|  |  |
| --- | --- |
| MECHATRONICS INTL-PROGRAM | ASSIGNMENT COVERSHEET |

|  |  |  |
| --- | --- | --- |
| STUDENT DETAILS | | |
| STUDENT NAME | 于航 | |
| STUDENT NAME (PINYIN) | YU HANG | |
| STUDENT ID | 17222054 | |
| CLASS NUMBER | 1702 | |
| ASSIGNMENT DETAILS | | |
| COURSE NAME |  | |
| COURSE NUMBER |  | |
| ASSIGNMENT NAME |  | |
| LECTURER/TUTOR |  | |
| SUBMISSION DETAILS | | |
| DATE DUE | 06/02/2020 | |
| SUBMITTION DATE | 06/01/2020 | |
| STAFF ONLY | | |
| RECEIVED BY |  | DATE/TIME |
| MARKER |  | GRADE |

Declaration and Acknowledgement

By submitting this, I declare that:

1. This assignment meets all the requirements of the subject as student in the relevant subject outline, which I have read.
2. (a) This assessment item is entirely my own work, except where I have included fully-documented references to the work of others.

(b) The material contained in this assessment item has not previously been submitted for assessment.

1. I acknowledge that:

(a) The marker of this assessment item may, for the purpose of assessing this assignment, reproduce this assignment and provide a copy to another member of academic staff.

(b) If required to do so, I will provide an electronic copy of this assessment item to the marker.

1. I am aware that late submission without an authorized extension from the subject coordinator may incur a penalty.

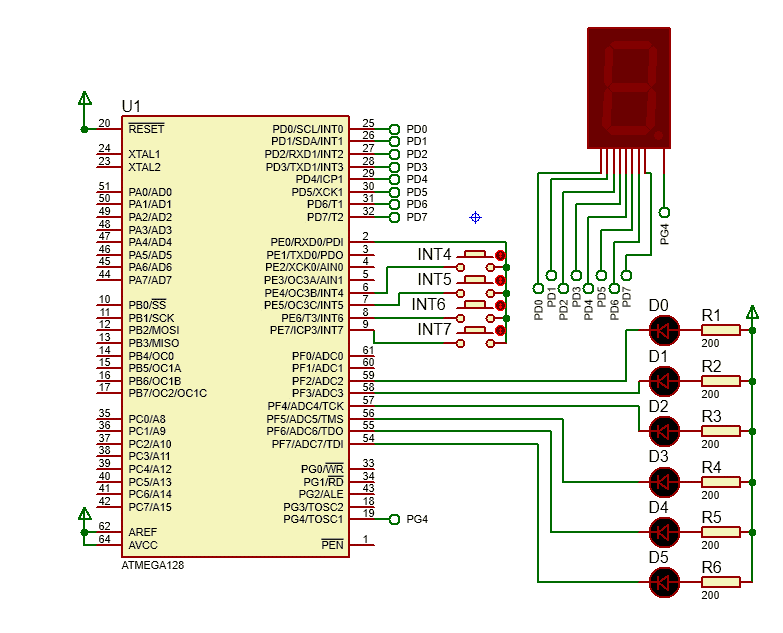
Please note: Assignments are not to be submitted by fax and must be submitted during Lectures/tutorials/laboratories or directly to the academic. Only under special circumstances will the Administrative Staff collect assignments

