目录

[74HC595驱动 2](#_Toc487267517)

[Usart0基本例程 3](#_Toc487267518)

[Usart0中断方式 4](#_Toc487267519)

[INT0外部中断 5](#_Toc487267520)

[INT1外部中断 6](#_Toc487267521)

[PCINT引脚中断 7](#_Toc487267522)

[Timer0\_PWMA 8](#_Toc487267523)

[Timer0\_PWMA&PWMB 9](#_Toc487267524)

[Timer0\_Config 9](#_Toc487267525)

[Timer1\_PWMA 10](#_Toc487267526)

[Timer1\_PWMB 11](#_Toc487267527)

[Timer1\_Config 12](#_Toc487267528)

[Timer1\_Take\_Config 13](#_Toc487267529)

[Timer2\_PWMA 14](#_Toc487267530)

[Timer2\_PWMB 14](#_Toc487267531)

[Timer2\_Config 15](#_Toc487267532)

[ADC\_Chnel 16](#_Toc487267533)

## 74HC595驱动

|  |
| --- |
| /\*  \* AVRGCC1.c  \*  \* Created: 2016/10/23 17:11:04  \* Author: xuyuntonge  \*/  #include <avr/io.h>  #include <avr/pgmspace.h>  #include <avr/interrupt.h>  #include <avr/signal.h>  #include <string.h>  #include <stdio.h>  #define F\_CPU 8000000UL //初始化晶振频率  #include <util/delay.h>  #include"helloavr.h"  #define SDA\_L PC5\_LOW  #define SDA\_H PC5\_HIGH  #define SCK\_L PC4\_LOW  #define SCK\_H PC4\_HIGH  #define RCK\_L PC3\_LOW  #define RCK\_H PC3\_HIGH  void HC595\_LATCH(void);  void HC595\_WR(unsigned char dat) ;  int main(void)  {  DDRC=0xff;  DDRD=0xff;  while (1)  {  HC595\_WR(0XFF);  HC595\_LATCH();  *\_delay\_ms*(500);  HC595\_WR(0X00);  HC595\_LATCH();  *\_delay\_ms*(500);  }  }  void HC595\_WR(unsigned char dat) //595写数据  {  unsigned char i;  for(i=0;i<8;i++) {  if(dat&0x01)  {  SDA\_H;  }  else SDA\_L;  SCK\_L;  SCK\_H;  dat>>=1;  }  }  void HC595\_LATCH(void)  {  RCK\_L;  RCK\_H;  } |

## Usart0基本例程

|  |
| --- |
| /\*  \* AVRGCC1.c  \*  \* Created: 2016/10/23 17:11:04  \* Author: xuyuntonge  \*/  #include <avr/io.h>  #include <avr/pgmspace.h>  #include <avr/interrupt.h>  #include <string.h>  #include <stdio.h>  # define F\_CPU 4000000UL  #include <util/delay.h>  void USART\_Init(void); //baud等于9600 8个数据位 0个停止位  void USART\_Transmit( unsigned char data );  unsigned char USART\_Receive( void );  void USART\_Init(void)  {  UBRR0H =0;  UBRR0L =25; //设置串口波特率为9600  UCSR0B = (1<<RXEN0)|(1<<TXEN0); //使能串口发送接收  UCSR0C = (3<<UCSZ00)| (0<<USBS0); //停止位为1位数据位为8位  }  void USART\_Transmit( unsigned char data )  {  while ( !( UCSR0A & (1<<UDRE0)) );  UDR0 = data;  }  unsigned char USART\_Receive( void )  {  while ( !(UCSR0A & (1<<RXC0)) );  return UDR0;  }  int main(void)  {  USART\_Init();  while (1)  {  USART\_Transmit(0xf3);  *\_delay\_ms*(300);  }  } |

## Usart0中断方式

|  |
| --- |
| /\*  \* AVRGCC1.c  \*  \* Created: 2016/10/23 17:11:04  \* Author: xuyuntonge  \*/  #include <avr/io.h>  #include <avr/pgmspace.h>  #include <avr/interrupt.h>  #include <string.h>  #include <stdio.h>  # define F\_CPU 4000000UL  #include <util/delay.h>  void USART\_Init(void); //baud等于9600 8个数据位 0个停止位  void USART\_Transmit( unsigned char data );  void USART\_Init(void)  {  cli();  DDRD |= 0b00000010;  PORTD |= 0b0000010;  UBRR0H =0;  UBRR0L =25;  UCSR0B = (1<<RXEN0)|(1<<TXEN0)|(1<<RXCIE0);  UCSR0C = (3<<UCSZ00)| (0<<USBS0);  sei();  }  void USART\_Transmit( unsigned char data )  {  while ( !( UCSR0A & (1<<UDRE0)) );  UDR0 = data;  }  int main(void)  {  USART\_Init();  while (1)  {  *\_delay\_ms*(300);  }  }  ISR(USART\_RX\_vect)  {  unsigned char Date;  Date=UDR0;  USART\_Transmit(Date);  } |

