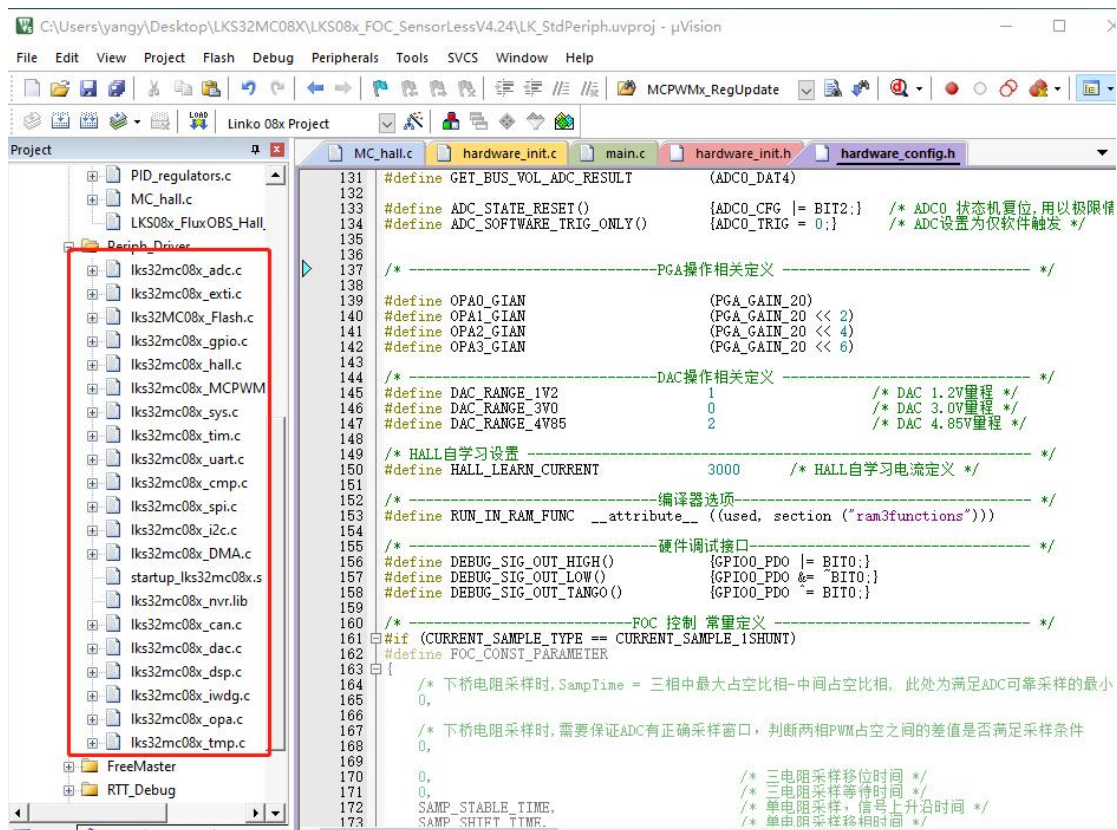


## LKS08x\_FOC\_SensorLessV4.25 之前库函数升级操作步骤

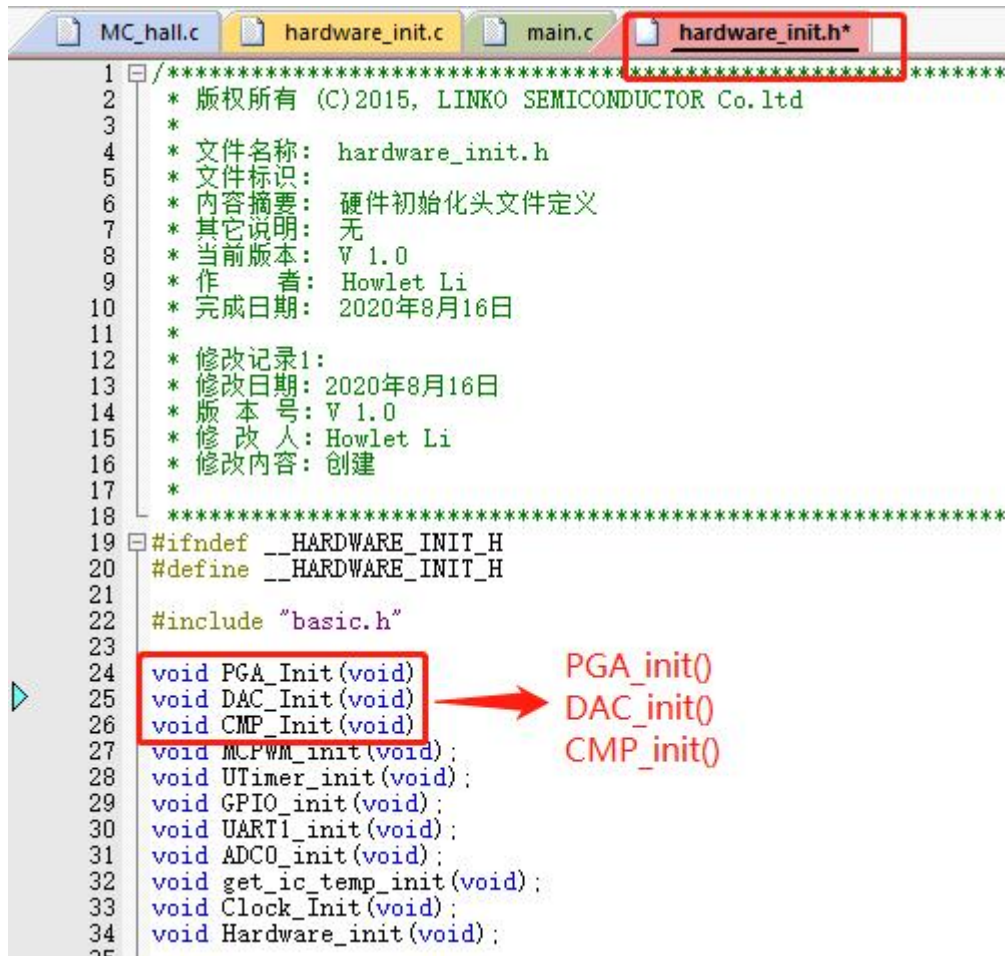
- 1、首先保证全工程能够编译通过。
- 2、将 LKS08x\_FOC\_SensorLessV4.24 文件内 LKS32MC08x\_Periph\_Driver 内的 Include 和 Source 替换为新的库函数。



- 3、将文件所有的 lks32cm08x\_xxx.c 库函数放入 keil 内。



- 4、将 hardware\_init.h 声明的 PGA\_Init、DAC\_Init、CMP\_Init 的 Init 大写的 I 改为小写的 i，即 PGA\_init、DAC\_init、CMP\_init。因为最新库函数将库函数定义的模块初始化名称 xxx\_Inti() (xxx 可以是 PGA，CMP 等)的 init 首字母大写，这样统一后用户使用不会混乱，所以用户自定义的模块初始化配置函数为 xxx\_init()的 init 就为小写开头。



```
1  /*****  
2  * 版权所有 (C)2015, LINKO SEMICONDUCTOR Co.,Ltd  
3  *  
4  * 文件名称: hardware_init.h  
5  * 文件标识:  
6  * 内容摘要: 硬件初始化头文件定义  
7  * 其它说明: 无  
8  * 当前版本: V 1.0  
9  * 作    者: Howlet Li  
10 * 完成日期: 2020年8月16日  
11 *  
12 * 修改记录1:  
13 * 修改日期: 2020年8月16日  
14 * 版 本 号: V 1.0  
15 * 修 改 人: Howlet Li  
16 * 修改内容: 创建  
17 *  
18 *****/  
19 #ifndef __HARDWARE_INIT_H  
20 #define __HARDWARE_INIT_H  
21  
22 #include "basic.h"  
23  
24 void PGA_Init(void)  
25 void DAC_Init(void)  
26 void CMP_Init(void)  
27 void MCPWM_init(void);  
28 void UTimer_init(void);  
29 void GPIO_init(void);  
30 void UART1_init(void);  
31 void ADC0_init(void);  
32 void get_ic_temp_init(void);  
33 void Clock_Init(void);  
34 void Hardware_init(void);  
35
```

PGA\_init()  
DAC\_init()  
CMP\_init()