Experiment 3:Interrupt subsystem

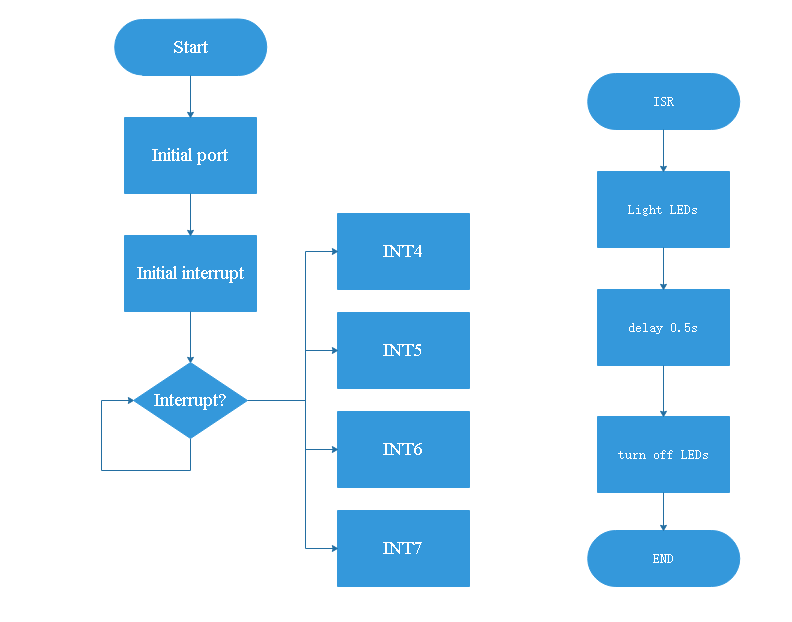
1. Object of experiment
   1. Be able to correctly identify the pins of external interrupt sources of ATmega128 and explain the role and using method of Interrupt direction scale
   2. Be able to use the principle of interrupt, according to the initial setup steps design interrupt service program, design, write and debug the interrupt initialization program and interrupt service routines.
2. Experiment content
   1. basic experiment
      1. From Atmeg128 INT4 - INT7 (PE4 - PE7) acquisition state of switch 4 to 7, INT4 - INT7 were set into a low level, any level flip, falling down, rising along the trigger, when the corresponding switch state changes, in response to the interrupt service program, and LED light.
   2. extended experiment

When press the corresponding key, using digital tube display the corresponding switch interrupt service routine.

1. Experimental schematic diagram



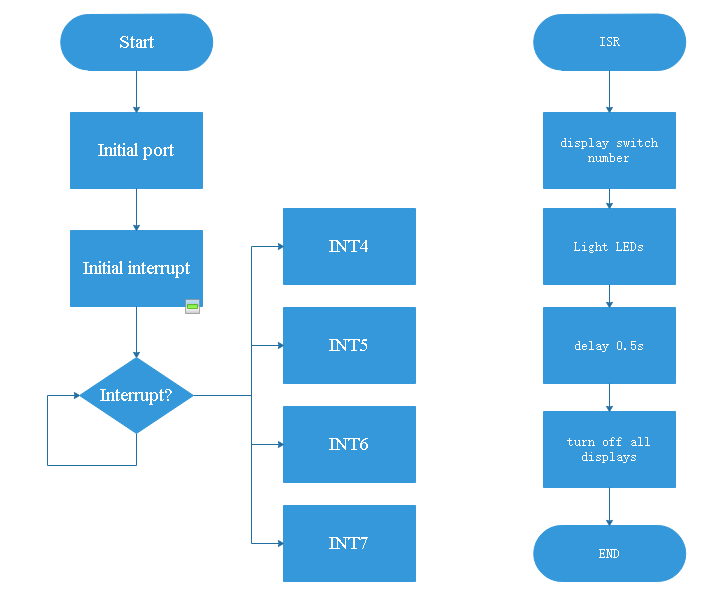
1. Code and flow diagram
   1. basic experiment I
      1. flow diagram



* + 1. code

|  |
| --- |
| 1. #include<iom128v.h> 2. #include<macros.h> 4. #pragma interrupt\_handler int4:iv\_INT4 5. #pragma interrupt\_handler int5:iv\_INT5 6. #pragma interrupt\_handler int6:iv\_INT6 7. #pragma interrupt\_handler int7:iv\_INT7 9. **void** init\_int(**void**){ 10. asm("cli"); 11. EICRB=0x1B;//7 low level 6 any logical change 12. //5 falling eage 4rising eage 13. EIMSK=0xF0;//enable INT4~7 14. asm("sei");//enable global 15. } 17. **void** init\_port(**void**){ 18. DDRF =0xFF;//led 19. PORTF=0xFF;//pull-up 20. DDRE =0x01;//key 21. PORTE|=0xFE;//low level output and pull-up input 22. } 23. //lights the LED lamp 0.5s 24. **void** int4(**void**){ 25. PORTF=0x00; 26. delay\_ms(500); 27. PORTF=0xFF; 28. } 30. **void** int5(**void**){ 31. PORTF=0x00; 32. delay\_ms(500); 33. PORTF=0xFF; 34. } 36. **void** int6(**void**){ 37. PORTF=0x00; 38. delay\_ms(500); 39. PORTF=0xFF; 40. } 42. **void** int7(**void**){ 43. PORTF=0x00; 44. delay\_ms(500); 45. PORTF=0xFF; 46. } 48. **void** main(**void**){ 49. init\_port(); 50. init\_int(); 51. **while**(1){;} 52. } |

