|  |  |
| --- | --- |
| 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 | 06/09/2020 | |
| SUBMITTION DATE | 06/09/2020 | |
| 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 5:Timer

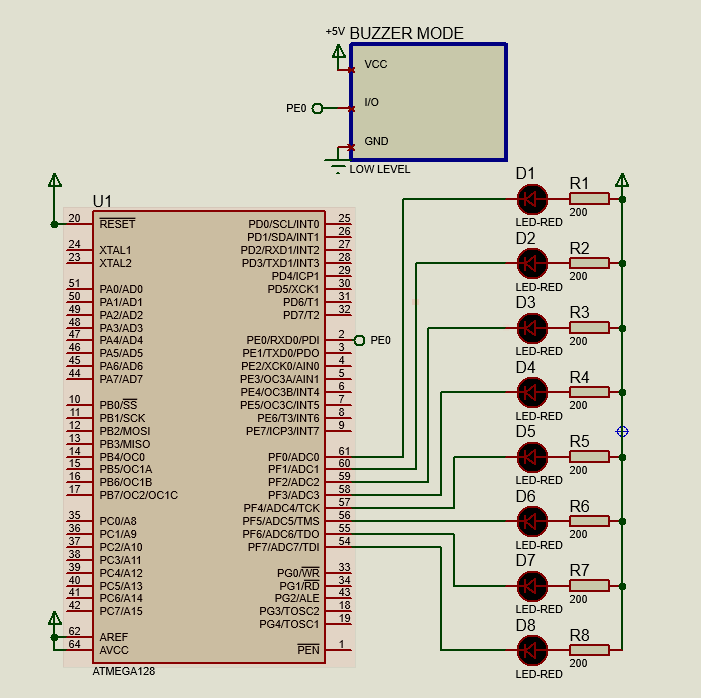
1. Object of experiment
   1. Be able to explain the structure, function and working principle of timer and recognize the number and pin of timer in ATmega128.
   2. Be able to design, write and debug the initial and extended program of timer/counter.
2. Experiment content
   1. basic experiment

Use A group of timer1 to generate timing of 10ms in CTC mode. In ISR of timer1 TCCR1A Compare matching interrupt, PF generate 2s square wave output cycle, which drive LEDs to display the PF level changes.

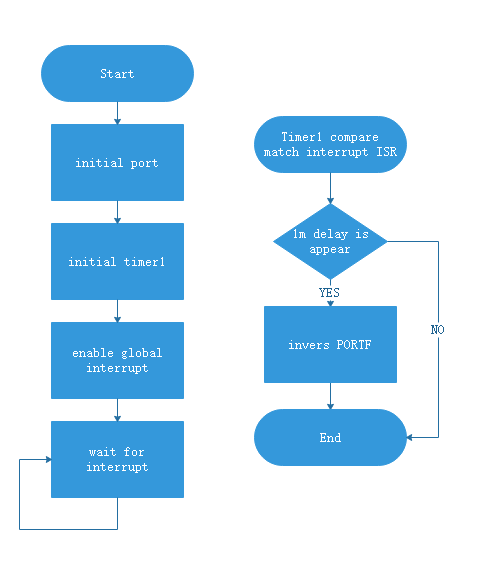
* 1. extended experiment

Consult relevant materials, write notes 1, 2, 3… of the tone play program.

1. Experimental schematic diagram



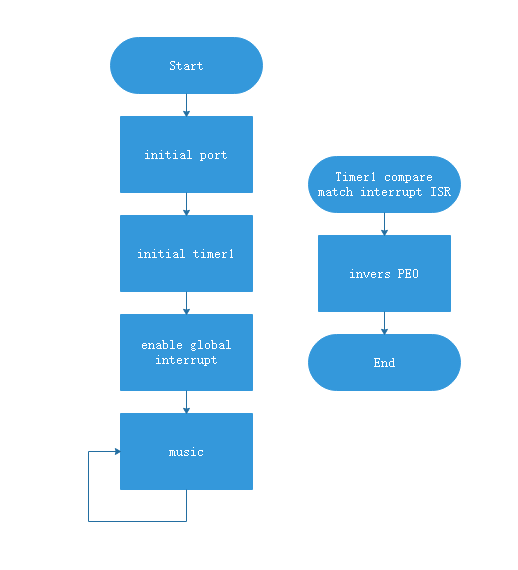
1. Code and flow diagram
   1. basic experiment
      1. flow diagram



