FUJITSU MICROELECTRONICS (SHANGHAI) CO., LTD.
Unit 1501, Jin Mao Tower, 88 Century Ave., Pudong New Area, Shanghai, P.R. China
Tel: 61007800 Fax: 61007802

MCU-AN-500089-E-10

F²MC-8FX FAMILY
8-BIT MICROCONTROLLER
MB95F430 SERIES

Induction Heater C Library User Manual

Revision History

Version	Date	Comment	Charge
1.0	2010-03-28	Create.	Folix
1.1	2010-08-11	Modified: MCU Pin Assignment;	Folix

Contents

1. System Introduce

This document introduce how to use the C library API to develop the Induction Heater based MB95F434K.

2. System Hardware Environment

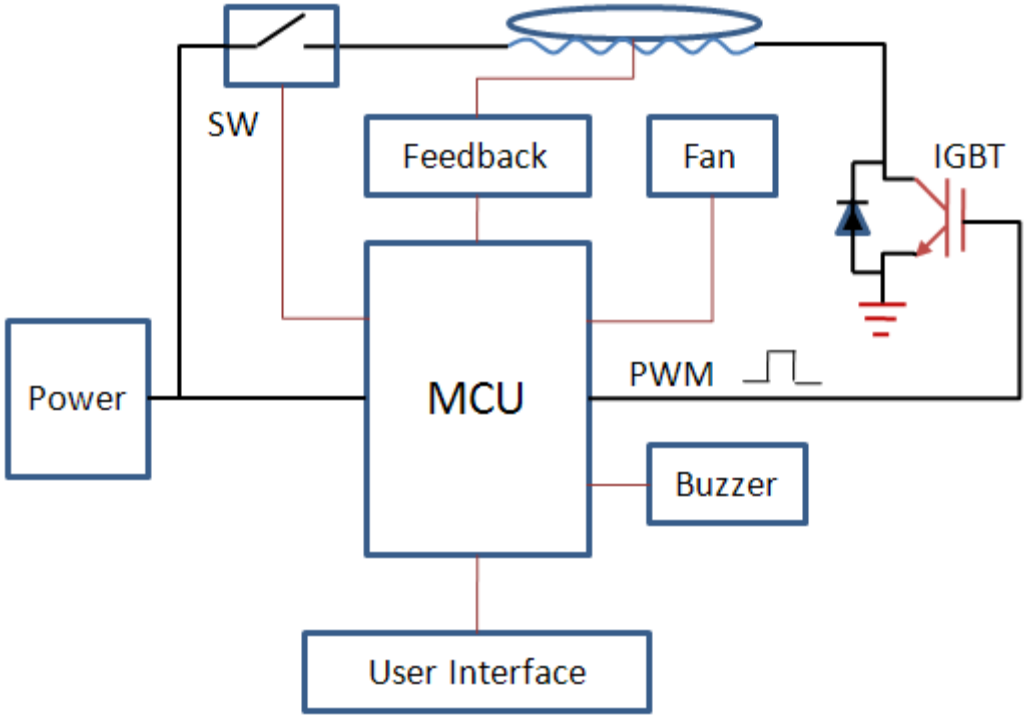
- ➢ CPU Chip: Fujitsu MB95F434K;
- ➢ CPU Frequency: 16.25MHz;
- ➢ Minimum Instruction Time: 61.5ns;
- ➢ Ram Space: 496Bytes;
- ➢ Code Space: 20KBytes;

3. Development Environment and Tools

Name	Description	Manufacturer	Notes
Windows XP Pro	PC OS	Microsoft	SP2
Softune V3	Software Developing IDE	Fujitsu	For FFMC-8L
MB95F434K Emulator	MCU Emulator	Fujitsu	---
Cadence SPB	Design Schematic and PCB	Cadence	V16.0

4. Hardware Design

4.1. System Hardware Block Diagram



The electromagnetic eddy current heating is our system theory. We use the PWM to control the IGBT to generate the heating eddy current. We can adjust the PWM width to adjust the power. The fan is use to cool the IGBT. The feedback signal is used to protect the system and calculate power. The feedback signal includes current, voltage, surge and temperature.