5、hardware\_init.h 声明的 PGA\_Init、DAC\_Init、CMP\_Init 更改过后，对应的 hardware\_init.c 也需要更改。



```
40 2015/11/5 V1.0 Howlet Li 创建  
41 *****/  
42 void Hardware_init(void)  
43 {  
44     __disable_irq(); /* 关闭中断 中断总开关 */  
45     SYS_WR_PROTECT = 0x7a83;  
46  
47     FLASH_CFG |= 0x00080000; /* enable prefetch */  
48  
49     DSP_Init(); /* DSP模块初始化*/  
50     UART_init(); /* 串口初始化UART0*/  
51     ADC0_init(); /* ADC初始化 */  
52     MCPWM_init(); /* PWM初始化 */  
53     UTimer_init(); /* 通用计数器初始化 */  
54     GPIO_init(); /* GPIO初始化 */  
55     DAC_Init(); /* DAC 初始化 */  
56     PGA_Init(); /* PGA 初始化 */  
57     CMP_Init(); /* 比较器初始化 */  
58     HALL_Perip_Init(); /* HALL模块初始化 */  
59     TempSensor_Init(); /* 温度传感器初始化 */  
60  
61     SoftDelay(100); /* 延时等待硬件初始化稳定 */  
62
```

DAC\_init()  
PGA\_init()  
CMP\_init()



```
MC_hall.c hardware_init.c main.c hardware_init.h
214 输出参数: 无
215 返回值: 无
216 其它说明:
217 修改日期 版本号 修改人 修改内容
218 -----
219 2016/3/15 V1.0 Howlet Li 创建
220 *****/
221 void PGA_Init(void) void PGA_init(void)
222 {
223     SYS_AnalogModuleClockCmd(SYS_AnalogModule_OPA0, ENABLE);
224     SYS_AnalogModuleClockCmd(SYS_AnalogModule_OPA1, ENABLE);
225     SYS_AnalogModuleClockCmd(SYS_AnalogModule_OPA2, ENABLE);
226     SYS_AnalogModuleClockCmd(SYS_AnalogModule_OPA3, ENABLE);
227
228     SYS_AFE_REG0 = OPA0_GIAN | OPA1_GIAN | OPA2_GIAN | OPA3_GIAN; /* 4个OPA增益设置 */
229 }
230
231 /*****
232 函数名称: void CMP_Init(void)
233 功能描述: 比较器初始化
234 输入参数: 无
235 输出参数: 无
236 返回值: 无
237 其它说明:
238 修改日期 版本号 修改人 修改内容
239 -----
240 2016/3/15 V1.0 Howlet Li 创建
241 *****/
242 void CMP_Init(void) void CMP_init(void)
243 {
244     CMP_InitTypeDef CMP_InitStruct;
245
246     CMP_StructInit(&CMP_InitStruct);
247
248     CMP_InitStruct.CMP0_EN = ENABLE; /* 比较器0开关 */
249     CMP_InitStruct.CMP0_SELN = SELN_DAC;
250     CMP_InitStruct.CMP0_SELP = SELP_CMP_OPA0IP; /* CMP0_P CMP0_OPA0IP */
251     CMP_InitStruct.CMP0_InEnable = ENABLE; /* 比较器信号输入使能 */
252     CMP_InitStruct.CMP0_IE = DISABLE; /* 比较器0信号中断使能 */
253
254     CMP_InitStruct.CMP1_EN = ENABLE; /* 比较器1开关 */
255     CMP_InitStruct.CMP1_SELN = SELN_DAC; /* CMP1_N 内部参考 */
256 }
```

```
MC_hall.c hardware_init.c main.c hardware_init.h
277 2016/3/23 V1.0 Howlet Li 创建
278 *****/
279 void DAC_Init(void) void DAC_init(void)
280 {
281     SYS_AnalogModuleClockCmd(SYS_AnalogModule_DAC, ENABLE);
282     SYS_WR_PROTECT = 0x7a83;
283
284     SYS_AFE_REG1 &= ~(u32)0x03 << 6);
285     SYS_AFE_REG1 |= (DAC_RANGE_1V2 << 6); /* 设置DAC满量程为1.2V; 00:3V| 01:1.2V| 10:4.8
286
287     if(((SYS_AFE_REG1 >> 6)&0x03) == DAC_RANGE_1V2)
288     { /* 加载DAC 1.2V量程校正值 */
289         SYS_AFE_DAC_DC = Read_Trim(0x00000344);
290         SYS_AFE_DAC_AMC = Read_Trim(0x00000334);
291     }
292     else if(((SYS_AFE_REG1 >> 6)&0x03) == DAC_RANGE_3V0)
293     { /* 加载DAC 3.0V量程校正值 */
294         SYS_AFE_DAC_DC = Read_Trim(0x00000340);
295         SYS_AFE_DAC_AMC = Read_Trim(0x00000330);
296     }
297     else if(((SYS_AFE_REG1 >> 6)&0x03) == DAC_RANGE_4V85)
298     { /* 加载DAC 4.85V量程校正值 */
299         SYS_AFE_DAC_DC = Read_Trim(0x00000348);
300         SYS_AFE_DAC_AMC = Read_Trim(0x00000338);
301     }
302
303     SYS_AFE_DAC = 512; /* 1.2*512/4096/0.05 = 3A, 其中0.05为母线采样电
304 }
305
306 /*****
```

