

```

1
2 #include "one_wire.h"
3 #include "ds18b20.h"
4
5
6 #define DS_QUANTITY      3
7
8
9 static LL_GPIO_InitTypeDef GPIO_InitStruct = {0};
10
11
12 static uint8_t dt[DS_QUANTITY][8];
13
14
15 static uint8_t DS18B20_Reset(void);
16 static uint8_t DS18B20_ReadBit(void);
17 static uint8_t DS18B20_ReadByte(void);
18 static void DS18B20_WriteBit(uint8_t bit);
19 static void DS18B20_WriteByte(uint8_t dt);
20
21 /* */
22 void DS18B20_PortInit(void) {
23
24     GPIO_InitStruct.Pin = LL_GPIO_PIN_12;
25     GPIO_InitStruct.Mode = LL_GPIO_MODE_OUTPUT;
26     GPIO_InitStruct.Speed = LL_GPIO_SPEED_FREQ_MEDIUM;
27     GPIO_InitStruct.OutputType = LL_GPIO_OUTPUT_OPENDRAIN;
28     GPIO_InitStruct.Pull = LL_GPIO_PULL_NO;
29     LL_GPIO_Init(GPIOB, &GPIO_InitStruct);
30
31 }
32
33
34 /* */
35 uint8_t DS18B20_Init(uint8_t mode) {
36
37     if(DS18B20_Reset()) return 1;
38
39     if( mode==SKIP_ROM ) {
40
41         DS18B20_WriteByte(0xCC);    // Skip ROM
42     }
43
44     DS18B20_WriteByte(0x4E);    // Write stratchpad
45
46     DS18B20_WriteByte(0x64);    // TH register = 100C
47     DS18B20_WriteByte(0x9E);    // TL register = -30C
48     //DS18B20_WriteByte(RESOLUTION_9BIT);
49
50     return 0;
51 }
52
53
54
55 /* */
56 static uint8_t DS18B20_Reset(void) {
57
58     uint8_t state = 0;
59
60     LL_GPIO_ResetOutputPin(GPIOB, LL_GPIO_PIN_12);
61     Delay_us(480);
62     LL_GPIO_SetOutputPin(GPIOB, LL_GPIO_PIN_12);
63
64     Delay_us(65);
65
66     state = ( LL_GPIO_IsInputPinSet(GPIOB, LL_GPIO_PIN_12) ? 1 : 0 );

```

```

67
68     Delay_us(420);
69
70     return state;
71 }
72
73
74
75 /* */
76 static uint8_t DS18B20_ReadBit(void) {
77
78     uint8_t bit = 0;
79
80     LL_GPIO_ResetOutputPin(GPIOB, LL_GPIO_PIN_12);
81     Delay_us(3);
82     LL_GPIO_SetOutputPin(GPIOB, LL_GPIO_PIN_12);
83
84     Delay_us(15);
85
86     bit = ( LL_GPIO_IsInputPinSet(GPIOB, LL_GPIO_PIN_12) ? 1 : 0);
87
88     Delay_us(50);
89
90     return bit;
91 }
92
93
94 /* */
95 static uint8_t DS18B20_ReadByte(void) {
96
97     uint8_t data = 0;
98
99     for (uint8_t i = 0; i <= 7; i++)
100
101         data += DS18B20_ReadBit() << i;
102
103     return data;
104 }
105
106
107
108 /* */
109 static void DS18B20_WriteBit(uint8_t bit) {
110
111     LL_GPIO_ResetOutputPin(GPIOB, LL_GPIO_PIN_12);
112     Delay_us(bit ? 3 : 65);
113
114     LL_GPIO_SetOutputPin(GPIOB, LL_GPIO_PIN_12);
115     Delay_us(bit ? 65 : 3);
116 }
117
118
119 /* */
120 static void DS18B20_WriteByte(uint8_t dt) {
121
122     for (uint8_t i = 0; i < 8; i++) {
123
124         DS18B20_WriteBit(dt >> i & 1);
125         Delay_us(5);
126     }
127 }
128
129
130 /* */
131 void DS18B20_MeasureTempCmd(uint8_t dev, uint8_t mode) {
132

```

```

133     DS18B20_Reset();
134
135     if(mode==SKIP_ROM) {
136
137         DS18B20_WriteByte(0xCC);    // Skip ROM
138     }
139
140     DS18B20_WriteByte(0x44);    // Convert T
141 }
142
143
144
145 /* */
146 void DS18B20_ReadStratchpad(uint8_t dev, uint8_t mode) {
147
148     uint8_t i;
149
150     DS18B20_Reset();
151
152     if(mode==SKIP_ROM) {
153
154         DS18B20_WriteByte(0xCC);    // Skip ROM
155     }
156
157     DS18B20_WriteByte(0xBE);    // Read stratchpad
158
159     for(i=0; i<8; i++) {
160
161         dt[dev][i] = DS18B20_ReadByte();
162     }
163 }
164
165
166
167 /* */
168 uint8_t DS18B20_GetSign(uint8_t dev) {
169
170     return(dt[dev][1] & 0x01) ? 1 : 0;
171 }
172
173
174 /* */
175 float DS18B20_Convert(uint8_t dev) {
176
177     float t = (dt[dev][1] & 0x07)<<4 | dt[dev][0]>>4;
178
179     if(dt[dev][0] & 0x0F) t += 0.5;
180
181     return t;
182 }

```