

**FM/PSOC FAMILY**  
**32-BIT MICROCONTROLLER**  
**FM0+/PSOC4100S SERIES**

---

**DC FAN MOTOR CUSTOMER  
INTERFACE**

**APPLICATION NOTE**



## Revision History

Version	Date	Updated by	Modifications
0.1.0	2013-1-25	Larry Yang	First Draft
0.2.0	2013-9-29	Devin Zhang	Add the content and change the template
0.3.0	2014-11-21	Devin Zhang	Add new parameters
0.4.0	2016-11-29	Devin Zhang	Change new template
0.5.0	2017-11-15	Devin Zhang	Change cypress template and update information

This manual contains 40 pages.

Specifications are subject to change without notice. For further information please contact each office.

**All Rights Reserved.**

This document states the current technical specifications regarding the Spansion product(s) described herein. Each product described herein may be designated as Advance Information, Preliminary, or Full Production.

Copyright © Spansion All rights reserved.

## Contents

<b>REVISION HISTORY .....</b>	<b>2</b>
<b>CONTENTS .....</b>	<b>4</b>
<b>1 INTRODUCTION .....</b>	<b>7</b>
1.1 Purpose.....	7
1.2 Variable Naming Rule .....	7
1.3 System Files Structure .....	7
<b>2 SYSTEM BUILD.....</b>	<b>9</b>
2.1 Hardware Connect .....	9
2.2 System Files Structure .....	10
2.2.1 Configure Motor Parameters.....	12
2.2.2 Define PI Parameters .....	13
2.2.3 Define Start-up Parameters .....	13
2.2.4 Define Motor Protect Parameters.....	14
2.2.5 Define Motor Running Parameters.....	14
2.2.6 Define Hardware Parameters.....	14
2.3 Motor Start-up .....	15
2.3.1 Code Build.....	15
2.3.2 Download and Debug .....	16
2.3.3 Run Program and Motor .....	17
2.4 Motor Performance Debug .....	19
2.4.1 Start-up Debug .....	19
2.4.2 PI Parameters Debug .....	21
<b>3 VARIABLE DESCRIPTION.....</b>	<b>24</b>
3.1 Customer Interface Variable List .....	24
3.2 Motor1_pole_pairs.....	25
3.3 Motor1_Ld .....	25
3.4 Motor1_Lq .....	25
3.5 Motor1_Res .....	27
3.6 Motor1_Ke .....	27
3.7 Motor1_Kt .....	27
3.8 Motor1_Kj .....	27
3.9 Motor1_dki .....	27
3.10 Motor1_dkp .....	27
3.11 Motor1_qki .....	28
3.12 Motor1_qkp .....	28
3.13 Motor1_ski.....	28

3.14	Motor1_skp .....	28
3.15	Motor1_Speed_Inc_Acceleration_Hz .....	28
3.16	Motor1_Speed_Dec_Acceleration_Hz.....	28
3.17	Motor1_OpenLoop_Inc_CurrentAPS .....	28
3.18	Motor1_OpenLoop_CurrentA .....	28
3.19	Motor1_Orient_TimeS .....	29
3.20	Motor1_Force_Inc_Speed_Hz.....	29
3.21	Motor1_CloseLoop_Speed_Hz.....	29
3.22	Motor1_Initial_Speed_Pre_Close_Hz.....	29
3.23	Motor1_Detect_Direction_TimeS.....	29
3.24	Motor1_Inverse_Rotor_Brake_TimeS .....	29
3.25	Motor1_Pre_CloseLoop_TimeS .....	29
3.26	Motor1_Pre_CloseLoop_Current_MinA.....	29
3.27	Motor1_Pre_CloseLoop_Current_MaxA.....	29
3.28	Motor1_Inc_SpeedAcc_TimeS .....	30
3.29	Motor1_Inc_Speed_MAX_Hz .....	30
3.30	Motor1_PWM_Brake_Inc_TimeS .....	30
3.31	Motor1_Over_CurrentA .....	30
3.32	Motor1_max_dcvoltagEV.....	30
3.33	Motor1_min_dcvoltagEV.....	30
3.34	Motor1_Error_keep_timeS .....	30
3.35	Motor1_Field_Value .....	30
3.36	Motor1_Comp_MaxA .....	30
3.37	Motor1_is_MaxA .....	31
3.38	Motor1_dead_timeUS .....	31
3.39	Motor1_PWM_Carry_Frequency .....	31
3.40	Motor1_Current_Carry_Frequency .....	31
3.41	Motor1_speed_Max_RPM.....	31
3.42	Motor1_speed_Min_RPM.....	31
3.43	Motor1_Rotor_direction.....	31
3.44	Motor1_Running_Level .....	31
3.45	Motor1_Dynamic_PWM_Enable.....	31
3.46	Motor1_PWM_Table .....	32
<b>4</b>	<b>HARDWARE CONFIGURATION .....</b>	<b>33</b>
4.1	AD_OFFSET_NUM .....	33
4.2	AD_OFFSET_MAX_VALUE .....	33
4.3	Motor1_Control_mode.....	33
4.4	MOTOR1_PWM_POLAR_AVAILABLE .....	33

4.5	SYS_CLOK .....	33
4.6	MOTOR1_SHUNT_NUMBER .....	33
4.7	SPEED_FROM_OUT .....	33
4.8	MOTOR1_U_PIN .....	33
4.9	MOTOR1_V_PIN.....	34
4.10	MOTOR1_W_PIN.....	34
4.11	DC_V_PIN.....	34
4.12	MOTOR1_SPEED_PIN .....	34
4.13	ADC_Digit .....	34
4.14	ADC_REF .....	34
4.15	VDC_Amplifier_Multiple.....	34
4.16	Motor1_Current_Rs.....	35
4.17	Motor1_Current_Amplifier_Multiple .....	35
4.18	MOTOR1_BEMF_LPFK_MIN.....	36
4.19	MOTOR1_BEMF_LPFK_MAX.....	36
4.20	MOTOR1_BEMF_LPFK_MIN_HZ .....	36
4.21	MOTOR1_BEMF_LPFK_MAX_HZ .....	36
<b>5</b>	<b>SYSTEM ERROR CODE DEFINE.....</b>	<b>37</b>
<b>6</b>	<b>ADDITIONAL INFORMATION .....</b>	<b>38</b>
<b>7</b>	<b>APPENDIX.....</b>	<b>39</b>
7.1	List of Figures and Tables .....	39

# 1 Introduction

## 1.1 Purpose

This document describes user interface of sensor-less PMSM control, which is based on FM3/FM0+ series MCU. The description includes variable meaning and how to use the customer interface.

## 1.2 Variable Naming Rule

Note: Add fan motor variable naming rule, for multi-motor control, main motor variable still use prefix Motor0\_.

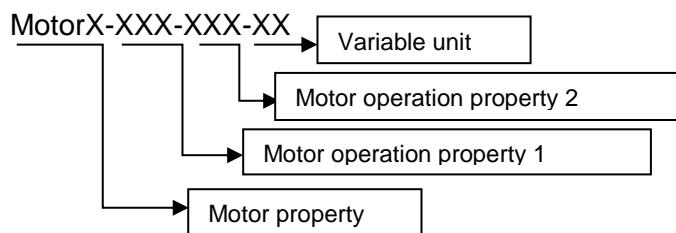


Figure 1-1: define the variables format

## 1.3 System Files Structure

DC fan motor files structure is listed as below.

