

MYIR-ZYNQ7000系列-zturn教程(12)：用i2c接口读取温度传感器STLM75

原创 虚无缥缈vs威武 最后发布于2018-03-03 11:34:08 阅读数 2149 ☆ 收藏

开发板环境：vivado 2017.1，开发板型号xc7z020clg400-1，这个工程主要用I2C接口读取STLM75的温度，同时也会对其它的相关寄存器进行读写以验证程序的正确性。

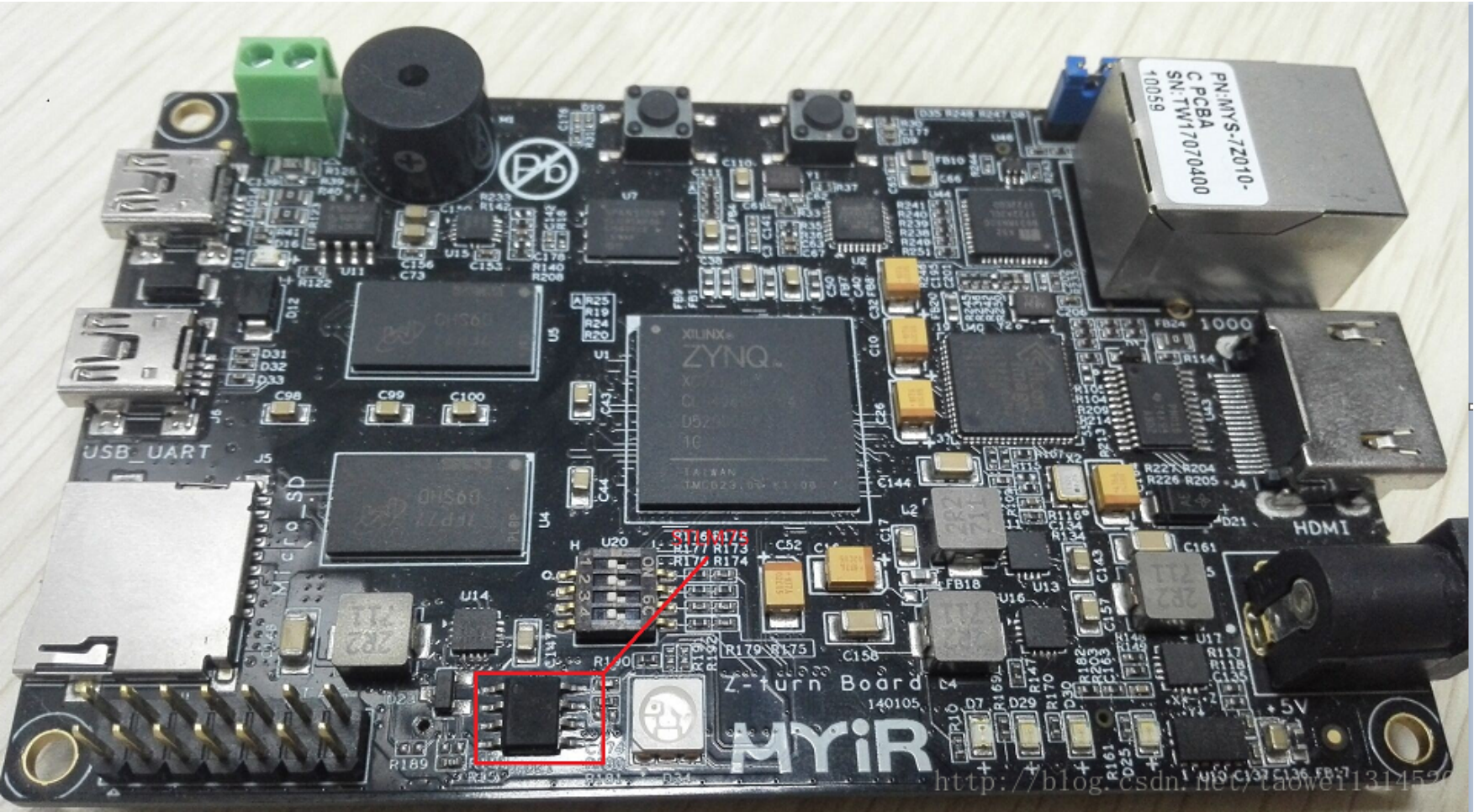
链接：<https://pan.baidu.com/s/1WyJmJ2ZoMmwHoN3gHBN65A> 提取码：uf8p

下面的这个截图是STLM75的一些管脚介绍：

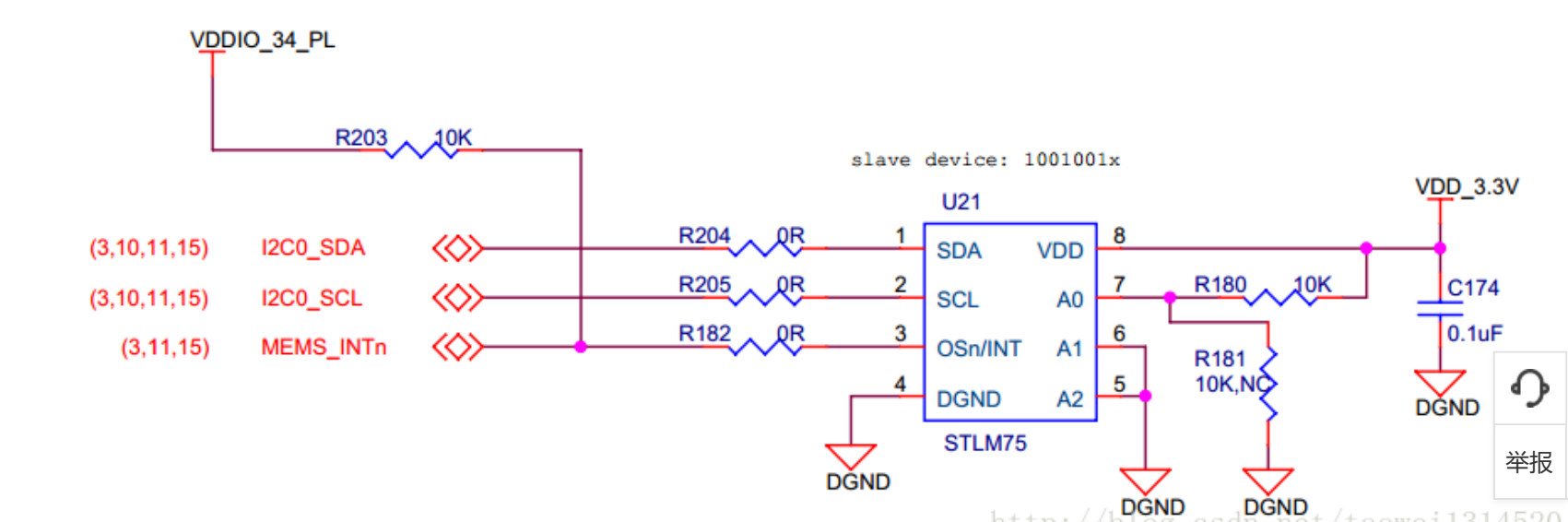
Table 1. Signal names			
Pin	Sym	Type/direction	Description
1	SDA ⁽¹⁾	Input/output	Serial data input/output
2	SCL	Input	Serial clock input
3	OS/INT ⁽¹⁾	Output	Overlimit signal/interrupt alert output
4	GND	Supply ground	Ground
5	A ₂	Input	Address2 input
6	A ₁	Input	Address1 input
7	A ₀	Input	Address0 input
8	V _{DD}	Supply power	Supply voltage (2.7 V to 5.5 V)

1. SDA and OS/INT are open drain.

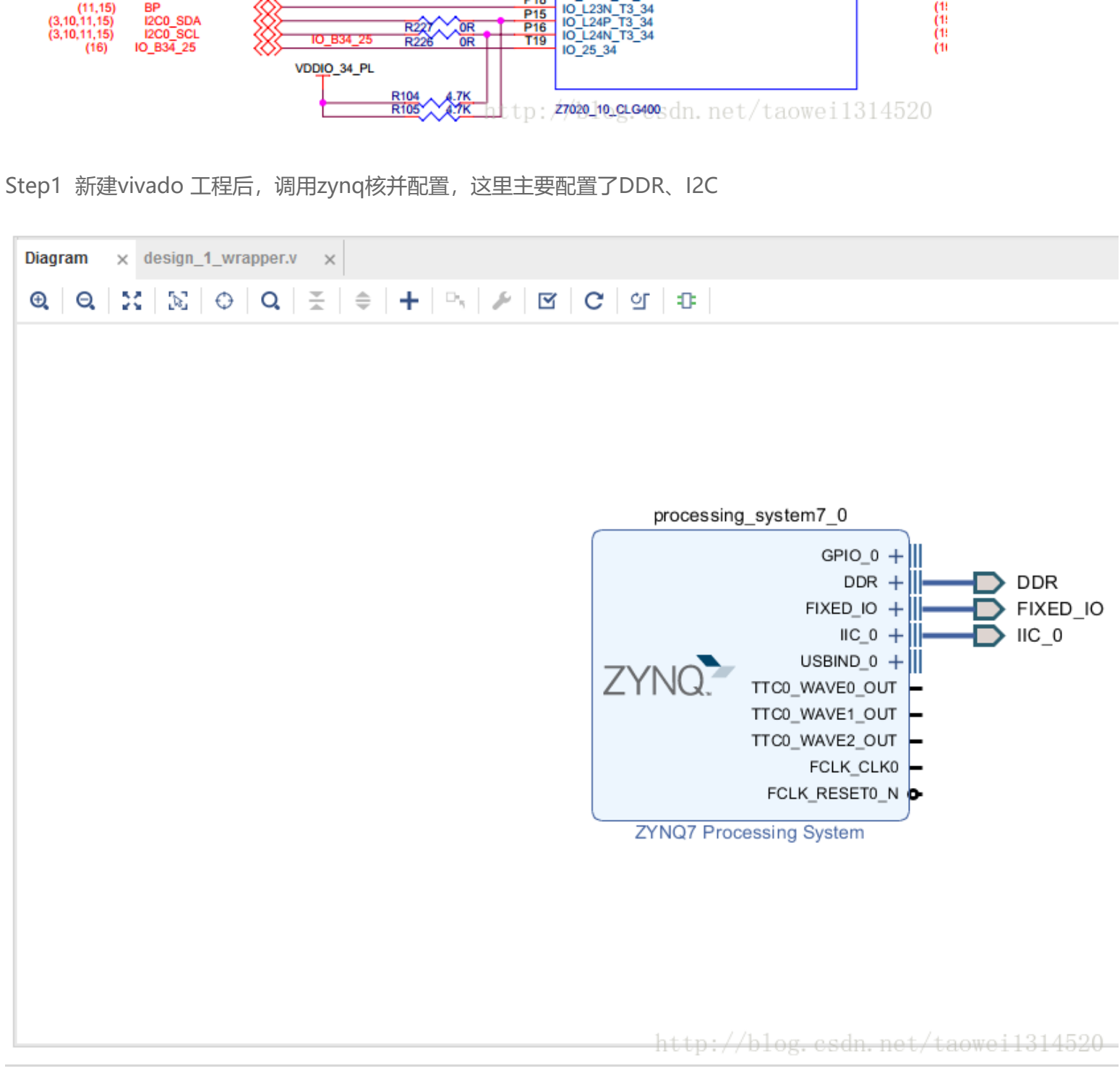
<http://blog.csdn.net/taowei1314520>



下面的截图是z-turn开发板上STLM75的硬件连接情况，可以看出这个STLM75的设备地址是 1001001x(最后一位是读写控制位)