## INT0外部中断

|  |
| --- |
| /\*  \* AVRGCC1.c  \*  \* Created: 2016/10/23 17:11:04  \* Author: xuyuntonge  \*/  #include <avr/io.h>  #include <avr/pgmspace.h>  #include <avr/interrupt.h>  #include <string.h>  #include <stdio.h>  # define F\_CPU 4000000UL  #include <util/delay.h>  void Int0\_Config()  {  cli();  EICRA=(1<<ISC00); //任意电平变换引发中断  EIMSK=(1<<INT0); //使能外部中断0  sei();  }  int main(void)  {  DDRC=0xff;  DDRD=0x00;  Int0\_Config();  while (1)  {    }  }  ISR(INT0\_vect ) //外部中断0入口  {  PORTC=0XFF;  } |

## INT1外部中断

|  |
| --- |
| /\*  \* AVRGCC1.c  \*  \* Created: 2016/10/23 17:11:04  \* Author: xuyuntonge  \*/  #include <avr/io.h>  #include <avr/pgmspace.h>  #include <avr/interrupt.h>  #include <string.h>  #include <stdio.h>  # define F\_CPU 4000000UL  #include <util/delay.h>  void Int0\_Config()  {  cli();  EICRA=(1<<ISC10); //任意电平变换引发中断  EIMSK=(1<<INT1); //使能外部中断1  sei();  }  int main(void)  {  DDRC=0xff;  DDRD=0x00;  Int0\_Config();  while (1)  {    }  }  ISR(INT1\_vect ) //外部中断1入口  {  PORTC=0XFF;  } |

## PCINT引脚中断

|  |
| --- |
| /\*  \* AVRGCC1.c  \*  \* Created: 2016/10/23 17:11:04  \* Author: xuyuntonge  \*/  #include <avr/io.h>  #include <avr/pgmspace.h>  #include <avr/interrupt.h>  #include <string.h>  #include <stdio.h>  # define F\_CPU 4000000UL  #include <util/delay.h>  void PcInt\_Config()  {  cli();  PCICR=(1<<PCIE0); //PCINT7-0  PCIFR=(0<<PCIF0); //清除中断标志位  PCMSK0=(1<<PCIF0); //使能PCINT1中断  sei();  }  int main(void)  {  DDRC=0xff;  DDRD=0x00;  PcInt\_Config();  while (1)  {  }  }  ISR(PCINT0\_vect) //外部中断1入口  {  PCIFR=(0<<PCIF0);  PORTC=0XFF;  } |

## Timer0\_PWMA

|  |
| --- |
| /\*  \* AVRGCC1.c  \*  \* Created: 2016/10/23 17:11:04  \* Author: xuyuntonge  \*/  #include <avr/io.h>  #include <avr/pgmspace.h>  #include <avr/interrupt.h>  #include <string.h>  #include <stdio.h>  # define F\_CPU 4000000UL  #include <util/delay.h>  void Timer0\_PWMA(unsigned int PWM\_Date)  {  TCCR0A=(1<<COM0A1)|(1<<WGM00)|(1<<WGM01);//|(1<<WGM02)|  TCCR0B=(1<<CS00);  OCR0A=PWM\_Date; //占空比设置  }  int main(void)  {  DDRD=0xff;  Timer0\_PWMA(120);  while (1)  {  }  } |

## Timer0\_PWMA&PWMB

|  |
| --- |
| /\*  \* AVRGCC1.c  \*  \* Created: 2016/10/23 17:11:04  \* Author: xuyuntonge  \*/  #include <avr/io.h>  #include <avr/pgmspace.h>  #include <avr/interrupt.h>  #include <string.h>  #include <stdio.h>  # define F\_CPU 4000000UL  #include <util/delay.h>  void Timer0\_PWMB(unsigned int PWM\_Date)  {  TCCR0A|=(1<<COM0B1)|(1<<WGM00)|(1<<WGM01);//|(1<<WGM02)|  TCCR0B|=(1<<CS00);  OCR0B=PWM\_Date; //占空比设置  }  void Timer0\_PWMA(unsigned int PWM\_Date)  {  TCCR0A|=(1<<COM0A1)|(1<<WGM00)|(1<<WGM01);//|(1<<WGM02)|  TCCR0B|=(1<<CS00);  OCR0A=PWM\_Date; //占空比设置  }  int main(void)  {  DDRD=0xff;  Timer0\_PWMB(120);  Timer0\_PWMA(60);  while (1)  {    }  } |