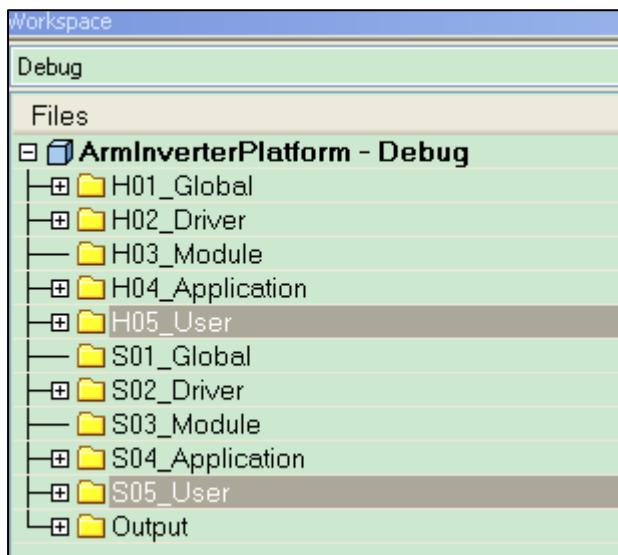


Figure 1-2: DC Fan Motor Files Structure

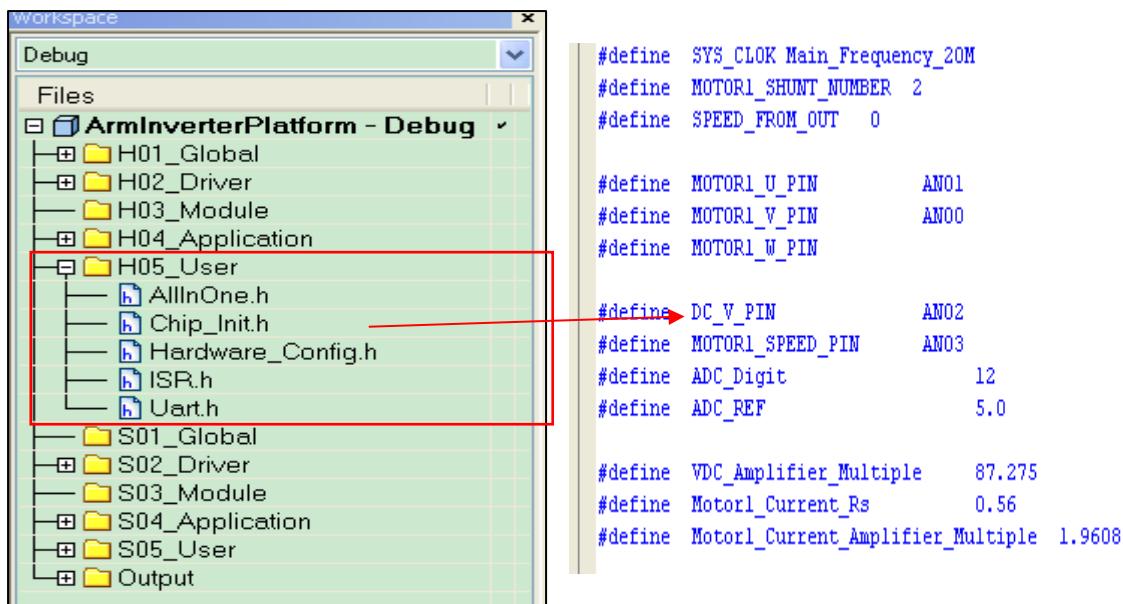


Figure 1-3: DC Fan Motor User H files Structure

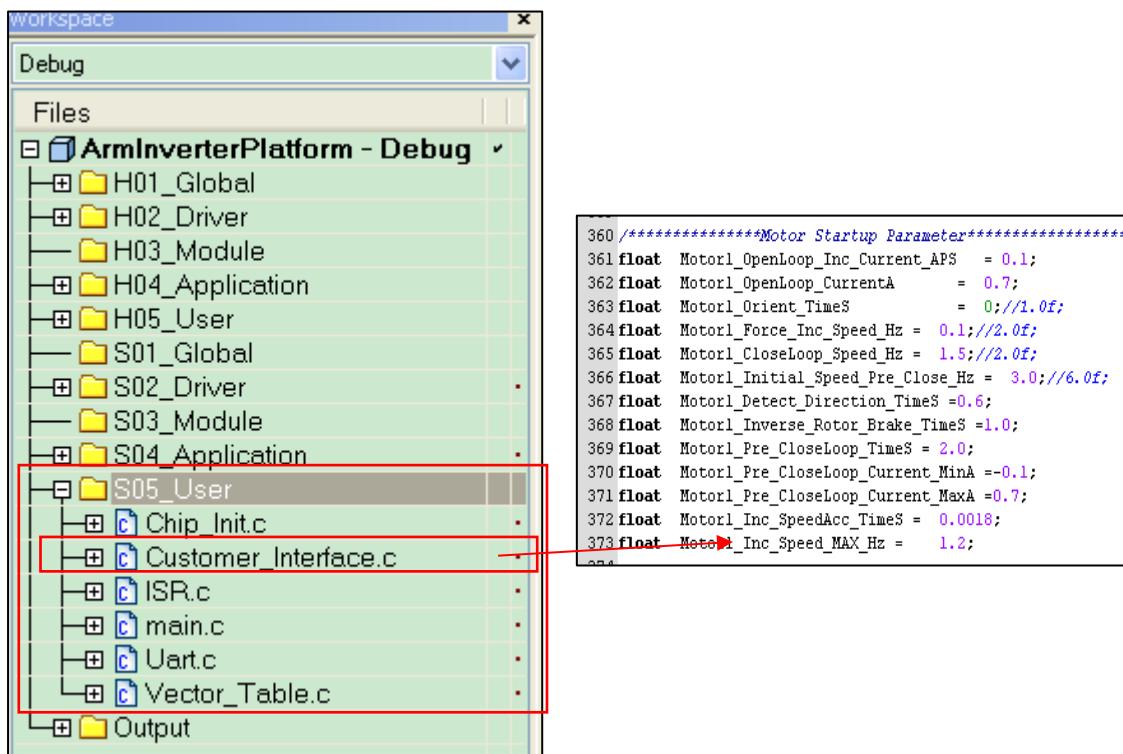


Figure 1-4: DC Fan Motor User S files Structure

If using M0、M4 chips need change H02\_Driver Files, you can requirement the file from Spansion engineers;

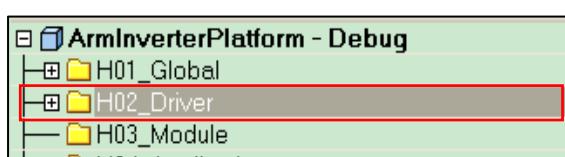


Figure 1-5: DC Fan Motor Driver H files Structure

## 2 System Build

### 2.1 Hardware Connect

When you get the Fan Motor EVB as show in Figure 2-1, please connect as below:

- ① AC power supply input interface;
- ② Emulator debug interface;
- ③ Motor three phase line interface;
- ④ Slide rheostat control speed interface (used in the middle of the interface of two pin);
- ⑤ GUI debug interface;

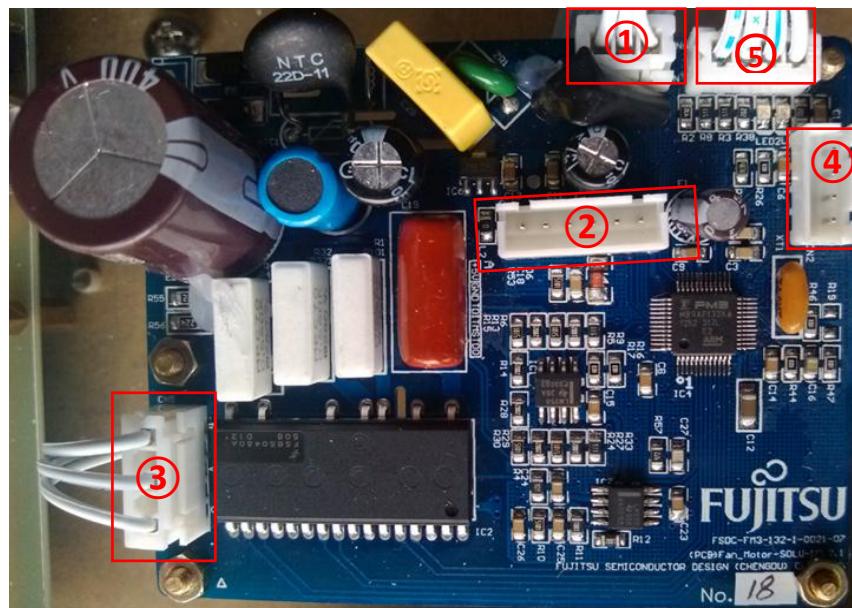


Figure 2-1: Appearance of EVB

## 2.2 System Files Structure

Open “IAR Embedded Workbench”, and click the “File>Open>Workspace”, the follow as show in Figure 2-2.