4.2. MCU Resource Usage

MB95F434K Resource Usage			
No.	**Resource**	**Usage Info**	**Remark**
1	VCC	5.0V	
2	ROM	15KBytes	
3	RAM	300Bytes	
4	OSC Clock	16.25MHz	
5	CPU Clock	8.125MHz	
6	GPIO	6 Pins	LED COM(5),Start
7	INTxx	1 Pins (Internal)	Pad osc
8	8/16Bit PPG(Timer)	1 Pins(PPG)	
9	ANxx	4 Pins	
10	UART0	LED Display	
11	Debug	1 Pins	
12	Watchdog Timer	Monitor System	
13	16Bit Timer0	Keyboard Sample	
14	16Bit Timer1	10MS Timer	Protect,Error

4.3. MCU Pin Assignment

No.	Name	Define	Note	Remark
1	PG2/PPG0/X1A/OUT1	sub oscillator	LED COM1	
2	PG1/TRG0/ADTG/X0A/BZ/OUT0	sub oscillator	Fan	
3	VCC	VCC	Vcc	
4	C	C	C	
5	P60/OPAMP_P	OPAMP_P	Current by Resistor	
6	P61/OPAMP_N	OPAMP_N		
7	P62/OPAMP_O	P62	LED COM2	
8	P12/EC0/UI/DBG	P12/DBG	Debug	
9	P00/INT00/AN00	P00	Start IGBT	
10	P01/INT01/AN01/BZ	BZ	Buzzer	
11	P02/INT02/AN02/UCK/TRG0	UCK	Serial Clock	
12	P03/INT03/AN03/UO/PPG0	UO	Serial Data	
13	P04/INT04/AN04/UI/HCLK1	AN04	AC voltage Measuring	
14	P05/INT05/AN05/TO0/HCLK2	AN05	AC Current Measuring	
15	P06/INT06/AN06/TO1	AN06	Temperature for IGBT	
16	P07/INT07/AN07/EC0	AN07	Temperature for plate	
17	P70/CMP0_O/OUT0/TRG0	P70	LED COM4	
18	P71/CMP0_P	CMP0_P	PAD-L+ Input	
19	P72/CMP0_N	CMP0_N	PAD-L- Input	
20	P73/CMP1_O/OUT1/PPG0	PPG0	PPG-IGBT	
21	P74/CMP1_P	CMP1_P	OverCurrent Reference Input	
22	P75/CMP1_N	CMP1_n	OverCurrent Compare Input	
23	P76/CMP2_O/UCK	P76	LED COM3	
24	P63/CMP2_P	CMP2_P	OverVoltage Reference Input	
25	P64/CMP2_N	CMP2_N	OverVoltage Compare Input	
26	P65/CMP3_O/UO	P65	LED COM0	
27	P66/CMP3_P	CMP3_P	Surge Reference Input	
28	P67/CMP3_N	CMP3_N	Surge Compare Input	
29	PF2/RSTX	PF2	Reset	
30	PF0/X0	X0	oscillator	
31	PF1/X1	X1	oscillator	
32	VSS	VSS	Vss	

5. System Function

5.1. Global Variable

Variable	Description	Note
BYTE dVal[4];	Display for 4-bit digital.	Used in Display function.
BYTE dLed;	Display for 8 led.	Used in Display function.

5.2. Function List

Prototype	Description	Note
void InitSys(void)	Initalize system.	call in the system at first
int GetKey(void)	Get the user pushed key.	
int ParseKey(int key)	Get the system event.	
void EventProc(int evt)	Process the event by the event value.	
void Display()	Display process	

5.3. Function Prototype

5.3.1. InitSys ()

Prototype	void InitSys(void)
Output	void
Input	void
Description	Initalize system.

5.3.2. GetKey ()

Prototype	int GetKey(void)
Output	int, return the key value.
Input	void
Description	Get the user pushed key.

5.3.3. ParseKey ()

Prototype	int ParseKey(int key)
Output	int, return the event value.
Input	key, the user pushed key.
Description	Get the system event.

5.3.4. EventProc ()

Prototype	void EventProc(int evt)
Output	void
Input	evt, the system event.
Description	Process the event by the event value.