6、CMP\_init 初始化内容需要更改，对应结构体成员名称发生改变。

CMP\_InitStruct.CMP0\_InEnable 改为 CMP\_InitStruct.CMP0\_IN\_EN

CMP\_InitStruct.CMP1\_InEnable 改为 CMP\_InitStruct.CMP1\_IN\_EN

CMP\_InitStruct.CMP\_FltCnt = 15; 改为 CMP\_InitStruct.FIL\_CLK10\_DIV16 = 15;

CMP\_InitStruct.CMP\_CLK\_EN = ENABLE; 改为 CMP\_InitStruct.CLK10\_EN = ENABLE;

Comparator\_init(&CMP\_InitStruct); 改为 CMP\_Init(&CMP\_InitStruct);

更改前:

```
240 2016/3/15 V1.0 Howlet Li 创建
241 ****
242 void CMP_init(void)
243 {
244     CMP_InitTypeDef CMP_InitStruct;
245
246     CMP_StructInit(&CMP_InitStruct);
247
248     CMP_InitStruct.CMP0_EN = ENABLE; /* 比较器0开关 */
249     CMP_InitStruct.CMP0_SELN = SELN_DAC;
250     CMP_InitStruct.CMP0_SELP = SELP_CMP_OPA0IP; /* CMP0_P CMP0_OPA0IP */
251     CMP_InitStruct.CMP0_InEnable = ENABLE; /* 比较器信号输入使能 */
252     CMP_InitStruct.CMP0_IE = DISABLE; /* 比较器0信号中断使能 */
253
254
255     CMP_InitStruct.CMP1_EN = ENABLE; /* 比较器1开关 */
256     CMP_InitStruct.CMP1_SELN = SELN_DAC; /* CMP1_N 内部参考 */
257     CMP_InitStruct.CMP1_SELP = SELP_CMP_IP0; /* CMP1_P CMP1_OPA3 */
258     CMP_InitStruct.CMP1_InEnable = ENABLE; /* 比较器信号输入使能 */
259     CMP_InitStruct.CMP1_IE = ENABLE; /* 比较器1信号中断使能 */
260
261     CMP_InitStruct.CMP_FltCnt = 15; /* 即滤波宽度=tclk 周期*16*CMP_FltCnt */
262     CMP_InitStruct.CMP_CLK_EN = ENABLE; /* 时钟使能 */
263
264     Comparator_init(&CMP_InitStruct);
265 }
266
267
```

更改后:

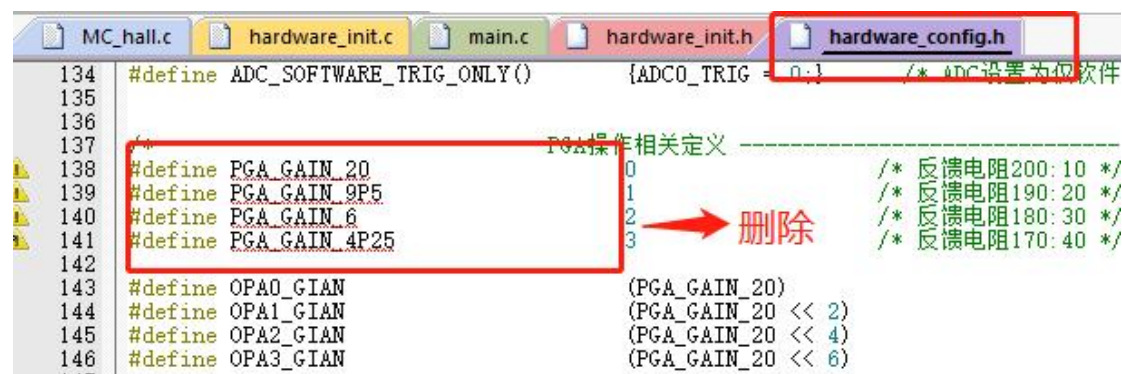
```
235 输出参数: 无
236 返回值: 无
237 其它说明:
238 修改日期 版本号 修改人 修改内容
239 -----
240 2016/3/15 V1.0 Howlet Li 创建
241 ****
242 void CMP_init(void)
243 {
244     CMP_InitTypeDef CMP_InitStruct;
245
246     CMP_StructInit(&CMP_InitStruct);
247
248     CMP_InitStruct.CMP0_EN = ENABLE; /* 比较器0开关 */
249     CMP_InitStruct.CMP0_SELN = SELN_DAC;
250     CMP_InitStruct.CMP0_SELP = SELP_CMP_OPA0IP; /* CMP0_P CMP0_OPA0IP */
251     CMP_InitStruct.CMP0_IN_EN = ENABLE; /* 比较器信号输入使能 */
252     CMP_InitStruct.CMP0_IE = DISABLE; /* 比较器0信号中断使能 */
253
254
255     CMP_InitStruct.CMP1_EN = ENABLE; /* 比较器1开关 */
256     CMP_InitStruct.CMP1_SELN = SELN_DAC; /* CMP1_N 内部参考 */
257     CMP_InitStruct.CMP1_SELP = SELP_CMP_IP0; /* CMP1_P CMP1_OPA3 */
258     CMP_InitStruct.CMP1_IN_EN = ENABLE; /* 比较器信号输入使能 */
259     CMP_InitStruct.CMP1_IE = ENABLE; /* 比较器1信号中断使能 */
260
261     CMP_InitStruct.FIL_CLK10_DIV16 = 15; /* 即滤波宽度=tclk 周期*16*CMP_FltCnt */
262     CMP_InitStruct.CLK10_EN = ENABLE; /* 时钟使能 */
263
264     CMP_Init(&CMP_InitStruct);
265 }
266
267
```

7、新的库函数将 OPA 的放大倍数放置在 lks32mc08x\_opa.h 文件内, 所以导致



hardware\_config.h 内的放大倍数宏定义重复，所以将 hardware\_config.h 内的 OPA 放大倍数宏定义删除。

更改前：



```
134 #define ADC_SOFTWARE_TRIG_ONLY() {ADCO_TRIG = 0;} /* ADC设置为仅软件 */
135
136
137 /* -----PGA操作相关定义 ----- */
138 #define PGA_GAIN_20 0 /* 反馈电阻200:10 */
139 #define PGA_GAIN_9P5 1 /* 反馈电阻190:20 */
140 #define PGA_GAIN_6 2 /* 反馈电阻180:30 */
141 #define PGA_GAIN_4P25 3 /* 反馈电阻170:40 */
142
143 #define OPA0_GIAN (PGA_GAIN_20)
144 #define OPA1_GIAN (PGA_GAIN_20 << 2)
145 #define OPA2_GIAN (PGA_GAIN_20 << 4)
146 #define OPA3_GIAN (PGA_GAIN_20 << 6)
```