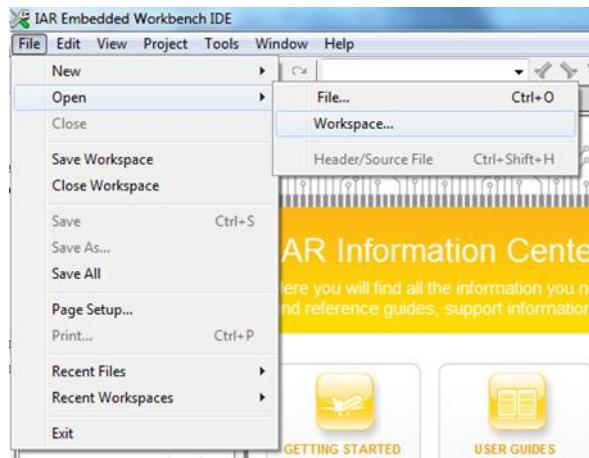


Figure 2-2: IAR Embedded Workbench project open follow

According to the project file storage path and open it as follow (it same as “[FWLIB]\Inverter Platform-XXX-XXX-XX\editor\EWARM\ArmInverterPlatform.eww”), Figure 2-3 show the IAR Embedded Workbench project Interface.

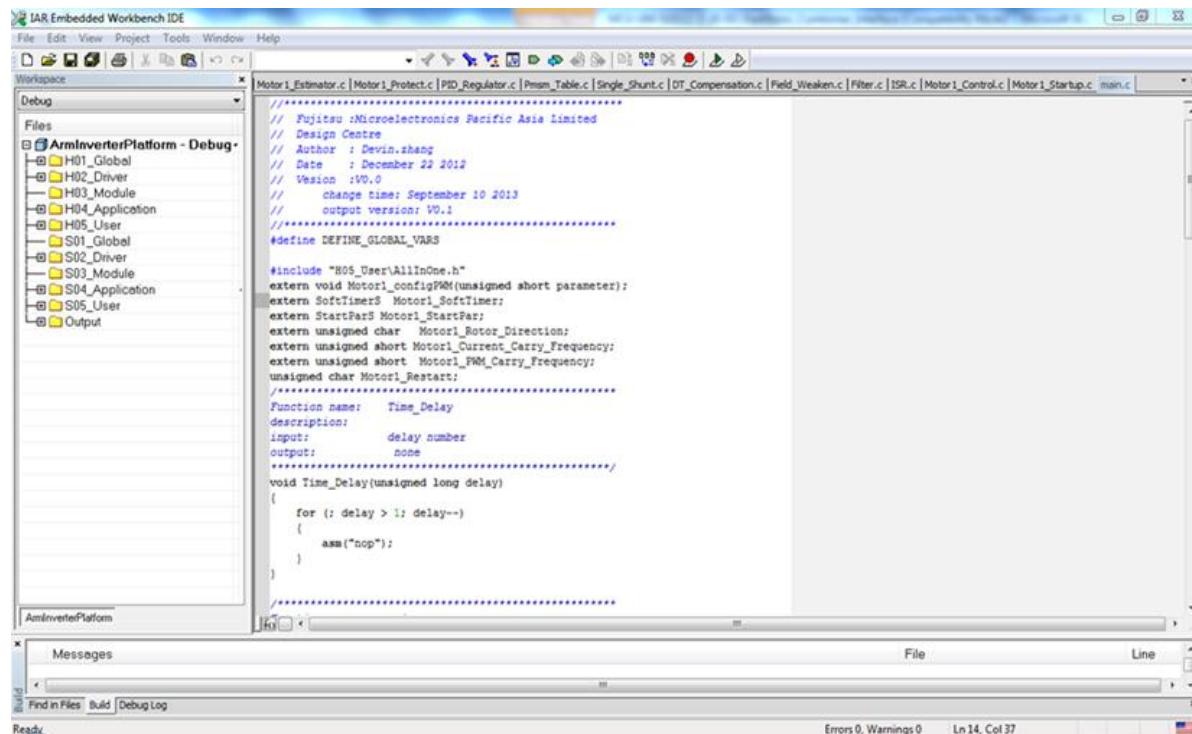


Figure 2-3: IAR Embedded Workbench project Interface

Figure 2-4 shows the firmware files structure, the file “Hardware\_Config.h” contain the hardware parameters, it is not necessary to change the hardware parameters usually, but if changed the Demo EVB hardware parameters or customer use his own hardware, please configure the hardware parameters and ensure that the parameters match the hardware.

“Customer\_Interface.c” used to configure the motor parameters, if the motor used as default with system, you not necessary to change the file too, but if you need, please configure it as follow section:

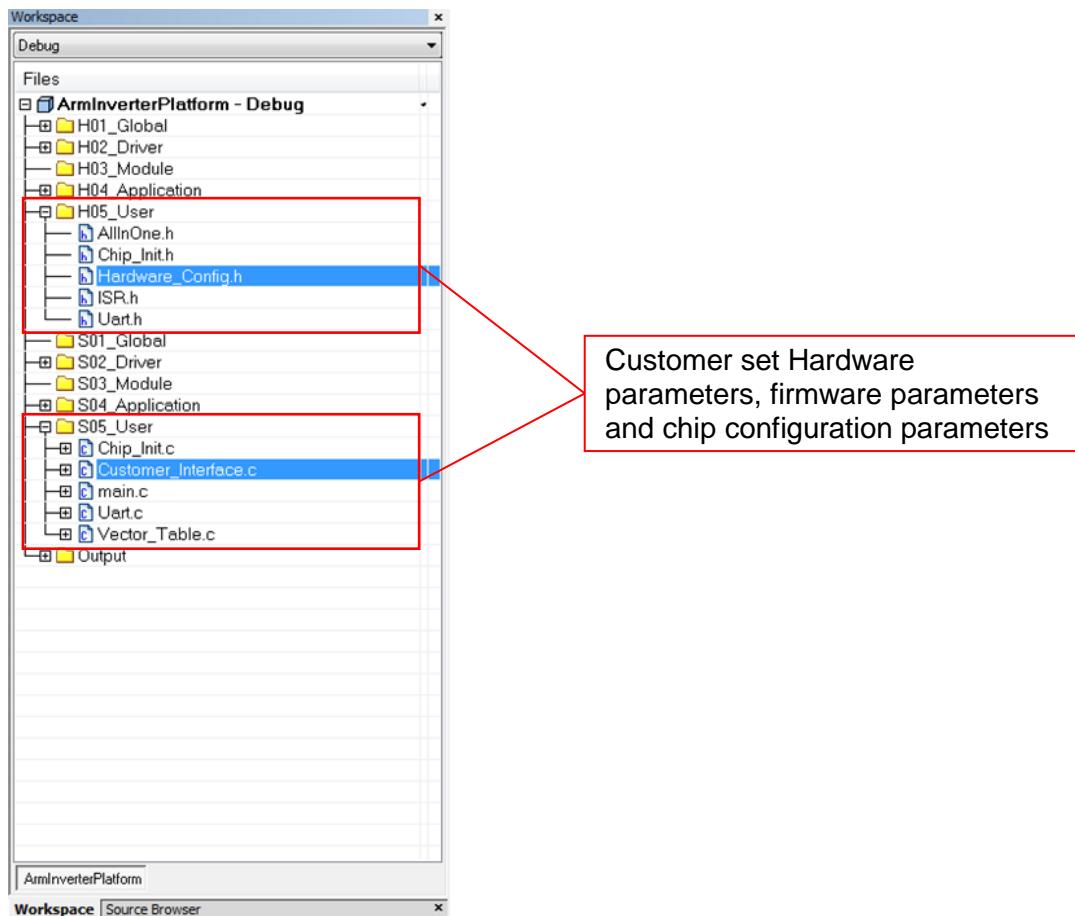
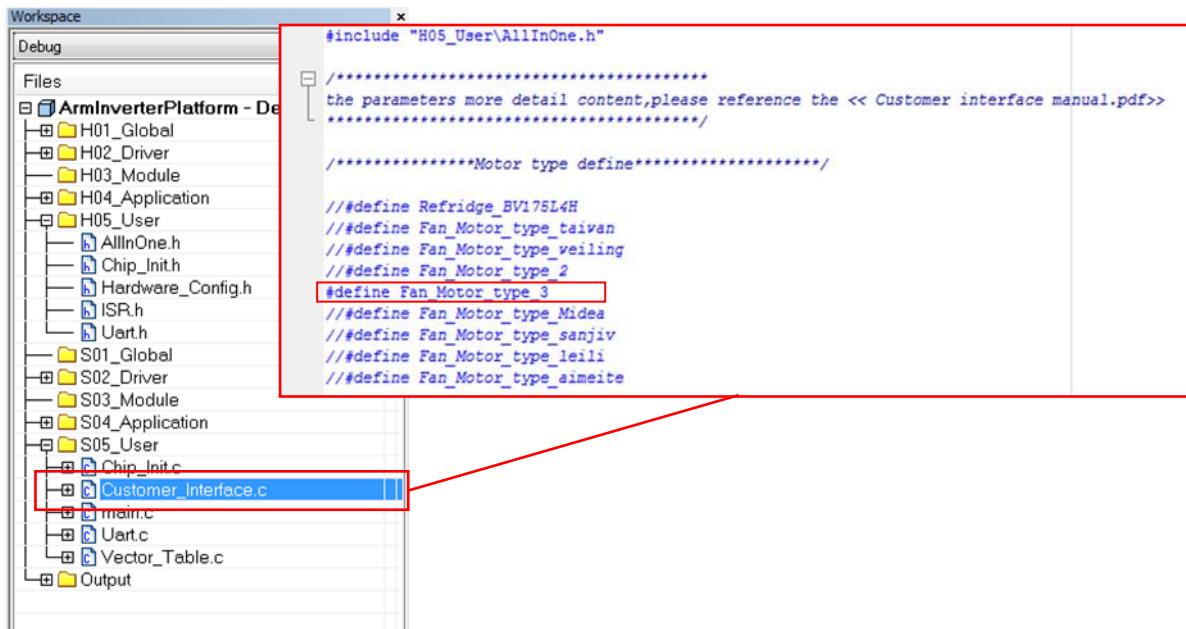