5.3.5. Display ()

Prototype	void Display()
Output	void
Input	void.
Description	Display.

6. Event Function

6.1. Function List

Prototype	Description	Note
void ModeCaoCai(void)	Change mode to CaoCai.	
void ModeHuoGuo(void)	Change mode to HuoGuo.	
void ModeDunCai(void)	Change mode to DunCai.	
void ModeBaoTang(void)	Change mode to BaoTang.	
void ModeBaoWen(void)	Change mode to BaoWen.	
void WattInc(void)	Increase the watt.	
void WattDec(void)	Decrease the watt.	
void TmrInc(void)	Increase the timer's value.	
void TmrDec(void)	Increase the timer's value.	
void TmrRst(void)	Reset and turn off the timer or start the timer.	
void TempInc(void)	Increase the temperature in stew mode.	
void TempDec(void)	Decrease the temperature in stew mode.	
void ConstTempProc(void)	Constant temperature heating process. It is used in empty event.	

6.2. Function Prototype

6.2.1. ModeCaoCai ()

Prototype	void ModeCaoCai(void)
Output	void
Input	void
Description	Change mode to CaoCai.

6.2.2. ModeHuoGuo ()

Prototype	void ModeHuoGuo(void)
Output	void
Input	void
Description	Change mode to HuoGuo.

6.2.3. ModeDunCai ()

Prototype	void ModeDunCai(void)
Output	void
Input	void
Description	Change mode to DunCai.

6.2.4. ModeBaoTang ()

Prototype	void ModeBaoTang(void)
Output	void
Input	void
Description	Change mode to BaoTang.

6.2.5. ModeBaoWen ()

Prototype	void ModeBaoWen(void)
Output	void
Input	void
Description	Change mode to BaoWen.

6.2.6. WattInc ()

Prototype	void WattInc(void)
Output	void
Input	void
Description	Increase the watt 100w. Range is from 200w to 2100w.

6.2.7. WattDec ()

Prototype	void WattDec(void)
Output	void
Input	void
Description	Decrease the watt 100w. Range is from 200w to 2100w.

6.2.8. TmrInc ()

Prototype	void TmrInc(void)
Output	void
Input	void
Description	Increase the timer’s value 1 minute. Range is from 1 to 180.

6.2.9. TmrDec ()

Prototype	void TmrDec(void)
Output	void
Input	void
Description	Increase the timer’s value 1 minute. Range is from 1 to 180.

6.2.10. TmrRst ()

Prototype	void TmrRst(void)
Output	void
Input	void
Description	Reset and turn off the timer or start the timer.

6.2.11. TempInc ()

Prototype	void TempInc(void)
Output	void
Input	void
Description	Increase the temperature 10℃. Range is from 60 to 180.

6.2.12. TempDec ()

Prototype	void TempDec(void)
Output	void
Input	void
Description	Decrease the temperature 10℃. Range is from 60 to 180.

6.2.13. ConstTempProc ()

Prototype	void ConstTempProc(void)
Output	void
Input	void
Description	Constant temperature heating process. It is used in empty event.

7. Driver Function

7.1. Function List

Prototype	Description	Note
void GetADC(int chl)	Get the ADC value.	
void BuzzerOn(void)	Turn on the buzzer.	
void BuzzerOff(void)	Turn off the buzzer.	
void PowerOn(void)	Turn on the system.	
void PowerOff(void)	Turn off the system.	
void PWMSet(int cycle,int duty)	Set the PWM.	
void PwmOn(void)	Turn on the PWM driver.	
void PwmOff(void)	Turn off the PWM driver.	
void FanOn(void)	Turn on the fan.	
void FanOff(void)	Turn off the fan.	
void VCInit(void)	Initialize all the VC.	
void OPAMPInit(void)	Initialize the OPAMP.	

7.2. Function Prototype

7.2.1. GetADC (int chl)

Prototype	void GetADC(int chl)
Output	void
Input	Channel, Range is 1~4. 1---Voltage (AN4), 2---Current (AN5), 3---Igbt temp (AN6), 4---Pad temp (AN7)
Description	get the ADC.

