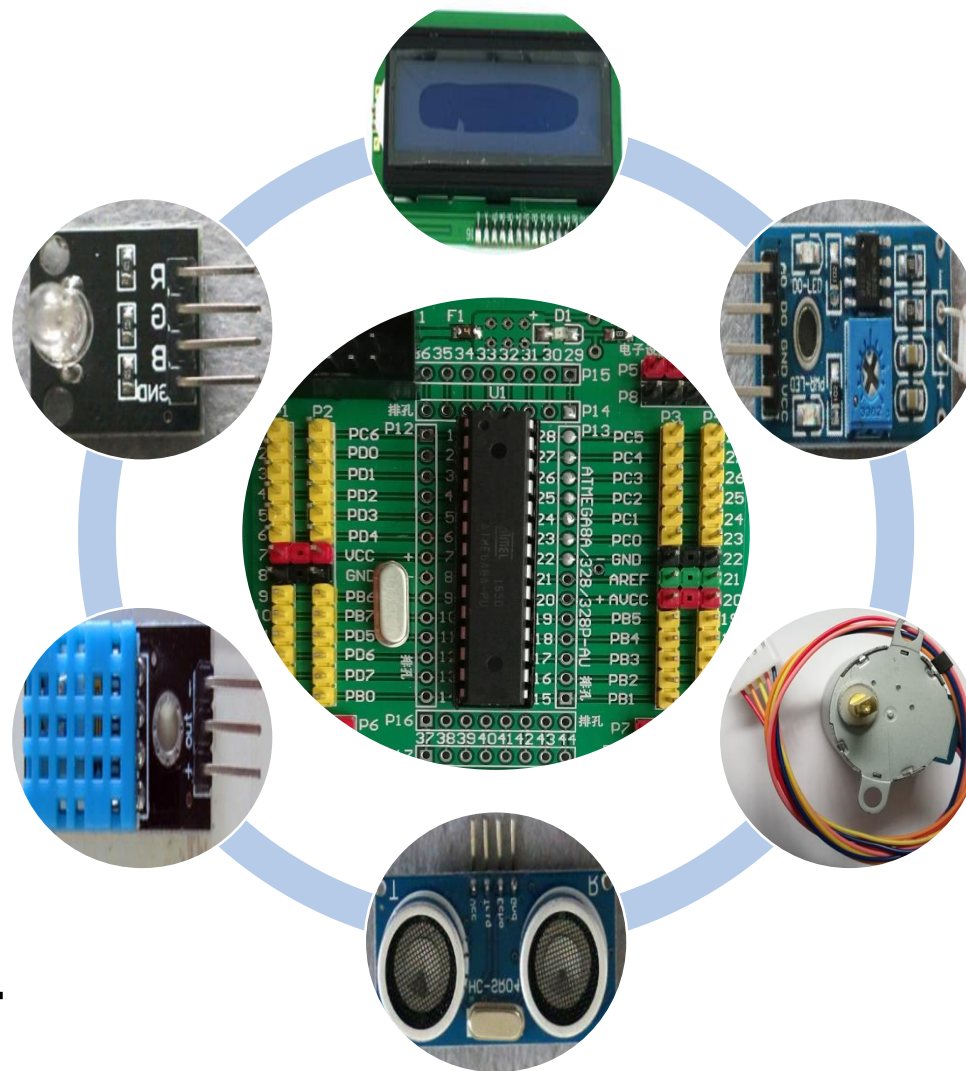


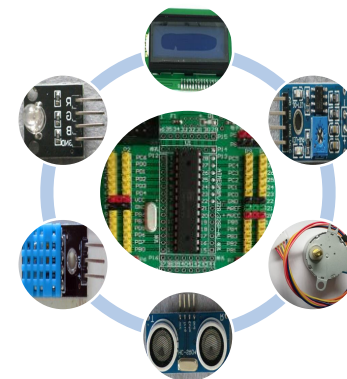
电子设计实践 基础

温湿度传感器、
超声波传感器、
MCU ADC与光敏电阻



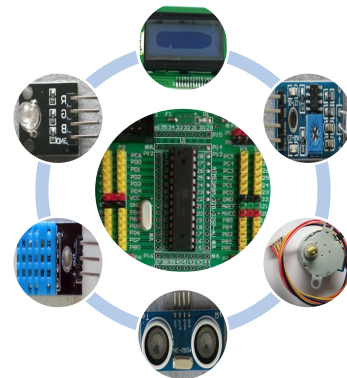
上节课内容回顾

- MCU TWI接口、LCD显示屏及其编程
 - TWI接口 (I^2C 总线) 及其编程
 - LCD1602模块 (I^2C 最快100K) 及其编程

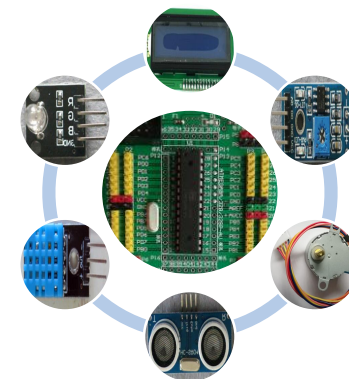
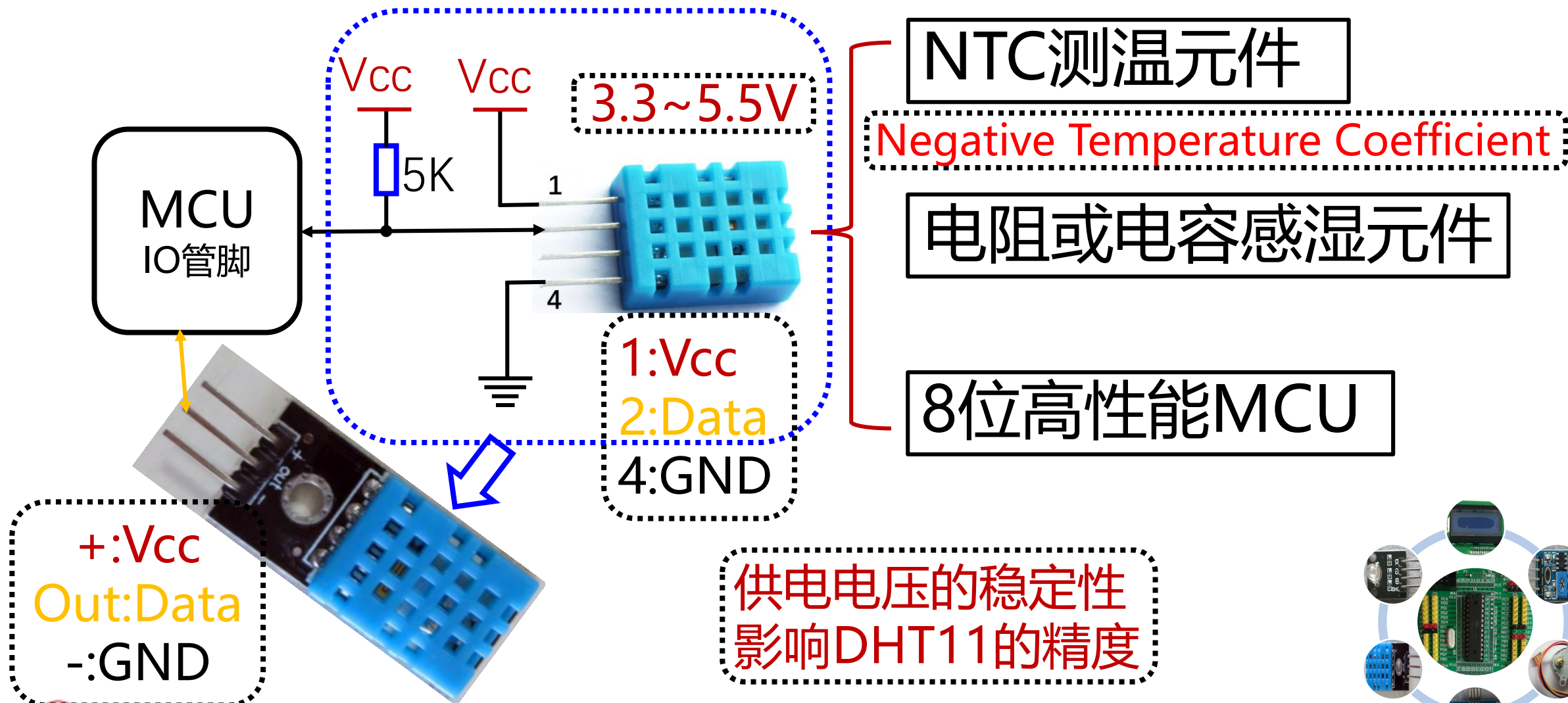