Figure 2-4: Project Files Structure

### 2.2.1 Configure Motor Parameters

If your motor parameters not in the default list.

Firstly, please define your motor type in the file “Customer\_Interface.c”, for example, your motor named as “Fan\_Motor\_Type\_3”, show as Figure 2-5.



```

Workspace x
Debug
Files
ArmlnverterPlatform - De
H01_Global
H02_Driver
H03_Module
H04_Application
H05_User
AllInOne.h
Chip_Init.h
Hardware_Config.h
ISP.h
Uart.h
S01_Global
S02_Driver
S03_Module
S04_Application
S05_User
Chip_Init.c
Customer_Interface.c
main.c
Uart.c
Vector_Table.c
Output
#include "H05_User\AllInOne.h"

*****
the parameters more detail content,please reference the << Customer interface manual.pdf>>
*****



*****Motor type define*****



#define Refridge_BV175L4H
#define Fan_Motor_type_taiwan
#define Fan_Motor_type_veiling
#define Fan_Motor_type_2
#define Fan_Motor_type_3
#define Fan_Motor_type_Midea
#define Fan_Motor_type_sanjiv
#define Fan_Motor_type_leili
#define Fan_Motor_type_aimeite

```

Figure 2-5: Define Motor Name

Then, you can copy a set of others motor defined data, and change the parameters as formats, such as motor pole pairs, inductance, resistance, inductive voltage constant, torque constant and moment of inertia constant, show as Figure 2-6. You can directly using the given parameters from motor factory, or measure it by yourself as the method as the follow chapter.

```

#ifndef Fan_Motor_type_3

*****Motor Paramters*****
unsigned char Motor1_Pole_Pairs = 4.0;           //the pole pairs of rotor
float Motor1_Ld = 107;                           //the d axis reductanc
float Motor1_Lq = 107;                           //the q axis reductance
float Motor1_Rs = 49;                            //the resistance between two phases(U,V eg.)
float Motor1_Ke = 120;                           //inductive voltage constant
float Motor1_Kt = 0.39;                          //torque constant
float Motor1_Kj = 0.685;                         //moment of inertia constant

```

Figure 2-6: Define Motor Parameters

## 2.2.2 Define PI Parameters

Usually, the initial value of PI regulate does not need to be modified, when the system running and program can adjust the parameters of speed loop and current loop according to system status automatically, and the parameters of speed loop depending on the moment of inertia, and the parameters of current loop depending on the value of resistance and inductive. Figure 2-7 show the PI regulate parameter interface, for the details, please reference the chapter of “PI parameters”.

```
*****PI Paramters Regulate information*****
float Motor1_Dki = 6.0;           //d axis current PI regulator integral constant
float Motor1_Dkp = 200.0;          //d axis current PI regulator proportion constant
float Motor1_Qki = 6.0;           //q axis current PI regulator integral constant
float Motor1_Qkp = 200.0;          //q axis current PI regulator proportion constant
float Motor1_Ski = 0.0003;         //speed PI regulator integral constant
float Motor1_Skp = 0.03;           //speed PI regulator proportion constant
float Motor1_Speed_Inc_Acceleration_Hz = 3.0; //the speed increase acceleration frequency value per second
float Motor1_Speed_Dec_Acceleration_Hz = 1.0; //the speed decrease acceleration frequency value per second
```

Figure 2-7: PI Regulate Parameters

## 2.2.3 Define Start-up Parameters

Customer can define the parameters according to what status of motor start-up, such as if you need orient and start-up with open loop, you need define the orient stage time and open loop stage maximum current. If not, please use it as default. The increase speed of open loop stage and the frequency of open loop change to close loop need to accordance with motor inertia and the lowest frequency to work properly. The motor detect direction times according to whether you need to start-up against the wind. If not need, please set it as 0. If you need, please set it as a nonzero, the number as smaller as better, and the typical values is 0.2S. After select prepare close loop start-up method, you need define prepare close stage maintain time and prepare close stage current value. And to define the open loop stage speed increase value and maximum value of open loop stage increase and PWM brake pulse increase time as show in Figure 2-8.

```
*****Motor Startup Parameter*****
float Motor1_OpenLoop_Inc_CurrentAPS = 0.1; //the openloop stage current increase value
float Motor1_OpenLoop_CurrentA = 0.3; //the openloop stage maximum current
float Motor1_Orient_TimeS = 0; //the motor start up orient stage time of openloop stage
float Motor1_Force_Inc_Speed_Hz = 0.1; //the increase speed of openloop stage
float Motor1_CloseLoop_Speed_Hz = 2.0; //the open loop frequency change to close loop
float Motor1_Initial_Speed_Fre_Close_Hz = 2; //the open loop increase speed and pre close module control
float Motor1_Detect_Direction_TimeS = 0.5; //the rotor initial status detection time
float Motor1_Inverse_Rotor_Brake_TimeS = 2; //the brake maximum time
float Motor1_Pre_CloseLoop_TimeS = 2.0; //the pre close stage maintain time
float Motor1_Pre_CloseLoop_Current_MinA = 0; //the close loop stage minimum current value
float Motor1_Pre_CloseLoop_Current_MaxA = 0.2; //the pre close stage maximum current value
float Motor1_Inc_SpeedAcc_TimeS = 0.0018; //the speed accelerator increase time
float Motor1_Inc_Speed_MAX_Hz = 1.2; //the speed accelerator maximum value
float Motor1_PWM_Brake_Inc_TimeS = 0.05; //the brake pwm pulse increase time
```