7.2.2. BuzzerOn ()

Prototype	void BuzzerOn(void)
Output	void
Input	void
Description	Turn on the buzzer.

7.2.3. BuzzerOff ()

Prototype	void BuzzerOff(void)
Output	void
Input	void
Description	Turn off the buzzer.

7.2.4. PowerOn ()

Prototype	void PowerOn(void)
Output	void
Input	void
Description	Turn on the system, and set in the braise mode.

7.2.5. PowerOff ()

Prototype	void PowerOff(void)
Output	void
Input	void
Description	Turn off the system.

7.2.6. PWMSet (int cycle,int duty)

Prototype	void PWMSet(int cycle,int duty)
Output	void
Input	Cycle,duty
Description	Set the PWM. The cycle must be greater than the duty.

7.2.7. PwmOn ()

Prototype	void PwmOn(void)
Output	void
Input	void
Description	Turn on the PWM driver.

7.2.8. PwmOff ()

Prototype	void PwmOff(void)
Output	void
Input	void
Description	Turn off the PWM driver.

7.2.9. FanOn ()

Prototype	void FanOn(void)
Output	void
Input	void
Description	Turn on the fan.

7.2.10. FanOff ()

Prototype	void FanOff(void)
Output	void
Input	void
Description	Turn off the fan.

7.2.11. VCInit ()

Prototype	void VCInit(void)
Output	void
Input	void
Description	Initialize all the VC.

7.2.12. OPAMPInit ()

Prototype	void OPAMPInit(void)
Output	void
Input	void
Description	Initialize the OPAMP.

8. Interrupt Function

8.1. Function List

Prototype	Description	Note
void Keyboard(void)	Sample keyboard and parse key value.	It is used in 10ms timer interrupt.
void ProtectIGBT(void)	Detect IGBT temp and start the fan.	It is used in 10ms timer interrupt.
void ProtectSys(void)	Detect error and protect system.	It is used in 10ms timer interrupt.
void IntCurOver(void)	External interrupt service for over current.	
void IntVolOver(void)	External interrupt service for over voltage.	
void IntSurge(void)	External interrupt service for surging.	
void IntKettle(void)	External interrupt service for no kettle.	

8.2. Function Prototype

8.2.1. Keyboard ()

Prototype	void Keyboard(void)
Output	void
Input	void
Description	Sample keyboard and parse key value. It is used in 10ms timer interrupt. Use 16bit timer 0 channel.

8.2.2. ProtectIGBT ()

Prototype	void ProtectIGBT(void)
Output	void
Input	void
Description	Detect IGBT temp and start the fan. It is used in 10ms timer interrupt. Use 16bit timer 1 channel.

8.2.3. ProtectSys ()

Prototype	void ProtectSys(void)
Output	void
Input	void
Description	Detect error and protect system. It is used in 10ms timer interrupt. Use 16bit timer 1 channel.

8.2.4. IntCurOver ()

Prototype	void IntCurOver(void)
Output	void
Input	void
Description	External interrupt service for over current. Use VC 1 channel.

8.2.5. IntVolOver ()

Prototype	void IntVolOver(void)
Output	void

Input	void
Description	External interrupt service for over voltage. Use VC 2 channel.

8.2.6. IntSurge ()

Prototype	void IntSurge(void)
Output	void
Input	void
Description	External interrupt service for surging. Use VC 3 channel.

8.2.7. IntKettle ()

Prototype	void IntKettle(void)
Output	void
Input	void
Description	External interrupt service for no kettle. Use VC 0 channel.

