High-Speed Hair Dryer Application Manual

3-phase Motor Control MCU
FU6862L

Fortior Technology (Shenzhen) Co., Ltd

Table of Contents

1 Overview

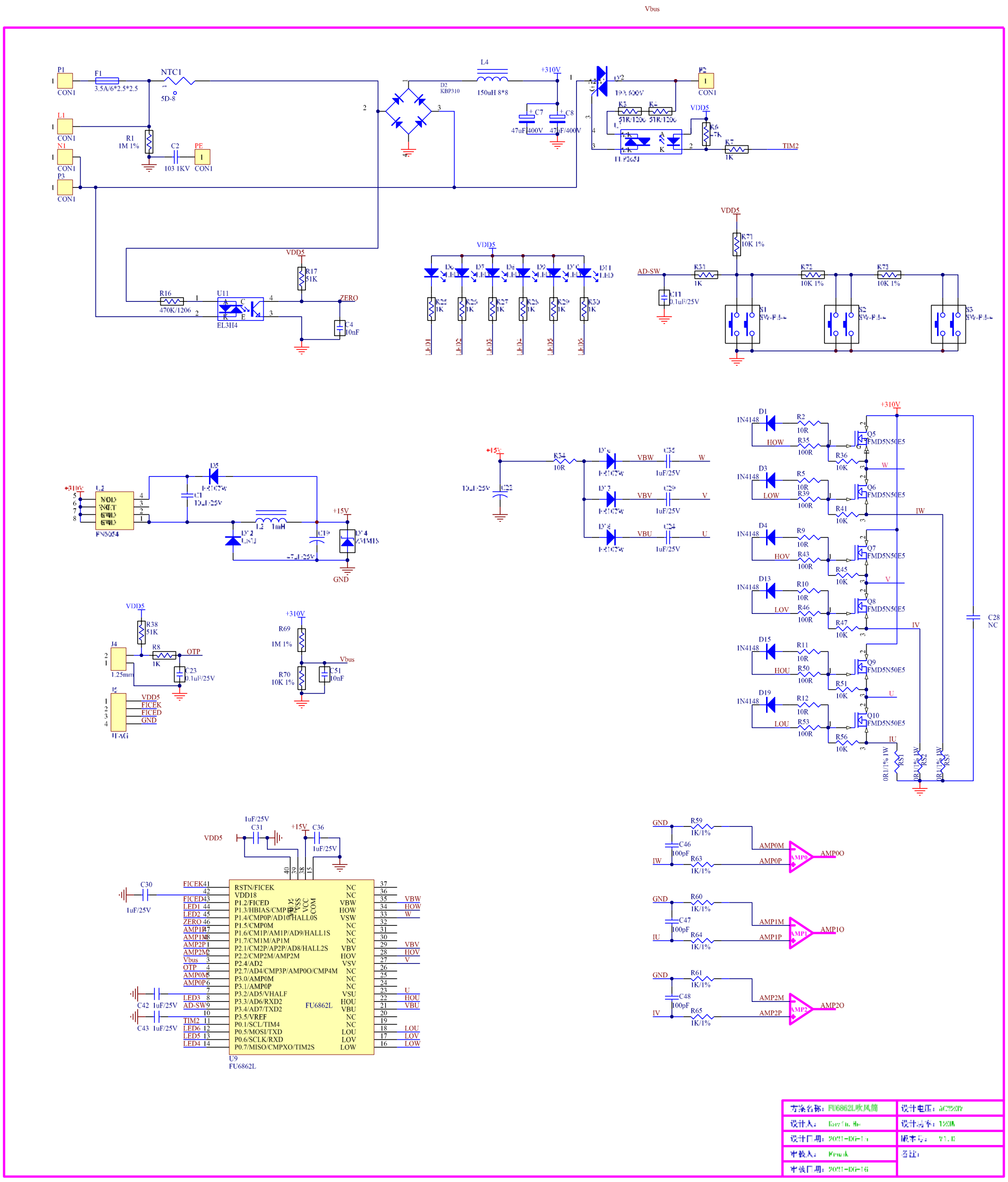
This debugging manual introduces in detail how to use the FU6862L chip of Fortior Technology to perform sensorless FOC driver control on BLDC high-speed hair dryer motor on the demo board exclusive for high-speed hair dryers. Users can have a quick browse of hardware principles in Chapter 2 and software principles in Chapter 3 first, then focus on the debugging steps in Chapter 4.

Software and Hardware Involved

Software/ hardware	Name	Related Chapters	Notes
Software	FU-AM-FU6862-B-059-SW-V1.0.00-20221212	All	Debugging is conducted on this software
Hardware	FU-AM-FU6862-B-059-HW-V1.0.00-20221212	All	Debugging is conducted on this hardware

2 Hardware

2.1 Hardware Schematic Diagram



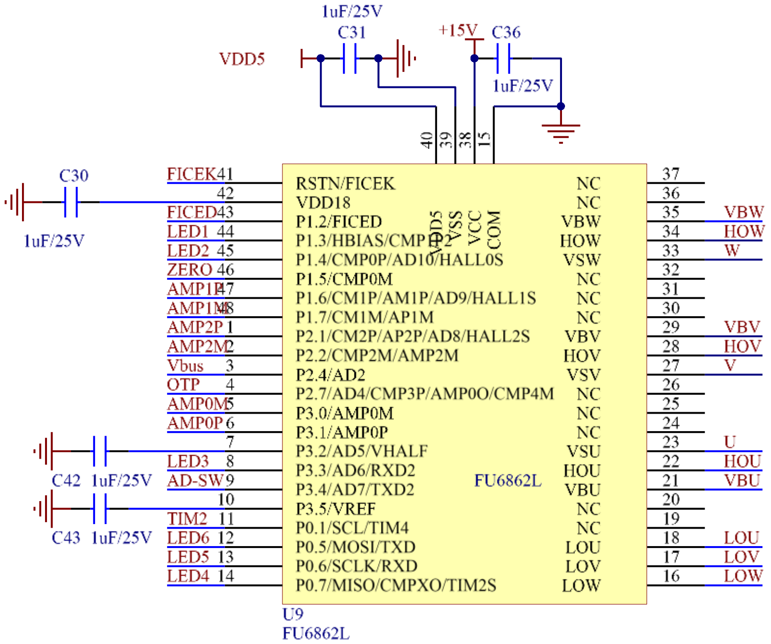
Hardware usage:

This demo board is exclusive for high-speed hair dryer application FU6862L three-resistor solution, which can be used directly after power-on.

Notes:

Users need to properly configure BUS voltage ratio, amplifier magnification, sampling resistance.

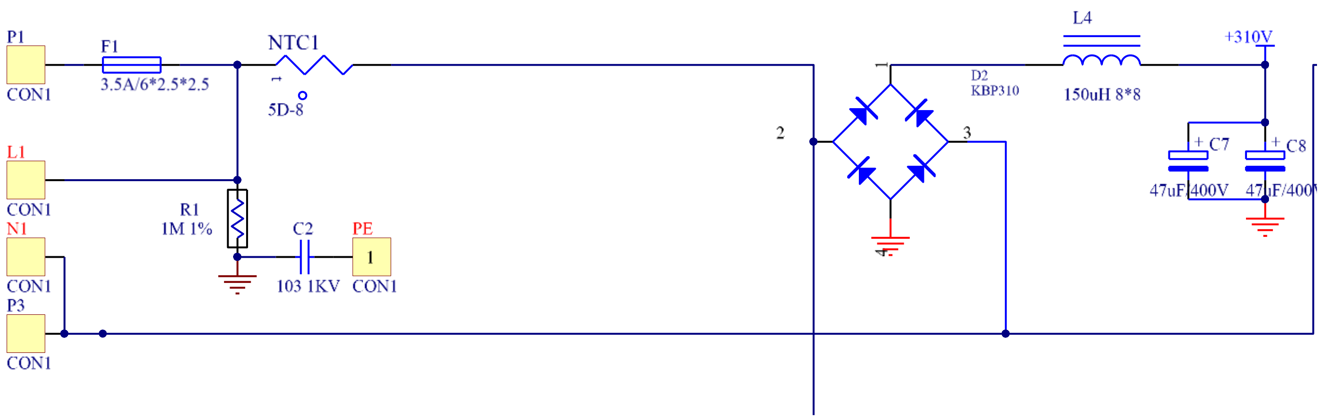
2.1.1 Chip Circuit



Instructions:

The FU6862L can be used in AC high voltage for 6 N-channel MOSFET driver applications.

2.1.2 Power Circuit



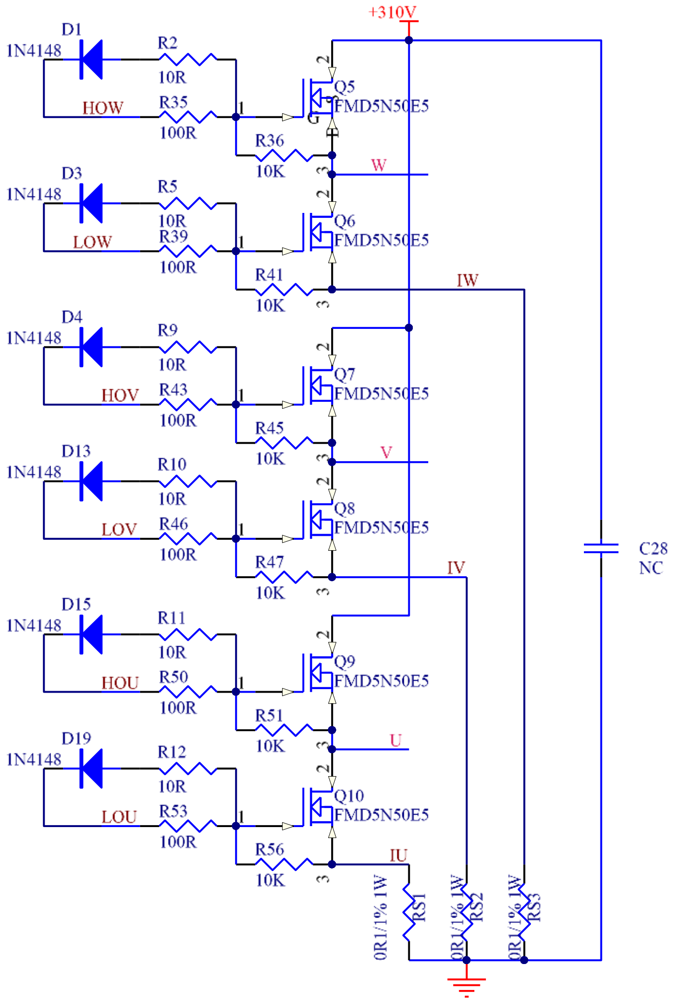
Instructions:

Connect AC power to both P1 and P3 pins.