本节课主要内容

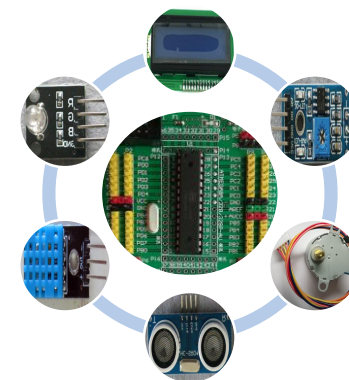
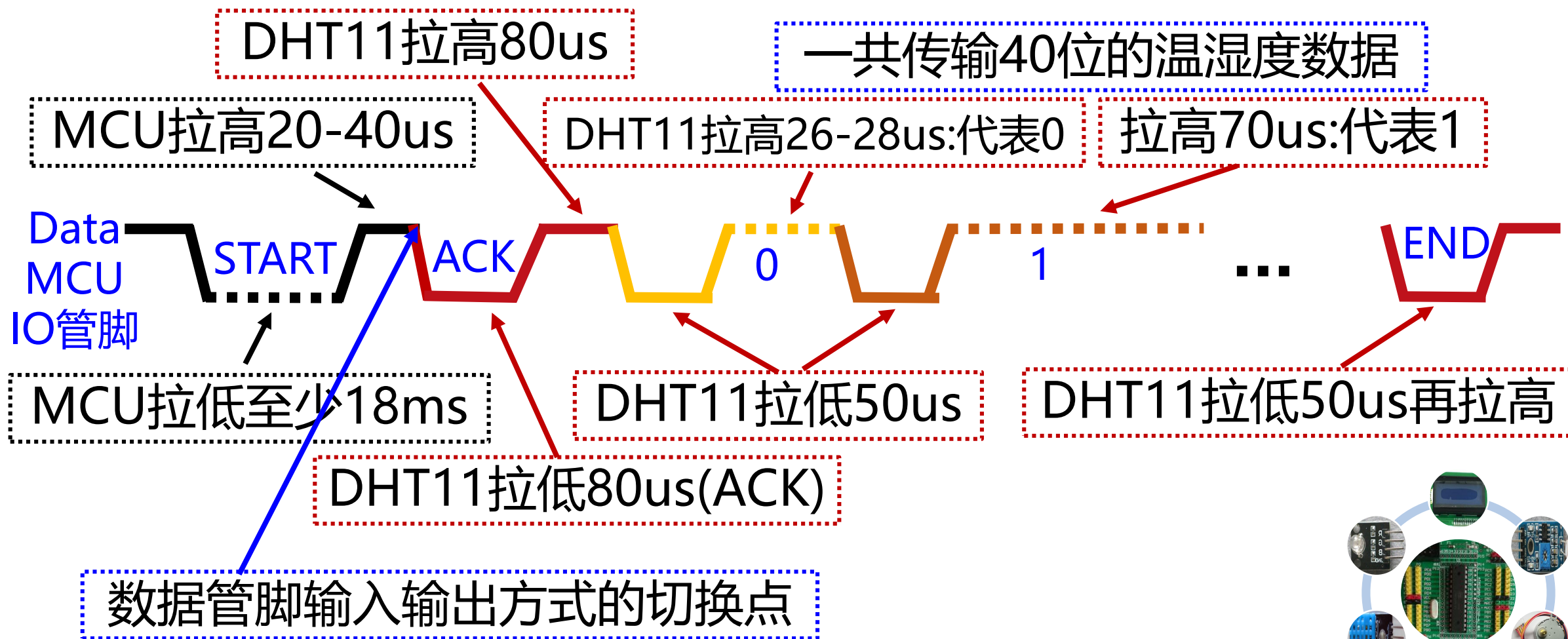
- 温湿度传感器
- 超声波传感器
- MCU ADC与光敏电阻



温湿度传感器：DHT11

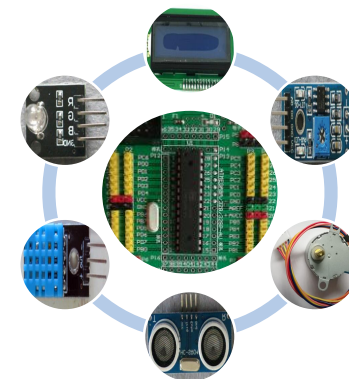
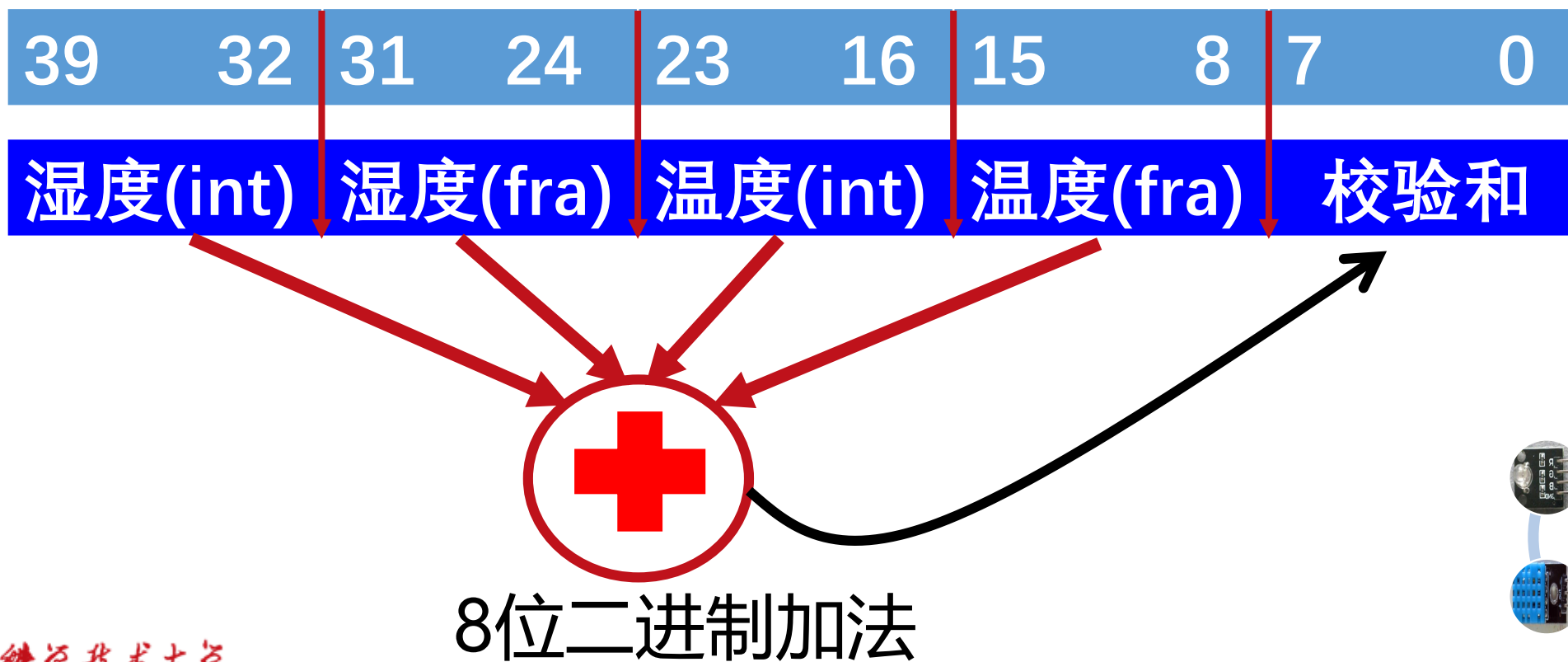