<http://blog.csdn.net/taowei1314520>



Step1 新建vivado 工程后，调用zynq核并配置，这里主要配置了DDR、I2C

Step2 新建一个xdc文件


```
1 set_property PACKAGE_PIN P15 [get_ports iic_0_sda_io]
2 set_property PACKAGE_PIN P16 [get_ports iic_0_scl_io]
3 set_property IOSTANDARD LVCMOS33 [get_ports iic_0_scl_io]
4 set_property IOSTANDARD LVCMOS33 [get_ports iic_0_sda_io]
5
6 set_property PULLUP true [get_ports iic_0_scl_io]
7 set_property PULLUP true [get_ports iic_0_sda_io]
```


这个STLM75是接在PL端所以用的emio来进行I2C的读写


Step3 进行综合、生成顶层文件，顶层文件如下所示(注意I2C一定要加下面的IOBUF)

```
1 //Copyright 1986-2017 Xilinx, Inc. All Rights Reserved.
2 //-----
3 //Tool Version: Vivado v.2017.1 (win64) Build 1846317 Fri Apr 14 18:55:03 MDT 2017
4 //Date      : Mon Feb 26 18:28:07 2018
5 //Host      : taowei running 64-bit Service Pack 1 (build 7601)
6 //Command   : generate_target design_1_wrapper.bd
7 //Design    : design_1_wrapper
8 //Purpose   : IP block netlist
9 //-----
10 `timescale 1 ps / 1 ps
11
12 module design_1_wrapper
13     (DDR_addr,
14      DDR_ba,
15      DDR_cas_n,
16      DDR_ck_n,
17      DDR_ck_p,
18      DDR_cke,
19      DDR_cs_n,
20      DDR_dm,
21      DDR_dq,
22      DDR_dqs_n,
23      DDR_dqs_p,
24      DDR_odt,
25      DDR_ras_n,
```


```
26     DDR_reset_n,
27     DDR_we_n,
28     FIXED_IO_dds_vrn,
29     FIXED_IO_dds_vrp,
30     FIXED_IO_mio,
31     FIXED_IO_ps_clk,
32     FIXED_IO_ps_porb,
33     FIXED_IO_ps_srstb,
34     iic_0_scl_io,
35     iic_0_sda_io);
36 inout [14:0]DDR_addr;
37 inout [2:0]DDR_ba;
38 inout DDR_cas_n;
39 inout DDR_ck_n;
40 inout DDR_ck_p;
41 inout DDR_cke;
42 inout DDR_cs_n;
43 inout [3:0]DDR_dm;
44 inout [31:0]DDR_dq;
45 inout [3:0]DDR_dqs_n;
46 inout [3:0]DDR_dqs_p;
47 inout DDR_odt;
48 inout DDR_ras_n;
49 inout DDR_reset_n;
50 inout DDR_we_n;
51 inout FIXED_IO_dds_vrn;
52 inout FIXED_IO_dds_vrp;
53 inout [53:0]FIXED_IO_mio;
54 inout FIXED_IO_ps_clk;
55 inout FIXED_IO_ps_porb;
56 inout FIXED_IO_ps_srstb;
57 inout iic_0_scl_io;
58 inout iic_0_sda_io;
59
60 wire [14:0]DDR_addr;
61 wire [2:0]DDR_ba;
62 wire DDR_cas_n;
63 wire DDR_ck_n;
64 wire DDR_ck_p;
65 wire DDR_cke;
66 wire DDR_cs_n;
67 wire [3:0]DDR_dm;
68 wire [31:0]DDR_dq;
69 wire [3:0]DDR_dqs_n;
70 wire [3:0]DDR_dqs_p;
71 wire DDR_odt;
72 wire DDR_ras_n;
73 wire DDR_reset_n;
74 wire DDR_we_n;
75 wire FIXED_IO_dds_vrn;
76 wire FIXED_IO_dds_vrp;
77 wire [53:0]FIXED_IO_mio;
78 wire FIXED_IO_ps_clk;
79 wire FIXED_IO_ps_porb;
80 wire FIXED_IO_ps_srstb;
81 wire iic_0_scl_i;
82 wire iic_0_scl_io;
83 wire iic_0_scl_o;
84 wire iic_0_scl_t;
85 wire iic_0_sda_i;
86 wire iic_0_sda_io;
87 wire iic_0_sda_o;
88 wire iic_0_sda_t;
89
90 design_1 design_1_i
91     (.DDR_addr(DDR_addr),
92     .DDR_ba(DDR_ba),
93     .DDR_cas_n(DDR_cas_n),
94     .DDR_ck_n(DDR_ck_n),
95     .DDR_ck_p(DDR_ck_p),
96     .DDR_cke(DDR_cke),
97     .DDR_cs_n(DDR_cs_n),
98     .DDR_dm(DDR_dm),
99     .DDR_dq(DDR_dq),
100     .DDR_dqs_n(DDR_dqs_n),
101     .DDR_dqs_p(DDR_dqs_p),
102     .DDR_odt(DDR_odt),
103     .DDR_ras_n(DDR_ras_n),
104     .DDR_reset_n(DDR_reset_n),
```











6













举报


```

105         .DDR_we_n(DDR_we_n),
106         .FIXED_IO_ddr_vrn(FIXED_IO_ddr_vrn),
107         .FIXED_IO_ddr_vrp(FIXED_IO_ddr_vrp),
108         .FIXED_IO_mio(FIXED_IO_mio),
109         .FIXED_IO_ps_clk(FIXED_IO_ps_clk),
110         .FIXED_IO_ps_porb(FIXED_IO_ps_porb),
111         .FIXED_IO_ps_srstb(FIXED_IO_ps_srstb),
112         .IIC_0_scl_i(iic_0_scl_i),
113         .IIC_0_scl_o(iic_0_scl_o),
114         .IIC_0_scl_t(iic_0_scl_t),
115         .IIC_0_sda_i(iic_0_sda_i),
116         .IIC_0_sda_o(iic_0_sda_o),
117         .IIC_0_sda_t(iic_0_sda_t));
118     IOBUF iic_0_scl_iobuf
119     (.I(iic_0_scl_o),
120      .IO(iic_0_scl_io),
121      .O(iic_0_scl_i),
122      .T(iic_0_scl_t));
123     IOBUF iic_0_sda_iobuf
124     (.I(iic_0_sda_o),
125      .IO(iic_0_sda_io),
126      .O(iic_0_sda_i),
127      .T(iic_0_sda_t));
128 endmodule

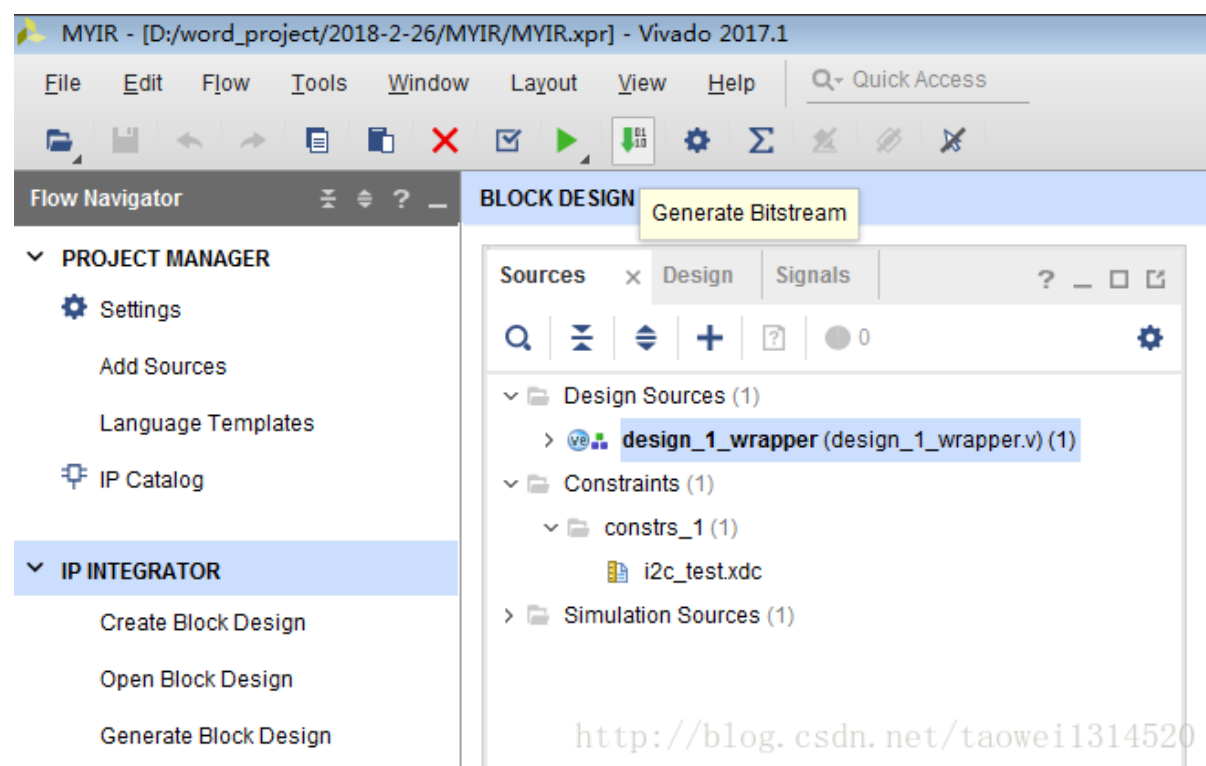
```



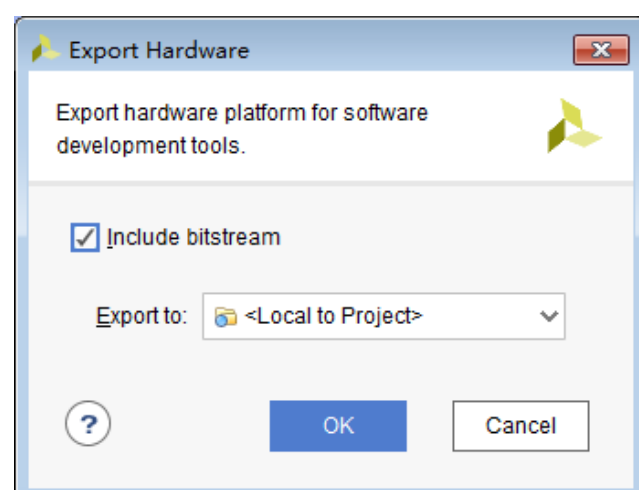
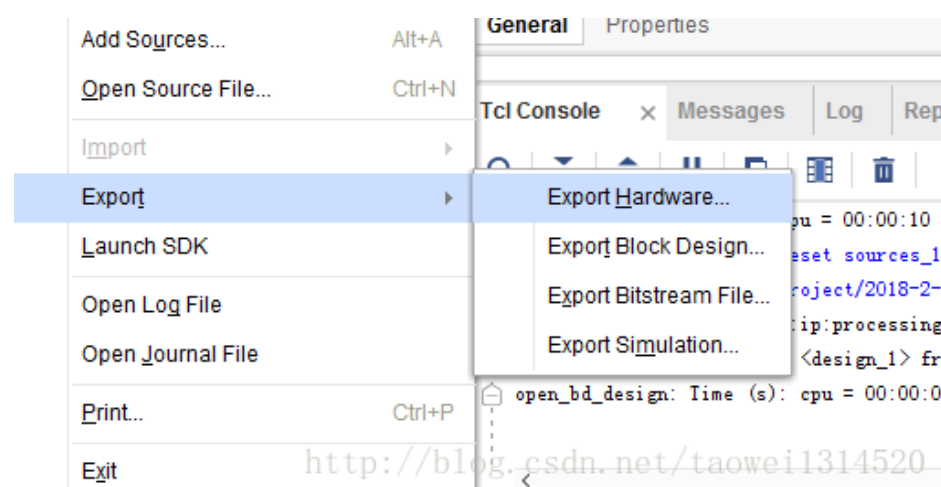
6