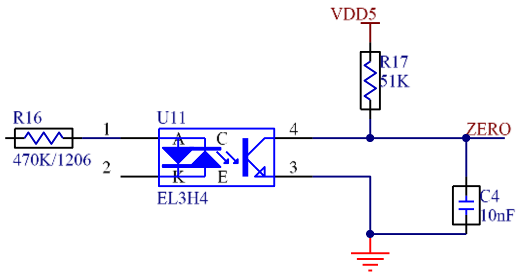
2.1.3 Power Drive Circuit

Notes:

When in maximum current conditions, the power of the sampling resistor cannot exceed 80% of the rated power.



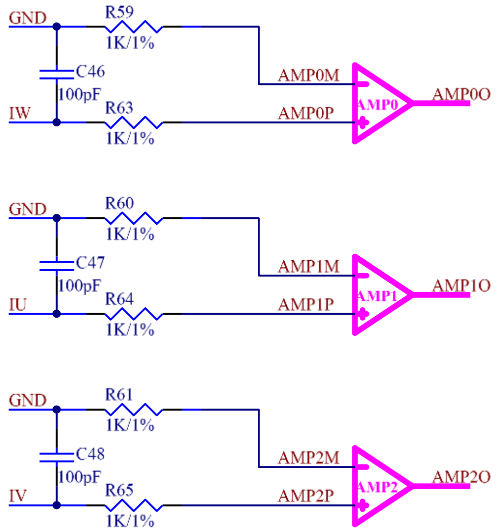
2.1.4 Zero-crossing Detection Circuit



Notes:

This circuit is used to detect zero-crossing signals. It can be triggered by an external interrupt. Therefore, the zero-crossing component needs to be connected to the input of the external interrupt. The C4 capacitor is best placed near the chip side to reduce interference.

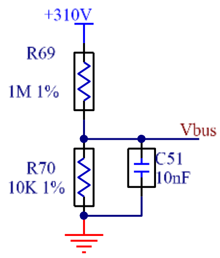
2.1.5 Op-amp Configuration Circuit



Notes:

1. C46, C47, C48, R59, R60, R61, R63, R64 and R65 require 1% precision resistor.
2. The schematic adopts differential input. The op-amp magnification is configurable in the program.
3. Maximum sampling current= (VREF - VHALF)/magnification/sampling resistor value;
4. The maximum sampling current is generally set to around 4 times of the maximum busbar current.

2.1.6 BUS Voltage Sampling Circuit



Notes:

1. R69、R70 should apply resistors with an accuracy of 1%;
2. Maximum sampling voltage = (R69 + R70)/(R70)*VREF；
3. In general, maximum sampling voltage is set as twice maximum application voltage, and the voltage at OVP should be lower than 0.8*VREF.

3 Software Architecture

3.1 Motor State Machine Flowchart

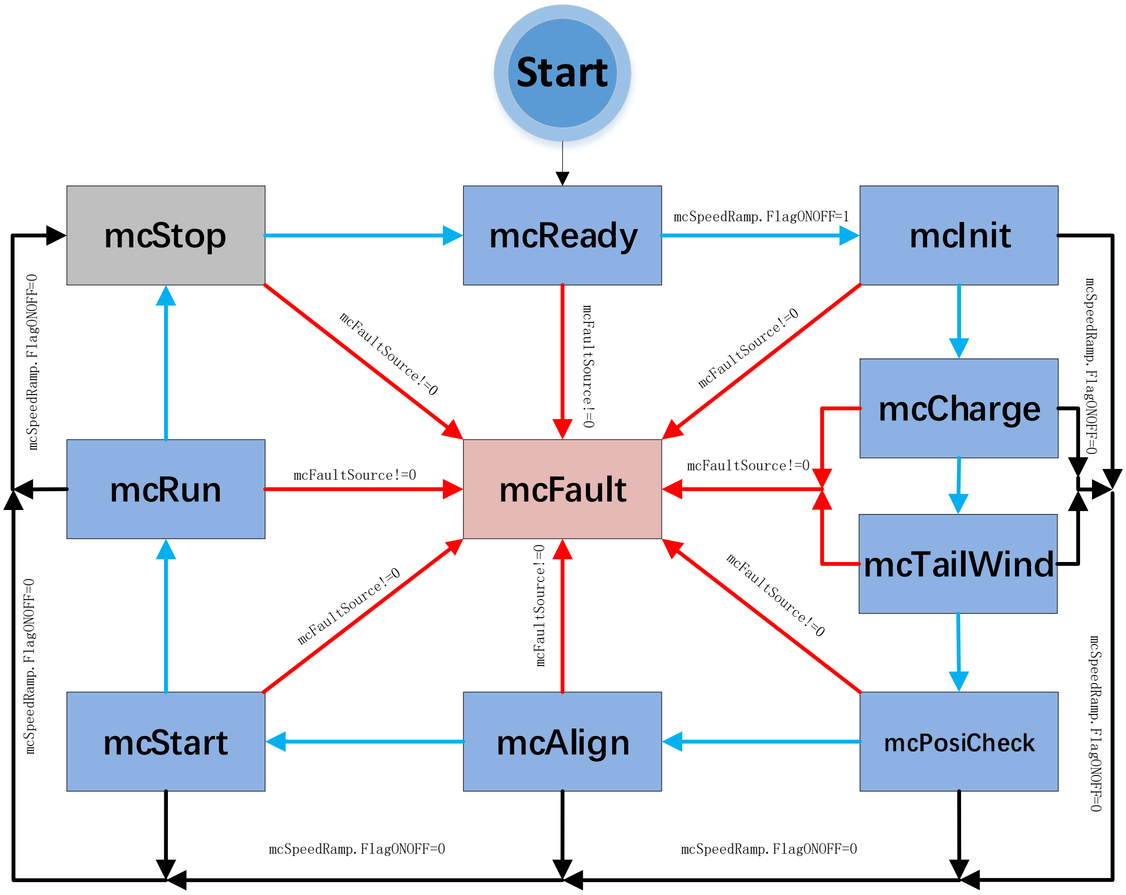


Figure 3-1 Motor State Machine Flowchart

Motor state machine is broken down into three paths as shown in the above figure:

1. Run: mcReady -> mcInit -> mcCharge -> mcTailWind -> mcPosiCheck -> mcAlign -> mcStart -> mcRun;
2. Stop:mcInit、mcCharge、mcTailWind、mcPosiCheck、mcAlign、mcStart、mcRun, once a shutdown signal is detected, it shifts to mcStop state to slow down then shutdown the motor;
3. Fault: if a fault occurs in any status, it jumps to mcFault state. As fault detection is not performed in mcFault state, concurrent reporting of multiple faults is unavailable.

Notes:

1. mcReady: ready state, waiting for start command. Upon start signal, it shifts to mcInit state;
2. mcInit: the state is to initialize related variables and PI, in which current and external ADC triggering for BUS sampling are switched off. It shifts to the next state once the operation is done.
3. mcCharge: The method used for pre charging is that the U phase upper and lower bridge arms are output in a complementary manner, followed by the U and V phases, and finally the U, V, and W phases are output in a complementary manner. The actual effective complementary output of the upper and lower bridge arms is the level output of the lower bridge arm, while the charging time is adjustable.

4. mcTailWind: The high-speed blower does not add a clockwise or counterclockwise judgment. Regardless of whether the wind is clockwise or counterclockwise, the high-speed blower directly stops by braking and then starts at a standstill.
5. mcPosiCheck: initial position detection state, which is to detect the initial position of a motor. This operation is done before normal startup procedure; The high-speed blower does not add initial position detection, but directly jumps to the pre-positioning state.
6. mcAlign: motor alignment state, in which controller outputs a constant current to drag the motor forcedly to a fixed angle. It shifts to mcStart once the operation is done. The pre-positioning time on the high-speed blower program is very short, which is equivalent to directly pulling and starting without positioning.
7. mcStart: start state, which is majorly to configure motor startup code. Once the configuration of related registers and variables is done, it enters the next state mcRun. Motor startup process is fulfilled in ME Core.
8. mcRun: run state covers both motor startup stage and running stage. Motor speed control is performed in this state.
9. mcStop: Shutdown state, which is used for shutdown operations. When the speed is reduced to a certain value, brake and switch to the mcBreak state.
10. mcBrake In the braking state, perform braking processing (braking has been processed in the mcStop state in the program), and switch to the mcReady state after braking;
11. mcFault: fault state. Upon protection occurrence, the program records the error source and shifts state machine to fault state to perform shutdown protection. When the error source is cleared, it enters mcReady state to wait for the next start command.

Notes:

1. The motor state machine supports 8 states, allowing only fixed transition among them. E.g., mcReady state can only switch to mcInit state and mcFault state;
2. In particular, the three states, mcTailWind, mcPosiCheck and mcAlign, all support enable bits. When they are not enabled, it skips to the next state directly. E.g., when neither mcPosiCheck nor mcAlign is enabled, it switches from mcTailWind to mcStart directly. "The high-speed hair dryer directly skips the mcTailWind, mcPosiCheck, and mcAlign states, which are only given 1ms, equivalent to direct Omega startup.".

3.2 Program Flowchart

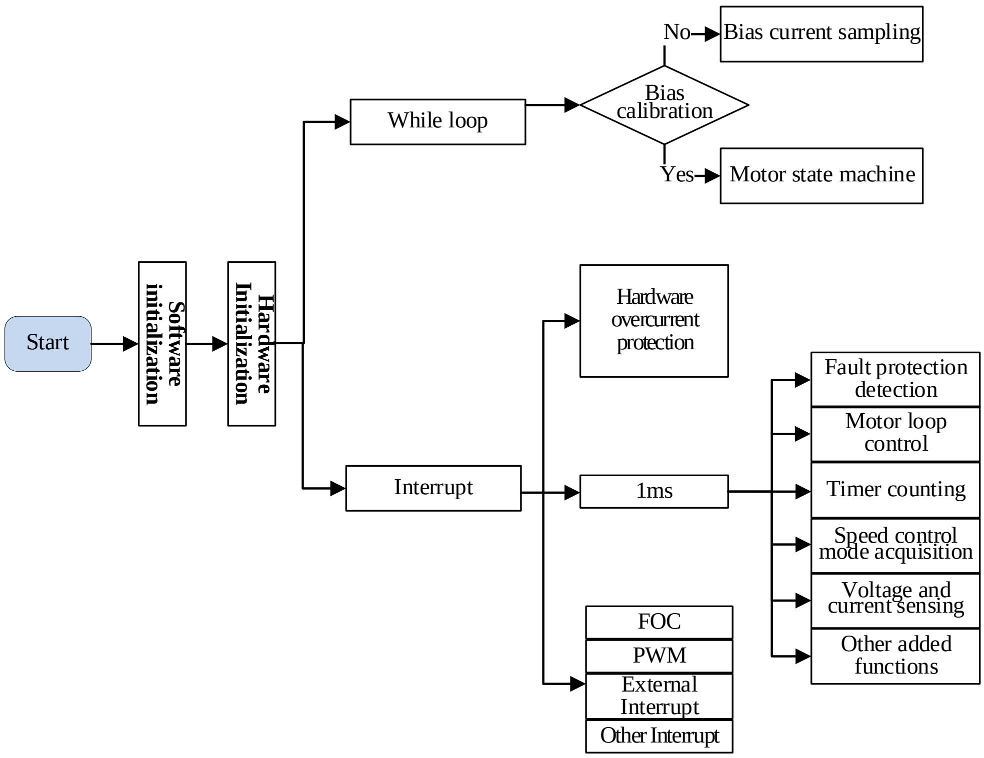


Figure 3-2 Program Flowchart

3.3 Program Description

3.3.1 Main Function

Program initialization -> GetCurrentOffset() for bias voltage detection + MC_Control() for motor control.