9. Demo System

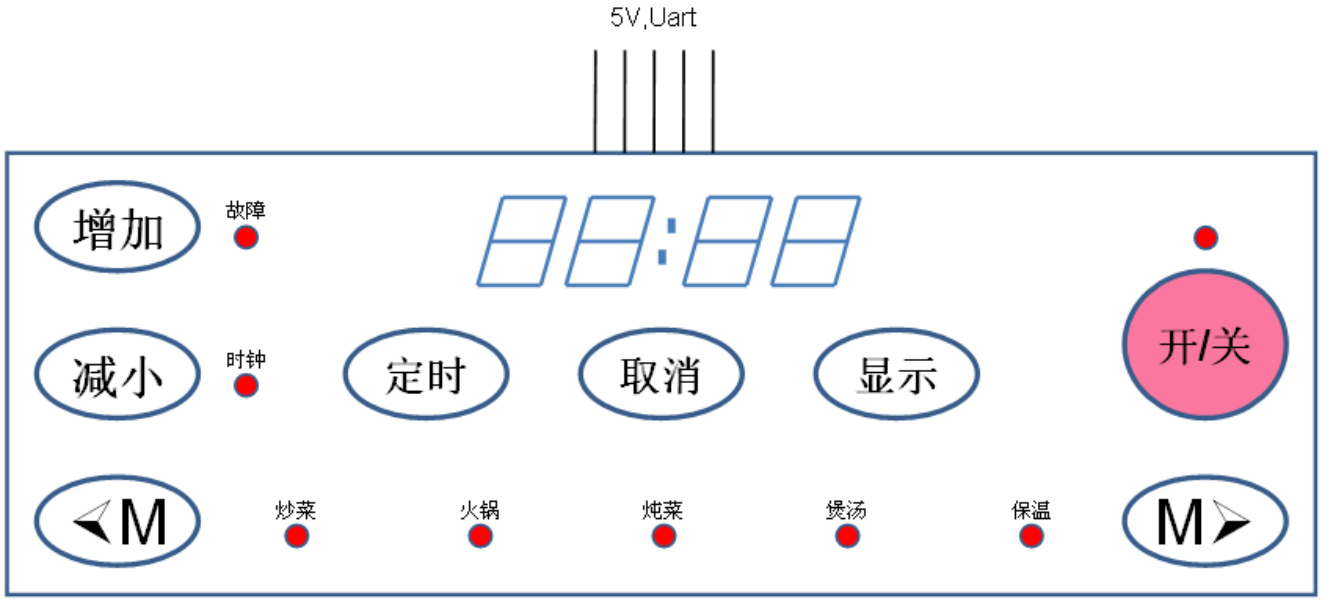
9.1. System Features

- Power Control
- Fan Control
- Fan Delay Off
- Buzz Control
- Led Display
- Keyboard
- Kettle Detect
- Kettle Empty Detect
- Timing Power Off(3Hours)
- Constant Power Heating
- Constant Temperature Heating
- IGBT Temp Detect
- Plate Temp Detect
- IGBT Temp Higher Protect(120℃)
- Plate Temp Higher Protect(300℃)
- Kettle Empty Protect(300℃)
- Sensor Open/Short Detection
- Over Current Protect(18A)
- Over Voltage Protect(270V)
- Lower Voltage Protect(150V)
- Surge Protect(1000V)
- No Operation for 2 Hours, auto turn off,if no timer

In the system design, we will use some feedback signals to protect system or calculate watt. The protect signal include over current, over voltage, surge voltage and kettle detect. Anyone of them appears, the system must stop work. If the IGBT temperature is raise over 60℃, the fan must turn on. If the IGBT temperature is raise over 120℃, the system must stop work. If the kettle is empty, the plate temp will raise. If the temp raise up to 300℃, the system must stop work.