Step4 生成bit文件

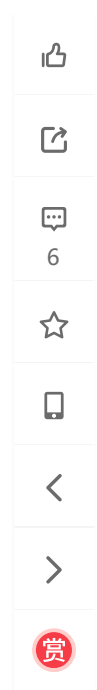
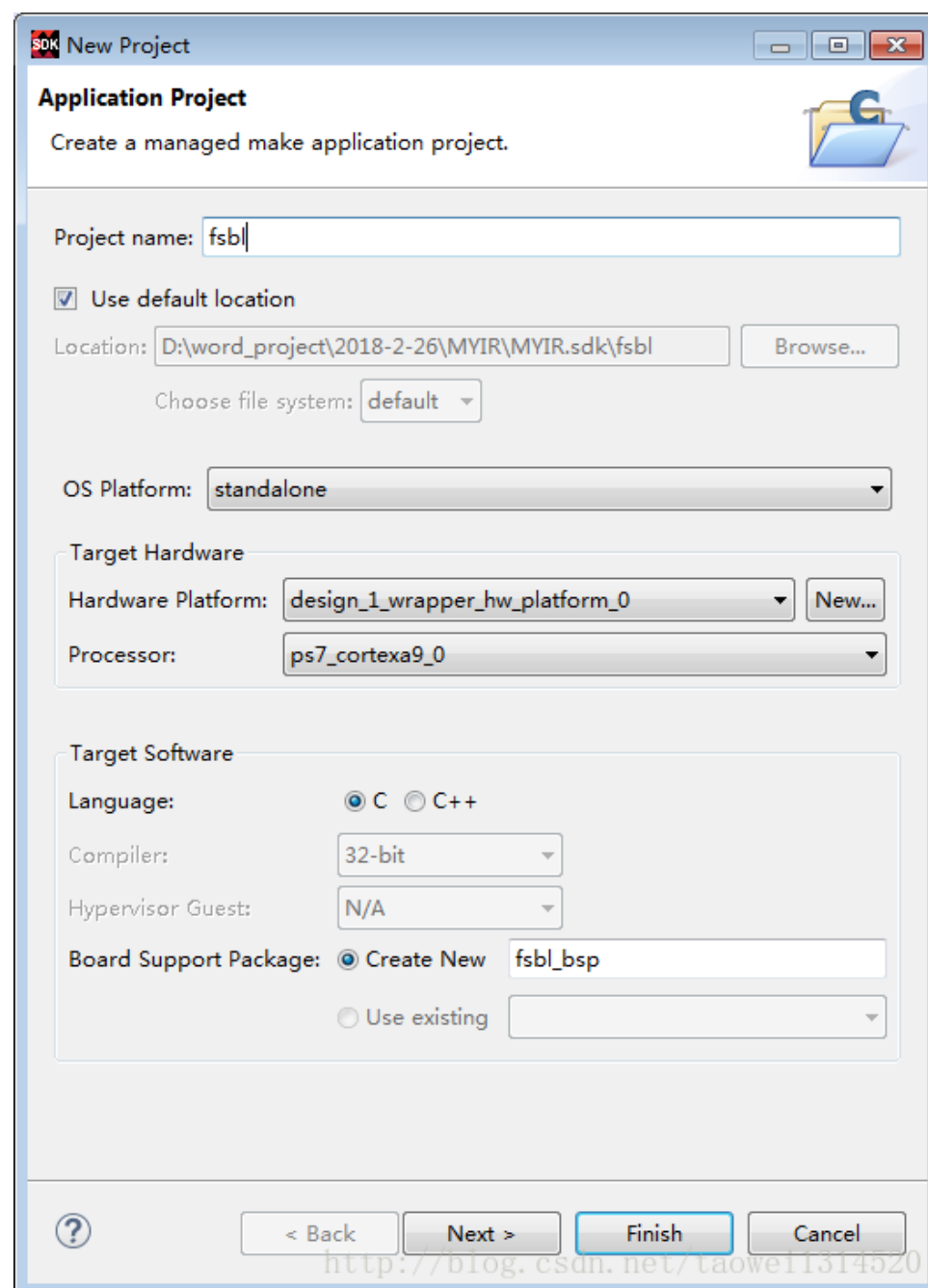
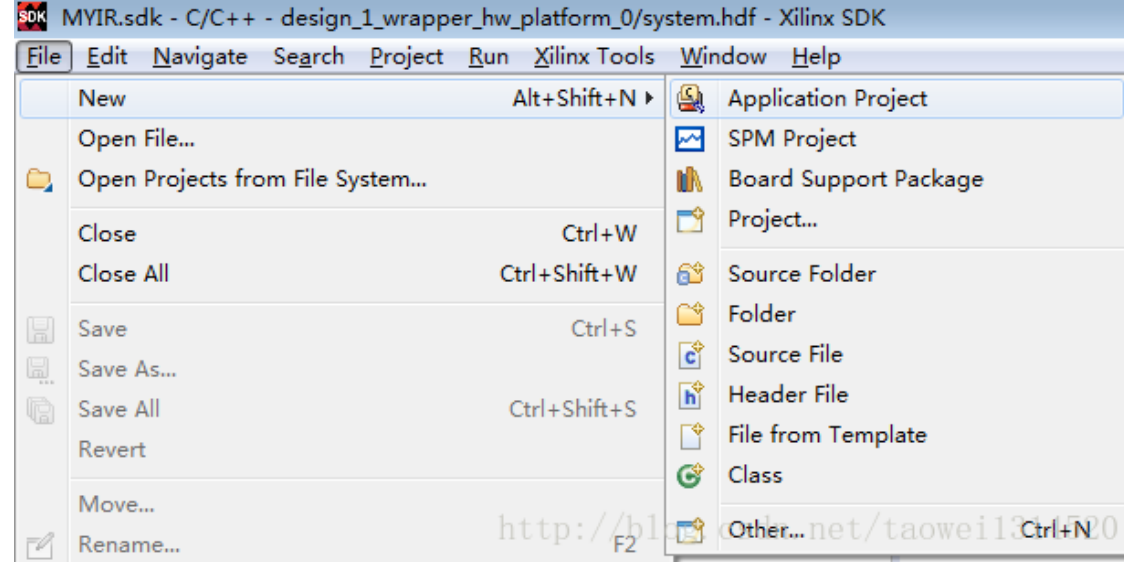


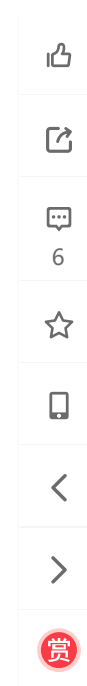
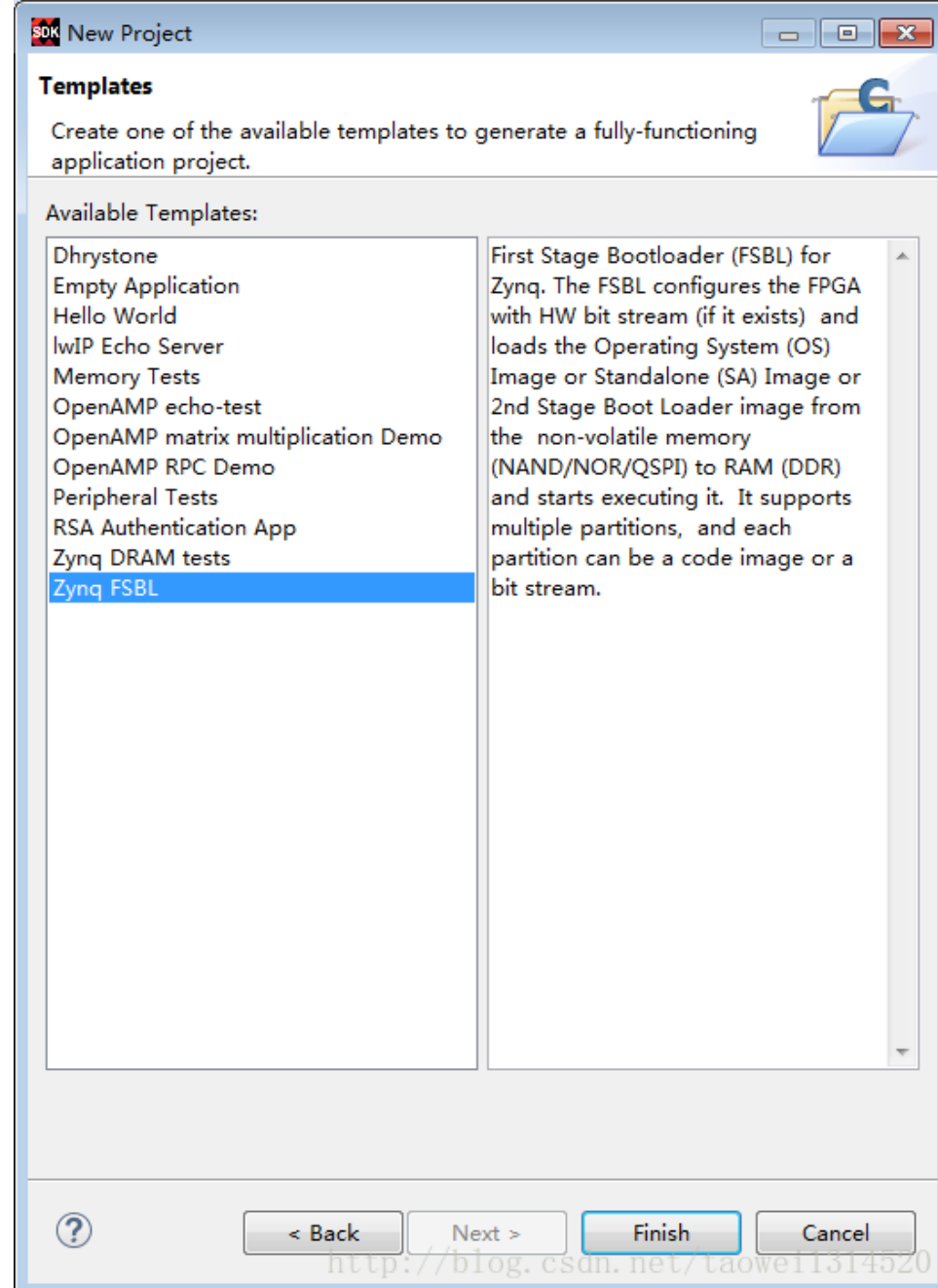
Step5 点击菜单栏上的 File->Export->Export Hardware 导出硬件配置文件