3.3.2 1ms Timer Interrupt

In the program, functions such as speed regulation, fault protection detection, BUS current sampling and BUS voltage sampling are all called in 1ms interrupt, with the following functions involved:

Speed_response();	// loop control function
TargetSpeed_Colletion ();	// Button based speed regulation function
StarRampDealwith();	// ATO ramping control during motor startup
LEDControl ();	// Rotation speed gear assignment, thermal gear assignment, lamp display
Fault_Detection ();	// fault detection

3.3.3 FOC Interrupt

FOC interrupt, namely carrier interrupt, is majorly to handle relatively fast-timing programs such as calling a divider.

3.3.4 CMP3 Interrupt

Comparator 3 interrupt is majorly to handle hardware overcurrent protection. Please refer to Section 5.2.1 for more details.

3.3.5 External Interrupt

The external interrupt is used to collect the zero-crossing signals. The reception of the AC zero-crossing signal triggers the external interrupt. The shutdown and opening of the GPIO port can be controlled through the TIM3 delay for heat control purpose.

3.3.6 Timer3 Interrupt

When the control method is chopping, Timer3 interrupt is mainly used as a delay to open the thyristor count after zero-crossing is detected. After zero-crossing triggers an external interrupt, Timer3 is turned on to start counting and delay turning on the heating wire. The delay time can be configured for heat control purpose.

4 Debugging Steps

4.1 Motor Parameter Configuration

4.1.1 Motor Parameters

1. The number of motor pole pairs: Pole_Pairs;
2. Motor phase resistance RS, phase inductance LD and LQ, and BEMF constant Ke;
3. Motor speed base MOTOR_SPEED_BASE = 2* rated motor speed.

4.1.2 Motor Parameter Measurement Method

1. The number of pole pairs Pole_Pair: the parameter value is given in design;
2. Phase resistance Rs: the 2-phase line resistance RL of a motor is measured through a multimeter or LCR; phase resistance Rs = RL/2;
3. Phase inductance Ls: the 2-phase line inductance LL at 1KHz frequency is measured through LCR; phase inductance Ls = LL/2; LD = LQ = Ls;
4. BEMF constant Ke: connect an oscilloscope probe to one phase of a motor, and connect ground to one of the other two terminals of the motor; rotate the load, and measure the BEMF waveform. Take a sine wave in the middle and measure the peak-to-peak Vpp and frequency f. The calculation is as follows:

$$\mathrm{Ke} = 1000 * \mathrm{P} * \frac{\mathrm{Vpp}}{2 * 1.732 * 60 * \mathrm{f}}$$

Where, P is the number of pole pairs of the motor.

As an example, the measured BEMF waveform is as follows:

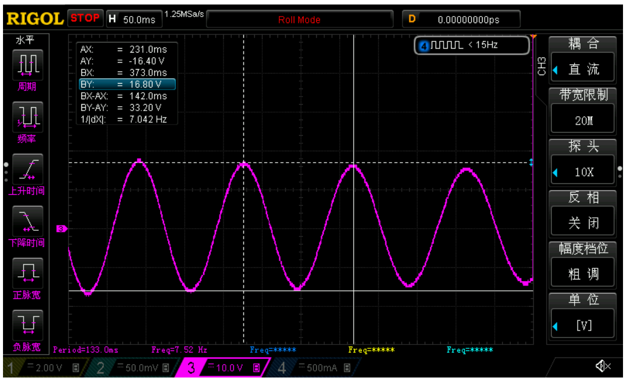


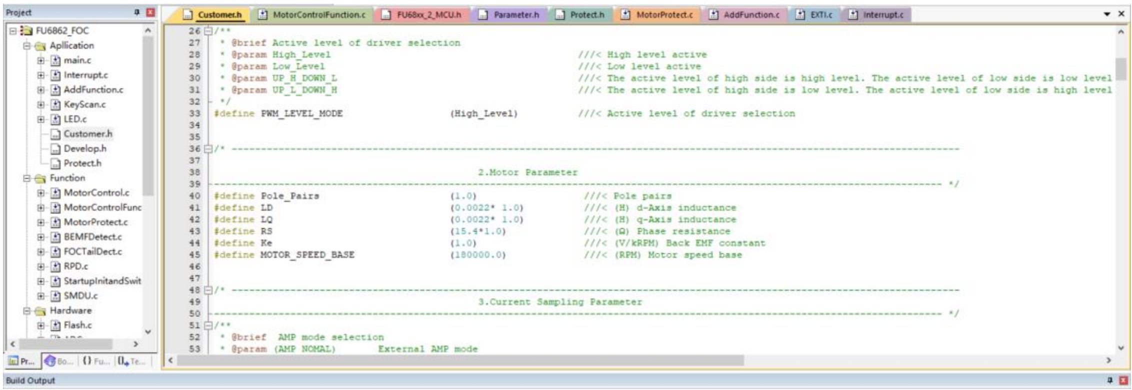
Figure 4-1 BEMF Waveform

The measured peak-to-peak Vpp is 33.2V, the frequency f is 7.042Hz, and the number of pole pairs P is 4, then:

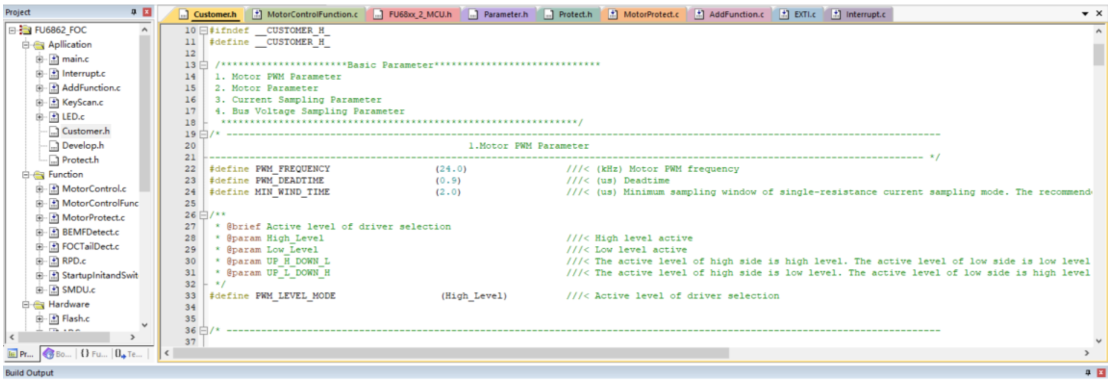
$$\mathrm{BEMF\ Ke} = 1000 * 4 * \frac{33.2}{2 * 1.732 * 60 * 7.042} = 90.73$$

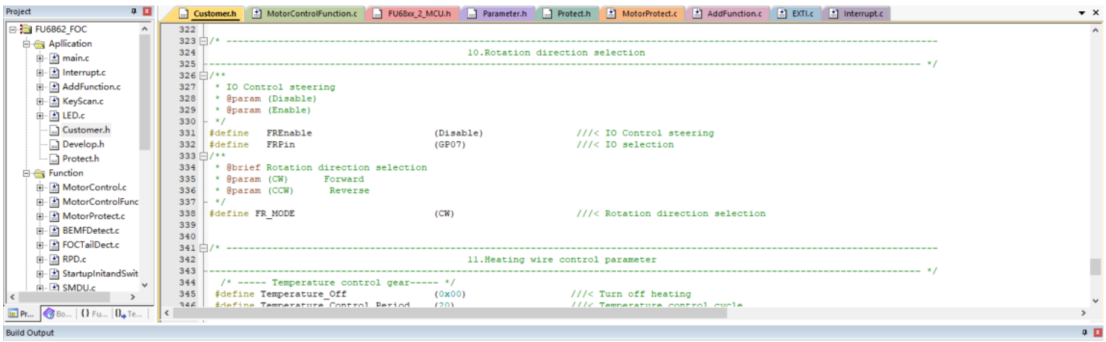
5. Speed base MOTOR_SPEED_BASE: In general, speed base is set as about 2 times maximum motor speed. As this value affects startup performance and so on, it should be determined beforehand and better unchanged later.

4.1.3 Corresponding Program Code



4.2 Chip Internal Parameter Configuration





Notes:

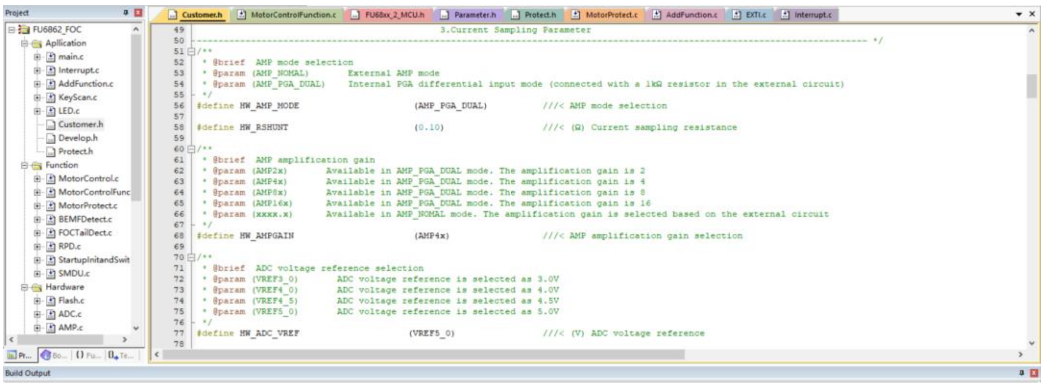
1. In general, carrier frequency needs to be set as about 10 times maximum electrical cycle. As carrier frequency affects startup, MOS temperature rising and so on, users need to select a proper carrier frequency before debugging. The blower generally has a high rotational speed, so the default 24K debugging can be used first;
2. Dead zone value is set according to actual MOS on/off speed to ensure no risk of shoot-through;
3. Minimum sampling window should be greater than 2 times dead zone and less than 1/16 of carrier cycle, i.e., 1000/16/PWM_FREQUENCY > MIN_WIND_TIME > 2*PWM_DEADTIME;
4. Forward and reverse rotation settings. According to the actual wiring settings, there will be high-frequency noise when the blower motor is reversed, and the air output is much smaller than that of forward rotation. If the motor is reversed, just reverse this bit.

