## 六 附件

### 附录A 传感器类型简介

1串口I/O

是存储器数据进出的通道，RAM通过输入/ 输出电路与微处理器交换信息。在进行读操作时这一接口是存储器的输出端；在写入时这一端口是输入端。一线进出二用，用读/写控制线控制。存储器中的输入/输出端数目决定于存储单元的位数。

2 Uart类

通用同步/异步接收/发射器（Universal Synchronous/Asynchronous Receiver/Transmitter）,它即是同步的。也是异步的！它有一个接收器，还有一个发射器！但它就是一个串口而已。

3 I2C类

I2C外围设备的SDA(Serial Data,串行数据)和SCL(Serial Clock,串行时钟)。IIC另一种写法，又叫Inter-IC（Inter-Integrated Circuit bus,IC间总线），是飞利浦（NXP）开发的一种通信协议，最初用在电视机中以减少芯片之间的连线。

4 SPI类

同步串行外设接口(SPI - Serial Peripheral Interface)是由摩托罗拉公司开发的全双工同步串行总线，该总线大量用在与EEPROM、ADC、FRAM和显示驱动器之类的慢速外设器件通信。

SPI（Serial Peripheral Interface）是一种串行串行同步通讯协议，由一个主设备和一个或多个从设备组成，主设备启动一个与从设备的同步通讯，从而完成数据的交换。SPI 接口由SDI（串行数据输入），SDO（串行数据输出），SCK（串行移位时钟），CS（从使能信号）四种信号构成，CS 决定了唯一的与主设备通信的从设备，如没有CS 信号，则只能存在一个从设备，主设备通过产生移位时钟来发起通讯。通讯时，数据由SDO 输出，SDI 输入，数据在时钟的上升或下降沿由SDO 输出，在紧接着的下降或上升沿由SDI 读入，这样经过8/16 次时钟的改变，完成8/16 位数据的传输。

### 附录B 数值换算

数值转换

1其他进制转换为十进制方法是，将其他进制安位展开，然后各项相加，就得到相应的十进制数。

例1：将二进制数（11011.101）2转换成十进制。

（11011.101）2 =1x24+1x23+0x22+1x21+1x20+1x2-1+0x2-2+1x2-3

=16+8+0+2+1+0.5+0+0.125 =（27.625）10

例2：将八进制数（136.524）8转换成十进制。

（136.524）8=1x82+3x81+6x80+5x8-1+2x8-2+4x8-3

=64+24+6+0.625+0.03125+0.0078125=（94.6640625）10

例3：将十六进制数（FF）16转换成十进制。

（FF）16=15x161+15x160=240+15=（255）10

例4：将十六进制数（13DF.B8）16 =（13DF.B8）H转换成十进制。

（13DF.B8）16 =1x163+3x162+13x161+15x160+11x16-1+8x16-2

=4096+768+208+15+0.6875+0.03125=（5087.71875）10

2十进制转换成其他进制

1. 整数部分（基数除法）。把要转换的 十进制数除以新的进制的基数，把余数作为新的进制的最低位。把所得的新的商再除与新的进制的基数，把余数作为新进制的次地位，继续上一步，直至商为零，这是余数就是进制的最高位。
2. 小数部分（基数乘法）。把要转换数的小数部分乘以新的进制的基数，把得到的整数部分作为新进制小数部分的最高位；把上一步得的小数部分再乘以新进制的基数，把整数部分作为新进制小数部分的次高位；继续上一步，直到小数部分变成零为止。或者到达预定的要求也可以。

例5：把十进制（25）10转换成二进制。

由于二进制基数为2，所以逐次除以2取其余数（0或1），即

|  |  |  |
| --- | --- | --- |
| 2  2  2  2  2 | 25 |  |
| 12 | ……余1 |
| 6 | ……余0 |
| 3 | ……余0 |
| 1 | ……余1 |
| 0 | ……余1 |

（25）10 =（11001）2，即25D=11001B。

例6：把十进制数173D转换成二进制数。

|  |  |  |
| --- | --- | --- |
| 2  2  2  2  2  2  2  2 | 173 |  |
| 86 | ……余1 |
| 43 | ……余0 |
| 21 | ……余1 |
| 10 | ……余1 |
| 5 | ……余0 |
| 2 | ……余1 |
| 1 | ……余0 |
| 0 | ……余1 |

（173）10=（10101101）2，即173D= 10101101B

例7:将十进制数（25）10转换成八进制。

|  |  |  |
| --- | --- | --- |
| 8  8 | 25 |  |
| 3 | ……余1 |
| 0 | ……余3 |

（25）10=（31）8，即25D=31O

例9：将十进制小数（0.375）10转换成二进制。

|  |  |
| --- | --- |
|  | 0.375 |
| x | 2 |
| [0]. | 750 |
|  | 2 |
| [1]. | 500 |
|  | 2 |
| [1]. | 000 |

（0.375）10=（0.011）2

3二进制与八进制和十六进制转换

八进制和十六进制的基数是8 =23、16=24，所以3位二进制数切好相当于一位八进制数，4位二进制数切好相当于一位十六进制数。

二进制数转换成八进制数的方法是，从小数点开始，分别向左、向右，将二进制数按每三位一组分组（不足三位补0），然后写出每一组的八进制数。

例10：求（0110 1111 010.1011）2的等值八进位数。

二进制001 101 111 010.101 100

八进制 1 5 7 2. 5 4

所以，（0110 1111 010.1011）2=（1572. 54）8

二进制数转换成十六进制数的方法是，从小数点开始，分别向左、向右，将二进制数按每四位一组分组（不足四位补0），然后写出每一组的十六进制数。

例11：(1101101011.101)2转换成十六进制。

二进制0011 0110 1011. 1010

十六进制3 6 B .A

所以，(1101101011.101)2=（36B.A）16

八进制、十六进制转换成二进制，正好可以相反，即按原来顺序将每一位八进制数（或十六位进制）相应用3位（或4位）二进制代替。

例12：将（678.A5）16转换成二进制。

十六进制 6 7 8 .A 5

二进制 0110 0111 1000 .1010 0101

所以，（678.A5）16=（0110 0111 1000 .1010 0101）2

一种心算法BCD码(即常用8421)。方法是将二进制分成前4位，后4位分开换算，先想象出4位二进制即（0x8）+（0x4）+（0x2）+（0x1），所得结果就是十进制数的0~9。