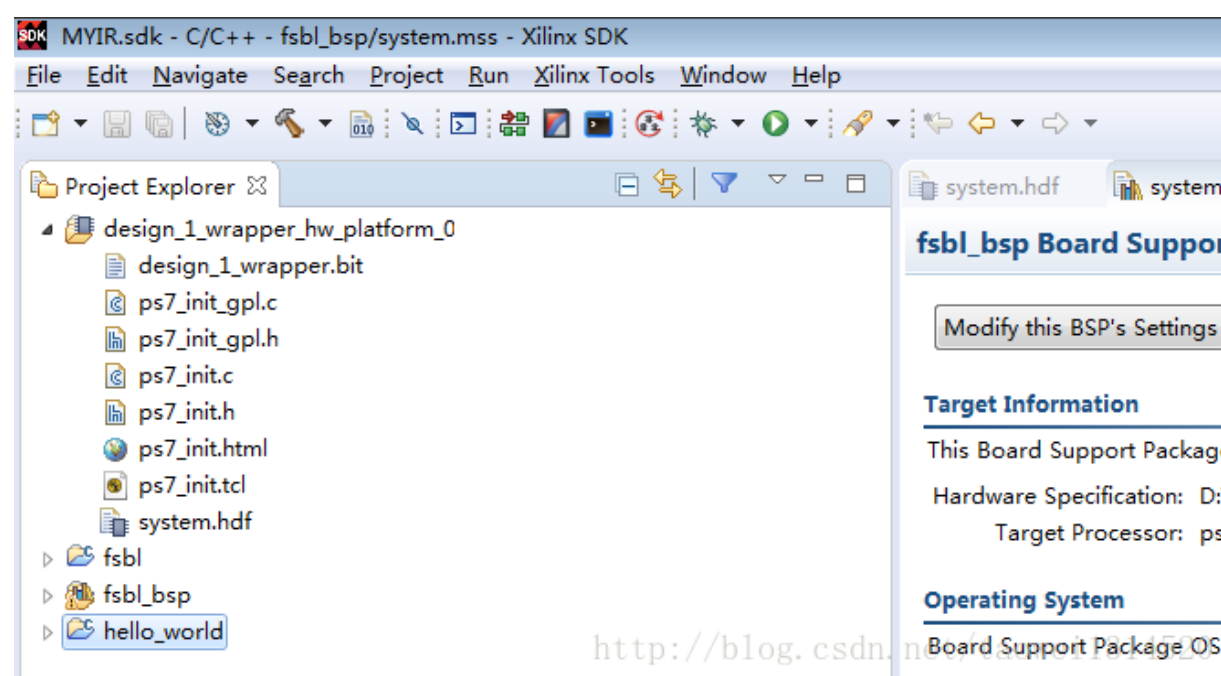
举报

Step 5 打开SDK, 然后新建一个fsbl





Step 6 新建一个hello_world模板工程



工程新建完成后，我们看手册可以看到这个STLM75一个有4个寄存器

Table 5. Register pointers selection summary

Pointer value (H)	P1	P0	Name	Description	Width (bits)	Type (R/W)	Power-on default	Comments
00	0	0	TEMP	Temperature register	16	Read-only	N/A	To store measured temperature data
01	0	1	CONF	Configuration register	8	R/W	00	
02	1	0	T _{HYS}	Hysteresis register	16	R/W	4B00	Default = 75 °C
03	1	1	T _{OS}	Overtemperature shutdown	16	R/W	5000	Set point for overtemperature shutdown (T _{OS}) limit default = 80 °C

当指针寄存器值不同时可以对不同的寄存器进行操作，当然这个指针寄存器上电后的初始状态是处在温度寄存器，指针寄存器数

00 可以对温度寄存器进行读(Temperature register)

01 可以对配置寄存器进行读写 (Configuration register)



- 10 可以对滞后寄存器进行读写 (Hysteresis register)
- 11 可以对超温阈值寄存器进行读写 (Overtemperature shutdown)

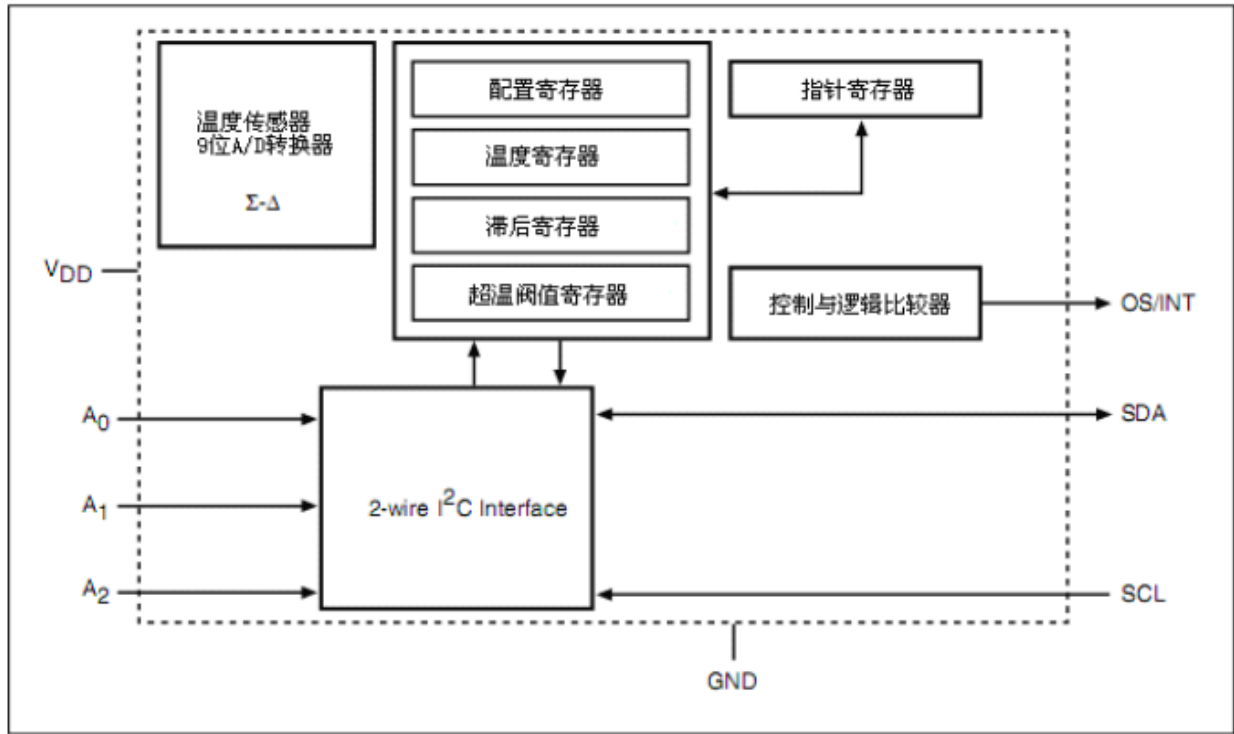
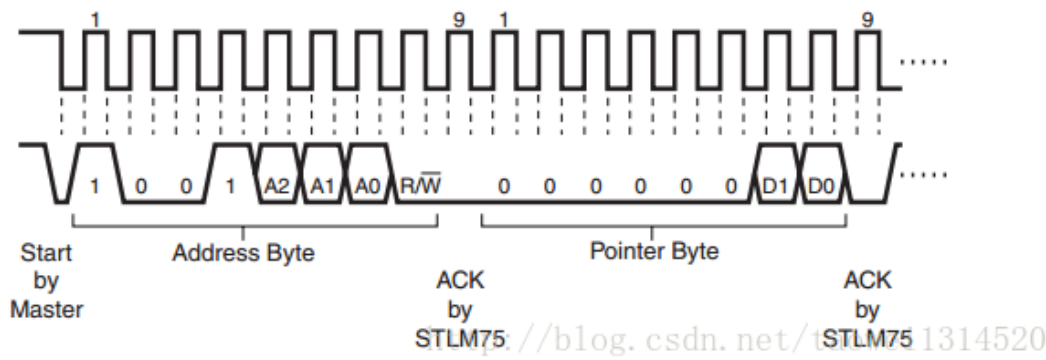


图 9-9 STLM75 功能框图

对这个指针寄存器进行写的波形图如下所示



这个D0和D1分别对应手册上的P0和P1，只要对指针寄存器写入不同的值就可以对上面四个寄存器进行访问

这个对指针寄存器进行写的过程中要注意这个D2~D7要都为0，D0和D1可以是00、01、10、11中的一个就行

我这里先对配置寄存器进行读，下面是hello_world工程的主程序

```
1  /*****
2  *
3  * Copyright (C) 2009 - 2014 Xilinx, Inc. All rights reserved.
4  *
5  * Permission is hereby granted, free of charge, to any person obtaining a copy
6  * of this software and associated documentation files (the "Software"), to deal
7  * in the Software without restriction, including without limitation the rights
8  * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
9  * copies of the Software, and to permit persons to whom the Software is
10 * furnished to do so, subject to the following conditions:
11 *
12 * The above copyright notice and this permission notice shall be included in
13 * all copies or substantial portions of the Software.
14 *
15 * Use of the Software is limited solely to applications:
16 * (a) running on a Xilinx device, or
17 * (b) that interact with a Xilinx device through a bus or interconnect.
18 *
19 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
20 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
21 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
22 * XILINX BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
23 * WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF
24 * OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
25 * SOFTWARE.
26 *
27 * Except as contained in this notice, the name of the Xilinx shall not be used
28 * in advertising or otherwise to promote the sale, use or other dealings in
29 * this Software without prior written authorization from Xilinx.
30 *
31 *****/
32
33 /*
34  * helloworld.c: simple test application
```

👍

🔗

💬
6

☆

📱

<

>

赏

🔊

举报

```
35 * 36 | *This application configures UART 16550 to baud rate 9600.
37 * PS7 UART (Zynq) is not initialized by this application, since
38 * bootrom/bsp configures it to baud rate 115200
39 *
40 * -----
41 * | UART TYPE   BAUD RATE                                |
42 * -----
43 *   uartns550   9600
44 *   uartlite    Configurable only in HW design
45 *   ps7_uart    115200 (configured by bootrom/bsp)
46 */
47
48 #include <stdio.h>
49 #include "platform.h"
50 #include "xil_printf.h"
51
52
53 #include "sleep.h"
54 #include "xiicps.h"
55
56 XIicPs IicInstance;      /* The instance of the IIC device. */
57
58 #define IIC_DEVICE_ID    XPAR_XIICPS_0_DEVICE_ID
59
60 u8 WriteBuffer[1];
61
62 u16 ReadBuffer[1];      /* Read buffer for reading a page. */
63
64
65
66
67 int iic_master_init(void)
68 {
69     int Status;
70     XIicPs_Config *ConfigPtr;    /* Pointer to configuration data */
71
72     ConfigPtr = XIicPs_LookupConfig(IIC_DEVICE_ID);
73     if (ConfigPtr == NULL) {
74         return XST_FAILURE;
75     }
76
77     Status = XIicPs_CfgInitialize(&IicInstance, ConfigPtr,
78                                   ConfigPtr->BaseAddress);
79     if (Status != XST_SUCCESS) {
80         return XST_FAILURE;
81     }
82
83     XIicPs_SetSCLK(&IicInstance, 400000);
84
85     return XST_SUCCESS;
86 }
87
88
89
90 int iic_write_read_8(u8 Device_Address,u8 Pointer_Address)
91 {
92     int Status;
93
94     WriteBuffer[0] =    Pointer_Address; //0x01
95
96
97     Status = XIicPs_MasterSendPolled(&IicInstance, WriteBuffer,
98                                       1, Device_Address>>1);
99     if (Status != XST_SUCCESS) {
100         return XST_FAILURE;
101     }
102
103     while (XIicPs_BusIsBusy(&IicInstance));
104
105     usleep(2500);
106     if (Status != XST_SUCCESS) {
107         return XST_FAILURE;
108     }
109
110
111
112     Status = XIicPs_MasterRecvPolled(&IicInstance, ReadBuffer,
113                                       1, Device_Address>>1);
```