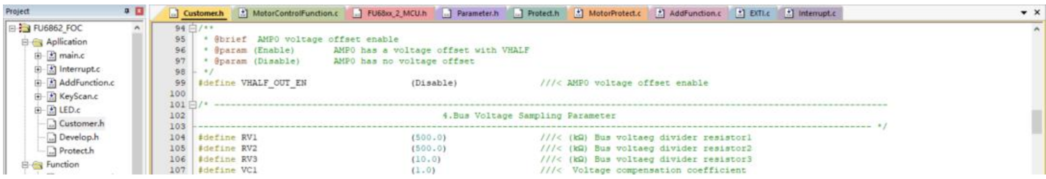
4.3 Hardware Parameter Configuration

1. Figure out BUS voltage divider ratio, sampling resistance value, and magnification according to the voltage and power ranges of a motor.
2. Resistance and magnification selection rules:
   1) BUS voltage divider resistance:
      - Voltage divider ratio should not be too small: In general, suggest maximum sampling voltage is 0.8*VREF. E.g., for a motor with maximum voltage being 30V and ADC reference VREF being 4.5V, suggest the voltage divider ratio is no less than 30/0.8/4.5 = 8.33; If the voltage divider ratio is too small a value like 5, when the voltage is 30V, the voltage at the AD port is 6V after voltage division, then it overflows.
      - Voltage divider ratio should not be too large: If so, the AD sampling voltage accuracy is insufficient. E.g., If voltage divider ratio is 40, when maximum voltage is 30V, the voltage at AD port is 30V/40V = 0.75V; when maximum voltage is 28V, the voltage at AD port is 0.7V. Thus, the accuracy is quite low. Moreover, the AD still has a margin of 4.5 - 0.75 = 3.75V.
   2) Sampling resistance and magnification:
      Maximum sampling current = VREF/HW_RSHUNT/HW_AMPGAIN; it should be noted that maximum sampling current is not the current displayed on power supply (i.e., the current after filtering), but the current through a sampling resistor.
      - Sampling resistance should not be too large: If so, it is easy to cause sampling overflow or resistor power out of range; sampling resistors of 2512 packaging commonly support 1W or 2W power; sampling resistors of 1206 packaging commonly support 1/4W; users should make sure the power I^2R through a sampling resistor does not exceed the corresponding power value.
      - Sampling resistance should not be too small: If so, the accuracy is insufficient.
      - Magnification is adjusted according to sampling resistance. Determine sampling resistance first, then adjust magnification.

HW_RSHUNT represents sampling resistance and HW_AMPGAIN represents magnification.

3. Fill the values of BUS voltage divider ratio, sampling resistance and magnification into the program code (in the Customer.h file) accordingly.





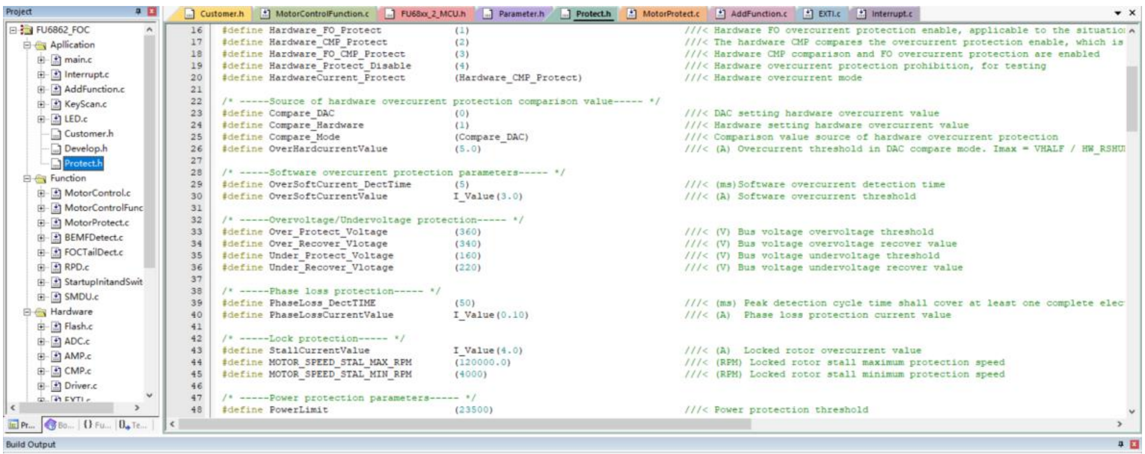
Where,

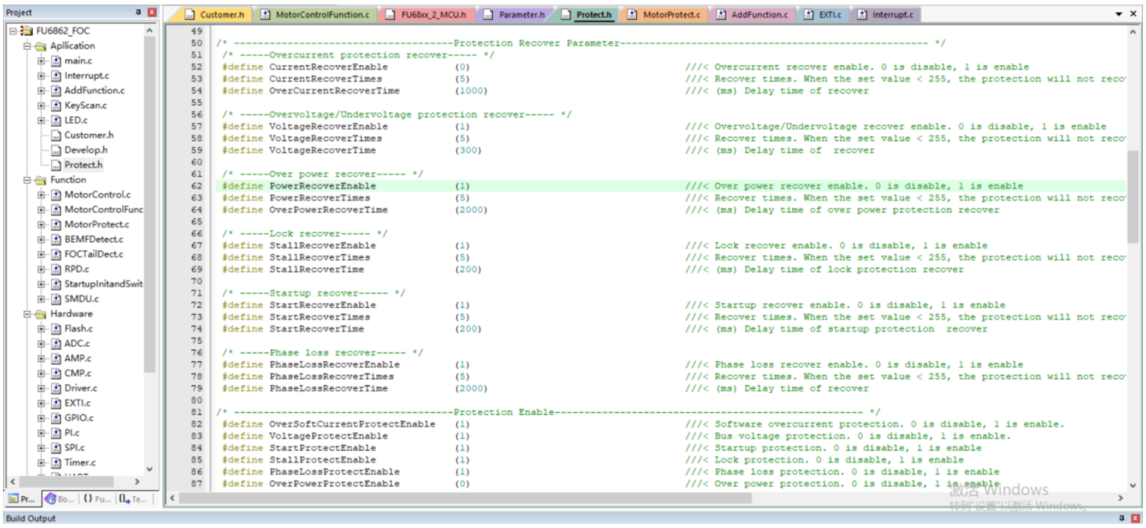
1) BUS voltage divider ratio = (RV1 + RV2 + RV3)/RV3;

2) VC1 is voltage compensation coefficient. It is used in startup stage only. Just leave it unchanged so far.

4.4 Protection Parameter Configuration

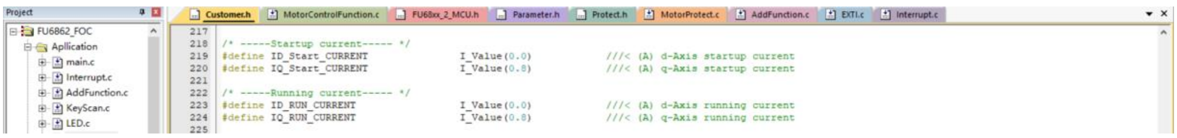
1. Current protection settings:
   - Hardware overcurrent: Set hardware overcurrent protection value according to the maximum current value of a power device. In general, set hardware overcurrent protection value OverHardcurrentValue larger than maximum BUS current, and less than the maximum current value of the power device.
   - Software overcurrent: In general, set OverSoftCurrentValue a little smaller than hardware overcurrent. Software overcurrent is triggered by software, and protection time is less than that of hardware overcurrent.
2. Set overvoltage/undervoltage protection and protection recovery parameters. Please refer to Section 5.2.2 for details;
3. Turn off all protections except the above to prevent false triggering during startup. Apply other protections later when they are required. As for overcurrent protection, it is always on with no enable bit.
4. Fill the parameters into the program code accordingly (in the Protect.h file).





4.5 Startup Parameter Configuration

Users can take all startup default settings first then adjust the values when encountering startup problems or difficulties. Please refer to Section 5.1 for parameter adjustment details to settle down common startup issues.



1. Startup current: In general, ID_Start_CURRENT is fixed to 0 and IQ_Start_CURRENT is set according to actual motor settings;

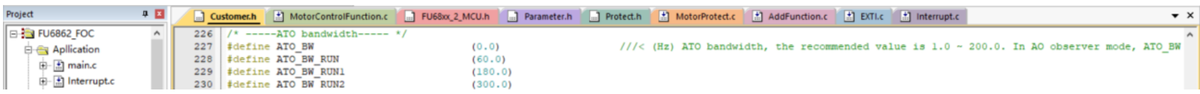
Notes:

IQ_Start_CURRENT should not be too small. If so, the starting torque gets too small to start the motor normally.

IQ_Start_CURRENT should not be too large. If so, overshoot in startup is encountered and startup noise is introduced.

2. Switching current: IQ_RUN_CURRENT determines transient current. By observing actual phase current upon IO port reverse, users can figure out whether current is smooth at loop switching moments. Tune IQ_RUN_CURRENT accordingly if required.

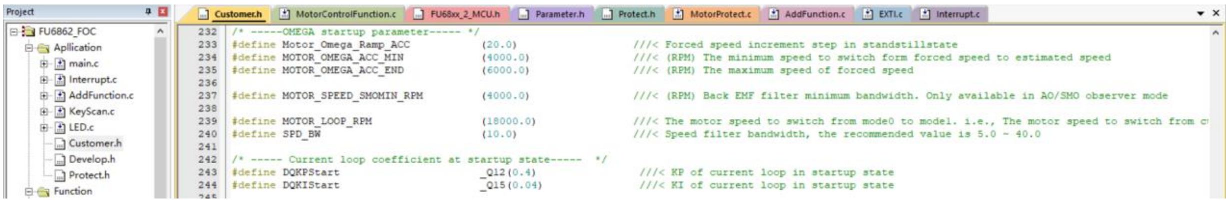
3. Startup ATO: Since inaccuracy output of FOC estimator in the case of lower speed, it is necessary to set ATO_BW (speed bandwidth filtering value) to limit the maximum speed output of FOC estimator;



Notes:

For high-speed air blowers, the impact of the first three ATO starts is relatively obvious, and needs to be adjusted according to the actual situation. Because the AO observer is turned on, the first ATO_ BW does not need to be too large.

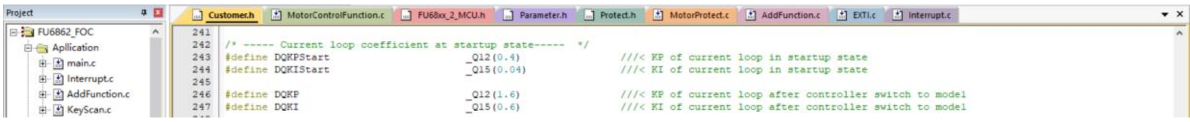
4. Omega startup settings affect startup current frequency, namely motor startup acceleration;



Notes:

1) Motor_Omega_Ramp_ACC reference value range is from 10 to 50;
2) MOTOR_OMEGA_ACC_MIN reference value range is from 2000 to 10000;
3) MOTOR_OMEGA_ACC_END reference value range is from 3000 to 16000;
4) MOTOR_LOOP_RPM shall be bigger than MOTOR_OMEGA_ACC_END, and reference value range is from 6000 to 16000;
5) MOTOR_SPEED_SMOMIN_RPM reference value range is from 3000 to 8000.