Figure 2-8: Motor Start-up Parameters

## 2.2.4 Define Motor Protect Parameters

The motor protection parameters base on the system features, the maximum soft over current value and maximum voltage and minimum voltage protection value must ensure that the motor and EVB hardware are not damaged and normal operation. Usually, the voltage protection value and error keeping time don't need to modify. The motor protection parameters show in Figure 2-9.

```
*****motor1 protection parameters*****
float      Motor1_Over_CurrentA = 0.9;           //the maximum soft over current protection
unsigned short Motor1_Max_Dc_VoltageV = 420;       //the maximum value of DC voltage is 400V
unsigned short Motor1_Min_Dc_VoltageV = 20;         //the minimum value of DC voltage is 140V
unsigned char  Motor1_Error_Keep_TimeS = 10;        // the maximum time error maintain
```

Figure 2-9: Motor Protection Parameters

## 2.2.5 Define Motor Running Parameters

To define the motor field value need according to the motor parameters and system operation voltage, if the power supply can drive motor running in maximum speed, the motor field value can set as zero. If not, please define the value as what you need, as proposal, the maximum current field weaken value don't more than sixty percent of the total current;

The value of dead time compensation need according to the motor parameters, if the motor's electrical characteristics as good as we need, we can ignore this value;

The dead time value according to the electrical characteristics of IPM or MOSFET and hardware layout, to meet demand, the value as smaller as possible, because it can influence of harmonic and control error;

Motor running maximum torque current according to the power of system;

To define the motor start running PWM carry frequency and the current implement frequency according motor electrical characteristics, if the electromagnetic noise and the characteristics of start-up are meet the requirements, the value does not need to modify. If enable dynamic change carry frequency function, system can matching the suitable frequency list in PWM carry frequency table to eliminate the noise.

In the same application domain, not need to modify the range of motor speed.

All the parameters in Figure 2-10 support online modification.

```
*****motor1 running parameters*****
float      Motor1_Field_Value = 0;           //the maximum current of field value
float      Motor1_Comp_MaxA = 0.2;           //Dead time compensation offset banlance current.
float      Motor1_Is_MaxA = 0.7;             //the motor running maximum torque current
float      Motor1_Dead_TimeUS = 2.0;          //the dead time value
unsigned short Motor1_PWM_Carry_Frequency =15000; //the motor start running PWM carry frequency
unsigned short Motor1_Current_Carry_Frequency=5000; //the current implement frequency
unsigned short Motor1_Speed_Max_RPM = 1000;    //the motor running maximum speed of RPM
unsigned short Motor1_Speed_Min_RPM = 180;      //the motor running minimum speed of RPM
unsigned char  Motor1_Rotor_Direction = 1;       //the motor running derrection 0:CW 1:CCW
unsigned char  Motor1_Running_Level = 4;          //the motor running stage
unsigned char  Motor1_Dynamic_PWM_Enable =0;     //the motor carry frequency switch 0:disable 1:enable
unsigned short Motor1_PWM_Table[30] = {           //Motor1 PWM Carry Frequency/1000
```

Figure 2-10: Motor Running Parameters

## 2.2.6 Define Hardware Parameters

For to define the hardware parameters, please reference the detail of hardware parameters configuration and calculation method in bellow chapter.

## 2.3 Motor Start-up

### 2.3.1 Code Build

When all the parameters configure finished, we can get into the simulation and debug interface. Firstly, we need build the project, there are two ways to achieve it, like show in Figure 2-11 or Figure 2-12.

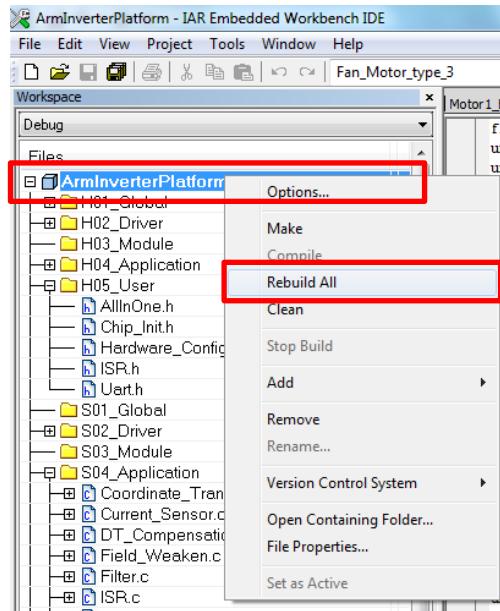


Figure 2-11: Code Build (1)

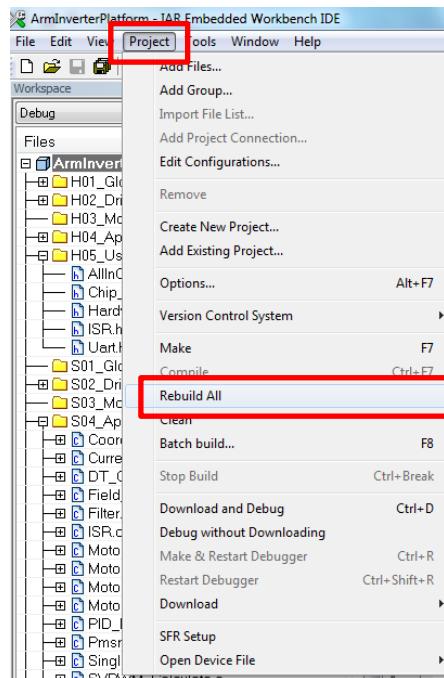


Figure 2-12: Code Build (2)

### 2.3.2 Download and Debug

Through the following ways achieve download code and enter debug interface.

The first way: like show in Figure 2-13;

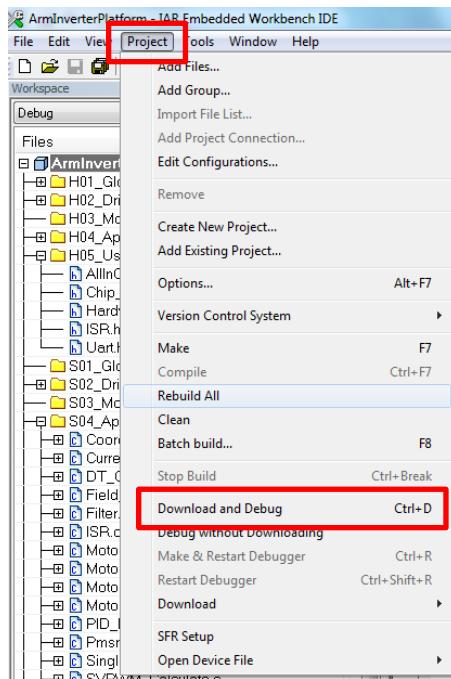


Figure 2-13: Download Code (1)

The second way: you can click on the shortcut button like show in Figure 2-14;

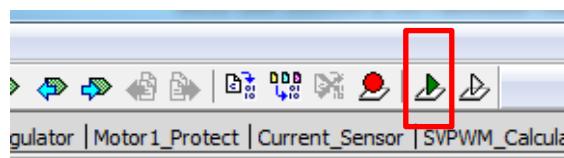


Figure 2-14: Download Code (2)