6



举报

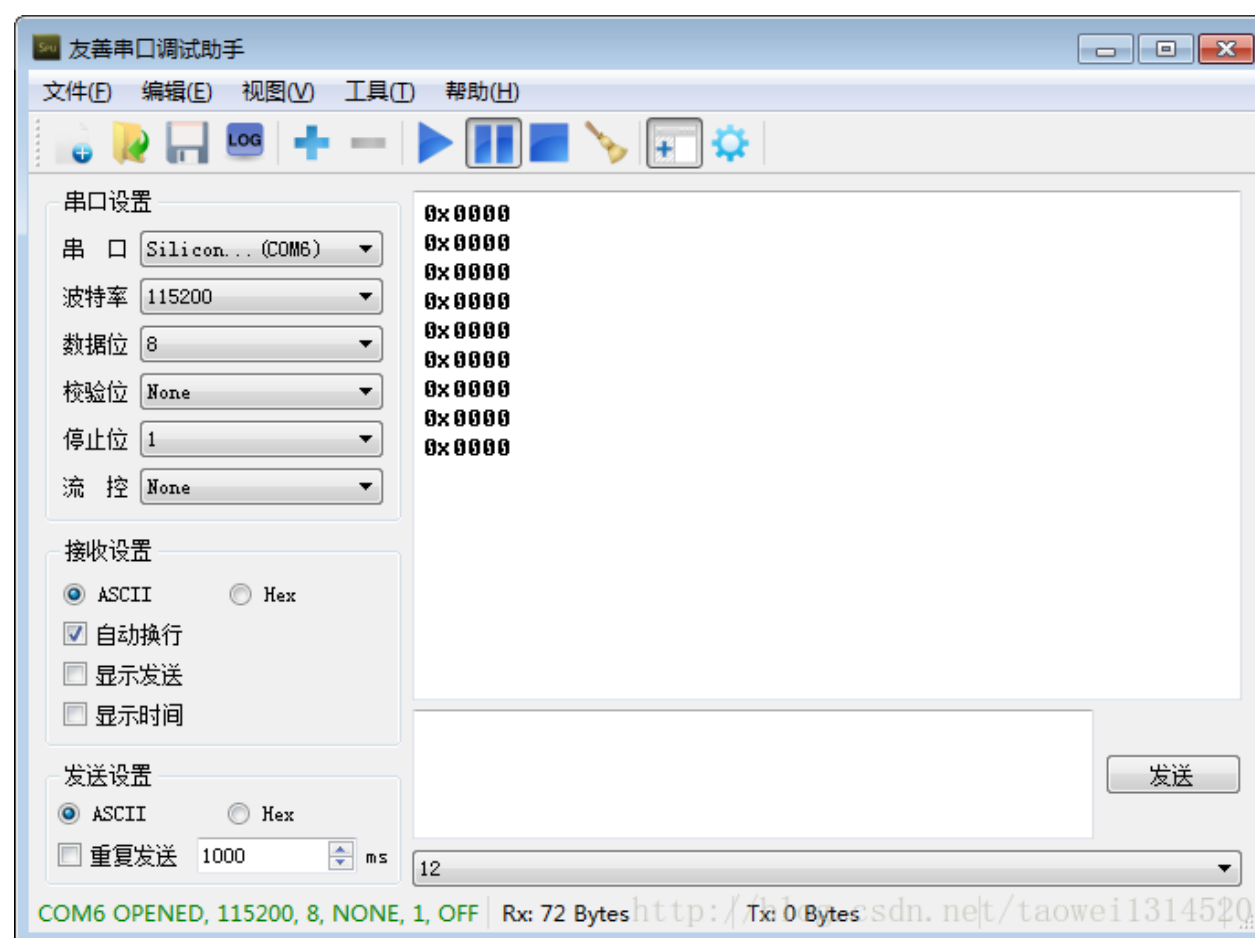

```
114     if (Status != XST_SUCCESS) {115         return XST_FAILURE;
116     }
117     while (XIicPs_BusIsBusy(&IicInstance));
118     xil_printf("0x%04x\r\n",ReadBuffer[0]);
119     return 0;
120 }
121
122
123 int main(void)
124 {
125
126     u8 Device_Address;
127
128     Device_Address = 0x93; //10010011
129
130
131     iic_master_init();
132
133
134     while(1)
135     {
136
137         iic_write_read_8(Device_Address,0x01);
138         sleep(1);
139     }
140
141
142     return 0;
143 }
```



6



下面是打印的寄存器的值



可以看到和手册里的一致

Pointer value (H)	P1	P0	Name	Description	Width (bits)	Type (R/W)	Power-on default	Comments
00	0	0	TEMP	Temperature register	16	Read-only	N/A	To store measured temperature data
01	0	1	CONF	Configuration register	8	R/W	00	
02	1	0	T _{HYS}	Hysteresis register	16	R/W	4B00	Default = 75 °C
03	1	1	T _{OS}	Overtemperature shutdown	16	R/W	5000	Set point for overtemperature shutdown (T _{OS}) limit default °C

接下来是对滞后寄存器进行读，下面是hello_world工程的主程序

```
1 | /*****
```

举报

```
2  * 3 | * Copyright (C) 2009 - 2014 Xilinx, Inc. All rights reserved.
3
4  *
5  * Permission is hereby granted, free of charge, to any person obtaining a copy
6  * of this software and associated documentation files (the "Software"), to deal
7  * in the Software without restriction, including without limitation the rights
8  * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
9  * copies of the Software, and to permit persons to whom the Software is
10 * furnished to do so, subject to the following conditions:
11 *
12 * The above copyright notice and this permission notice shall be included in
13 * all copies or substantial portions of the Software.
14 *
15 * Use of the Software is limited solely to applications:
16 * (a) running on a Xilinx device, or
17 * (b) that interact with a Xilinx device through a bus or interconnect.
18 *
19 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
20 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
21 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
22 * XILINX BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
23 * WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF
24 * OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
25 * SOFTWARE.
26 *
27 * Except as contained in this notice, the name of the Xilinx shall not be used
28 * in advertising or otherwise to promote the sale, use or other dealings in
29 * this Software without prior written authorization from Xilinx.
30 *
31 *****/
32
33 /*
34  * helloworld.c: simple test application
35  *
36  * This application configures UART 16550 to baud rate 9600.
37  * PS7 UART (Zynq) is not initialized by this application, since
38  * bootrom/bsp configures it to baud rate 115200
39  *
40  * -----
41  * | UART TYPE   BAUD RATE                                |
42  * -----
43  *  uartns550   9600
44  *  uartlite    Configurable only in HW design
45  *  ps7_uart    115200 (configured by bootrom/bsp)
46  */
47
48 #include <stdio.h>
49 #include "platform.h"
50 #include "xil_printf.h"
51
52
53 #include "sleep.h"
54 #include "xiicps.h"
55
56 XIicPs IicInstance;      /* The instance of the IIC device. */
57
58 #define IIC_DEVICE_ID    XPAR_XIICPS_0_DEVICE_ID
59
60 u8 WriteBuffer[1];
61
62 u16 ReadBuffer[1];      /* Read buffer for reading a page. */
63
64
65
66
67 int iic_master_init(void)
68 {
69     int Status;
70     XIicPs_Config *ConfigPtr;    /* Pointer to configuration data */
71
72     ConfigPtr = XIicPs_LookupConfig(IIC_DEVICE_ID);
73     if (ConfigPtr == NULL) {
74         return XST_FAILURE;
75     }
76
77     Status = XIicPs_CfgInitialize(&IicInstance, ConfigPtr,
78                                   ConfigPtr->BaseAddress);
79     if (Status != XST_SUCCESS) {
80         return XST_FAILURE;
```



6



举报

```
81     } 82  
83     XIicPs_SetSCLk(&IicInstance, 400000);  
84  
85     return XST_SUCCESS;  
86 }  
87  
88  
89  
90 int iic_write_read_8(u8 Device_Address,u8 Pointer_Address)  
91 {  
92     int Status;  
93  
94     WriteBuffer[0] = Pointer_Address; //0x02  
95  
96  
97     Status = XIicPs_MasterSendPolled(&IicInstance, WriteBuffer,  
98                                     1, Device_Address>>1);  
99     if (Status != XST_SUCCESS) {  
100         return XST_FAILURE;  
101     }  
102  
103     while (XIicPs_BusIsBusy(&IicInstance));  
104  
105     usleep(2500);  
106     if (Status != XST_SUCCESS) {  
107         return XST_FAILURE;  
108     }  
109  
110  
111  
112     Status = XIicPs_MasterRecvPolled(&IicInstance, ReadBuffer,  
113                                     1, Device_Address>>1);  
114     if (Status != XST_SUCCESS) {  
115         return XST_FAILURE;  
116     }  
117     while (XIicPs_BusIsBusy(&IicInstance));  
118     xil_printf("0x%04x\r\n",ReadBuffer[0]);  
119     return 0;  
120 }  
121  
122  
123 int main(void)  
124 {  
125  
126     u8 Device_Address;  
127  
128     Device_Address = 0x93; //10010011  
129  
130  
131     iic_master_init();  
132  
133  
134     while(1)  
135     {  
136  
137         iic_write_read_8(Device_Address,0x02);  
138         sleep(1);  
139     }  
140  
141  
142     return 0;  
143 }
```