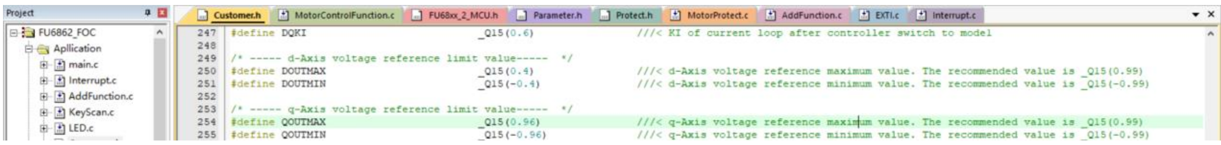
5. Current Loop PI: The current loop PI is divided into the starting current loop PI and the running current loop PI;



Notes:

1) The starting current loop PI affects the starting of the motor.
2) The running current loop PI affects the stability of the current and its efficiency as well. Phase current will oscillate when the PI is too small.
3) The recommended range of DQKP is between 0.4 to 2.0;
4) The recommended range of DQKI is between 0.05 to 0.6.

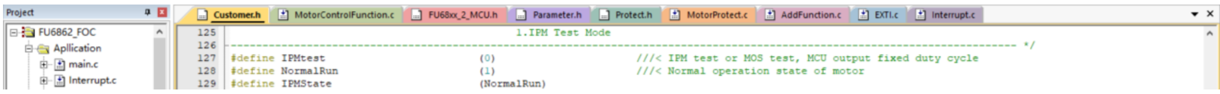
6. Maximum output limit of DQ axes: D axis affects the magnetic flux of the motor, while Q axis affects the torque of the motor.



Notes:

1) FOC__UQ feedbacks whether the output of the motor is saturated;
2) The more positive the FOC__UD is, the more advanced the angle is. The motor angle can be advanced by increasing the compensation angle (FOC_THECOMP). The maximum rotating speed can be increased at this time. FOC__UD is a positive value.
3) Excessive advanced angles can lead to current overshoot during shutdown. This can be handled either by low-voltage warning shutdown or fast undervoltage protection method.
4) Excessive advanced angles can lead to poor efficiency. Under the same power condition, the phase current amplitude will be larger. Hence, it is necessary to set feasible compensation angle.

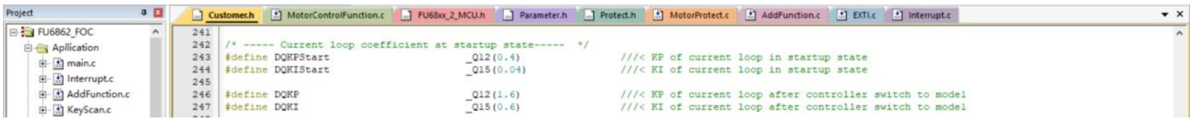
4.6 Hardware Driver Circuit Detection



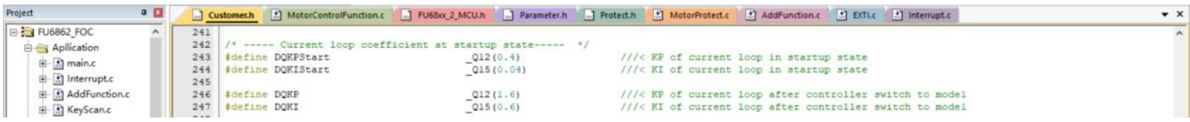
Select IPMtest for IPMState mode, and the motor will run the pre-positioning state at this moment. The three UVW phases will have a regular PWM waveform output, and the hardware driver circuit is seen normal. If not, check for any hardware issue.

4.7 Current Loop Debugging

1. When the current loop starts, set smaller KPKI value. Increase the KPKI value when it is into the closed loop.



2. During operation, the busbar voltage will fluctuate with the magnitude of the load. The phase current will fluctuate and it is necessary to increase the KPKI value during operation. KP is generally set to 1.5, and KI is generally set to 0.4. (The value is for reference only and can be set based on the real situation.)



Common problems and solutions:

1) During operation, overcurrent protection is triggered.

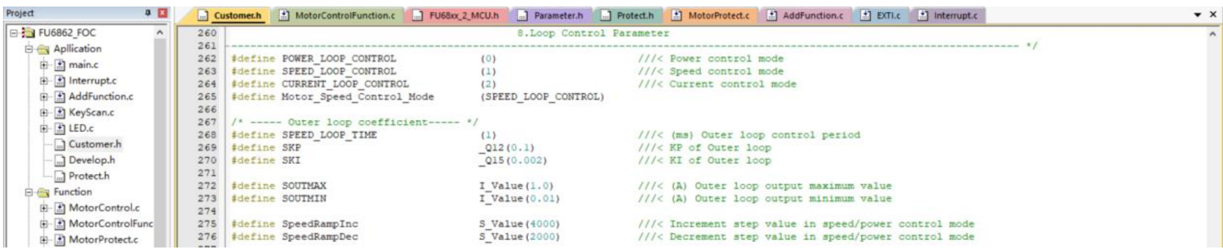
   Solution: Check whether the phase current waveform is abnormal. Check whether the set value is relatively small that triggers overcurrent protection normally. If no abnormal problems are found, check whether there is any hardware wiring issue.

2) There is jitter in the phase current waveform.

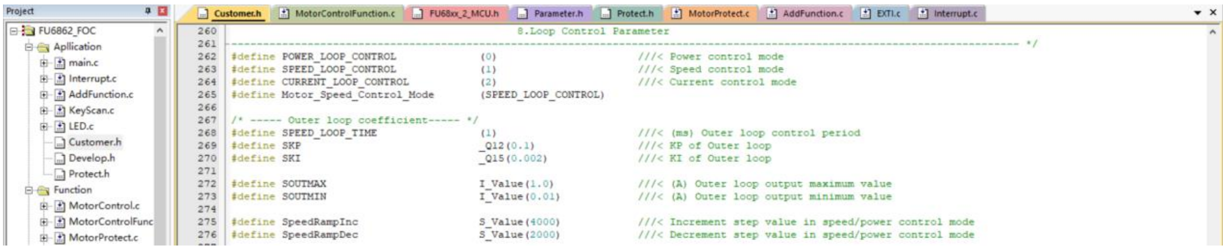
   Solution: Adjust the current loop PI value (that is, DQKP, DQKI). The current loop PI and current sampling have greater influence on the stability of the current waveform.

4.8 Speed Loop Debugging

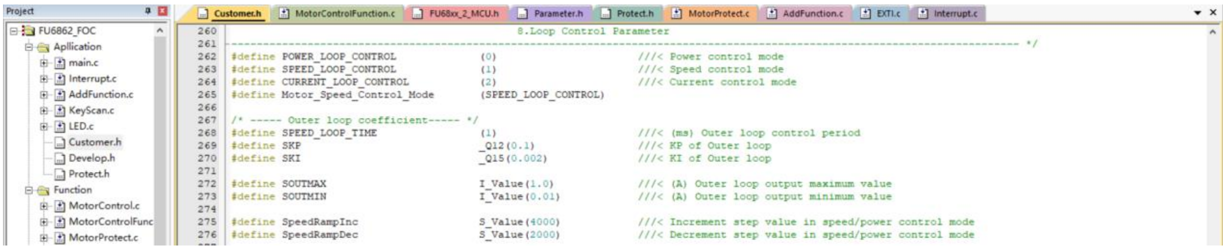
1. Generally, high-speed hair dryer adopts constant speed loop. Hence, select speed loop for loop control.



2. Set the maximum value of the speed loop limit SOUTMAX. If the set value is too small, it will not speed up. The limit should be set based on the real situation;



3. Adjust the speed loop SKP, SKI, speed loop ramp increment and speed adjustment cycle to ensure the stability of the speed loop and fast start-up response without overshoot.



4.9 Add Button Feature

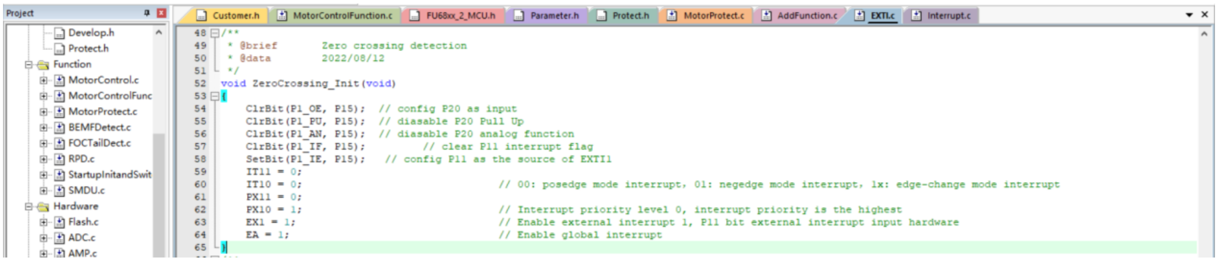
1. Generally, hair dryer speed can be adjusted by the button;
   1) In the hair dryer program, speed adjustment is only available through button speed adjustment method. When the board is under normal condition, the motor will operate after it is powered on.

   2) The program logic can be modified according to customer's function requirements. The current button logic is as follows: Wind speed: 1->2->3->2->1...... and repeats in this way; wind temperature: 0->1->2->3->2->1->0->1...... and repeats in this way. (0 represents cold air); One-button cold air: Press and hold the button for cool air all the time. Release the button to restore to the original air temperature.