ATmega8A从DHT11获取数据的过程控制



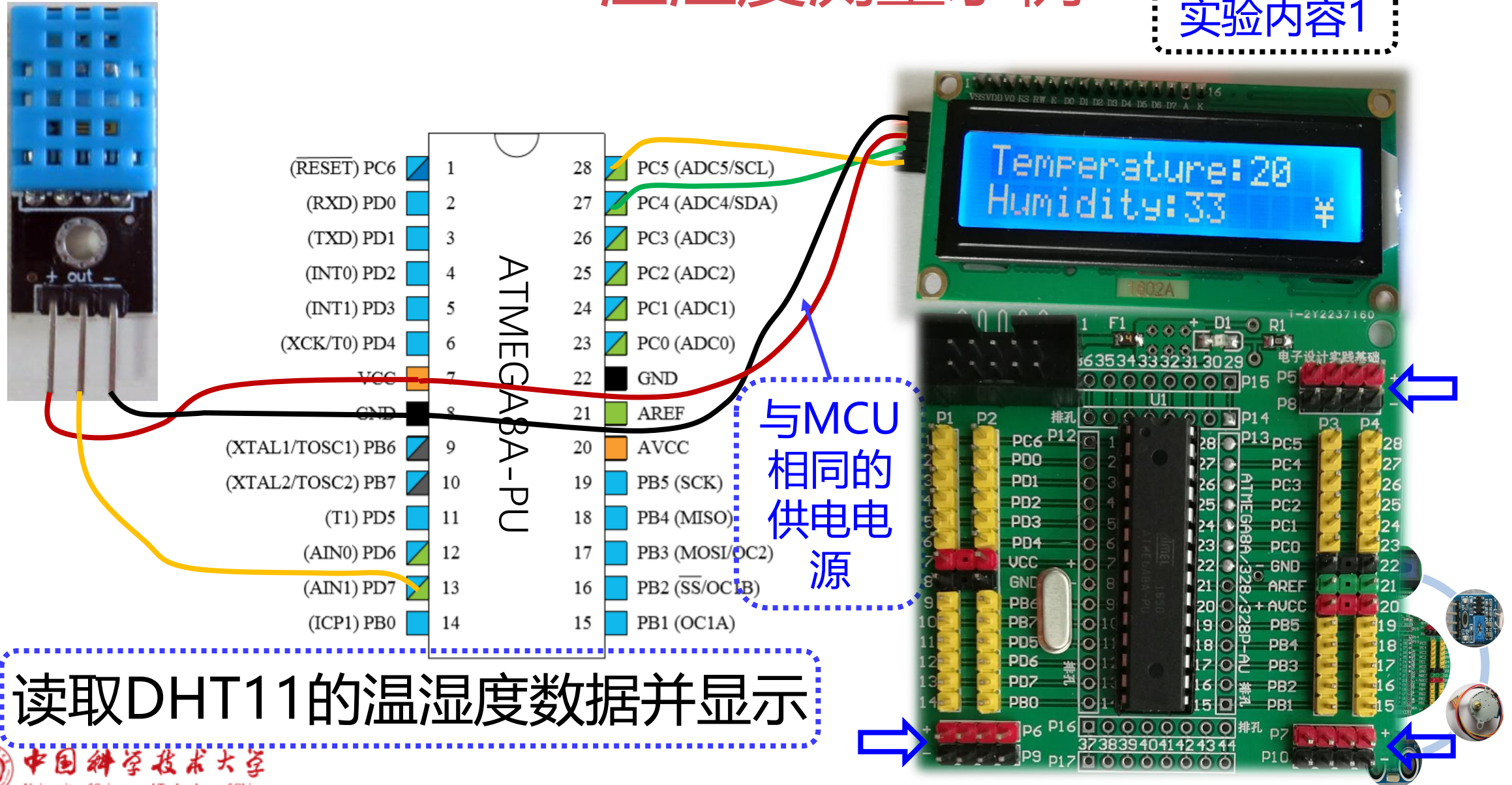
DHT11 40位温湿度数据的构成

- 40位数据
- 为上次采集结果：采集间隔最小2秒
- 先传最高位



DHT11温湿度测量示例

实验内容1



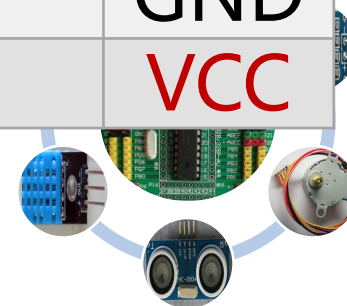
DHT11温湿度测量编程-DHT11头文件(1)

实验内容1

DHT11
控制与
数据传
输函数
写在头
文件
dht11.h
里，以
便后期
使用

```
/*  
 * dht11.h  
 * DHT11温湿度传感器数据的读取，使用ATMEGA8A PD7  
 * Created: 2023/4  
 * Author: lyh  
 */  
  
#ifndef DHT11_H_  
#define DHT11_H_  
#ifndef F_CPU  
#define F_CPU 1000000UL  
#endif  
#include <avr/io.h>  
#include <util/delay.h>
```

MCU与DHT11和LCD的连接			
DTH11	MCU	LCD1602	MCU
+	VCC	SCL	SCL
Out	PD7	SDA	SDA
-	GND	GND	GND
		VCC	VCC



DHT11温湿度测量编程-DHT11头文件(2)

```
unsigned char byteReadDHT11(void) //从DHT11读取一个字节的数据
{
    unsigned char oneBit,oneByte=0; //每次接收1位，最后完成8位接收并返回
    unsigned char i,uc_cnt; //循环变量，超时计数（脉冲宽度统计）
    for(i=0;i<8;i++) //接收8位数据，先接收的是最高位
    {
        //1bit由50us的低电平开始，后跟26~28us的高电平为0，70us的高电平为1
        uc_cnt = 1; //超时计数初值，统计50us低电平
        while((PIND & (1<<PIND7))==0) //在50us的低电平
        {
            uc_cnt++; //超时计数
            if(uc_cnt==0) break; /*计数溢出：超时，跳过*/
        }
        _delay_us(30); //跳过30us的高电平，如还是高电平收到'1'，否则收到'0'
        oneBit = 0; //假定收到'0'
        if((PIND & (1<<PIND7))!=0) oneBit = 1; //还是高电平，收到'1'
        uc_cnt = 1; //超时计数初值，统计70us高电平
        while((PIND & (1<<PIND7))!=0) //在70us的高电平
        {
            uc_cnt++; /*超时计数*/
            if(uc_cnt==0) break; /*计数溢出：超时，跳过*/
        }
        oneByte <<=1; //收到1位后，之前收到的为高位，故左移一位
        oneByte |= oneBit; /*新收到的合并到字节中*/
    }
    return oneByte; /*返回收到的1字节*/
}
```

禁止
任何
中断



实验内容1



DHT11温湿度测量编程-DHT11头文件(3)

```
void DHT11_Run(unsigned char * uc_data) //与DHT11通信的设置和数据传输等
{
    unsigned char uc_cnt; //超时计数 (脉冲宽度统计)
    DDRD |= (1<<DDRD7); //PD7为输出模式
    PORTD &= ~(1<<PORTD7); /*PD7输出0*/ _delay_ms(20); //至少18ms的START信号
    PORTD |= (1<<PORTD7); /*PD7输出1:20~40us*/ _delay_us(20); //20us高电平
    DDRD &= ~(1<<DDRD7); /*PD7为输入模式*/ _delay_us(20); //等待DHT11响应
    if((PIND & (1<<PIND7)) != 0) return; //DHT11没有响应
    uc_cnt = 1; //超时计数初值, 统计80us低电平响应
    while((PIND & (1<<PIND7)) == 0) //在80us的低电平
    {
        uc_cnt++; //超时计数
        if(uc_cnt == 0) break; /*计数溢出: 超时, 跳过*/
    }
    uc_cnt = 1; //超时计数初值, 统计80us高电平响应
    while((PIND & (1<<PIND7)) != 0) //在80us的高电平
    {
        uc_cnt++; /*超时计数*/ if(uc_cnt == 0) break; /*计数溢出: 超时, 跳过*/
    }
    for(uc_cnt = 0; uc_cnt < 5; uc_cnt++) //接收DHT11发送的40位温湿度数据
    {
        uc_data[uc_cnt] = byteReadDHT11(); //0-湿度整/小数, 温度整/小数, 校验-4
    }
}

#endif /* DHT11_H_ */
```

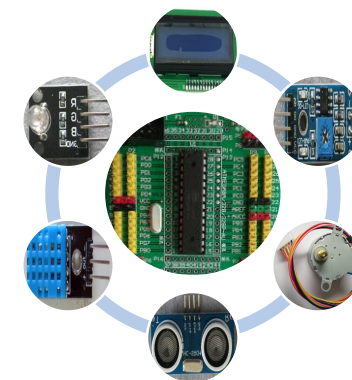
禁止
任何
中断

DHT11温湿度测量编程-主程序(1)

实验内容1

```
#include <avr/io.h>
#include "twi_lcd.h"
#include "dht11.h"

int main(void)
{
    unsigned char toggle=1,dht11_data[5]={0};
    TWI_Init();
    LCD_Init();
```