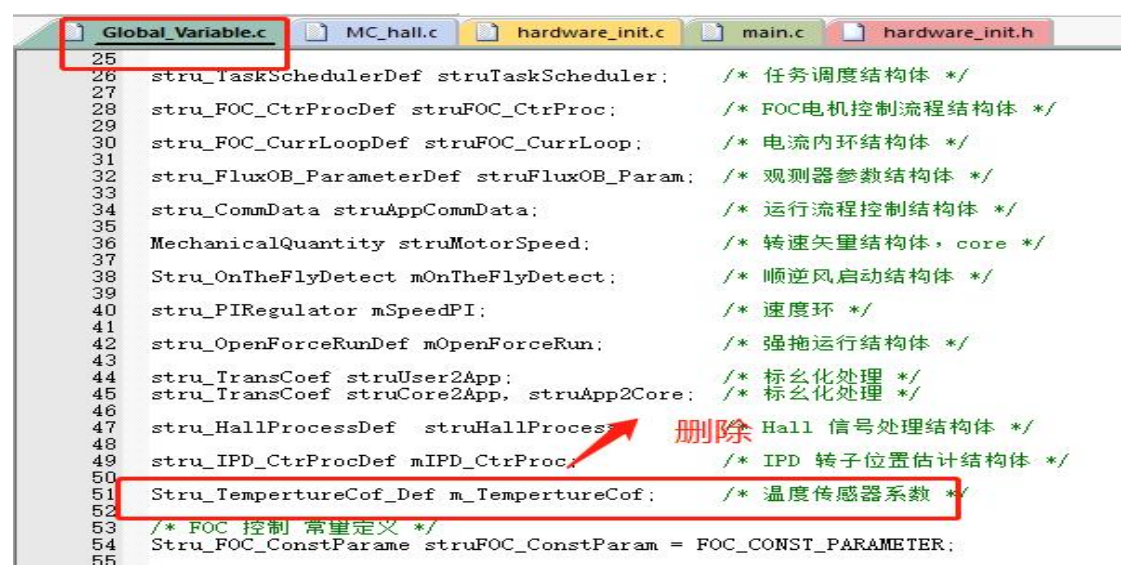
更改后：



```
134 #define ADC_SOFTWARE_TRIG_ONLY() {ADCO_TRIG = 0;} /* ADC设置为仅软件触发 */
135
136
137 /* -----PGA操作相关定义 ----- */
138
139 #define OPA0_GIAN (PGA_GAIN_20)
140 #define OPA1_GIAN (PGA_GAIN_20 << 2)
141 #define OPA2_GIAN (PGA_GAIN_20 << 4)
142 #define OPA3_GIAN (PGA_GAIN_20 << 6)
143
144 /* -----DAC操作相关定义 ----- */
145 #define DAC_RANGE_1V2 1 /* DAC 1.2V量程 */
```

8、因为新的库函数将芯片内置温度模块单独做了一个.c 和.h 文件，所以导致与 global\_variable.c、 hardware\_init.c 和 process\_control.c 内定义的相关温度传感器的函数重复，因此我们需要将

1) global\_variable.c 内的【Stru\_TempertureCof\_Def m\_TempertureCof; /\*温度传感器系数 \*/】 删除；



```
25
26 stru_TaskSchedulerDef struTaskScheduler; /* 任务调度结构体 */
27
28 stru_FOC_CtrProcDef struFOC_CtrProc; /* FOC电机控制流程结构体 */
29
30 stru_FOC_CurrLoopDef struFOC_CurrLoop; /* 电流内环结构体 */
31
32 stru_FluxOB_ParameterDef struFluxOB_Param; /* 观测器参数结构体 */
33
34 stru_CommData struAppCommData; /* 运行流程控制结构体 */
35
36 MechanicalQuantity struMotorSpeed; /* 转速矢量结构体，core */
37
38 Stru_OnTheFlyDetect mOnTheFlyDetect; /* 顺逆风启动结构体 */
39
40 stru_PIRegulator mSpeedPI; /* 速度环 */
41
42 stru_OpenForceRunDef mOpenForceRun; /* 强拖运行结构体 */
43
44 stru_TransCof struUser2App; /* 标么化处理 */
45 stru_TransCof struCore2App, struApp2Core; /* 标么化处理 */
46
47 stru_HallProcessDef struHallProcess; /* Hall 信号处理结构体 */
48
49 stru_IPD_CtrProcDef mIPD_CtrProc; /* IPD 转子位置估计结构体 */
50
51 Stru_TempertureCof_Def m_TempertureCof; /* 温度传感器系数 */
52
53 /* FOC 控制 常量定义 */
54 Stru_FOC_ConstParame struFOC_ConstParam = FOC_CONST_PARAMETER;
55
```