* 1. extended experiment
     1. flow diagram



* + 1. code

|  |
| --- |
| 1. #include<iom128v.h> 2. #include<macros.h> 4. #pragma interrupt\_handler int4:iv\_INT4 5. #pragma interrupt\_handler int5:iv\_INT5 6. #pragma interrupt\_handler int6:iv\_INT6 7. #pragma interrupt\_handler int7:iv\_INT7 9. unsigned **char** tab[10]={0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82, 0xF8,0x80, 0x90}; 11. **void** init\_int(**void**){ 12. asm("cli"); 13. EICRB=0x1B;//7 low level 6 any logical change 14. //5falling edge 4rising edge 15. EIMSK=0xF0;//enable INT4~7 16. asm("sei"); //enable global 17. } 19. **void** init\_port(**void**){ 20. DDRF =0xFF;//led 21. PORTF=0xFF;//pull-up 22. DDRE =0x01;//key 23. PORTE|=0xFE;//low level output and pull-up input 24. DDRD  = 0XFF;//Digital tube 25. PORTD = 0XFF;//pull-up 26. PORTG = 0x10; 27. DDRG  = 0x10;//Digital tube number selection 28. } 29. //The digital tube displays the switch number and lights the LED lamp 30. **void** int4(**void**){ 31. PORTD = tab[4]; 32. PORTF=0x00; 33. delay\_ms(500); 34. PORTF=0xFF; 35. PORTD = 0XFF; 36. } 38. **void** int5(**void**){ 39. PORTD = tab[5]; 40. PORTF=0x00; 41. delay\_ms(500); 42. PORTF=0xFF; 43. PORTD = 0XFF; 44. } 46. **void** int6(**void**){ 48. PORTD = tab[6]; 49. PORTF=0x00; 50. delay\_ms(500); 51. PORTF=0xFF; 52. PORTD = 0XFF; 53. } 55. **void** int7(**void**){ 57. PORTD = tab[7]; 58. PORTF=0x00; 59. delay\_ms(500); 60. PORTF=0xFF; 61. PORTD = 0XFF; 62. } 64. **void** main(**void**){ 65. init\_port(); 66. init\_int(); 67. **while**(1){;} 68. } |

1. Experiment Result
   1. Basic experiment I

LED will be lighten 0.5s when button 7 is pressed and if the button is not released the led will light all the time.

LED will be lighten 0.5s when the button 6 is pressed and if the button is released LED will be lighten 0.5s again

LED will be lighten 0.5s when the button 5 is pressed

LED will be lighten 0.5s when the button 6 is released

* 1. Extended experiment

LED will be lighten and digital tube will show “7” for 0.5s when button 7 is pressed and if the button is not released the LED and digital tube will light all the time.

LED will be lighten and digital tube will show “6” for 0.5s when the button 6 is pressed and if the button is released LED will be lighten and digital tube will show “6” for 0.5s again

LED will be lighten and digital tube will show “5” for 0.5when the button 5 is pressed

LED will be lighten and digital tube will show “6” for 0.5s when the button 6 is released

1. Reflection

To initial interrupt subsystem we should do following:

* 1. Enable a type of interruption, such as INT4.
  2. Set the interrupt trigger condition. For instance, the rising edge triggers the interrupt.
  3. Enable global interrupt.

To write a interrupt service routine in iccavr, we should write “# pragma interrupt\_handler (the name of ISR): Interrupt vector number” at first. Then write ISR as a subfunction.

In an electronic clock, an external interrupt can be used to change state current time or alarm clock in which we can adjust the present time or alarm clock time. an external can be used to select either hour or minute will be adjust. and two other button can control the number of hour or minute increase and decrease.

1. Experience

In this experiment, I have a deeper understanding of interrupt vector diagram and understand the priority relationship of different interrupts. A clearer understanding of the triggering conditions of interrupts, especially external interrupts. At the same time, I have learned how to write a simple interrupt service program and how to apply interrupts to solve some problems.