DHT11温湿度测量编程-主程序(2)

实验内容1

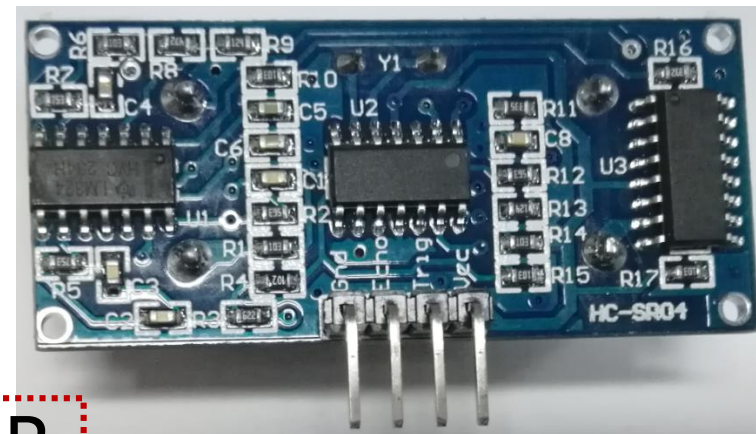
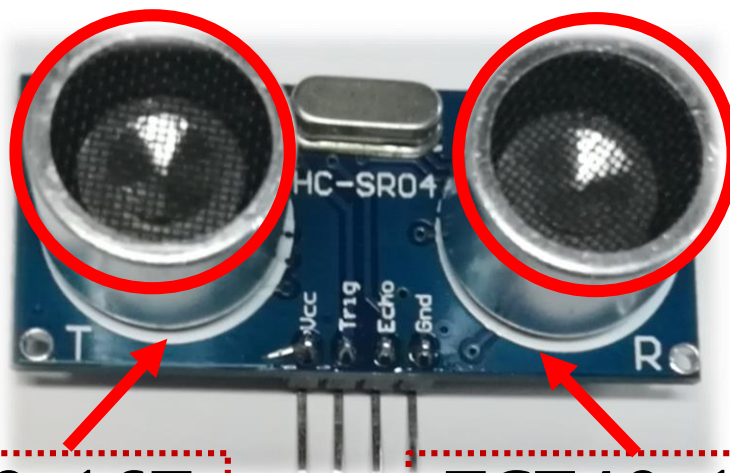
```
while (1)
{
    DHT11_Run(dht11_data);
    if(dht11_data[4]==(dht11_data[0]+dht11_data[1]+dht11_data[2]+dht11_data[3]))
    {
        LCD_Write_String(0,0,"Temperature:");//
        LCD_Write_NewChar(dht11_data[2]/10%10+0x30);
        LCD_Write_NewChar(dht11_data[2]%10+0x30);
        LCD_Write_String(1,0,"Humidity:");//
        LCD_Write_NewChar(dht11_data[0]/10%10+0x30);
        LCD_Write_NewChar(dht11_data[0]%10+0x30);
    }
    if(toggle)
    {
        LCD_Write_Char(1,15,0x5c);//显示符号
        toggle = 0;}
    else
    {
        LCD_Write_Char(1,15,0x20);//不显示符号
        toggle = 1;}
    _delay_ms(1500);
}
```



超声波收发模块：HC_SR04

■超声波

- 大于20KHz声波
- 距离测量等

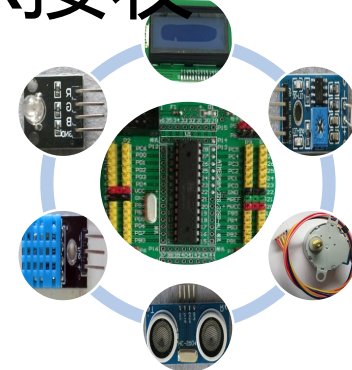


■HC_SR04

TCT40-16T

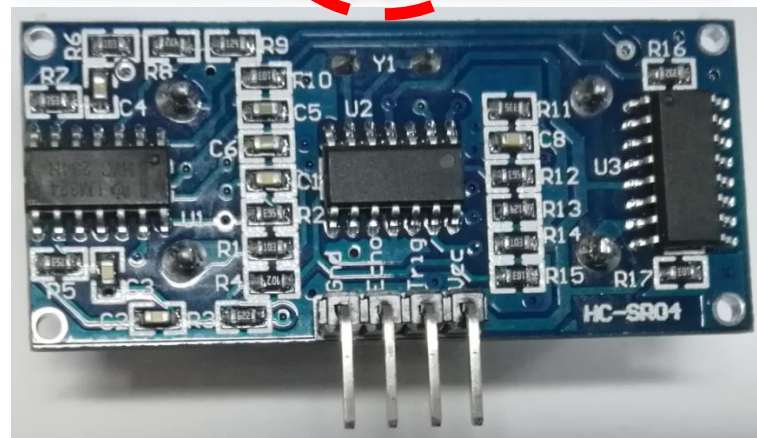
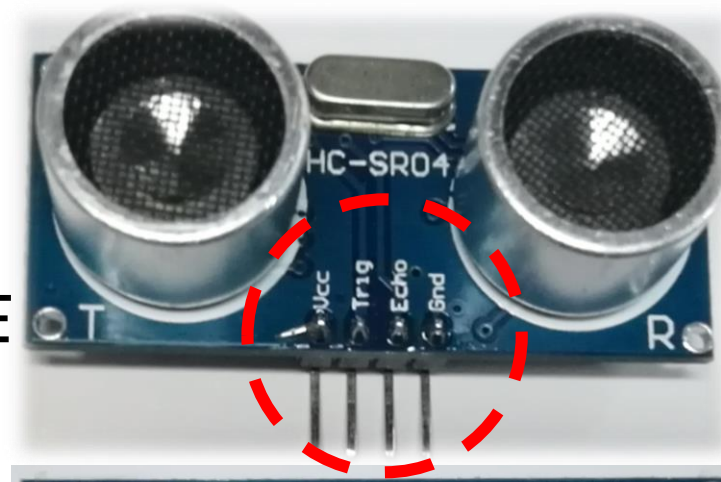
TCT40-16R

- TCT40-16T和TCT40-16R及其相关电路构成
- 从T发送的超声波在遇到被测物体反射回来后被R接收
- $2 \times \text{距离} = \text{(接收时刻 - 发送时刻)} \times \text{声速}$



超声波收发模块：HC_SR04

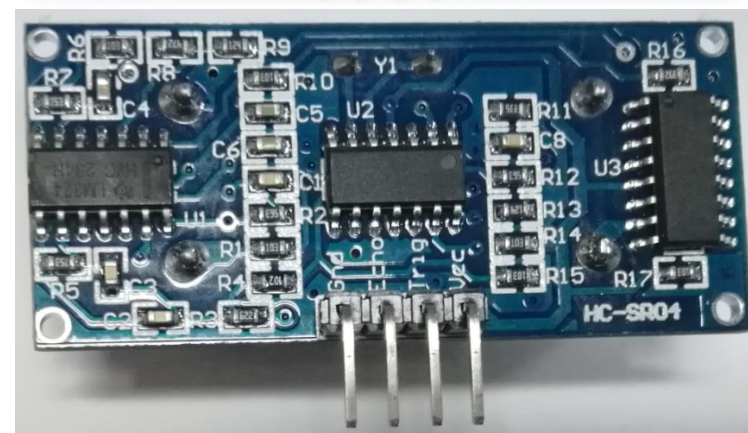
- 4管脚：Vcc, Trig, Echo, Gnd
- +5V工作电压, Trig-触发, Echo-回波指示
- 测距范围：2~700cm, $\pm 0.3\text{cm}$
- 注意
 - 不要带电连接，否则会影响正常工作
 - 被测物体的表面积不少于0.5平方米，否则会影响测量结果