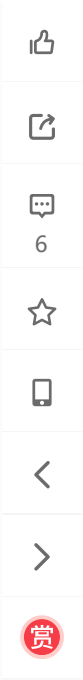
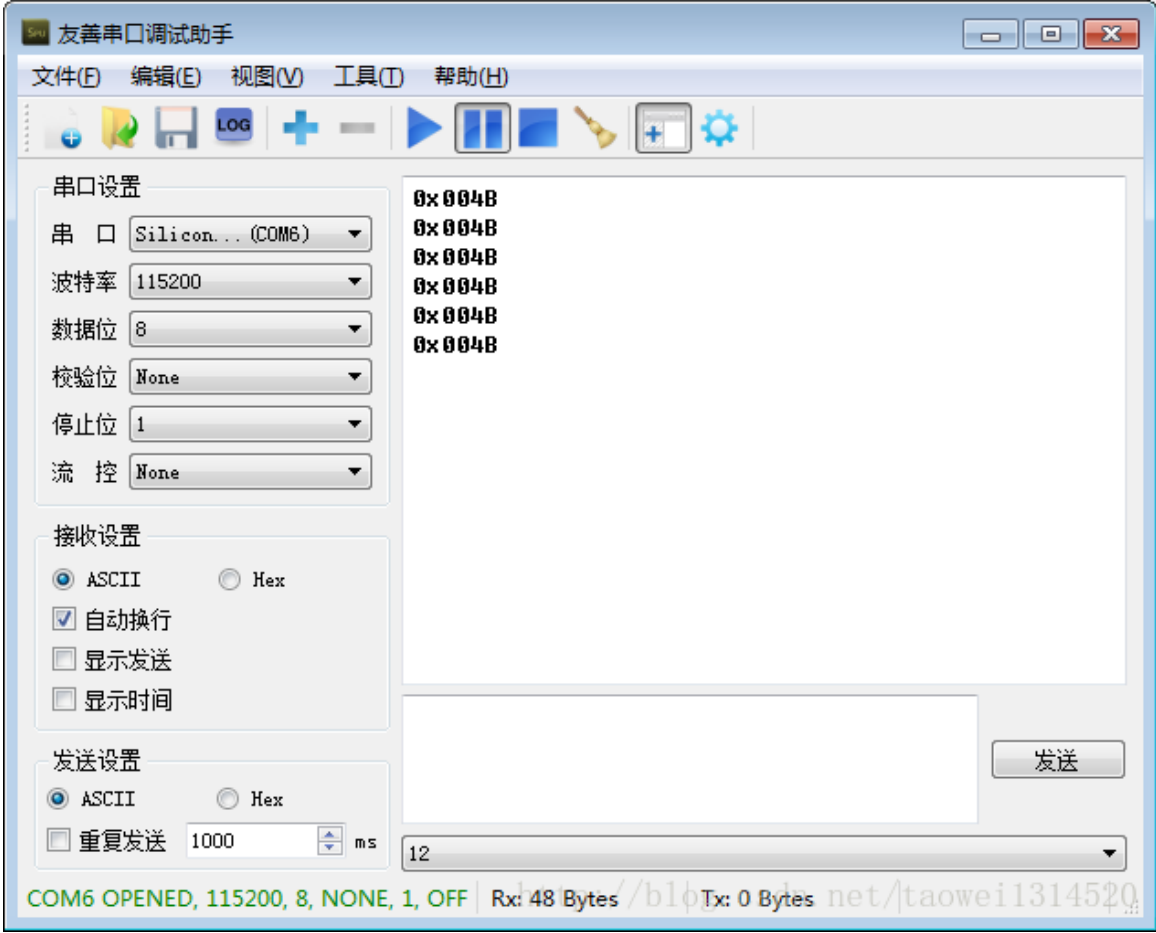
下面是打印的寄存器的值



6



举报



可以看到和手册里的一致

Pointer value (H)	P1	P0	Name	Description	Width (bits)	Type (R/W)	Power-on default	Comments
00	0	0	TEMP	Temperature register	16	Read-only	N/A	To store measured temperature data
01	0	1	CONF	Configuration register	8	R/W	00	
02	1	0	T _{HYS}	Hysteresis register	16	R/W	4B00	Default = 75 °C
03	1	1	T _{OS}	Overtemperature shutdown	16	R/W	5000	Set point for overtemperature shutdown (T _{OS}) limit default = 80 °C

再对超温阈值寄存器进行读,下面是hello_world工程的主程序

```
1  /*****
2  *
3  * Copyright (C) 2009 - 2014 Xilinx, Inc. All rights reserved.
4  *
5  * Permission is hereby granted, free of charge, to any person obtaining a copy
6  * of this software and associated documentation files (the "Software"), to deal
7  * in the Software without restriction, including without limitation the rights
8  * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
9  * copies of the Software, and to permit persons to whom the Software is
10 * furnished to do so, subject to the following conditions:
11 *
12 * The above copyright notice and this permission notice shall be included in
13 * all copies or substantial portions of the Software.
14 *
15 * Use of the Software is limited solely to applications:
16 * (a) running on a Xilinx device, or
17 * (b) that interact with a Xilinx device through a bus or interconnect.
18 *
19 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
20 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
21 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
22 * XILINX BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
23 * WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF
24 * OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
25 * SOFTWARE.
26 *
27 * Except as contained in this notice, the name of the Xilinx shall not be used
28 * in advertising or otherwise to promote the sale, use or other dealings in
29 * this Software without prior written authorization from Xilinx.
30 *
31 *****/
32
33 /*
34  * helloworld.c: simple test application
```




```
35  * 36 | * This application configures UART 16550 to baud rate 9600.
37  * PS7 UART (Zynq) is not initialized by this application, since
38  * bootrom/bsp configures it to baud rate 115200
39  *
40  * -----
41  * | UART TYPE      BAUD RATE      |
42  * -----
43  *   uartns550      9600
44  *   uartlite        Configurable only in HW design
45  *   ps7_uart        115200 (configured by bootrom/bsp)
46  */
47
48 #include <stdio.h>
49 #include "platform.h"
50 #include "xil_printf.h"
51
52
53 #include "sleep.h"
54 #include "xiicps.h"
55
56 XIicPs IicInstance;      /* The instance of the IIC device. */
57
58 #define IIC_DEVICE_ID    XPAR_XIICPS_0_DEVICE_ID
59
60 u8 WriteBuffer[1];
61
62 u16 ReadBuffer[1];      /* Read buffer for reading a page. */
63
64
65
66
67 int iic_master_init(void)
68 {
69     int Status;
70     XIicPs_Config *ConfigPtr;    /* Pointer to configuration data */
71
72     ConfigPtr = XIicPs_LookupConfig(IIC_DEVICE_ID);
73     if (ConfigPtr == NULL) {
74         return XST_FAILURE;
75     }
76
77     Status = XIicPs_CfgInitialize(&IicInstance, ConfigPtr,
78                                   ConfigPtr->BaseAddress);
79     if (Status != XST_SUCCESS) {
80         return XST_FAILURE;
81     }
82
83     XIicPs_SetSCLK(&IicInstance, 400000);
84
85     return XST_SUCCESS;
86 }
87
88
89
90 int iic_write_read_8(u8 Device_Address,u8 Pointer_Address)
91 {
92     int Status;
93
94     WriteBuffer[0] =    Pointer_Address; //0x03
95
96
97     Status = XIicPs_MasterSendPolled(&IicInstance, WriteBuffer,
98                                       1, Device_Address>>1);
99     if (Status != XST_SUCCESS) {
100         return XST_FAILURE;
101     }
102
103     while (XIicPs_BusIsBusy(&IicInstance));
104
105     usleep(2500);
106     if (Status != XST_SUCCESS) {
107         return XST_FAILURE;
108     }
109
110
111
112     Status = XIicPs_MasterRecvPolled(&IicInstance, ReadBuffer,
113                                       1, Device_Address>>1);
```



6



举报

```

114     if (Status != XST_SUCCESS) {115         return XST_FAILURE;
116     }
117     while (XIicPs_BusIsBusy(&IicInstance));
118     xil_printf("0x%04x\r\n",ReadBuffer[0]);
119     return 0;
120 }
121
122
123 int main(void)
124 {
125     u8 Device_Address;
126
127     Device_Address = 0x93; //10010011
128
129
130
131 iic_master_init();
132
133
134 while(1)
135 {
136
137     iic_write_read_8(Device_Address,0x03);
138     sleep(1);
139 }
140
141
142     return 0;
143 }

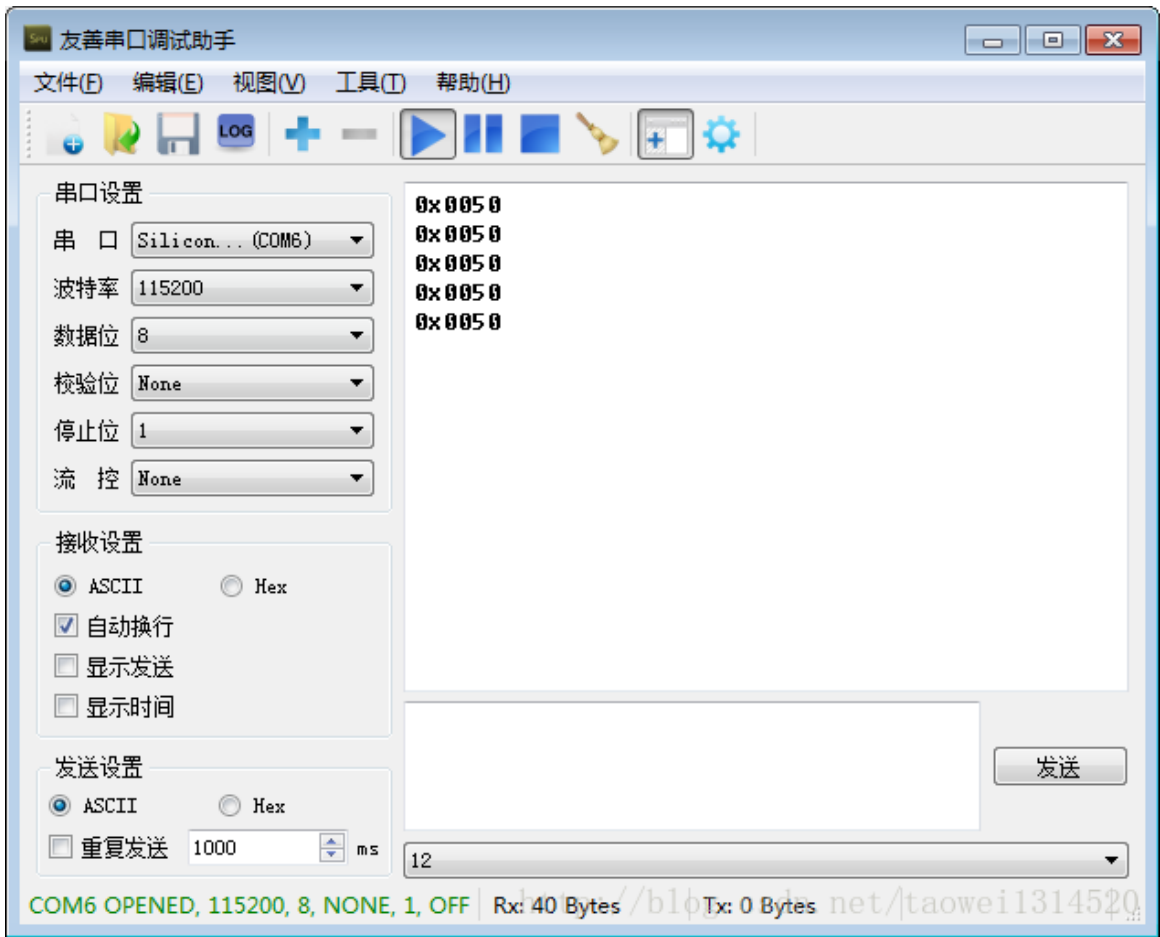
```



6



下面是打印的寄存器的值



可以看到和手册里的一致

Pointer value (H)	P1	P0	Name	Description	Width (bits)	Type (R/W)	Power-on default	Comments
00	0	0	TEMP	Temperature register	16	Read-only	N/A	To store measured temperature data
01	0	1	CONF	Configuration register	8	R/W	00	
02	1	0	T _{HYS}	Hysteresis register	16	R/W	4B00	Default = 75 °C
03	1	1	T _{OS}	Overtemperature shutdown	16	R/W	5000	Set point for overtemperature shutdown (T _{OS}) limit default = 80