2) hardware\_init.c 内的【void TempSensor\_Init(void)】函数删除。

```
196
197 2020/11/5 V1.0 Howlet Li 创建
198 *****
199 void TempSensor_Init(void)
200 {
201     SYS_WR_PROTECT = 0x7a83; /* 解除系统寄存器写保护 */
202     SYS_AFE_REG5 |= BIT13; /* 打开温度传感器开关 */
203
204     m_TempertureCof.nCofA = Read_Trim(0x00000398);
205     m_TempertureCof.nOffsetB = Read_Trim(0x0000039C);
206 }
207
208 /*****
209 函数名称: void PGA_Init(void)
210 功能描述: PGA初始化
211 *****/
```

3) process\_control.c 内的【s16 GetCurrentTemperature(s16 ADC\_value)】函数删除。

```
97 }
98 }
99
100 /*****
101 函数名称: s16 GetCurrentTemperature(s16 ADC_value)
102 功能描述: 得到当前温度值
103 输入参数: ADC_value: ADC通道14为温度传感器, 参数输入需要以右对齐格式,
104           当ADC设置为左对齐时, 需要右移4位
105 输出参数: 无
106 返回值: t_Temperture: 当前温度值, 单位: 1个Lsb代表0.1度
107 其它说明:
108 修改日期 版本号 修改人 修改内容
109
110 2020/11/5 V1.0 Howlet Li 创建
111 *****/
112 s16 GetCurrentTemperature(s16 ADC_value)
113 {
114     s16 t_Temperture;
115     t_Temperture = (m_TempertureCof.nOffsetB - ((s32)m_TempertureCof.nCofA * ADC_value))
116
117     return t_Temperture;
118 }
119
120 /*****
```

》》》 到此刻在编译就不会报错啦!!!

9、最后还有 5 个警告需要去掉，主要原因是占空比寄存器是有符号型，但负数赋值都强制转为无符号型，所以我们将强制转换语句去掉即可。更改 hardware\_init.c 和 SignalChain\_Sample.c 文件。

```
346 修改日期 版本号 修改人 修改内容
347
348 2020/8/5 V1.0 Howlet Li 创建
349 *****
350 void ADC_NormalModeCFG(void)
351 {
352     ADC0_CHN0 = ADC0_CURRETN_A_CHANNEL | (ADC0_CURRETN_B_CHANNEL << 8);
353     ADC0_CHN1 = ADC0_CURRETN_A_CHANNEL | (ADC0_CURRETN_B_CHANNEL << 8);
354     ADC0_CHN2 = ADC0_BUS_VOL_CHANNEL | (ADC0_BUS_VOL_CHANNEL << 8); //
355
356     MCPWM_TMR0 = (u16)(10-PWM_PERIOD); // 100-PWM_PERIOD
357 }
358
359 #endif
360
```

```
Process_Control.c MC_hall.c hardware_init.c main.c hardware_init.h
2017/11/5 V1.0 Howlet Li 创建
*****
661 void seekPosADCsetting(void)
662 {
663     ADC0_TRIG = BIT1; /*硬件触发ADC MCPWM T1被使能 */
664     ADC0_CFG |= BIT2; /*状态机复位, idle状态, 完成后自动清0 */
665     MCPWM_TMR0 = PWM_PERIOD; /*ADC采样时刻0设置为下桥关断中心点 */
666     MCPWM_TMR1 = (u16)(-600); /*ADC采样时刻1设置为PWM中心点向后偏600个Clock*/
667 }
668
669 /*****
670 函数名称: void DebugPWM_OutputFunction(void)
671 功能描述: PWM输出功能调试 输出25%占空比
672 输入参数: 无
673 输出参数: 无
674 返回值: 无
675 其它说明:
676 修改日期 版本号 修改人 修改内容
677 2017/11/5 V1.0 Howlet Li 创建
678 *****
679 void DebugPWM_OutputFunction(void)
680 {
681     MCPWM_TH00 = (u16)(-(PWM_PERIOD >> 2));
682     MCPWM_TH01 = (PWM_PERIOD >> 2);
683     MCPWM_TH10 = (u16)(-(PWM_PERIOD >> 2));
684     MCPWM_TH11 = (PWM_PERIOD >> 2);
685     MCPWM_TH20 = (u16)(-(PWM_PERIOD >> 2));
686     MCPWM_TH21 = (u16)(-(PWM_PERIOD >> 2));
687     PWMOutputs(ENABLE);
688     while(1)
689     {
690     }
691 }
692
693
694
```