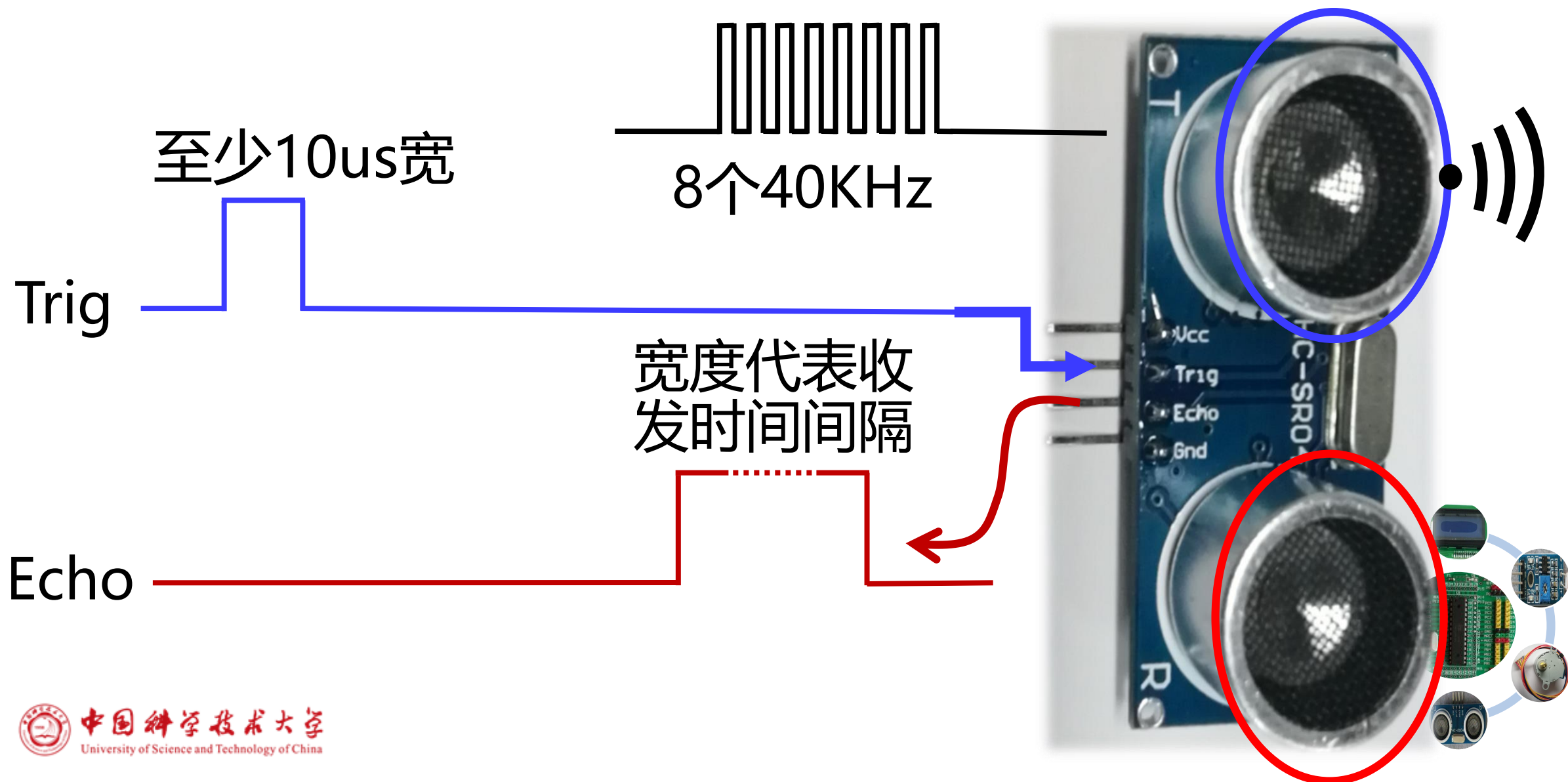
超声波收发模块：HC_SR04

■ 工作过程

- 给Trig管脚一个至少10us的高电平，开启测量
- HC_SR04发出超声波，并检测回波，若有回波，就通过Echo管脚输出高电平，其高电平持续时间就是超声波发送和返回的时间
- 距离 = $(\text{Echo高电平时间} \times 340\text{m/s}) / 2$

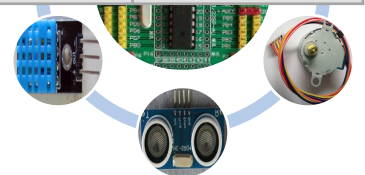
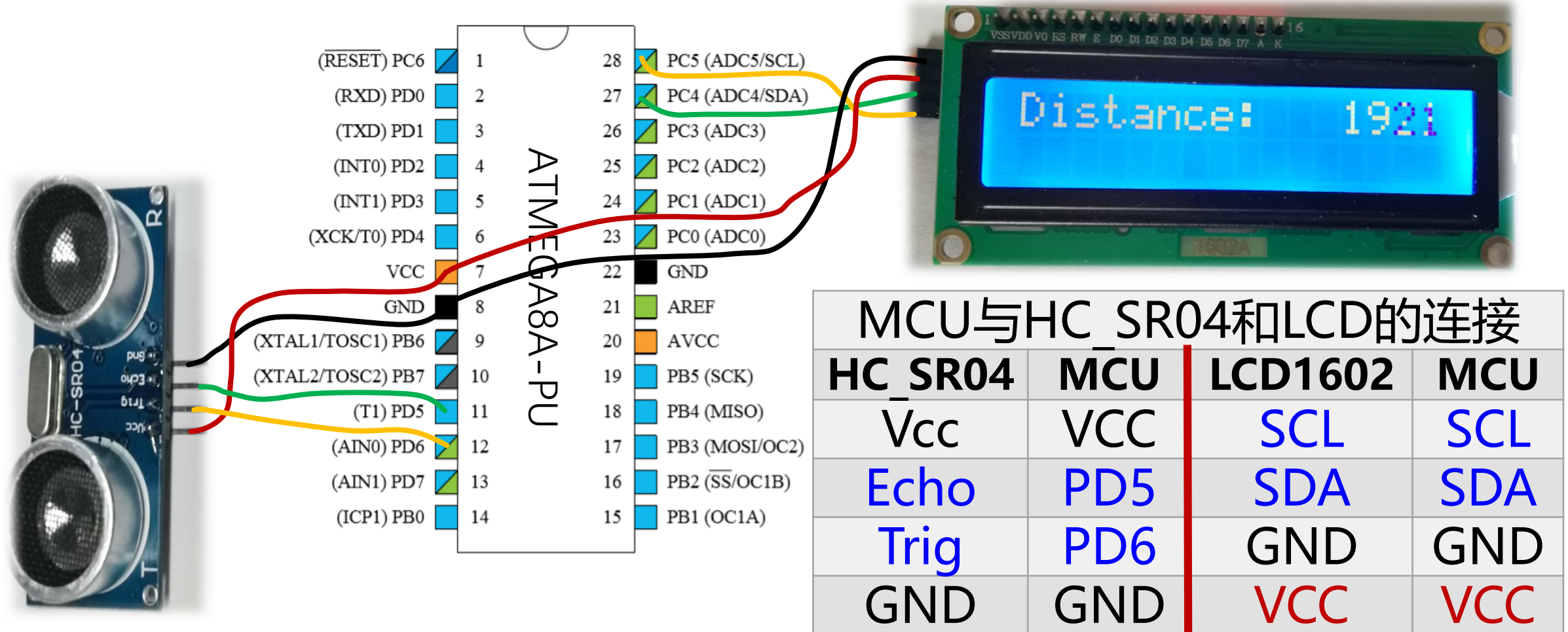


超声波收发模块：HC_SR04



超声波收发模块(HC_SR04)测距示例

实验内容2

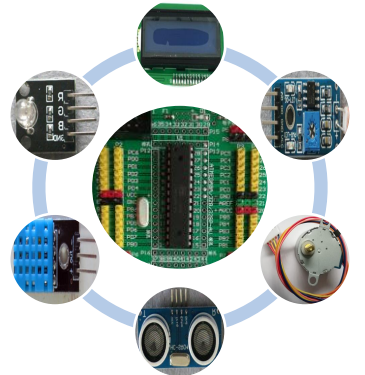


HC_SR04测距示例的编程-hcsr04.h头文件(1)

实验内容2

HCSR04控制
函数写在头
文件hcsr04.h
里，方便后
期使用

```
#ifndef HC_SR04_H_
#define HC_SR04_H_
#ifndef F_CPU
#define F_CPU 1000000UL
#endif
#include <avr/io.h>
#include <util/delay.h>
```



HC SR04测距示例的编程-hcsr04.h头文件(2)