Through the above operation can enter the interface as show in Figure 2-15

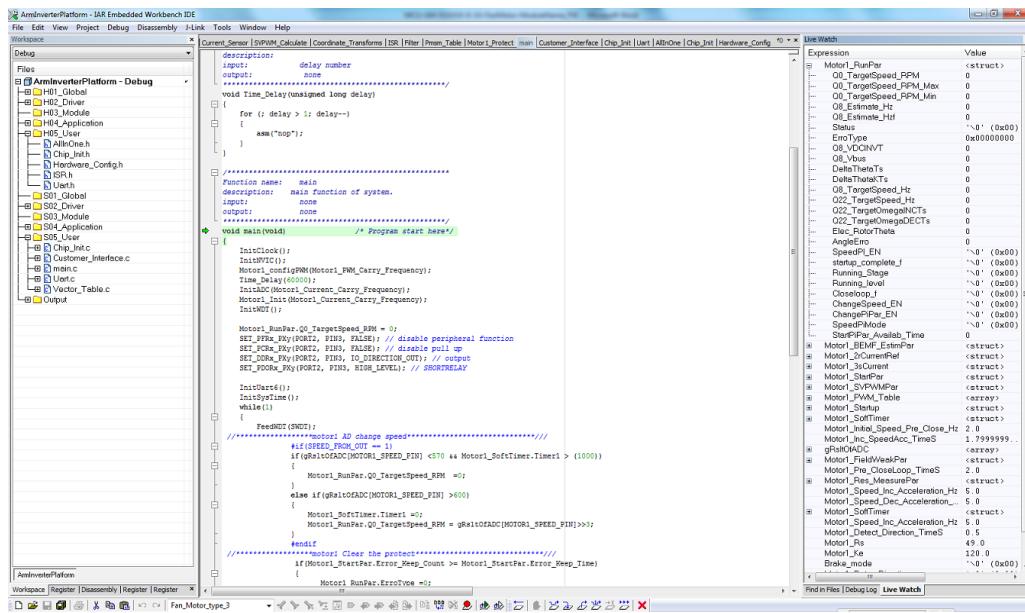


Figure 2-15: Debug Interface

### 2.3.3 Run Program and Motor

Click the button in the Figure 2-16 “” or press “F5” on the keyboard can make the program running, in this mode, program can response the instructions what we send and collect data and feedback the program state.



Figure 2-16: Program Control Button

Through “View>Live Watch” enter the variables monitored interface as show in Figure 2-17, type in the global variable name in “Expression” bar. For example, add the architecture “Motor1\_RunPar” and unfold it, set the “Q0\_TargetSpeed\_RPM” value as a nonzero number and press “Enter”, the motor start-up finished. After motor start-up, we need modify the parameters according to the motor running state and the current waveform display on the oscilloscope.

Expression	Value	Location
Motor1_RunPar	<struct>	0x200000DC
Q0_TargetSpeed_RPM	1	0x200000DC
Q0_TargetSpeed_RPM_M...	1000	0x200000E0
Q0_TargetSpeed_RPM_Min	180	0x200000E4
Q8_Estimate_Hz	0	0x200000E8
Q8_Estimate_Hzf	0	0x200000EC
Status	'\0' (0x00)	0x200000F0
ErrorType	0x00000000	0x200000F4
Q8_VDCINVVT	539	0x200000F8
Q8_Vbus	80924	0x200000FC
DeltaThetaTs	64701	0x20000100
DeltaThetaKTs	3355	0x20000104
Q8_TargetSpeed_Hz	768	0x20000108
Q22_TargetSpeed_Hz	12603846	0x2000010C
Q22_TargetOmegaINCTs	62914	0x20000110
Q22_TargetOmegaDECTs	62914	0x20000114
Elec_RotorTheta	1639339	0x20000118
AngleError	0	0x2000011C
SpeedPi_EN	'.' (0x01)	0x20000120
startup_complete_f	'.' (0x01)	0x20000121
Running_Stage	'\0' (0x00)	0x20000122
Running_level	'\0' (0x00)	0x20000123
Closeloop_f	'.' (0x02)	0x20000124
ChangeSpeed_EN	'.' (0x01)	0x20000125
ChangePiPar_EN	'\0' (0x00)	0x20000126
SpeedPiMode	'\0' (0x00)	0x20000127
StartPiPar_Available_Time	0	0x20000128
Motor1_BEMF_EstimPar	<struct>	0x20000000
Motor1_2rCurrentRef	<struct>	0x20000294
Motor1_3sCurrent	<struct>	0x20000288
Motor1_StartPar	<struct>	0x2000008C
Motor1_SVPWMPar	<struct>	0x2000012C
Motor1_PWM_Table	<array>	0x20000318

Figure 2-17: Variables Monitored Interface

## 2.4 Motor Performance Debug

### 2.4.1 Start-up Debug

For different start-up mode, you can configuration the parameters as follow:

1. Prepare close loop start-up mode;

The parameters configuration is show in Figure 2-18.

Live Watch	
Expression	Value
+ Motor1_RunPar	<struct>
Motor1_OpenLoop_Inc_Current...	1.00000001E-1
Motor1_OpenLoop_CurrentA	3.00000012E-1
Motor1_Orient_TimeS	0.0
Motor1_Force_Inc_Speed_Hz	1.00000001E-1
Motor1_CloseLoop_Speed_Hz	2.0
Motor1_Initial_Speed_Pre_Clos...	2.0
Motor1_Detect_Direction_TimeS	0.5
Motor1_Pre_CloseLoop_Curren...	2.00000003E-1

Figure 2-18: Prepare Close Loop Start-up Mode

The current waveform is show in Figure 2-19.

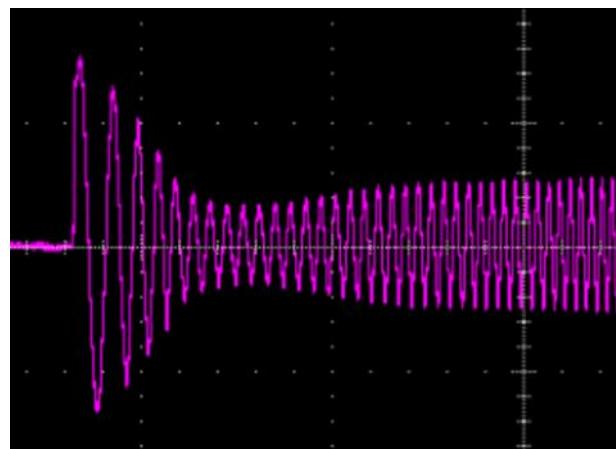


Figure 2-19: Prepare Close Loop Start-up Mode (100mA/1s/scale)

2. No step orient and open loop start-up mode;

The parameters configuration is show in Figure 2-20.

Live Watch	
Expression	Value
+ Motor1_RunPar	<struct>
Motor1_OpenLoop_Inc_Current...	1.00000001E-1
Motor1_OpenLoop_CurrentA	3.00000012E-1
Motor1_Orient_TimeS	1.0
Motor1_Force_Inc_Speed_Hz	1.00000001E-1
Motor1_CloseLoop_Speed_Hz	2.0
Motor1_Initial_Speed_Pre_Clos...	1.0
Motor1_Detect_Direction_TimeS	0.5
Motor1_Pre_CloseLoop_Curren...	2.00000003E-1

Figure 2-20: No Step Orient and Open Loop Start-up Mode

The current waveform is show in Figure 2-21.

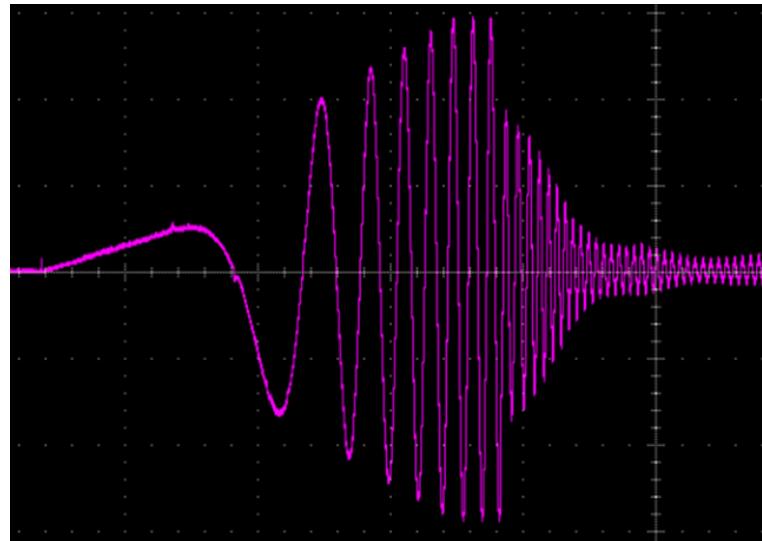


Figure 2-21: No Step Orient and Open Loop Start-up Mode (100mA/1s/scale)

3. Step orient and open loop start-up mode;

The parameters configuration is show in Figure 2-22.