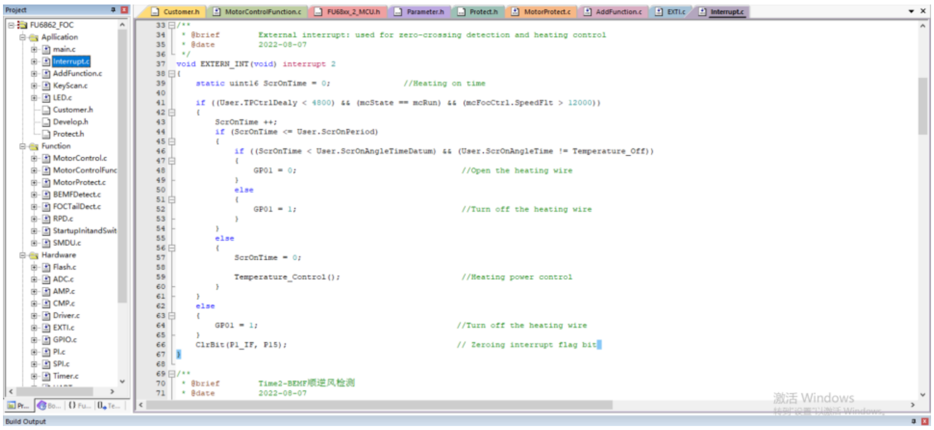
4.10 Add Heating Function

1. When setting PORT0.0-PORT0.6 as digital I/O input, or enabling comparator CMP4, users can set EX0 = 1 to make it as external 0 (INT0). When setting PORT1.0-1.7, PORT2.0-2.7 as digital I/O input, users can set EX1 = 1 and correspond to P1IE/P2IE to share external interrupt 1 (INT1). Set external interrupt 0 as enabling bit EX0, interrupt flag bit IF0, interrupt level trigger control IT0. The source of external interrupt 0 is specified by EXT0CFG in the register LVSR. These sources can be any one from PORT0.0 to PORT0.6 input and comparator CMP4 output. All external interrupt 0 interrupt sources share one interrupt entry and one interrupt flag bit. Set external interrupt 1 as enabling bit EX1, the interrupt enabler of 16 PINs is controlled by registers P1IE and P2IE. The corresponding interrupt flag bits are P1IF and P2IF, and the interrupt level trigger control is IT1;

2. External interrupt configuration can be achieved by setting zero-crossing trigger I/O port according to hardware design;



After an interrupt is triggered, the corresponding interrupt flag shall be cleared.



4.11 Reliability Test

4.11.1 Reliability of Function

After adding all functions, test the functions in the customer requirement list and make sure no abnormality occurs.

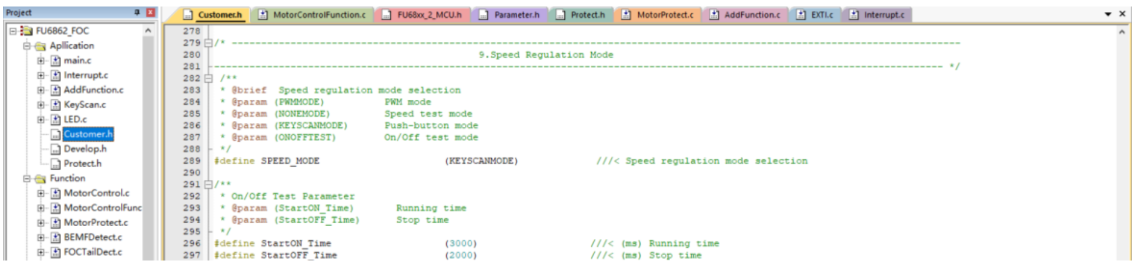
4.11.2 Reliability of Protection

After adding protection functions, verify each protection can be triggered normally and no protection is triggered falsely while motor is in normal operation. E.g., upon improper motor locked protection settings, it may cause the motor to report locked protection falsely during normal operation; or it fails to report motor locked protection upon actual motor locked occurrence.

4.11.3 Stability of Startup

When finishing functional debugging, users can carry on startup reliability test. Manual test is performed first. Then perform aging test after manual test is passed.

1. Select ONOFFTEST mode for SPEED_MODE.
2. Set the running time StartON_Time and stop time StartOFF_Time according to the actual situation;
3. Adjust Motor_Speed_High value to modify the starting and stop speed;
4. Block the motor using a tool then power up it. Check if a motor-locked protection is triggered normally, and no motor restart after the protection. It is to verify no restart occurs upon protection being triggered during motor startup or stop;
5. Power on again and perform aging test. Finally, judge startup failure by checking whether the motor remains stopped. Upon startup failure, the motor remains stopped with no more restart. In general, more than 3000 (the more the better if time allows) consecutively successful startup operations can secure startup reliability.



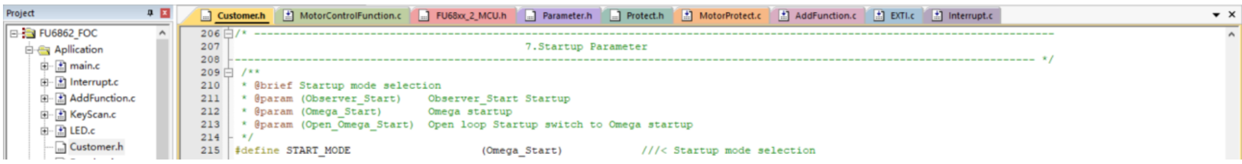
5 Function Introduction

When users get the original program, configure motor parameters and hardware parameters, then send start signal to a target motor, it usually can start normally. If encountering abnormal startup, users should check and settle down hardware problems first, then adjust startup settings.

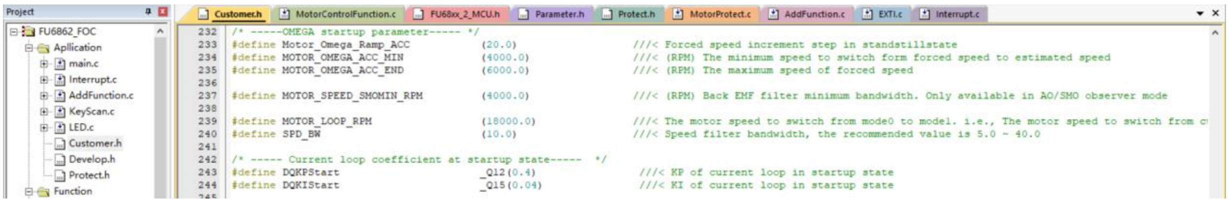
5.1 Startup Debugging

5.1.1 Omega Startup

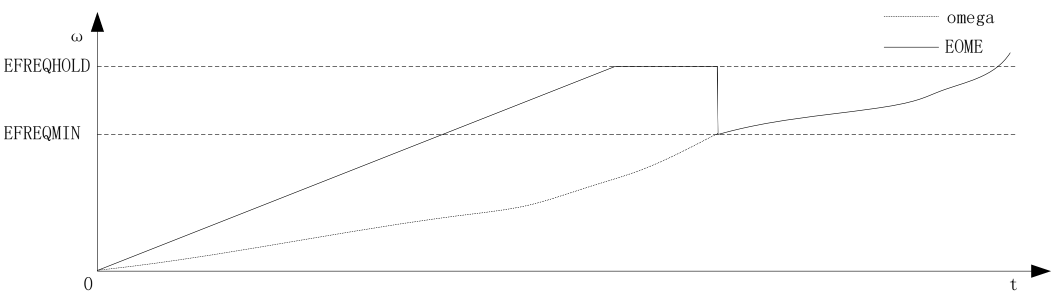
Select Omega to turn on the hair dryer, and this method is set by default by the program.



When the estimated speed OMEGA of FOC estimator is less than the minimum value FOC_EFREQMIN (corresponding to the parameter MOTOR_OMEGA_ACC_MIN) set by user, the forced speed starts from 0. It is added up with the incremental speed value FOC_EFREQACC (the parameter Motor_Omega_Ramp_ACC) and meanwhile it is limited by the maximum value FOC_EFREQACC (corresponding to the parameter Motor_Omega_Ramp_ACC) in each operation cycle. The forced speed is output as the final speed EOME, which is applied by angle calculation module to calculate estimator angle ETHETA; when the estimated speed OMEGA of FOC estimator is greater than or equal to EFREQMIN, the estimated speed OMEGA is output as the final speed EOME.



Startup procedure is shown in the figure below.



5.1.2 Common Issues & Solutions in Starting

Common Issues	Solutions
The motor rotates at an instant then stops with input current all the time.	1. A possible reason is startup current being too small, which cannot drive the motor to the next commutation. The solution is to increase IQ_Start_CURRENT; 2. Another possible reason is the output speed of FOC estimator being too small, which cannot drive the motor to the next commutation. If Item 1 is ruled out, then increase ATO_BW, ATO_BW_RUN, ATO_BW_RUN1 and ATO_BW_RUN2 sequentially; 3. If item 1 and item 2 are ruled out, check whether the hardware circuit of AMP0 encounters problems, which leads to inaccurate current sampling and the false estimation of FOC estimator; 4. It's also possible that the frequency of omega acceleration is too high. The solution is to reduce Motor_Omega_Ramp_ACC.
The motor rotates at an instant then stops and keeps jittering.	1. It's most probably due to ATO_BW being too large, which causes the output speed of FOC estimator being high, and makes the motor run out during startup. The solution is to reduce ATO_BW, ATO_BW_RUN, ATO_BW_RUN1 and ATO_BW_RUN2 sequentially; 2. Another possible reason is improper Omega startup settings .
The motor starts and rotates forward for a certain angle, then is stuck and locked at an instant, then rotates normally.	1. Users can estimate the time from startup to freezing, and then set ATO_BW accordingly. E.g., the motor starts and operates for 1s then freezes for at an instant then operates normally. The period 1s is corresponding to ATO_BW_RUN1 and ATO_BW_RUN2. The issue is caused by ATO_BW being relatively small, which limits motor speeding up. The solution is to increase ATO_BW. 2. Omega acceleration being too small can cause the stuck and locked as well. The solution is to increase Motor_Omega_Ramp_ACC.
The motor starts and rotates reversely, then when turning to rotate forward, it gets to jitter constantly.	1. It takes a long time for the motor to rotate reversely at an instant upon startup then rotate forward. At the time, ATO_BW is already increased to a relatively large value. The solution is to reduce ATO_BW; 2. Another possible reason is the frequency of omega acceleration being too high. The solution is to adjust Motor_Omega_Ramp_ACC.

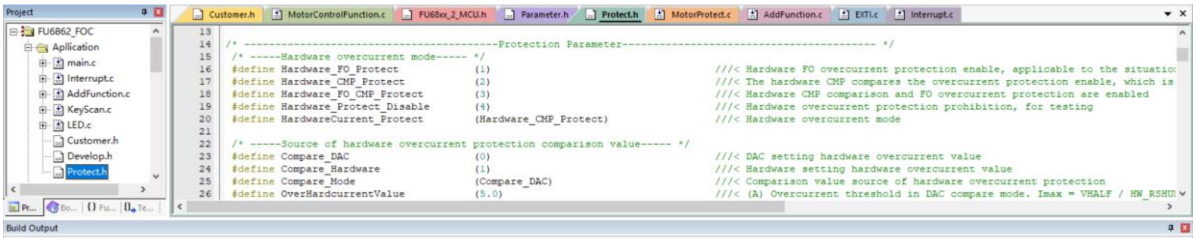
5.2 Introduction to Protection Functions

Protection values vary in different projects, motors and boards. Parameters for various protection functions need to be adjusted according to real projects. Upon the occurrence of expected motor locked protection or fault protection not being reported, or protection being falsely triggered in normal operation, it indicates improper setting of protection parameters, users need to adjust them accordingly.