实验内容2

```
unsigned int HCSR04_Run(void)
{
    unsigned int i_cnt=0; //统计echo的高电平(固定时钟下的次数)
    unsigned int uc_cnt=1; //超时计数 (统计在一定时间内没有响应)
    /*为了测量的准确, 如系统中使用了中断, 这里要禁止中断 cli(); */
    DDRD |= (1<<DDRD6); //PD6为输出 (到HC_SR04的Trig管脚)
    DDRD &= ~(1<<DDRD5); //PD5为输入 (来自HC_SR04的Echo管脚)
    PORTD |= (1<<PORTD6); //PD6输出高电平到Trig告诉HC_SR04准备发送超声波
    _delay_us(20); //持续20us(>10us)
    PORTD &= ~(1<<PORTD6); //PD6输出低电平到Trig结束通知
    while((PIND&(1<<PIND5))==0) //等待Echo为高电平(即收到超声回波)
    {
        uc_cnt++;
        if(uc_cnt>9000) break; //计数溢出: 超时, 跳过
    }
    i_cnt = 2; //统计Echo的初值=检测到echo信号+循环的判断约2个CPU时钟周期
    while((PIND&(1<<PIND5))!=0) //继续统计ECHO信号高电平的持续时间
    {
        i_cnt++; //每次=数据加载+&运算+判断+循环+加约5个时钟周期 (默认1MHz)
    }
    /*若之前禁用了中断, 这里可以开中断了 sei(); */
    return(i_cnt*5.0/100.0*17.0); //返回距离, 单位mm
}

#endif /* HC_SR04_H_ */
```

禁止
任何
中断

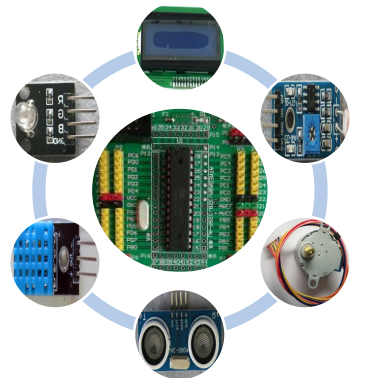


HC_SR04测距示例的编程-主程序

实验内容2

```
#include <avr/io.h>
#include "twi_lcd.h"
#include "hc_sr04.h"
..... //根据需要补充完整

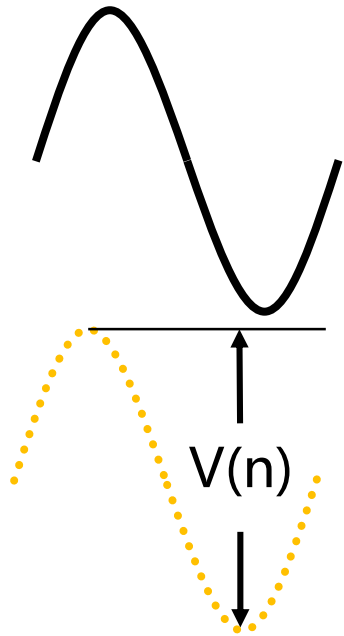
..... //补充完整
unsigned int distance=0;
unsigned char i=15,uc_d=0;
TWI_Init(); LCD_Init();
while (1)
{ distance = HCSR04_Run();
  LCD_Write_String(0,0,"Distance:");
  i=15; //从最低位开始转换和显示, 最多7位
  while(distance>0)
  { uc_d = distance % 10+0x30;
    LCD_Write_Char(0,i,uc_d);
    i--; distance /=10;}
  while(i>8) //高位没有数字时不显示
  { LCD_Write_Char(0,i,0x20);
    i--; }
  _delay_ms(100);//
}
..... //补充完整
```



ATMEGA8A 模拟数字转换器：ADC简述 (1)

Analog to Digital Converter

$$\blacksquare v(t) = U \sin(\omega t + \varphi)$$



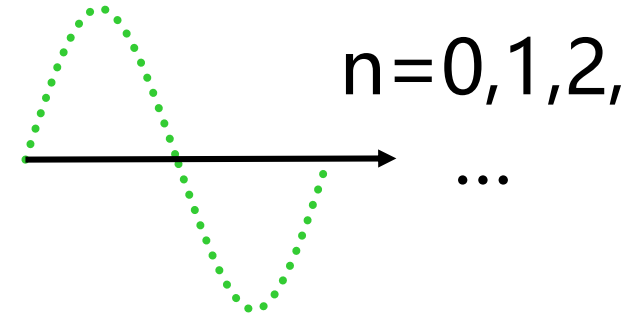
幅值离散

如量化幅度用10位二进制数表示:

$$V(n) = 0x000, 0x001, \dots, 0x3ff$$

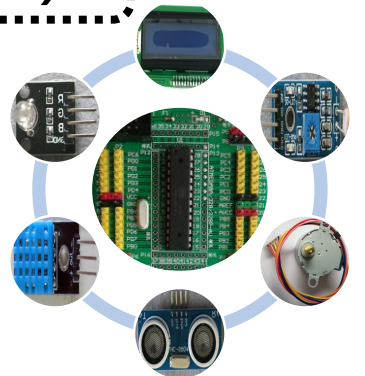
精度 (位数)

时间离散



采样 (速率)

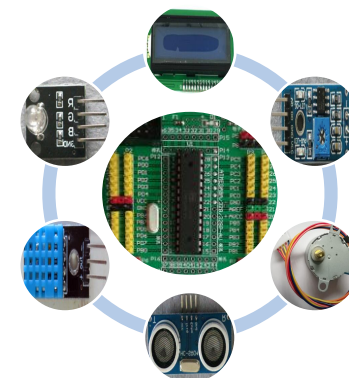
$$\bullet V(n) = U \sin(\omega n + \varphi)$$



ATMEGA8A 模拟数字转换器：ADC简述 (2)

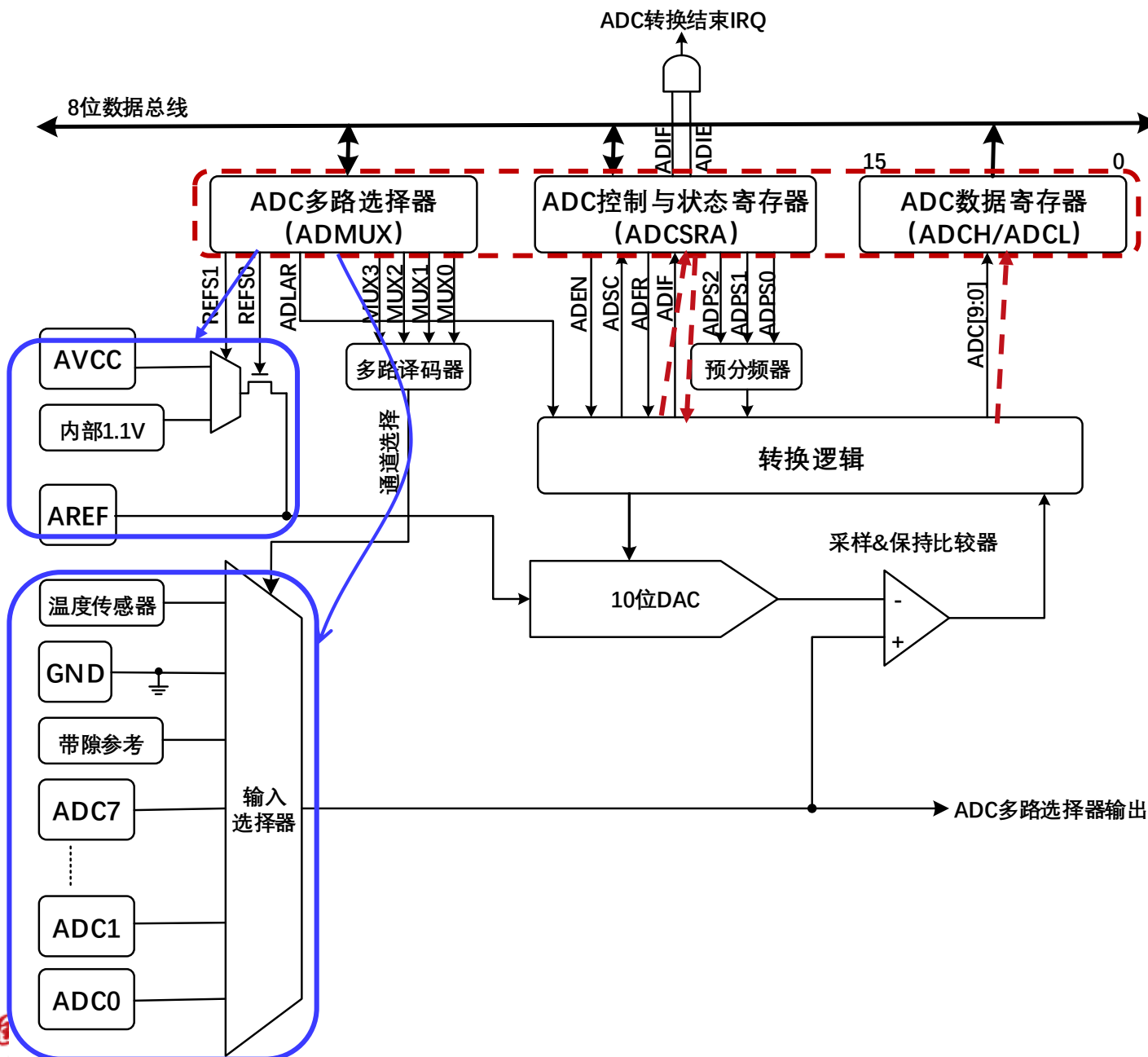
■ATMEGA8A的ADC

- 10位量化长度表示一个点
- 绝对精度： ± 2 LSB (LSB-Least Significant Bit)
- 转换时间：13~260us
- 最高采样速率15kSPS (SPS-Sample Per Second)
- 6路复用单端输入（其实仅1个ADC，可对6路之一采样）
- 输入电压范围：0~VCC (Max: +5V)
- 可选2.56V 参考电压
- 单次或连续转换模式；转换结果左对齐等。