## Timer0\_Config

|  |
| --- |
| /\*  /\*  \* AVRGCC1.c  \*  \* Created: 2016/10/23 17:11:04  \* Author: xuyuntonge  \*/  #include <avr/io.h>  #include <avr/pgmspace.h>  #include <avr/interrupt.h>  #include <string.h>  #include <stdio.h>  # define F\_CPU 4000000UL  #include <util/delay.h>  void Timer0\_Config()  {  cli();  TIMSK0=(1<<TOIE0); //使能溢出中断  TCCR0A=(1<<WGM01); //使能为CTC模式  TCCR0B=(1<<CS01); //8分频  TCNT0=100; //定时器初值  sei();  }  int main(void)  {  Timer0\_Config();  while (1)  {  }  }  ISR(TIMER0\_OVF\_vect) //定时器0溢出  {  TCNT0=100; //定时器重装初值  //here add your code  } |

## Timer1\_PWMA

|  |
| --- |
| /\*  \* AVRGCC1.c  \*  \* Created: 2016/10/23 17:11:04  \* Author: xuyuntonge  \*/  #include <avr/io.h>  #include <avr/pgmspace.h>  #include <avr/interrupt.h>  #include <string.h>  #include <stdio.h>  # define F\_CPU 4000000UL  #include <util/delay.h>  void Timer1\_PWMA(unsigned int PWM\_Date) //MAX=0x03ff  {  TCCR1A|=(1<<COM1A1)|(1<<WGM11)|(1<<WGM10); //10位快速PWM 匹配发生时候取反  TCCR1B|=(1<<CS10)|(1<<WGM12); //不分频  OCR1AH=PWM\_Date/256;  OCR1AL=PWM\_Date%256;  }  int main(void)  {  DDRB=0XFF;  Timer1\_PWMA(0x0300);  while (1)  {  }  } |

## Timer1\_PWMB

|  |
| --- |
| /\*  \* AVRGCC1.c  \*  \* Created: 2016/10/23 17:11:04  \* Author: xuyuntonge  \*/  #include <avr/io.h>  #include <avr/pgmspace.h>  #include <avr/interrupt.h>  #include <string.h>  #include <stdio.h>  # define F\_CPU 4000000UL  #include <util/delay.h>  void Timer1\_PWMB(unsigned int PWM\_Date) //MAX=0x03ff  {  TCCR1A|=(1<<COM1B1)|(1<<WGM11)|(1<<WGM10); //10位快速PWM 匹配发生时候取反  TCCR1B|=(1<<CS10)|(1<<WGM12); //不分频  OCR1BH=PWM\_Date/256;  OCR1BL=PWM\_Date%256;  }  int main(void)  {  DDRB=0XFF;  Timer1\_PWMB(0x0300);  while (1)  {  }  } |

## Timer1\_Config

|  |
| --- |
| /\*  \* AVRGCC1.c  \*  \* Created: 2016/10/23 17:11:04  \* Author: xuyuntonge  \*/  #include <avr/io.h>  #include <avr/pgmspace.h>  #include <avr/interrupt.h>  #include <string.h>  #include <stdio.h>  # define F\_CPU 4000000UL  #include <util/delay.h>  void Timer1\_Config(unsigned int Timer\_Date) //设置时间  {  cli();  TCCR1B=(1<<CS11)|(1<<CS10); //设置分频系数64分频  TCNT1H=(65535-Timer\_Date)/256;  TCNT1L=(65535-Timer\_Date)%256;  TIMSK1=(1<<TOIE1); //设置允许中断溢出  sei();  }  int main(void)  {    Timer1\_Config(5000);  while (1)  {    }  }  SIGNAL(TIMER1\_OVF\_vect) //定时器1  {  TCNT1H=(65535-5000)/256;  TCNT1L=(65535-5000)%256;  //here add your code  } |

## Timer1\_Take\_Config

|  |
| --- |
| /\*  \* AVRGCC1.c  \*  \* Created: 2016/10/23 17:11:04  \* Author: xuyuntonge  \*/  #include <avr/io.h>  #include <avr/pgmspace.h>  #include <avr/interrupt.h>  #include <string.h>  #include <stdio.h>  # define F\_CPU 4000000UL  #include <util/delay.h>  void Timer1\_Take() //设置时间  {  cli();  TCCR1B=(1<<CS11)|(1<<CS10); //设置分频系数64分频 下降沿触发 关闭滤波  TIMSK1=(1<<ICIE1); //设置允许捕捉中断  TIFR1=(0<<ICF1);  sei();  }  int main(void)  {  DDRB=0X00; //设置ICP1为输入模式即可  Timer1\_Take();  while (1)  {    }  }  SIGNAL(TIMER1\_CAPT\_vect) //捕捉  {  TIFR1=(0<<ICF1); //清除捕捉标志位  //here add your code  } |