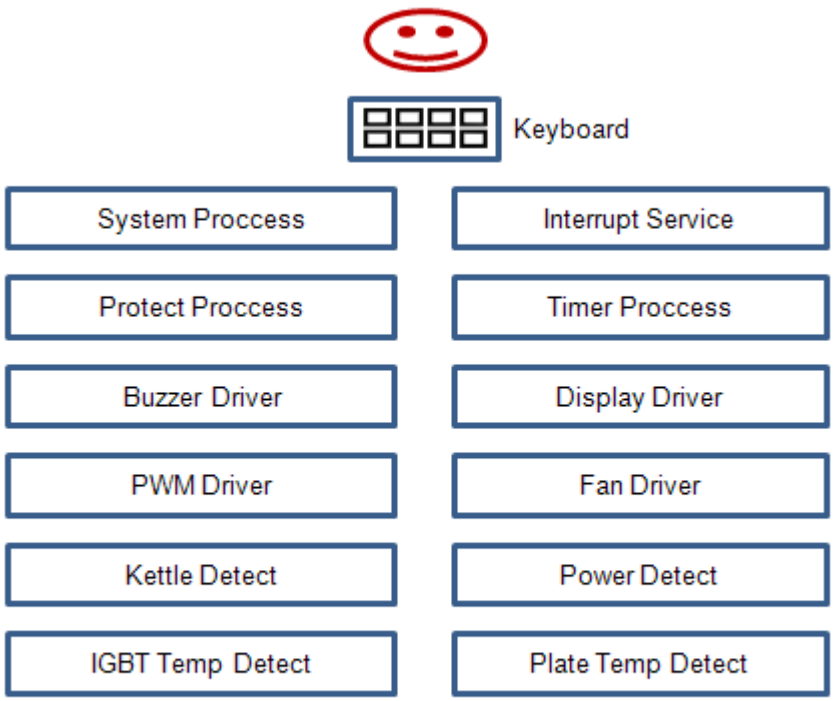
9.2. Software Design

9.2.1. User Interface



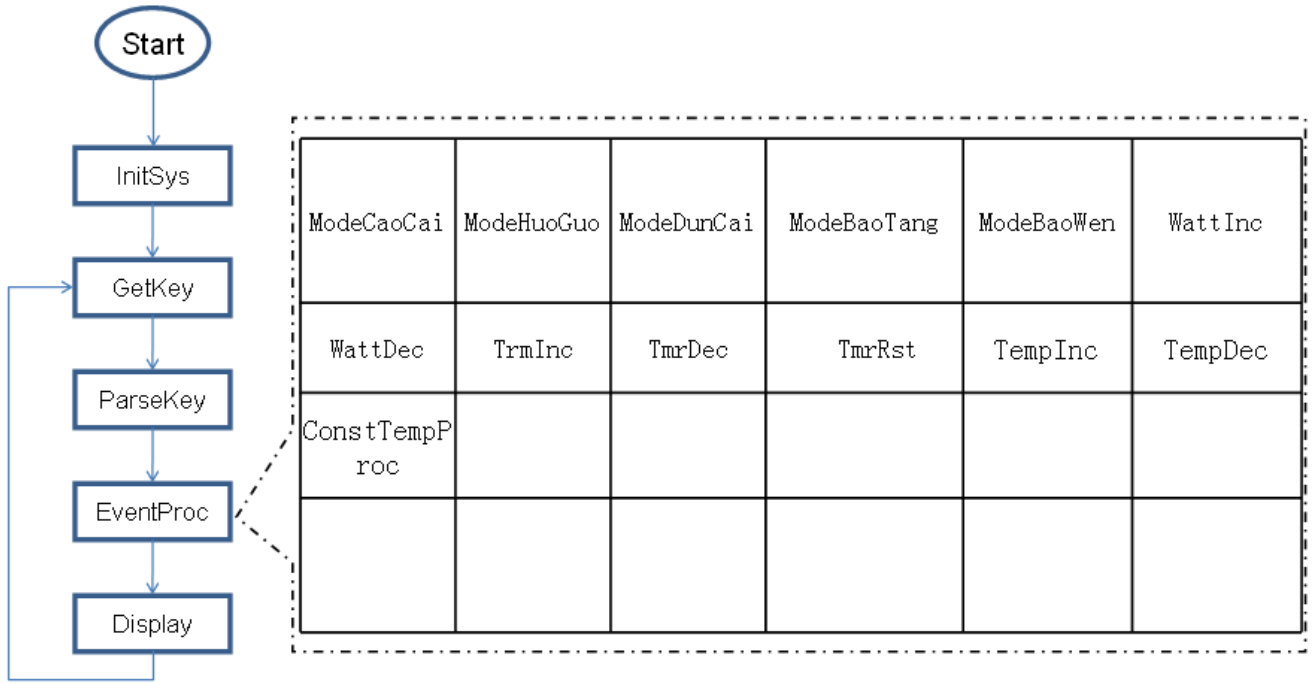
The system has six work mode. You can push the “M” (left) and “M” (right) to change the work mode. When power on, the system is standby. When you push the “开/关”, it will work under the “解冻” mode. In any mode, you can push the “开/关” to turn off or push the “增加” or “减小” to change the watt or the time when you are setting the timer. When you want to heat food by timer, you can push “定时” to set and start the timer. And you can push the “取消” to cancel the timer. You can push the “显示” to view the timer. If the time is zero, the timer is not started.

