mov a, r5  movc a, @a+dptr  mov p2, a  mov a, r6  movc a, @a+dptr  mov p1, a  jb p3.2, loopView  reti ;завершение обработки прерывания  ;обработчик прерывания INT1  input:  loopInput:  jb p3.3, loopInput  ;сохранение реальных чисел  mov a, xl  jnz null  mov xl, r2  jmp nil  null:  mov xl, #0  nil:  reti ;завершение обработки прерывания  ; Таблица кодов индикатора  tab: db 3fh, 06h, 5bh, 4fh, 66h, 6dh, 7dh, 07h, 7fh, 6fh, 77h, 7ch, 39h, 5eh, 79h, 71h  ; 0 1 2 3 4 5 6 7 8 9 A B C D E F  end |

Рисунок 3.1. – Листинг кода программы

1. Результаты работы программы



Рисунок 4.1. – Окружение микроконтроллера после запуска моделирования

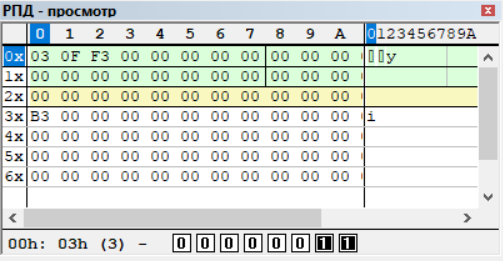


Рисунок 4.2. – РПД после запуска моделирования; в ячейке 02h (**r2**) хранится число введенное с помощью АЦП; в ячейке 30h (**yl**) хранится первоначальное число Y в данном случае оно равно 17910 или B316

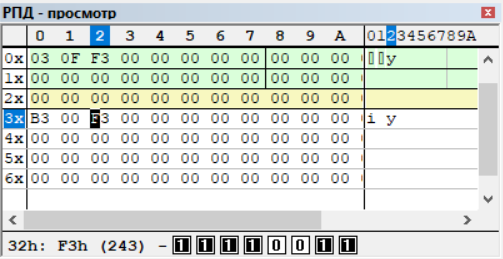


Рисунок 4.3. – РПД после нажатия на кнопку ввод; введенное число сохранилось в ячейке 32h (**xl**)

Также как и в ЛР5, после ввода числа возможность его изменения блокируется до следующего нажатия на кнопку «ввод X».

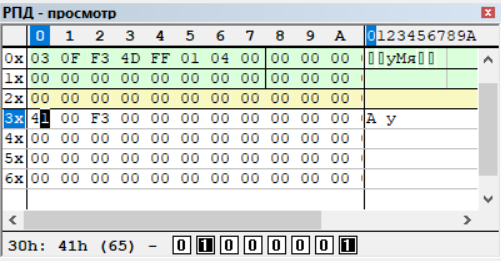


Рисунок 4.4. – РПД после нажатия на кнопку «Формула»; в ячейках 04h и 05h (r3 и r4 соответственно) появляется число -Y; в ячейках 05h и 06h (r5 и r6 соответственно) поразрядно разложенное число Y (уже после работы формулы) для вывода на экран; в ячейке 30h итоговое число Y после работы формулы



Рисунок 4.5. – Окружение микроконтроллера после работы формулы

Проверка результатов работы представлена в таблице 2:

Таблица 2. Результаты работы программы.

|  |  |  |
| --- | --- | --- |
| Y, hex | -Y, hex | X, hex |
| B3 | FF4D | F3 |
| 41 | FFBF | E1 |
| A1 | FF5F | B3 |
| 13 | … | … |



Рисунок 4.6. – Проверка числа -Y для при втором вводе

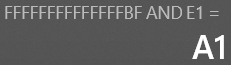


Рисунок 4.7. – Проверка итогового числа Y после работы формулы



Рисунок 4.8. - Итоговое число Y после работы формулы

1. Вывод

В результате выполнения данной лабораторной работы, основные приемы настройки аппаратной части микроконтроллера и разработки подпрограмм обработки прерываний от внешних событий. Как показала практика, использование подпрограмм обработки прерываний, позволяет упростить логику работы программы, при прежней работоспособности программы.