Live Watch	
Expression	Value
Motor1_RunPar	<struct>
Motor1_OpenLoop_Inc_Current...	10.0
Motor1_OpenLoop_CurrentA	3.00000012E-1
Motor1_Orient_TimeS	1.0
Motor1_Force_Inc_Speed_Hz	1.00000001E-1
Motor1_CloseLoop_Speed_Hz	2.0
Motor1_Initial_Speed_Pre_Clos...	1.0
Motor1_Detect_Direction_TimeS	0.5
Motor1_Pre_CloseLoop_Curren...	2.00000003E-1

Figure 2-22: Step Orient and Open Loop Start-up Mode

The current waveform is show in Figure 2-23.

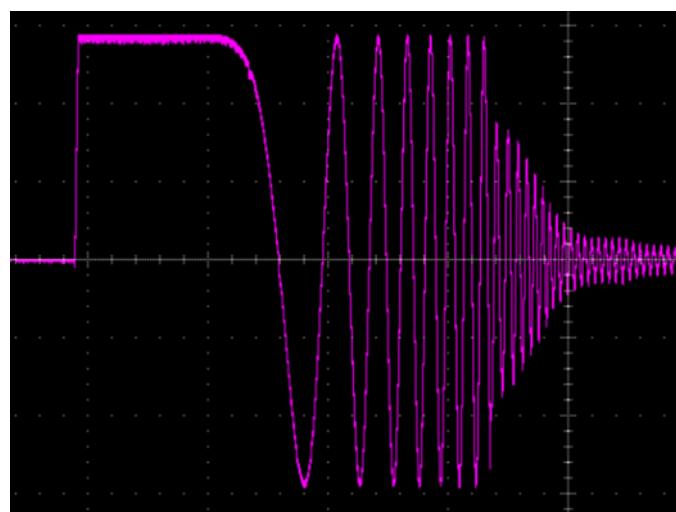


Figure 2-23: Step Orient and Open Loop Start-up Mode (100mA/1s/scale)

## 2.4.2 PI Parameters Debug

Modify parameters of speed loop and current loop according to motor current waveform and state.

1. Speed loop;

The parameters configuration is show in Figure 2-24.

Live Watch	
Expression	Value
+ Motor1_RunPar	<struct>
Motor1_Skp	1.0
Motor1_Ski	4.99999988E-3

Figure 2-24: Speed Loop Parameters

The current waveform is show in Figure 2-25.

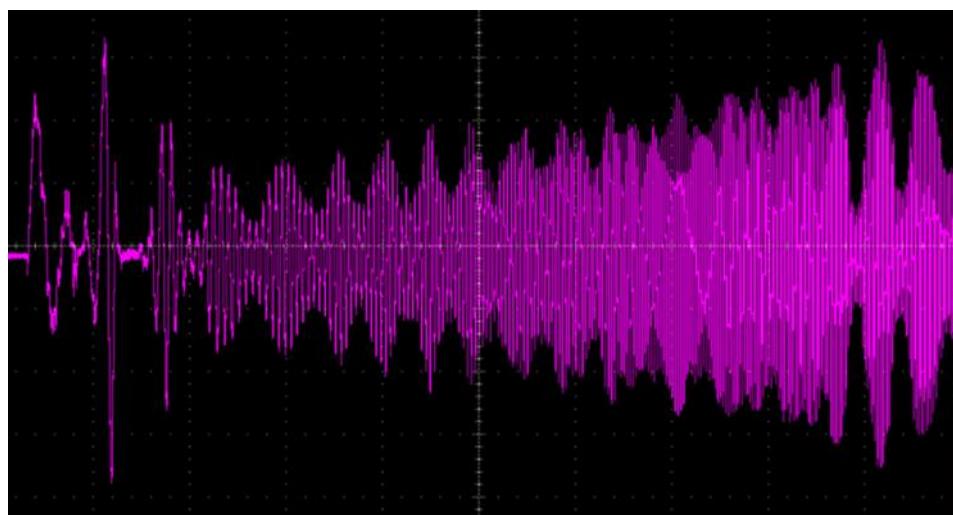


Figure 2-25: Current Waveform (100mA/1s/scale)

Modify the parameters is show in Figure 2-26

Live Watch	
Expression	Value
+ Motor1_RunPar	<struct>
Motor1_Skp	1.00000001E-1
Motor1_Ski	5.00000023E-4

Figure 2-26: Speed Loop Parameters

Now, the current waveform is show in Figure 2-27;

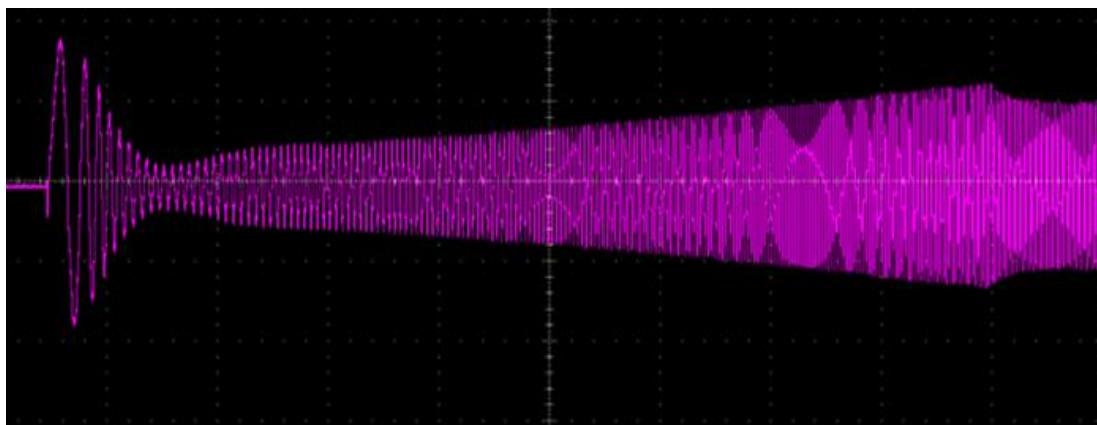


Figure 2-27: Current Waveform (100mA/1s/scale)

## 2. Current loop;

Modify the overlarge value to d and q component PI parameters, as show in Figure 2-28;

Live Watch	
Expression	Value
Motor1_RunPar	<struct>
Motor1_Dkp	2000.0
Motor1_Dki	42.0
Motor1_Qkp	2000.0
Motor1_Qki	42.0

Figure 2-28: Current Loop Parameters

Then, the current waveform is show in Figure 2-29;

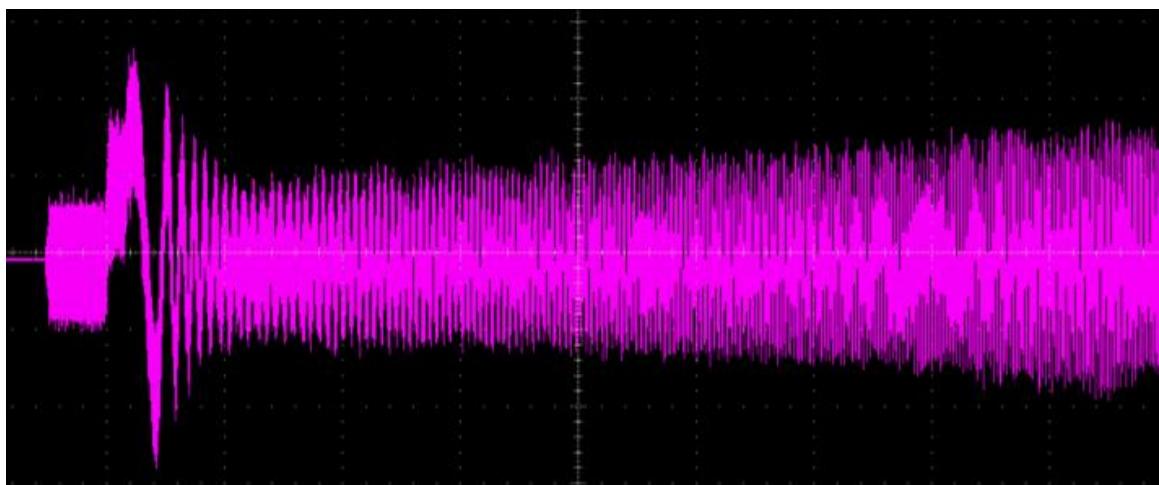


Figure 2-29: Current Waveform (100mA/1s/scale)

We can see that, current ripple and electromagnetic noise are very large.