9.2.2. System Module



System Function Module

9.2.3. System Main Loop



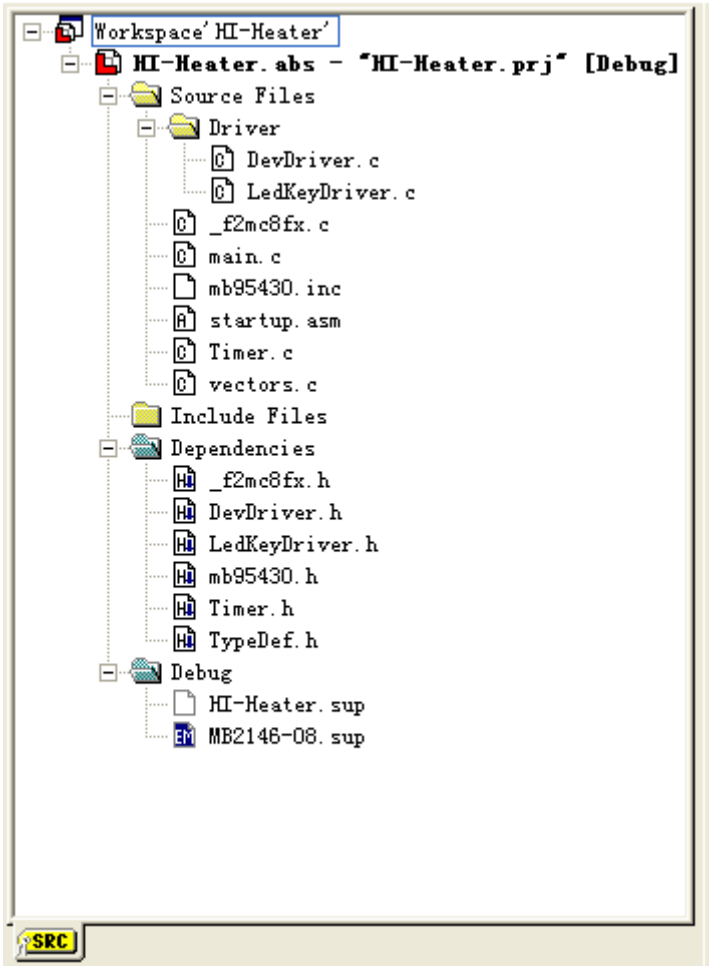
System Main Loop

Note: The main loop time is about 10ms.

9.2.4. System Files

In this application, there are 4 files: main.c, vectors.c, devdriver.c, ledkeydriver.c and startup.asm.
main.c, includes main loop and MCU control functions.
vectors.c, includes interrupt functions.
devdriver.c, includes device drivers.
ledkeydriver.c, includes led and key drivers.
startup.asm, is MCU startup codes.

9.2.5. System Files Structure



Appendix

- **Display Content**

Mode	Display Content	Notes
System	Power Led, Timer Led	
CaoCai Mode	CaoCai Led, Watt Display	
HuoGuo Mode	HuoGuo Led, Watt Display	
DunCai Mode	DunCai Led, Watt Display	
BaoTang Mode	BaoTang Led, Watt Display	
BaoWen Mode	BaoWen Led, Temp Display	
JieDong Mode	JieDong Led, Temp Display	
Error Mode	Error Code Display, error led.	Display such as “Er:E3”.

- **Error Code**

Error Code	Description	Notes
0xE0	Power under voltage	
0xE1	Power over voltage	
0xE2	Power over current	
0xE3	Main sensor over 300℃	
0xE4	Main sensor break circuit	
0xE5	Main sensor short circuit	
0xE6	Main sensor temp not change	
0xE7	IGBT temp over 120℃	
0xE8	IGBT sensor break circuit	
0xE9	IGBT sensor short circuit	
0xF0	Kettle is empty	
0xF1		