ADC结构

ADC转换结果 = $V_{in} * 1024 / V_{ref}$



(RESET) PC6	1	28	PC5 (ADC5/SCL)
(RXD) PD0	2	27	PC4 (ADC4/SDA)
(TXD) PD1	3	26	PC3 (ADC3)
(INT0) PD2	4	25	PC2 (ADC2)
(INT1) PD3	5	24	PC1 (ADC1)
(XCK/T0) PD4	6	23	PC0 (ADC0)
VCC	7	22	GND
GND	8	21	AREF
(XTAL1/TOSC1) PB6	9	20	AVCC
(XTAL2/TOSC2) PB7	10	19	PB5 (SCK)
(T1) PD5	11	18	PB4 (MISO)
(AIN0) PD6	12	17	PB3 (MOSI/OC2)
(AIN1) PD7	13	16	PB2 (SS/OC1B)
(ICP1) PB0	14	15	PB1 (OC1A)

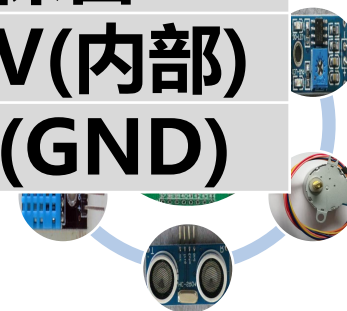
ATMEGA8A 模拟数字转换器：ADC选择寄存器

Bit	7	6	5	4	3	2	1	0
ADMUX	REFS1	REFS0	ADLAR		MUX3	MUX2	MUX1	MUX0
Access	R/W	R/W	R/W		R/W	R/W	R/W	R/W
Reset	0	0	0		0	0	0	0

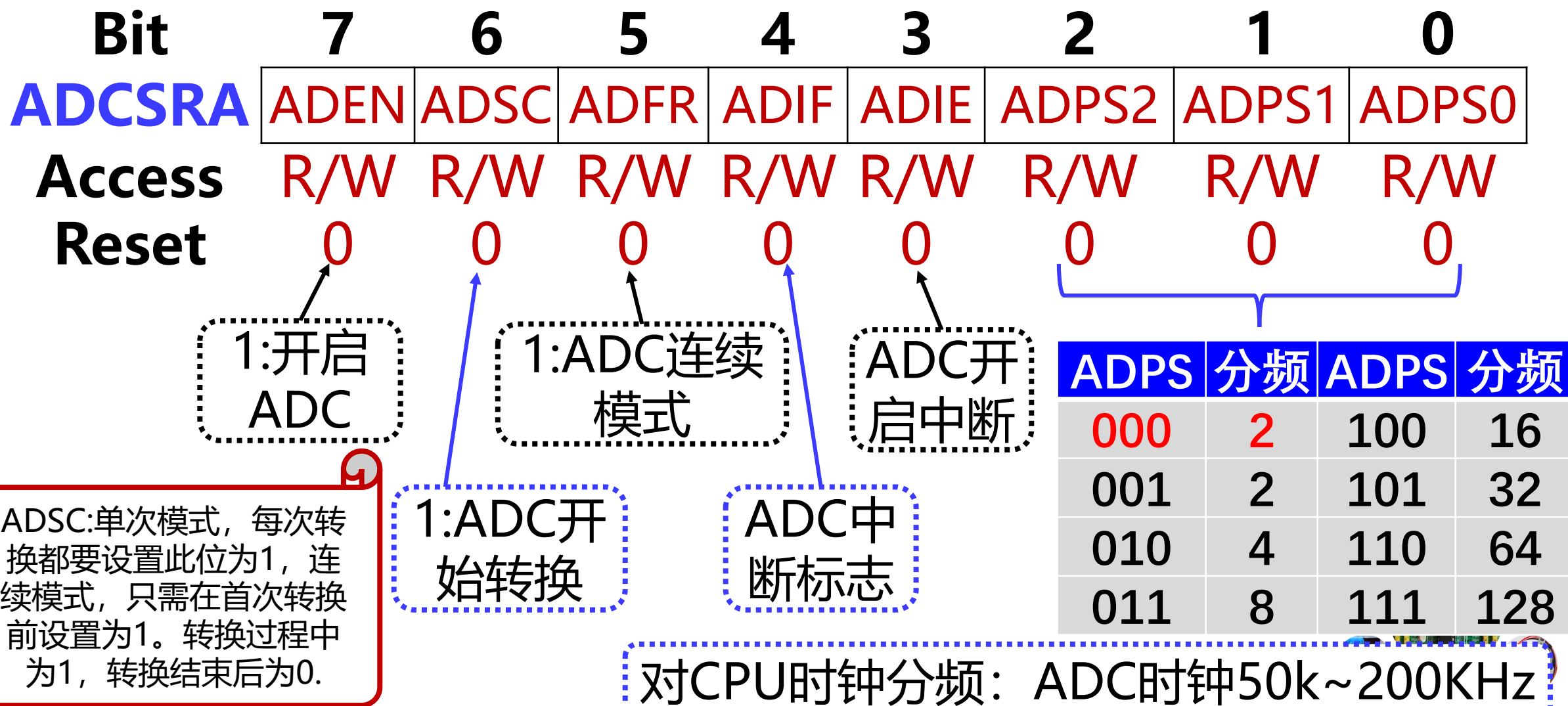
REFS1	REFS0	参考电压
0	0	AREF
0	1	AVCC
1	0	保留
1	1	2.56V

1:转换结果左对齐

MUX[3:0]	ADC输入
0000~0111	ADC0~7
1000~1101	保留
1110	1.3V(内部)
1111	0V(GND)



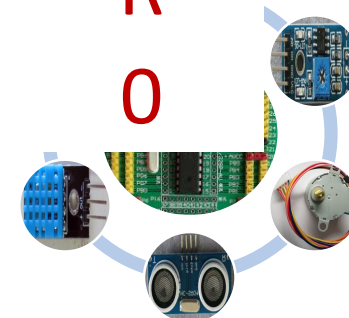
ATMEGA8A 模拟数字转换器：ADC控制状态寄存器



ADSC:单次模式，每次转换都要设置此位为1，连续模式，只需在首次转换前设置为1。转换过程中为1，转换结束后为0。

ATMEGA8A 模拟数字转换器：ADC数据寄存器

位	7	6	5	4	3	2	1	0
ADCH							ADC9	ADC8
读写	R	R	R	R	R	R	R	R
复位	0	0	0	0	0	0	0	0
	ADLAR=1						ADLAR=0	
位	7	6	5	4	3	2	1	0
ADCL	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0
读写	R	R	R	R	R	R	R	R
复位	0	0	0	0	0	0	0	0