例如1：二进制为1100 0101，转换成十六进制。

1100 0101=（1x8）+（1x4）+（0x2）+（0x1）

+（0x8）+（1x4）+（0x2）+（1x1）

=c +5=C5。

BCD码与其他进制数对照表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 十进制 | 十六进制 | BCD码 | | | |
| 8 | 4 | 2 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 2 | 2 | 0 | 0 | 1 | 0 |
| 3 | 3 | 0 | 0 | 1 | 1 |
| 4 | 4 | 0 | 1 | 0 | 0 |
| 5 | 5 | 0 | 1 | 0 | 1 |
| 6 | 6 | 0 | 1 | 1 | 0 |
| 7 | 7 | 0 | 1 | 1 | 1 |
| 8 | 8 | 1 | 0 | 0 | 0 |
| 9 | 9 | 1 | 0 | 0 | 1 |
|  | A | 1 | 0 | 1 | 0 |
|  | B | 1 | 0 | 1 | 1 |
|  | C | 1 | 1 | 0 | 0 |
|  | D | 1 | 1 | 0 | 1 |
|  | E | 1 | 1 | 1 | 0 |
|  | F | 1 | 1 | 1 | 1 |

附录C

常用关键字

U8定义一个8位无符号的数据。其能记录的数据范围为[0~255]。

U16定义一个16位无符号的数据。其能记录的数据范围为[0~65535]。

U32定义一个32位无符号的数据。其能记录的数据范围为[0~4294967295]。

S8定义一个8位有符号的数据。其能记录的数据范围为[-128~127]。

S16定义一个16位有符号的数据。其能记录的数据范围为[-32768~32767]。

S32定义一个32位有符号的数据。其能记录的数据范围为[-2147483648~2147483647]。

float定义一个32位浮点的数据。其能记录的数据范围为[-3.4\*10的负38~3.4\*10的38次方]。

enum定义枚举类型，用于补充u8、u16等数据类型，使程序看起来更简洁。常用于定义有穷个数且有规律的数据，比如：星期一到星期天，各个元素的值都是在上一个值加"1"。例如:

enum weeks =

{

monday=0x00,

tuesday,

wennesday,

thursday,

friday,

saturday,

sunday

};

struct定义结构体类型，用于补充u8、u16等数据类型，使程序看起来更简洁。常用于定义和某个对象相关的数据，比如：时间分为年月日时分秒。例如:

struct timmer =

{

u8 year;

u8 month;

u8 day;

u8 week;

u8 hour;

u8 minute;

u8 second;

};

void用于提示用户该位置为"空"的类型，用户在写程序的时候碰到的时候只需要将该字删除，不要操作。

break用于跳出某个函数，比如当达到一定条件，不需要再运行本程序的时候，使用此关键字就可以达到目的。

switch选择判断执行，常用于判断某个事件，而该事件又可以列举出。根据结果来执行相应事件中的程序。例如:

switch ()

{

case A:

do sth;

break;

case B:

do sth;

break;

default:

do sth;

break;

}

case匹配条件。常和switch一起使用。详情见switch关键字。

Default匹配条件。常和switch一起使用。详情见switch关键字。

for循环控制关键字。例如循环10次:

u8 a;

for (a=0;a<10;a++)

{

do sth;

}

while循环控制关键字。例如使用while来实现死循环:

while(1)

{

do sth;

}

if条件判断关键字，常和else关键字在一起使用。例如:

u8 a, b;

if (a>b)

{

do sth;

}

else

{

do sth;

}

else其他条件匹配。常和if关键在一起使用。详情见if关键字。

return函数返回的一个结果。

### 附录D函数概念

**程序概念**

/\*

"函数由:返回值类型、函数名、输入参数类型、以及函数体构成

一般写法:

返回值类型 函数名（输入参数）

{

函数体;

}

一般有四种方式：

1. 无返回，无传入参数

2. 无返回，有输入参数

3. 有返回，无输入参数

4. 有返回，有输入参数

实例:

void func (void)

{

do sth.;

}//定义的一个没有返回值，没有输入参数的函数名为func的函数"

\*/

**顺序执行**

/\*

"程序按正常顺序执行，中间不会发生其他任何事情"

**选择执行**

/\*

"程序选择性执行，中间会出现判断的语句，比如（if）等字样"

\*/

**循环执行**

/\*

"程序按用户设定的循环次数执行相同"

\*/

**函数调用方式**

/\*

"根据函数定义方式，然后使用相应方式去调用函数

如果遇到，无返回或无传入参数的时候需要将(void)关键字去掉

如果遇到有参数的情况，需要将相应的位置定义成相同参数类型的参数进行调用。

比如有返回值调用方法:Val=func();

有参数传入为:func(Val);

无传入无返回:func();"

### 附录E字符格式表达

格式字符 指定输出项的数据类型和输出的格式。

d 十进制整数

o 无符号八进制数

x 无符号十六进制数

u 不带符号的十进制整数