从上面可以看出对前面的几个寄存器进行读出的数据和手册一样说明，这个I2C读写是正确的，

最后对温度寄存器进行读，下面是hello_world工程的主程序

举报

```

1  /*****
2  *
3  * Copyright (C) 2009 - 2014 Xilinx, Inc. All rights reserved.
4  *
5  * Permission is hereby granted, free of charge, to any person obtaining a copy
6  * of this software and associated documentation files (the "Software"), to deal
7  * in the Software without restriction, including without limitation the rights
8  * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
9  * copies of the Software, and to permit persons to whom the Software is
10 * furnished to do so, subject to the following conditions:
11 *
12 * The above copyright notice and this permission notice shall be included in
13 * all copies or substantial portions of the Software.
14 *
15 * Use of the Software is limited solely to applications:
16 * (a) running on a Xilinx device, or
17 * (b) that interact with a Xilinx device through a bus or interconnect.
18 *
19 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
20 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
21 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
22 * XILINX BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
23 * WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF
24 * OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
25 * SOFTWARE.
26 *
27 * Except as contained in this notice, the name of the Xilinx shall not be used
28 * in advertising or otherwise to promote the sale, use or other dealings in
29 * this Software without prior written authorization from Xilinx.
30 *
31 *****/
32
33 /*
34  * helloworld.c: simple test application
35  *
36  * This application configures UART 16550 to baud rate 9600.
37  * PS7 UART (Zynq) is not initialized by this application, since
38  * bootrom/bsp configures it to baud rate 115200
39  *
40  * -----
41  * | UART TYPE   BAUD RATE                                |
42  * -----
43  *  uartns550   9600
44  *  uartlite    Configurable only in HW design
45  *  ps7_uart    115200 (configured by bootrom/bsp)
46  */
47
48 #include <stdio.h>
49 #include "platform.h"
50 #include "xil_printf.h"
51
52
53 #include "sleep.h"
54 #include "xiicps.h"
55
56 XIicPs IicInstance;    /* The instance of the IIC device. */
57
58 #define IIC_DEVICE_ID  XPAR_XIICPS_0_DEVICE_ID
59
60 u8 WriteBuffer[1];
61
62 u16 ReadBuffer[1];    /* Read buffer for reading a page. */
63
64
65
66
67 int iic_master_init(void)
68 {
69     int Status;
70     XIicPs_Config *ConfigPtr;    /* Pointer to configuration data */
71
72     ConfigPtr = XIicPs_LookupConfig(IIC_DEVICE_ID);
73     if (ConfigPtr == NULL) {
74         return XST_FAILURE;
75     }
76
77     Status = XIicPs_CfgInitialize(&IicInstance, ConfigPtr,
78                                   ConfigPtr->BaseAddress);

```



6



举报

```
79     if (Status != XST_SUCCESS) { 80 |         return XST_FAILURE;
81     }
82
83     XIicPs_SetSClk(&IicInstance, 400000);
84
85     return XST_SUCCESS;
86 }
87
88
89
90 int iic_write_read_8(u8 Device_Address,u8 Pointer_Address)
91 {
92     int Status;
93
94     WriteBuffer[0] =    Pointer_Address; //0x00
95
96
97     Status = XIicPs_MasterSendPolled(&IicInstance, WriteBuffer,
98                                     1, Device_Address>>1);
99     if (Status != XST_SUCCESS) {
100         return XST_FAILURE;
101     }
102
103     while (XIicPs_BusIsBusy(&IicInstance));
104
105     usleep(2500);
106     if (Status != XST_SUCCESS) {
107         return XST_FAILURE;
108     }
109
110
111
112     Status = XIicPs_MasterRecvPolled(&IicInstance, ReadBuffer,
113                                     1, Device_Address>>1);
114     if (Status != XST_SUCCESS) {
115         return XST_FAILURE;
116     }
117     while (XIicPs_BusIsBusy(&IicInstance));
118     xil_printf("0x%04x\r\n",ReadBuffer[0]);
119     return 0;
120 }
121
122
123 int main(void)
124 {
125
126     u8 Device_Address;
127
128     Device_Address = 0x93; //10010011
129
130
131     iic_master_init();
132
133
134     while(1)
135     {
136
137         iic_write_read_8(Device_Address,0x00);
138         sleep(1);
139     }
140
141
142     return 0;
143 }
```

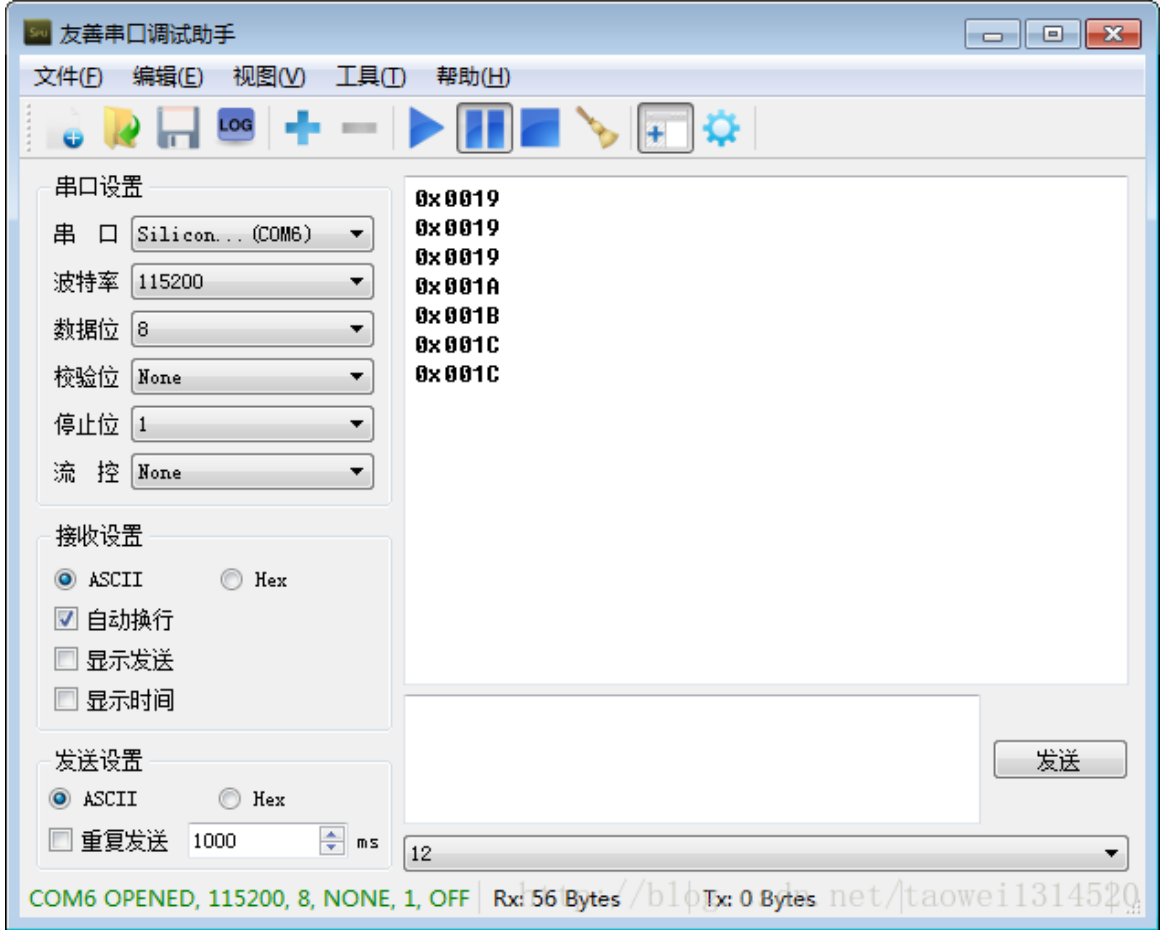
下面是打印的寄存器的值



6



举报



👍

🔗

💬

6

☆

📱

<

>

👑

赏

今天天气预报差不多是11度左右，将0x0019十六进制数化为10进制是25再除2差不多是12.5度这是在屋内所以差不多之后我将手放到STLM75上可以看到温度从0x0019上升到0x001A说明温度是在上升的。

我们再看这个温度寄存器0x0019的最高位是0所以温度是一个正的温度

Operation

After each temperature measurement and analog-to-digital conversion, the STLM75 stores the temperature as a 16-bit two’s complement number (see [Table 5: Register pointers selection summary on page 17](#)) in the 2-byte temperature register (see [Table 7 on page 18](#)). The most significant bit (S) indicates if the temperature is positive or negative:

- for positive numbers S = 0, and
- for negative numbers S = 1.

<http://blog.csdn.net/taowei1314520>

Table 7. Temperature register format

Bytes	HS byte								LS byte							
Bits	MSB	TMSB							TLSB							LSB
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STLM75	TD8 (Sign)	TD7 (TMSB)	TD6	TD5	TD4	TD3	TD2	TD1	TD0 (TLSB)	0	0	0	0	0	0	0

<http://blog.csdn.net/taowei1314520>

👍 点赞 ☆ 收藏 🔗 分享 ...



虚无缥缈vs威武

发布了37 篇原创文章 · 获赞 58 · 访问量 15万+

私信

关注



数据管理

发现称量样品及手工记录数据过程中可能发生的错误

广告 METTLER TOLEDO



想对作者说点什么



weixin_39495299

10个月前

查看回复(2)



请问device_address为什么要右移1位啊



yangsmiling

1年前

查看



举报



你好，请问之前的工程都有共享，这个教程的工程可以网盘共享出来吗。讲得很好谢谢了。

MYIR-ZYNQ7000系列-zturn教程(22)：用axi_iic对24C32进行读写 阅读数 1513

开发板环境：vivado 2017.4，开发板型号xc7z020clg400-1，这个工程主要用axi_iic对24C32进行读写链接：https... 博文 来自： taowe1314520的...

【原创】zynq-7010下运用I2C总线完成对LSM303D传感器的数据读取 5108

这是本人第一次写博客，我的毕设在用FPGA去读取LSM303D传感器的中的三轴的磁场强度数据，这也是我第一次用... 博文 来自： weixin_...

MYIR-ZYNQ7000系列-zturn教程(6)：uart_cycle 3194

开发板环境：vivado 2017.1，开发板型号xc7z020clg400-1，这个工程主要实现的功能是能在uart上进行数据的回... 博文 来自： taowe1314520的...

i2c-tools，i2cdump和i2cset等的使用总结 9684

1)察看I2C总线的bus和设备地址/sys/bus/i2c/devicesroot@zc706-zynq7:/sys/bus/i2c/devices# ls0-0074 2-00... 博文 来自： ambercc...