## Timer2\_PWMA

|  |
| --- |
| /\*  \* AVRGCC1.c  \*  \* Created: 2016/10/23 17:11:04  \* Author: xuyuntonge  \*/  #include <avr/io.h>  #include <avr/pgmspace.h>  #include <avr/interrupt.h>  #include <string.h>  #include <stdio.h>  # define F\_CPU 4000000UL  #include <util/delay.h>  #define BIT(x) (1<<(x))  void Timer2\_PWMA(unsigned int PWM\_Date) //MAX=0x03ff  {  TCCR2A|=(1<<COM2A1)|(1<<WGM21)|(1<<WGM20); //10位快速PWM 匹配发生时候取反|(1<<CS21)  TCCR2B|=(1<<CS20); //不分频  OCR2A=PWM\_Date;  }  int main(void)  {    DDRB=0XFF;  DDRC=0XFF;  DDRD=0XFF;  Timer2\_PWMA(200);  while (1)  {    }  } |

## Timer2\_PWMB

|  |
| --- |
| /\*  \* AVRGCC1.c  \*  \* Created: 2016/10/23 17:11:04  \* Author: xuyuntonge  \*/  #include <avr/io.h>  #include <avr/pgmspace.h>  #include <avr/interrupt.h>  #include <string.h>  #include <stdio.h>  # define F\_CPU 4000000UL  #include <util/delay.h>  #define BIT(x) (1<<(x))  void Timer2\_PWMB(unsigned int PWM\_Date) //MAX=0x03ff  {  TCCR2A|=(1<<COM2B1)|(1<<WGM21)|(1<<WGM20); //10位快速PWM 匹配发生时候取反|(1<<CS21)  TCCR2B|=(1<<CS20); //不分频  OCR2B=PWM\_Date;  }  int main(void)  {    DDRB=0XFF;  DDRC=0XFF;  DDRD=0XFF;  Timer2\_PWMB(200);  while (1)  {    }  } |

## Timer2\_Config

|  |
| --- |
| /\*  \* AVRGCC1.c  \*  \* Created: 2016/10/23 17:11:04  \* Author: xuyuntonge  \*/  #include <avr/io.h>  #include <avr/pgmspace.h>  #include <avr/interrupt.h>  #include <string.h>  #include <stdio.h>  # define F\_CPU 4000000UL  #include <util/delay.h>  #define BIT(x) (1<<(x))  void Timer2\_Config()  {  cli();  TIMSK2|=(1<<TOIE2);  TCCR2A|=(1<<WGM21);  TCCR2B|=(1<<CS21); //8分频  TCNT2=100;  sei();  }  int main(void)  {    DDRB=0XFF;  DDRC=0XFF;  DDRD=0XFF;  Timer2\_Config();  while (1)  {    }  }  ISR(TIMER2\_OVF\_vect) //定时器2溢出  {  TCNT2=100; //定时器重装初值  //here add your code  } |

## ADC\_Chnel

|  |
| --- |
| /\*  \* AVRGCC1.c  \*  \* Created: 2016/10/23 17:11:04  \* Author: xuyuntonge  \*/  #include <avr/io.h>  #include <avr/pgmspace.h>  #include <avr/interrupt.h>  #include <string.h>  #include <stdio.h>  # define F\_CPU 4000000UL  #include <util/delay.h>  #define BIT(x) (1<<(x))  unsigned int ADC\_Chnel() //设置时间  {  unsigned int date;  ADCSRA |= (1<<ADSC);  while(!(ADCSRA & (1 << ADIF))); //等待转换结束 // ADCSRA|=(1<<ADATE)|(1<<ADEN); (1<<ADATE)||(1<<ADSC)  ADCSRA |= (1 << ADIF);  return ADC;  }  void AD\_Init()  {  ADMUX |= (1 << REFS0)|(1 << REFS1); //ADC参考电压为AVcc，ADC结果右对齐，选择通道ADC0  ADCSRA |= (1 << ADEN) | (1 << ADPS2) | (1 << ADPS1); //使能AD转换，ADC时钟64分频  }  void USART\_Init(void)  {  UBRR0H =0;  UBRR0L =51; //设置串口波特率为9600  UCSR0B = (1<<RXEN0)|(1<<TXEN0); //使能串口发送接收  UCSR0C = (3<<UCSZ00)| (0<<USBS0); //停止位为1位数据位为8位  }  void USART\_Transmit( unsigned char data )  {  while ( !( UCSR0A & (1<<UDRE0)) );  UDR0 = data;  }  int main(void)  {    USART\_Init();  DDRB=0xff;  DDRC=0x00;  AD\_Init();  while (1)  {  USART\_Transmit(ADC\_Chnel());  *\_delay\_ms*(200);  }  } |