5.2.1 Overcurrent Protection

1. Hardware overcurrent protection;

Hardware overcurrent protection is fulfilled through comparator 3 in the chip. The detection method is: The BUS current flows through the sampling resistor, and a voltage is formed on the sampling resistor; the voltage is amplified by the operational amplifier and sent to the positive input of the comparator. A reference voltage generated by a DAC or by an external voltage divider (DAC is used currently) is input to the negative input of the comparator. When the BUS current is increasing to a certain value where the voltage of the comparator's positive input is higher than that of the negative input, a comparator interrupt in MCU is triggered. Upon this interrupt, MCU turns off the MOE automatically (whether it is automatic or not is configurable and it is automatic by default) to fulfill overcurrent protection. For hardware overcurrent protection, users only need to adjust OverHardcurrentValue.

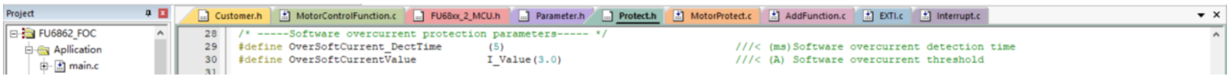


```
/* ---------------------------------------Protection Parameter--------------------------------------- */
/* -----Hardware overcurrent mode----- */
#define Hardware_FO_Protect            (1)                    ///< Hardware FO overcurrent protection enable, applicable to the situatio
#define Hardware_CMP_Protect           (2)                    ///< The hardware CMP compares the overcurrent protection enable, which is
#define Hardware_FO_CMP_Protect        (3)                    ///< Hardware CMP comparison and FO overcurrent protection are enabled
#define Hardware_Protect_Disable       (4)                    ///< Hardware overcurrent protection prohibition, for testing
#define HardwareCurrent_Protect        (Hardware_CMP_Protect) ///< Hardware overcurrent mode

/* -----Source of hardware overcurrent protection comparison value----- */
#define Compare_DAC                    (0)                    ///< DAC setting hardware overcurrent value
#define Compare_Hardware               (1)                    ///< Hardware setting hardware overcurrent value
#define Compare_Mode                   (Compare_DAC)          ///< Comparison value source of hardware overcurrent protection
#define OverHardcurrentValue           (5.0)                  ///< (A) Overcurrent threshold in DAC compare mode. Imax = VHALF / HW_RSHU
```

2. Software overcurrent protection

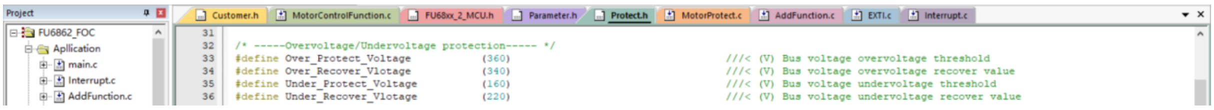
The program obtains the three-phase maximum current value. When the maximum current value exceeds the set software overcurrent protection value OverSoftCurrentValue, it counts once; When the count exceeds 3 times (modifiable),protection is triggered.



```
/* -----Software overcurrent protection parameters----- */
#define OverSoftCurrent_DectTime       (5)                    ///< (ms)Software overcurrent detection time
#define OverSoftCurrentValue           I_Value(3.0)           ///< (A) Software overcurrent threshold
```

5.2.2 Voltage Protection

The program detects the voltage through the AD2 port and reports overvoltage protection when the detected voltage exceeds a set value. Then when the voltage becomes lower than the overvoltage recovery value again, the overvoltage protection fault is cleared. When the voltage is lower than a set undervoltage value, an undervoltage protection is reported. Then when the voltage becomes higher than the undervoltage recovery value again, the undervoltage protection fault is cleared.



```
/* -----Overvoltage/Undervoltage protection----- */
#define Over_Protect_Voltage           (360)                  ///< (V) Bus voltage overvoltage threshold
#define Over_Recover_Vlotage           (340)                  ///< (V) Bus voltage overvoltage recover value
#define Under_Protect_Voltage          (160)                  ///< (V) Bus voltage undervoltage threshold
#define Under_Recover_Vlotage          (220)                  ///< (V) Bus voltage undervoltage recover value
```

5.2.3 Phase Loss Protection

3-phase current is asymmetrical in case of motor phase loss. Based on this, phase loss protection function detects the maximum values of 3-phase current within a certain period, and judges whether the maximum values of the 3-phase current are asymmetric.

Specific program implementation method: If it is detected that the maximum current of one phase is greater than 3 times the maximum current of the other phase (which can be modified according to the actual situation), and the maximum current of this phase is greater than the set PhaseLossCurrentValue value, it is determined as a phase loss.

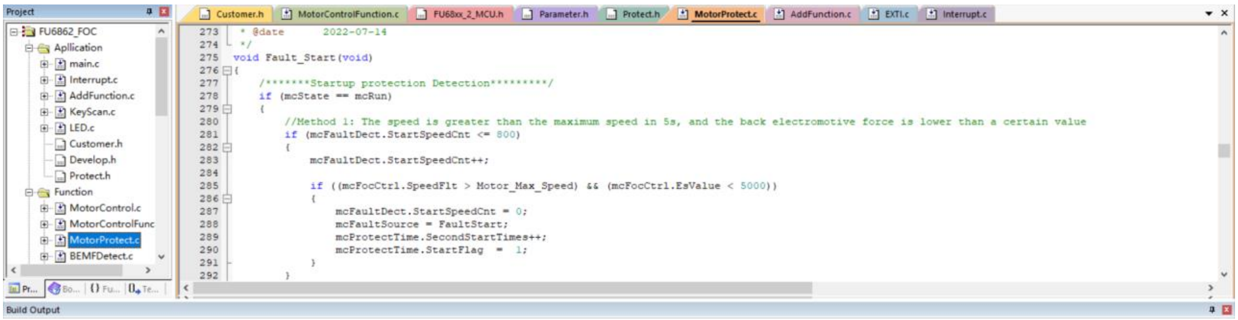
Notes:

In some cases, when a phase is missing, the signal of the missing phase will have burrs, which may cause the maximum current value collected to be about the same as those of the other two phases, which may not be detected by the above method. The phase loss detection method discussed above may fail in the case. Solution: Phase loss can be judged by comparing the accumulated current value within a certain period through integration method.

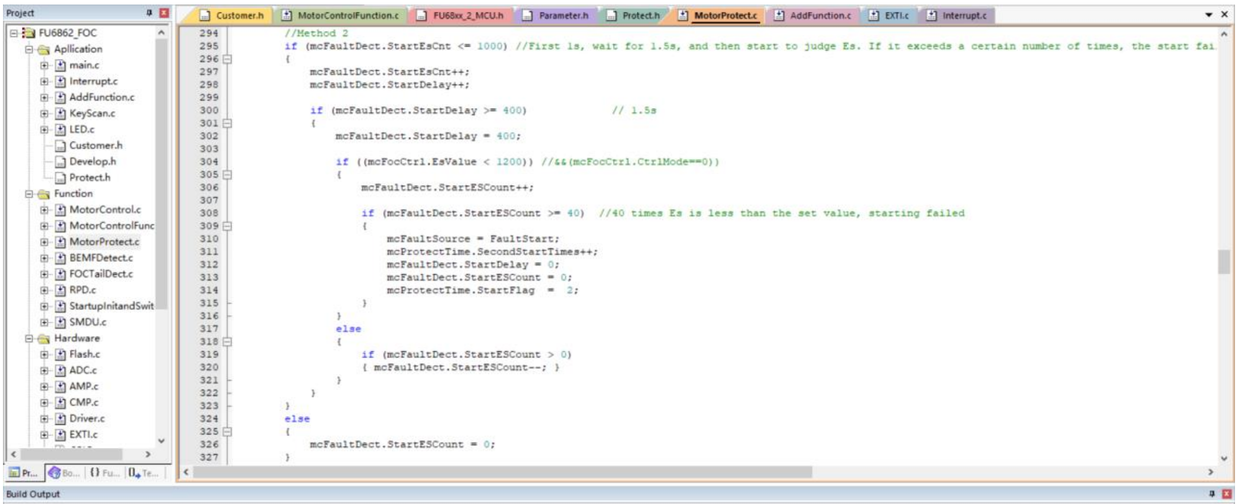
5.2.4 Startup Protection

There are three ways to detect startup protections.

1. The FOC_ESQU (the square of BEMF) calculated by the detection estimator can be used for detection. When under normal conditions, the higher the motor speed, the greater the FOC_ESQU is. When the motor is stalled and pull-out, the estimated speed can be very high; the FOC_ESQU can be very low. Therefore, the detection method can be changed; A specific way to implement the program can be: check whether the value of FOC_ESQU is still smaller than the set value Stall_DectEsValue1; or when the estimated speed is higher than the set value Motor_Max_Speed (this can be modified based on the actual situation), the value of FOC_ESQU is less than the set value 5000 (this can be modified according to the actual situation), then the system will regard it as a startup failure.

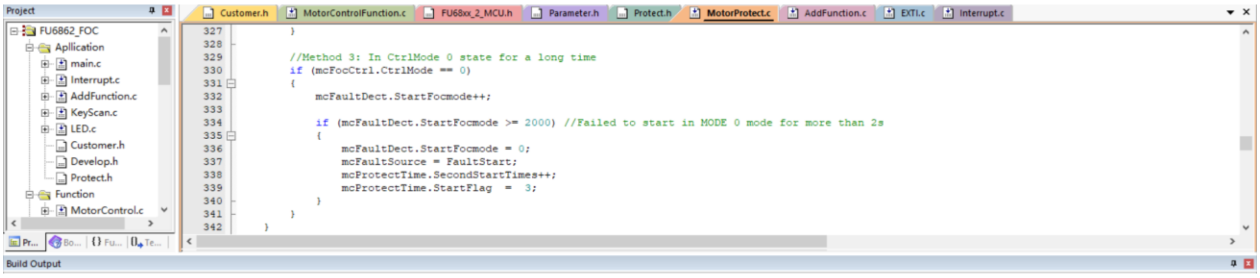


2. This can be checked by the calculated FOC_ESQU detection estimator. The estimated speed can be very high when it is stalled or pull-out. However, if the FOC_ESQU is very low, it will be considered as a stall or pull-out situation. Then the system will regard it as a startup failure.



3. When the motor starts up, if the estimated speed is greater than MOTOR_LOOP_RPM, the program will set its mode status from 0 to 1 to begin from a regular current and form a normal loop. In the meantime, its mode

status can be used to detect if a stall situation has occurred. If the mode is still at 0 after 2000ms (this can be modified according to the actual situation) after startup, then the system will regard it as a startup failure.