史上最详细的IDEA优雅整合Maven+SSM框架（详细思路+附带源码） 4万+

网上很多整合SSM博客文章并不能让初探ssm的同学思路完全的清晰，可以试着关掉整合教程，摇两下头骨，哈一大... 博文 来自： 程序员宜...

zynq学习笔记——EMIO方式模拟I2C时序对ADV7511进行读写 3422

创建硬件工程，很简单，PS接出两个EMIO和一个74.25M时钟管脚约束# ADV7511 I2C_SCLset_property PACKAG... 博文 来自： luotong86的专栏

zynq上IIC驱动 910

i2c总线驱动编写：/*filename: I2C.cauthor: pingbo andescription: EEPROM I2C*/#include <linux/module.h... 博文 来自： anpingbo的专栏

MYIR-ZYNQ7000系列-zturn教程(2)：Hello_World 3875

开发板环境：vivado 2017.1，开发板型号xc7z020clg400-1（工程末尾提供了工程源代码大家可以去网盘下载）ste... 博文 来自： taowe1314520的...

zynq PS端I2C的使用 2986

因为需要连接一款i2c接口的OLED，需要使用PS端的i2c接口。1、PL端勾选上i2c并通过EMIO分配引脚（PS可以通... 博文 来自： ma_cheng_yuan的...

刷了几千道算法题，这些我私藏的刷题网站都在这里了！ 11万+

遥想当年，机缘巧合入了 ACM 的坑，周边巨擘林立，从此过上了"天天被虐似死狗"的生活...然而我是谁，我可是死... 博文 来自： Rocky0429

Python——画一棵漂亮的樱花树（不同种樱花+玫瑰+圣诞树喔） 30万+

最近翻到一篇知乎，上面有不少用Python（大多是turtle库）绘制的树图，感觉很漂亮，我整理了一下，挑了一些我... 博文 来自： 碎片



Tyc_小胖
1篇文章
[关注](#) 排名:千里之外



ambercctv
174篇文章
[关注](#) 排名:千里之外



宜春
163篇文章
[关注](#) 排名:1000+

大学四年自学走来，这些私藏的实用工具/学习网站我贡献出来了 55万+

大学四年，看课本是不可能一直看课本的了，对于学习，特别是自学，善于搜索网上的一些资源来辅助，还是非常有... 博文 来自： 帅地

MYIR-ZYNQ7000系列-zturn教程(11)：i2c对24c32进行读写 2380

开发板环境：vivado 2017.1，开发板型号xc7z020clg400-1，这个工程主要是用i2c对24c32进行读写链接：https:/... 博文 来自： taowe1314520的...

Zynq7020 16位i2c地址的读写调试方法 555

以前我有篇博客写的是i2ctool的使用方法，博客链接：i2ctool的使用方法里面介绍的i2cset，i2cget等的指令，都是... 博文 来自： smile_5me的博客

【排序】插入类排序一(折半)插入排序、希尔排序 6325

前言在数据结构和算法中，排序是非常重要的一环，并且排序也是渗透编程的方方面面。你或许在写一个sql的order ... 博文 来自： bigsai

MYIR-ZYNQ7000系列-zturn教程(9)：将bit文件固化到QSPI_Flash 4060

开发板环境：vivado 2017.1，开发板型号xc7z020clg400-1。我们用FPGA最后生成的是二进制bit文件，bit文件下... 博文 来自： taowe1314520的...

Zynq-Linux移植学习笔记之31-用户自定义I2C驱动 1009

1、背景介绍板子上通过ZYNQ的I2C-0控制器连接了三片DBF芯片和一片Ti的226测功耗芯片，示意图如下：如上图... 博文 来自： 无知的我

MYIR-ZYNQ7000系列-zturn教程(17)：用axi_uart发送数据 4160

开发板环境：vivado 2017.1，开发板型号xc7z020clg400-1，这个工程主要用axi_uart发送数据，IP核设置的波特... 博文 来自： taowe1314520的...

MYIR-ZYNQ7000系列-zturn教程(25)：读取SD卡里的图片使用hdmi显示 662

开发板环境：vivado 2017.4，开发板型号xc7z010clg400-1，这个工程主要读取SD卡里的图片使用hdmi来进行显... 博文 来自： taowe1314520的...

MYIR-ZYNQ7000系列-zturn教程(14)：在PL中使用ILA进行调试 976

开发板环境：vivado 2017.1，开发板型号xc7z020clg400-1，这个工程主要是用ILA观测FPGA输出管脚的波形链接... 博文 来自： taowe1314520的...



662



举报

MYIR-ZYNQ7000系列-zturn教程(18)：基础教程gpio_mio做为输入口

开发板环境：vivado 2017.4，开发板型号xc7z020clg400-1，这个工程主要介绍怎样将gpio_mo做为输入口使用工...

博文 来自： taowei1314520的...

ZYNQ_IIC读写M24M01记录板子状态

ZYNQ_IIC读写M24M01记录板子状态1 M24M01特点1.1 特征1，兼容IIC的模式：1MHz；400kHz；100kHz；2, ...

博文 来自： 想做一条...

网页实现一个简单的音乐播放器（大佬别看。（☹_=_☹））

今天闲着无事，就想写点东西。然后听了下歌，就打算写个播放器。于是乎用h5 audio的加上js简单的播放器完工了...

博文 来自： qq_44211...

MYIR-ZYNQ7000系列-zturn教程(26)：自定义axi_lite IP点亮LED灯

开发板环境：vivado 2017.4，开发板型号xc7z020clg400-1，这个工程主要功能是自定义一个axi_lite IP然后在SDK...

博文 来自： taowei1314520的...

ZYNQ-I2C 调试

在zynq上成功移植I2c驱动，对eeprom进行读写访问。用示波器采样得到以下结论：MSB模式：1.先发送低地址，以...

博文 来自： 微微一笑

Python 植物大战僵尸代码实现(2):植物卡片选择和种植

这篇文章要介绍的是： - 上方植物卡片栏的实现。 - 点击植物卡片，鼠标切换为植物图片。 - 鼠标移动时，判断当前在...

博文 来自： marble_xu的博客

python学习方法总结(内附python全套学习资料)

不要再问我python好不好学了我之前做过半年少儿编程老师，一个小学四年级的小孩子都能在我的教学下独立完成p...

博文 来自： 一行数据

IIC (I2C)总线 FPGA Verilog HDL

IIC (I2C)总线 FPGA Verilog HDL配置文件：根据具体的IIC设备改一下时钟频率就可以产生正确的时钟波形`define ...

博文 来自： weixin_42965338...

Xilinx-ZYNQ7000系列-学习笔记（1）：用XADC测外部温度值

Xilinx-ZYNQ7000系列-学习笔记（2）：用XADC测外部温度值一、XADC简介Zynq器件XADC模块包括2个12比特1...

博文 来自： qq_42826337的博客

ZYNQ7000 pl330DMA vs CPU读DDR速率分析

ZYNQ7000 pl330DMA vs CPU读DDR速率分析；解决DMA在AMP环境下无法正常运行的问题；得出core0对内存...

博文 来自： Doriswang84的博客

STM32HAL库学习笔记--硬件I2C读写AT24C512

摘要：由于需要实现掉电存储功能，在无线传输模块上增加了一块EEPROM芯片，因为工程使用CubeMx配置并生成...

博文 来自： 苏铨雨的博客

《C++ Primer》学习笔记（五）：循环、分支、跳转和异常处理语句

专栏C++学习笔记《C++ Primer》学习笔记/习题答案 总目录https://blog.csdn.net/TeFuirnever/article/details/1...

博文 来自： 种树最好的时间是1...

linux系列之常用运维命令整理笔录

本博客记录工作中需要的linux运维命令，大学时候开始接触linux，会一些基本操作，可是都没有整理起来，加上是...

博文 来自： Nicky's blog

大学生活这样过，等着 Offer 飞来找

点击蓝色“沉默王二”关注我哟今天迟到了，非常抱歉，让大家久等了。01、开门见山由于我比较喜欢分享的原因，...

博文 来自： 代码GG陆晓明

Zynq----获取芯片内部温度数据

参照https://china.xilinx.com/html_docs/registers/ug1087/ug1087-zynq-ultrascale-registers.html可以知道温...

博文 来自： qq_31094099的博客

吐血推荐珍藏的Visual Studio Code插件

作为一名Java工程师，由于工作需要，最近一个月一直在写NodeJS，这种经历可以说是一部辛酸史了。好在有神器V...

博文 来自： K_Ohaha的博客

Xilinx FPGA Microblaze AXI_IIC使用方法及心得

Xilinx FPGA Microblaze AXI_IIC使用方法及心得前言最近公司要将主控程序从Cortex M系列的ARM上移植到Xilinx ...

博文 来自： 一路前行的博客

英特尔不为人知的 B 面

从 PC 时代至今，众人只知在 CPU、GPU、XPU、制程、工艺等战场中，英特尔在与同行硬件芯片制造商们的竞争中...

博文 来自： CSDN资讯

zynq-7000系列基于7015的linux下 I2C-bus switch的扩展使用

zynq-7000系列基于7015的linux下 I2C-bus switch的扩展使用

作者：卢浩

...

博文 来自： luhao806的专栏

python json java mysql pycharm android linux json格式



举报

371955816415万+

等级: 博客 4周排名: 3万+

积分: 1254总排名: 6万+

勋章:  

关注

私信

ARM單晶片物聯網系統開發

學會嵌入式單晶片 物聯網 疊
USB通訊韌體開發應用實作

最新文章

quartus II 12.1 使用教程（7） vga显示测试

MYIR-ZYNQ7000系列-zturn教程(27): lwip测试

quartus II 12.1 使用教程（6） ROM 测试

quartus II 12.1 使用教程（5） eeprom 读写测试

quartus II 12.1 使用教程（4） uart 测试

分类专栏

	VIVADO 安装教程	1篇
	quartus II	5篇
	三态门详解	
	quartus II 12.1 使用教程	1篇
	ZYNQ7000	27篇

归档

2019年12月	1篇
2019年9月	1篇
2019年8月	5篇
2019年7月	2篇
2019年4月	1篇
2019年3月	2篇
2019年1月	1篇
2018年11月	1篇

展开

热门文章

VIVADO 安装教程
阅读数 84216

三态门详解
阅读数 15398

quartus II 12.1 使用教程（1） 怎样调用 PLL 核
阅读数 7556

MYIR-ZYNQ7000系列-zturn教程(17): 用 axi_uart发送数据
阅读数 4156



6



举报

MYIR-ZYNQ7000系列-zturn教程(9)：将bit文件固化到QSPI_Flash

阅读数 4055

最新评论

VIVADO 安装教程

rq8866: 缺License的小伙伴 链接: https://pan.baidu.com/s/11mjkpyERdUH3q5C_TpfQxQ ...

FT232H如何使用jtag接口

taowei1314520: [reply]qq_42662835[/reply]我是直接对eeprom里写数据进去的，数据我已经 ...

FT232H如何使用jtag接口

taowei1314520: [reply]sssshhhhhhhh[/reply]这个vivado有这个usb驱动也需要安装一下， ...

FT232H如何使用jtag接口

sssshhhhhhhh: 你好，插上电脑以后显示 USB Serial Conventor （仅配置了USB和EEPROM ...

MYIR-ZYNQ7000系列-z...

kuyunge: SPI一次是通信一个字节码？

安全測試

您的高功率裝置

立即查看破缺

KEYSIGHT TECHNOLOGIES

QQ客服 kefu@csdn.net

客服论坛 400-660-0108

工作时间 8:30-22:00

关于我们 招聘 广告服务 网站地图

京ICP备19004658号 经营性网站备案信息

公安备案号 11010502030143

©1999-2020 北京创新乐知网络技术有限公司 网络110报警服务

北京互联网违法和不良信息举报中心

中国互联网举报中心 家长监护 版权申诉



6



举报