Temperature measurement application routine
V. 2.1
2018/7/2
New generation DS18B20

The temperature measurement digital chip has built-in high-speed ADC. It can achieve rapid temperature measurement - the highest update rate is 12ms, than the traditional DS18B20 750ms. Temperature measurement is much faster. In some common temperature measurement applications, in order to be compatible with temperature measurement and display equipment, it is recommended to add to the MCU program 500ms Delay waiting.

The following program provides a three-wire single-point temperature measurement application routine, the timing sequence is configured according to the most typical value.

```
int main(void)
{
float temp_result;
//
Decimal temperature measurement result
temp_result =DQ_Read_Temperature(temp);
//
Call temperature measurement function
printf("Temperature Get is: %f\n",temp_result);
//
Print temperature reading
}
float DQ_Read_Temperature()
//Temperature measurement, return result in demical
{
short s_tem;
//temporary variable
float result;
//measurement result
unsigned char temp[2];
//
Used to store raw readings
DQ_Rst();
DQ_Presence();
DQ_Write_Byte(0xcc);
//
Single point temperature measurement sending
skip ROM
instruction
DQ_Write_Byte (0x44);
//
Send temperature measurement instructions
DelayUS(500000);
//
For compatibility, it is recommended to delay
500ms
DQ_Rst();
DQ_Presence();
DQ_Write_Byte(0xcc);
   DQ_Write_Byte(0xbe); //
Send read temperature command
temp[0] = DQ_Read_Byte();
```

```
//
Low byte of temperature reading
LSB
temp[1] = DQ_Read_Byte();
//
Low byte of temperature reading
MSB
/ /
Convert binary reading to decimal
s_tem = temp[1]<<8;
s_tem = s_tem | temp[0];
s_tem = s_tem & 4095;
if( (s_tem & 2048) == 2048)
{
```

```
s_tem = (s_tem^4095) & 4095;
result = -1*(s_tem+1) * 0.0625;
}
else
result = s_tem * 0.0625;
return result;
}
Void DQ_RST (void)
{
// MCU IO
Set to push and pull
Pushpull
Stronger drive; open drain
Open Drain
also may
DQ = 0; // MCU IO
Output
0
Delay_us(600);
//
Delay
600us
, Typical range
: 480-960us
DQ = 1; // MCU IO
Output
1
Delay_us(15);
// MCU
wait
15us
There is a pulse
presence pulse
}
Void DQ_Presence(void)
{
//MCU IO
Set to input mode
, RST
After rising
60us
Left and right can check whether the pulse is low
```

```
Delay_us(200);
    //MCU
wait
200us
There is a pulse
}
unsigned char read_bit(void)
// read 1 bit
Subfunction
{
unsigned char result;
DQ = 0; //
Delay_us(5); // MCU
Delay
5us
DQ = 1; //
Delay_us(5); // MCU
Delay
5us
To sample sensor readings
result = DQ; // result
High reading
/
low
Delay_us(50); // MCU
wait
50us time slot
End of time slot
return(result); //
Return the character value
}
unsigned char DQ_Read_Byte(void)
// read 1 byte
Subfunction
{
unsigned char i, j;
unsigned char value = 0;
for (i=0;i<8;i++) {//
Read the low bit first, then the high bit
j = Read_bit();
```

```
value = (value) | (j<<i);
//
Read one at a time
bit
, Then move left
}
return(value);
//
Return the byte value
}
void write_bit(char bitval)
//
write
1-bit
Subfunction
, bitval
```

```
Value written for
{
DQ = 0; //
Delay_us(5);
//
if(bitval==1) DQ =1;
// if bit=1
,
then DQ
Output high
Delay_us (55);
//
Wait
55us
Let the sensor read
DQ = 1; // DQ
Output high
Delay_us (5);
//
Reserved
5us
Recovery Time
recovery time
}
void DQ_Write_Byte(char val)
//
write
1-byte
Subfunction
, val
Is the value to be written
{
unsigned char i;
unsigned char temp;
for (i=0; i<8; i++) {
temp = val>>i;
temp &= 0x01;
//
Write low byte first
(LSB)
write_bit(temp);
}
}
```