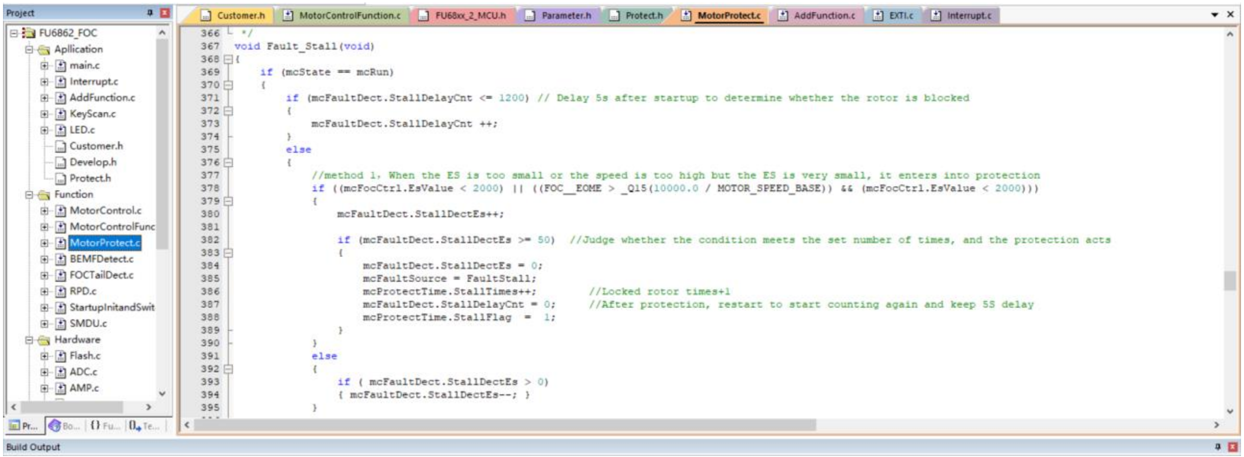


5.2.5 Stall Protection

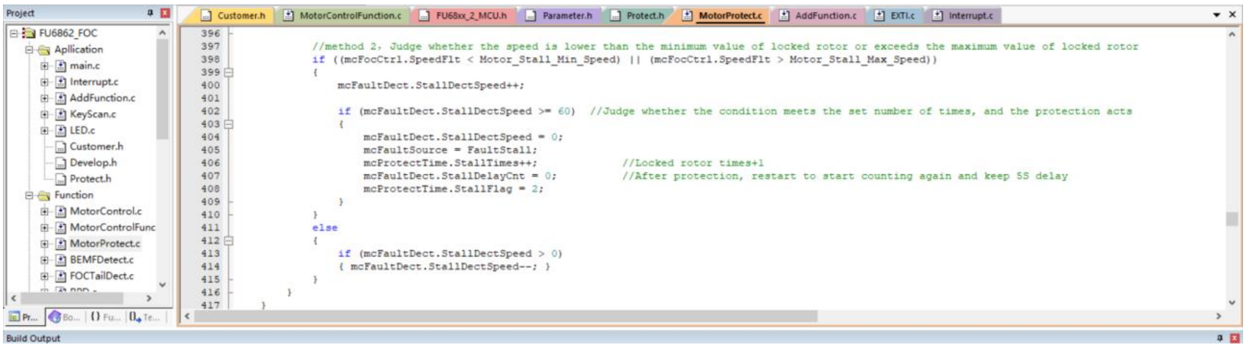
There are three ways to detect stall protections.

1. The FOC_ESQU (the square of BEMF) calculated by the detection estimator can be used for detection. When under normal conditions, the higher the motor speed, the greater the FOC_ESQU is. When the motor is stalled and pull-out, the estimated speed can be very high; the FOC_ESQU can be very low. Therefore, the detection method can be used.

A specific way to implement the program can be: When the power-on delay is 1200ms and the detected value of FOC_ESQU is less than the set value of 2000. In the mean time, when the estimated speed is higher than the set value 1200 rpm, or when the value of FOC_ESQU is smaller than the set value 2000, the count will be 50ms. When the count value is reached, the system will regard it as stall.

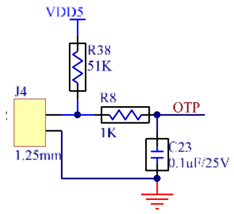


2. It can be determined by detecting the estimated speed. When the estimated speed exceeds the set speed Motor_Stall_Max_Speed or when the speed is lower than the set speed Motor_Stall_Min_Speed, the system will regard it as stall.



5.2.6 Over Temperature Protection

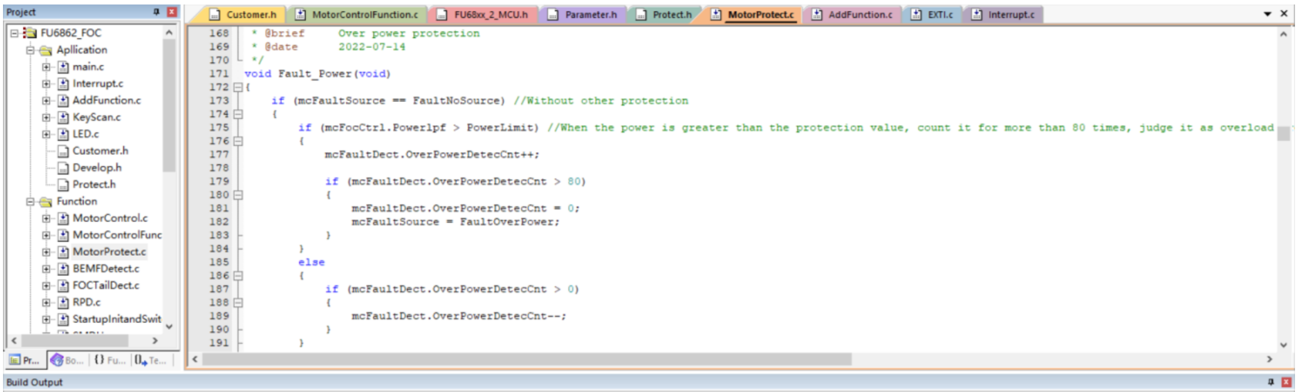
The common circuit diagram for overtemperature protection is shown below. The voltage dividing resistor usually uses an NTC resistor, which is placed at the air outlet. As the temperature rises, the resistance gradually decreases. There will be a corresponding resistance value at each temperature. The OTP is connected to an AD port of the chip. The program detects the voltage at the AD port. When the voltage is less than the voltage at the set temperature, it indicates that the NTC resistance temperature exceeds the set value and triggers protection.



This protection feature is not included in the standard program and can be added based on actual needs.

5.2.7 Power Protection

When the collected power filter value is greater than the power protection setting value PowerLimit, count for 80ms. When the count value is reached, the system will regard it as over power.



5.2.8 Other Protections

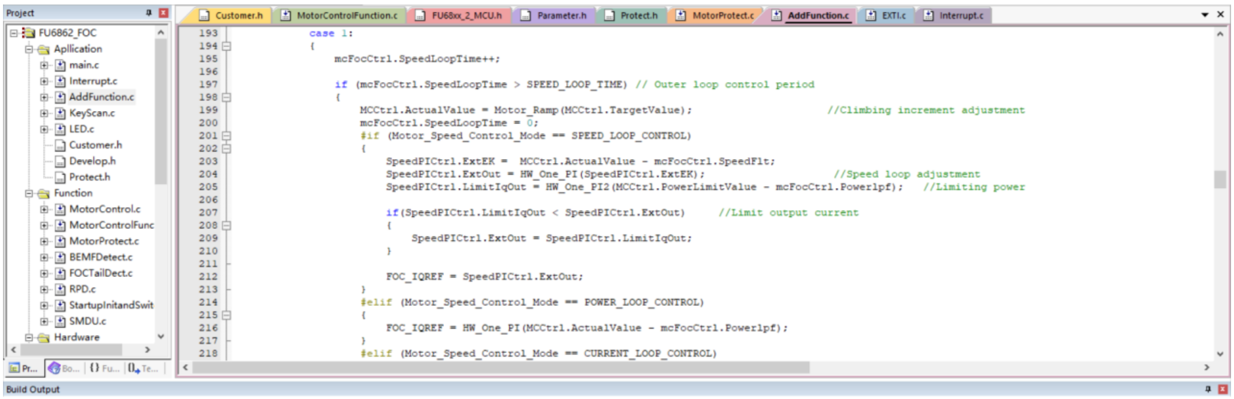
Users can add other protections per customer needs.

6 Other Common Function Debugging

6.1 Power Limiting Function

If the air inlet of a hair dryer is blocked when it is in constant speed control mode, the motor will run to a higher speed when the load becomes smaller. This can bring damage to the bearing. Moreover, poor heat dissipation of the motor can also bring damage to the motor. Therefore, the power limiting function is required to limit the power.

Power limiting methods: Dual power limiting methods for PI, including hardware PI1 for speed closed loop and hardware PI2 for power limit:



7 Key Issues and Solutions

Constant Power Debugging	
Common Issues	**Common Issues**
Startup problems are unsettled for a long time	Users is blocked by startup issue debugging. When software issues are ruled out, users should check hardware problems such as sampling and layout.
Startup aging test found startup failure	Parameters such as Omega startup parameters, ATO parameters, current loop startup KPKI
Motor speed response is slow or too fast	1. Debug the SKP and SKI of outer loop; 2. Adjust SPEED_LOOP_TIME; 3. If only the acceleration and deceleration are relatively slow, tune the incremental value of acceleration and deceleration.
Regular oscillation of phase current during operation	Due to bus voltage fluctuation, current loops KP and KI can be increased.
The rotational speed or power cannot meet the customer's requirements	1. When current is sine waveform, observe whether FOC__UQ is saturated. 2. If FOC_UQ is saturated and FOC__UD is relatively large, adjust compensation angle FOC_THECOMP (try both positive angle and negative angle); see whether customer needs are met. 3. FOC_ The UQ is not saturated. You can check whether the current loop limit QOUTMAX and outer loop limit SOUTMAX are not set sufficiently large, and check whether the maximum power limit POWERLPLIMIT is too small.
The motor is prone to high current after running at a high rotational speed	1. Adjust the compensation angle FOC_ THECOMP. 2. Check whether there is interference in hardware sampling. 3. Check whether the motor parameters are filled correctly.
Sinusoidal distortion of current waveform	1. Check whether the sampling bias reference is normal; 2. Modify the PI of the current loop, namely DQKP, DQKI; 3. Modify the carrier frequency (note that the modification will affect startup and operation).
Uncontrolled heating or large power pulsation	1. Check whether there is interference in zero-crossing sampling and whether external interrupts are triggered according to the configuration. If there are burrs in the zero-crossing signal, external interrupts will be continuously triggered, resulting in uncontrolled heating. The common cause is large ground wire interference. 2. Check whether the signal driving the silicon controlled rectifier is correct. If there is interference, the drive signal will output abnormally, resulting in uncontrolled heating. The common cause is large ground wire interference.
Notes: In general, all the parameter adjustment affects performance of startup and run. Users need to re-test and double check after problems are resolved.	

8 Revision History

Rev.	Changes	Effective Date	Revised by
V1.1	First release，Translated from Chinese manual V1.0.01.	2023/03/22	Kelly

Copyright Notice

If there are any differences between the Chinese and the English contents, please take the Chinese version as the standard.

Fortior Technology (Shenzhen) Co., Ltd.

Room203, 2/F, Building No.11, Keji Central Road2,

Software Park, High-Tech Industrial Park, Shenzhen, P.R. China 518057

Tel: 0755-26867710

Fax: 0755-26867715

URL: http://www.fortiortech.com