* + 1. code

|  |
| --- |
| 1. #include <iom128v.h> 2. #include <macros.h> 4. unsigned **char** a=0; 6. **void** init\_timer1(**void**){ 7. TCCR1A|=(0<<WGM11)|(0<<WGM10); 8. TCCR1B|=(1<<WGM12);//CTC 9. TCCR1A|=(0<<COM1A1)|(0<<COM1A0);//OC1A normal output 10. TCCR1B|=(0<<CS12)|(1<<CS11)|(0<<CS10);//8prescale 1US 11. OCR1A=9999;//10ms 12. TIMSK=0x10;//TIM1\_COMPC enable 13. } 15. **void** init\_port(**void**){ 16. DDRF=0XFF;//port F as output 17. PORTF=0X00;// low level 18. } 20. #pragma interrupt\_handler timer1:iv\_TIM1\_COMPC 21. **void** timer1(**void**){//ISR 22. a++; 23. **if**(a==100){//1s 24. a=0; 25. PORTF=~PORTF;//invert PORTF 26. } 27. } 29. **void** main(**void**){ 30. init\_port(); 31. init\_timer1(); 32. SEI();// enable global interrupt 33. **while**(1){;} 34. } |

* 1. extended experiment
     1. flow diagram



* + 1. code

|  |
| --- |
| 1. #include <iom128v.h> 2. #include <macros.h> 4. #define L1 262 5. #define L2 294 6. #define L3 330 7. #define L4 349 8. #define L5 392 9. #define L6 440 10. #define L7 494 11. #define M1 523 12. #define M2 587 13. #define M3 659 14. #define M4 698 15. #define M5 784 16. #define M6 880 17. #define M7 988 18. #define H1 1047 19. #define H2 1175 20. #define H3 1319 21. #define H4 1397 22. #define H5 1568 23. #define H6 1760 24. #define H7 1967 26. unsigned **int** star[]= { 27. H1,8, H1,8, H5,8, H5,8, H6,8, H6,8, H5,16, 28. H4,8, H4,8, H3,8, H3,8, H2,8, H2,8, H1,16, 29. H5,8, H5,8, H4,8, H4,8, H3,8, H3,8, H2,16, 30. H5,8, H5,8, H4,8, H4,8, H3,8, H3,8, H2,16, 31. H1,8, H1,8, H5,8, H5,8, H6,8, H6,8, H5,16, 32. H4,8, H4,8, H3,8, H3,8, H2,8, H2,8, H1,16, 33. 0xFF 34. }; 36. // to get 1 us delay 37. **void** delay\_us(unsigned **int** microsecond){ 38. **do**{ 39. microsecond--; 40. }**while** (microsecond>1); 41. } 43. // to get 1 ms delay 44. **void** delay\_ms(unsigned **int** millisecond){ 45. **while** (millisecond--){ 46. delay\_us(999); 47. } 48. } 50. **void** init\_timer1(**void**){ 51. TCCR1A|=(0<<WGM11)|(0<<WGM10); 52. TCCR1B|=(1<<WGM12);//CTC 53. TCCR1A|=(0<<COM1A1)|(0<<COM1A0);//OC1A NORMAL MODE OUTPUT 54. TCCR1B|=(0<<CS12)|(1<<CS11)|(0<<CS10);//8prescale 55. } 57. **void** init\_port(**void**){ 58. DDRE=0xFF; 59. PORTE=0XFF; 60. } 62. **void** music(unsigned **int** \*song){ 63. **while**(\*song!=0XFF){ 64. delay\_ms(10); 65. OCR1A=(unsigned **int**)(500000/(\*song)-1);//OCR1=8M/(8\*2\*f)-1 66. TIMSK=0x10;//enable timer1 compture interrupt 67. song++;//get time 68. delay\_ms(62\*(\*song)); 69. song++;//get another tune 70. TIMSK=0x00;//disable timer1 71. } 72. delay\_ms(1000); 73. } 74. #pragma interrupt\_handler timer1:iv\_TIM1\_COMPC 75. **void** timer1(**void**){ 76. PORTE^=BIT(0); 77. } 79. **void** main(**void**){ 80. init\_port(); 81. init\_timer1(); 82. SEI();//enable global interrupt 83. **while**(1){ 84. music(star); 85. } 86. } |

1. Experiment Result
   1. Basic experiment

Turn on the power, the LED light is on for 1s and then off for 1s, and the display is cyclic

* 1. Extended experiment

Turn on the power and the buzzer plays twinkle star

1. Reflection

to write initial program of timer we should which Waveform Generation Mode, Compare Match Output Mode and prescale are needed, whether an interruption is needed, and what frequency is needed.

After that we can set TCCRn, TIMSK and OCRn or TCNTn to initial the timer.

1. Experience

In this experiment, I learned the composition of timers in ATmega128 and mastered how to configure timers, write and debug timer initialization programs, and use timers to achieve specific functions, such as generating square waves and playing music.