感光模块简介

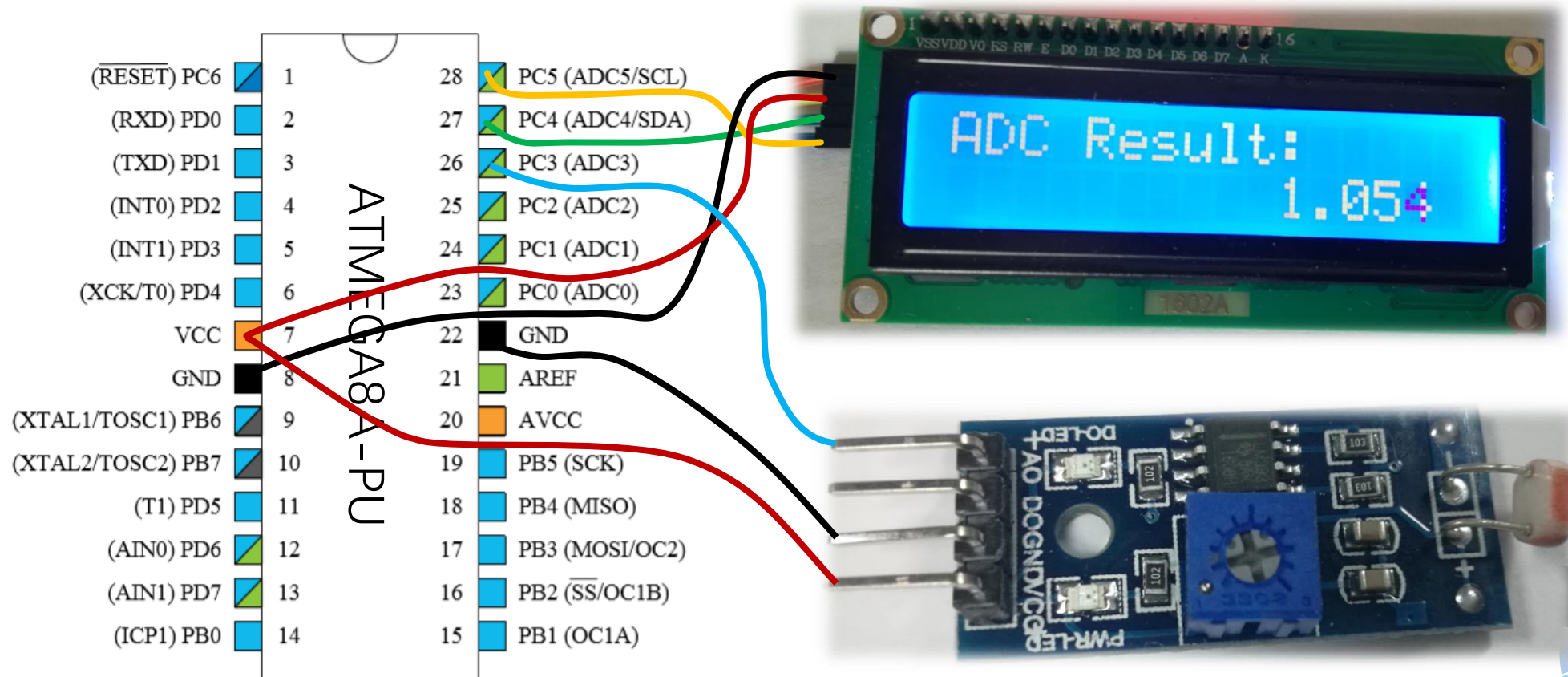
- 1, 光敏电阻：根据光线(可见光)其阻值会产生变化(NTP)，影响电压或电流
- 2, 比较器：比较两者电压，如比设定电压高则输出高电平，否则输出低电平
- 3, 模块有四个管脚，电源：Vcc和GND，输出：AO(光敏电阻上的模拟电压输出)，DO（输出高低电平：0或1）



ATMEGA8A 模拟数字转换器示例

实验内容3

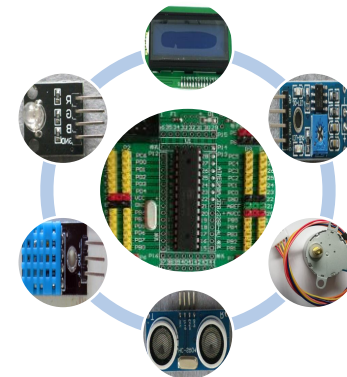
利用ATMEGA8A的ADC测量光敏电阻上的电压



ATMEGA8A 模拟数字转换器示例

实验内容3

MCU与光敏和LCD模块的连接			
光敏模块	MCU	LCD1602	MCU
Vcc	VCC	SCL	SCL
GND	GND	SDA	SDA
DO		GND	GND
AO	ADC3	VCC	VCC



ATMEGA8A 模拟数字转换器示例编程

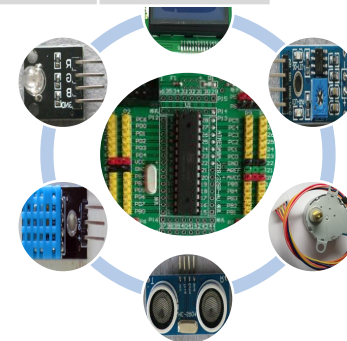
实验内容3

```
#include <avr/io.h>
#include "twi_lcd.h"
void ADC_Init(unsigned char adc_ch)//ADC初始化, 低4bit,adc0~7:0~7
{
    ADMUX = (1<<REFS0)|(adc_ch & 0x0f);//参考电压: AVCC, 低4bit为通道选择
    ADCSRA = (1<<ADEN)|(1<<ADFR)|(1<<ADPS1)|(1<<ADPS0);//开启ADC, 连续模式, 右对齐,
    //预分频8, 1000/8=125kHz

    ADCSRA |= (1<<ADSC);//ADC开始转换
}
```

```
int main(void)
{
    unsigned char adc_dh,adc_dl;
    //用于存储adch/1寄存器里的数据
    float adc_result;//adc采集结果
    unsigned char i,uc_int,uc_display[8];//临时变量
    LCD_Init();
    ADC_Init(3); //初始化adc, 对ADC3 (PC3) 采样, 参考电压...
```

ADPS	分频	ADPS	分频
000	2	100	16
001	2	101	32
010	4	110	64
011	8	111	128

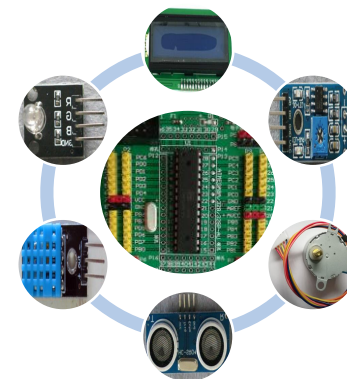


ATMEGA8A 模拟数字转换器示例编程

实验内容3

```
while (1)
{ while(!(ADCSRA & (1<<ADIF))); //等待ADC转换结束
  adc_d1 = ADCL; //先读低8位
  adc_dh = ADCH; //再读高8位
  adc_result = adc_dh *256.0+adc_d1; //合并
  adc_result *=5.0/1024.0; //计算电压

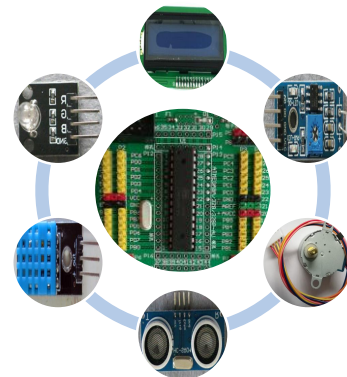
  for(i=0;i<8;i++) //初始化显示变量, 都不显示
  { uc_display[i]=0x20; }
  LCD_Write_String(0,0,"ADC Result:");
  uc_int = (unsigned char) adc_result; //取整数
  adc_result -=uc_int; //取小数
  i=2; //整数占2位
  uc_display[3]=0x30; //初始整数为0
  while(uc_int > 0 && i >0)
  { uc_display[i--]=uc_int%10+0x30; //最低位数字转换成字符
    uc_int/=10; /*去掉整数的最低位*/ }
```



ATMEGA8A 模拟数字转换器示例编程

实验内容3

```
uc_display[3]='.'; //小数点
i=4; //小数从元素4开始存储
while(adc_result>0 && i<7) //三位小数
{
    adc_result *=10; //第一位小数调整到整数
    uc_int = (unsigned char) adc_result; //取整数
    adc_result -=uc_int; //剩下的小数部分
    uc_display[i++]=uc_int+0x30; //当前的小数为转换为字符以显示
}
uc_display[7]=0; //字符结束
LCD_Write_String(1,8,uc_display); //显示采集转换后的电压数值
_delay_us(300);
}
```



本周实验内容

实验内容1：测量温湿度并显示在LCD1602上

实验内容2：利用超声波测量距离并显示在LCD1602上

实验内容3：利用ADC测量光敏电阻两端的电压并显示在LCD1602上

当次完成一个实验当场验收一个，总结下次交