If we set the PI parameters of d and q component are very small, as show in Figure 2-30;

Motor1_Dkp	1.00000001E-1
Motor1_Dki	2.1999998E-3
Motor1_Qkp	1.00000001E-1
Motor1_Qki	2.1999998E-3
Motor1_Pre_CloseLoop_Current...	6.9999988E-1
Motor1_2rCurrentRef	<struct>
Q12_d	0
Q12_q	998

Figure 2-30: Current Loop Parameters

Then, the current waveform is show in Figure 2-31.

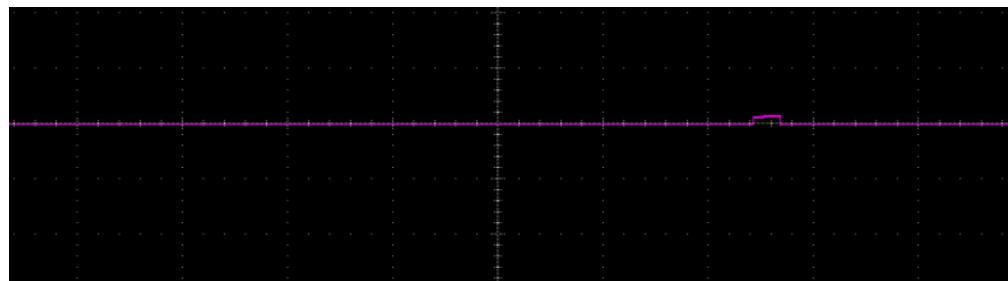


Figure 2-31: Current Waveform (100mA/1s/scale)

We can see that, current can't follow the value what the given current reference, system is abnormal.

If we set the PI parameters of d and q component are appropriate, as show in Figure 2-32;

Live Watch	
Expression	Value
Motor1_RunPar	<struct>
Motor1_Dkp	300.0
Motor1_Dki	6.0
Motor1_Qkp	300.0
Motor1_Qki	6.0
Motor1_Pre_CloseLoop_Current...	2.0000003E-1

Figure 2-32: Current Loop Parameters

Now, the current waveform as show in Figure 2-33;

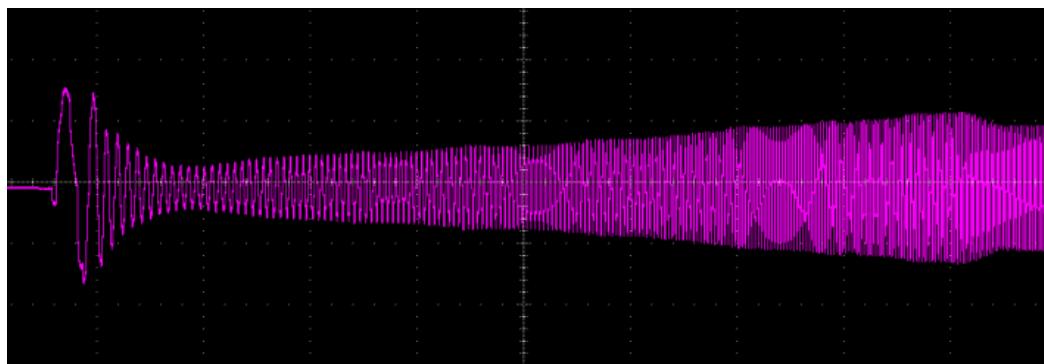


Figure 2-33: Current Waveform (100mA/1s/scale)

We can see that, the current ripple is very little and the current follow current reference very well.

### 3 Variable Description

#### 3.1 Customer Interface Variable List

##### Motor1: (Fan)

###### Motor Parameters

Motor1_Pole_Pairs	- Motor pole pairs
Motor1_Ld	- PMSM q axis phase inductance
Motor1_Lq	- PMSM d axis phase inductance
Motor1_Rs	-PMSM line resistor
Motor1_Ke	- PMSM Back Electromotive Force coefficient
Motor1_Kt	- PMSM torque const
Motor1_Kj	- PMSM moment of inertia

###### PI Parameters Regulate information

Motor1_dki	- D axis current loop integral parameter
Motor1_dkp	- D axis current loop proportion parameter
Motor1_qki	- Q axis current loop integral parameter
Motor1_qkp	- Q axis current loop proportion parameter
Motor1_ski	- Speed loop integral parameter
Motor1_skp	- Speed loop proportion parameter
Motor1_Speed_Inc_Acceleration_Hz	- Acceleration of speed loop
Motor1_Speed_Dec_Acceleration_Hz	- Deceleration of speed loop

###### Motor Startup Parameters

Motor1_OpenLoop_Inc_CurrentAPS	- Current vary step in open loop
Motor1_Openloop_CurrentA	- Current value in open loop
Motor1_Orient_TimeS	- Orient time
Motor1_Force_Inc_Speed_Hz	- Open loop acceleration
Motor1_CoseLoop_Speed_Hz	- Target speed when switching to closed loop
Motor1_Initial_Speed_Pre_Close_Hz	- Speed initial value before closed loop
Motor1_Detect_Direction_TimeS	- Detection time of rotational direction
Motor1_Inverse_Rotor_Brake_TimeS	- Brake time of motor reversion
Motor1_Pre_CloseLoop_TimeS	- Time before closed loop
Motor1_Pre_CloseLoop_Current_MinA	- Minimum current before closed loop
Motor1_Pre_CloseLoop_Current_MaxA	- Maximum current before closed loop
Motor1_Inc_SpeedAcc_TimeS	- Acceleration interval
Motor1_Inc_Speed_MAX_Hz	- Maximum speed acceleration
Motor1_PWM_Brake_Inc_TimeS	- PWM brake mode brake duty ratio pulse increase time

###### Motor1 protection parameters

Motor1_Over_CurrentA	- Software over current point
Motor1_max_dcvoltagEV	- DC bus over voltage point
Motor1_min_dcvoltagEV	- DC bus under voltage point
Motor1_Error_keep_timeS	- Fault maintain time

### **Motor1 running parameters**

Motor1_Field_Value	- Maximum current of field value
Motor1_Comp_MaxA	- Dead time compensation offset balance current
Motor1_is_MaxA	- Motor running maximum torque current
Motor1_dead_timeUS	- Dead time value
Motor1_PWM_Carry_Frequency	- Motor start running PWM carry frequency
Motor1_Current_Carry_Frequency	- Current implement frequency
Motor1_speed_Max_RPM	-Motor running maximum speed
Motor1_speed_Min_RPM	-Motor running minimum speed
Motor1_Rotor_direction	-Motor running direction set
Motor1_Running_Level	-Set motor running stage
Motor1_Dynamic_PWM_Enable	-Motor carrier dynamic change switching
Motor1_PWM_Table	-Motor carrier frequency table

### **3.2 Motor1\_pole\_pairs**

Data type: unsigned char

Description: PMSM pole pairs

Unit: none

Measure method: Turn motor rotor manually, test motor three-phase terminal line voltage waveform, record turning revolution N and sine wave period number M, so

$$\text{Motor1\_pole\_pairs} = M/N$$

### **3.3 Motor1\_Ld**

Data type: float

Description: PMSM d axis phase inductance

Unit: mH

Note: This parameter can be obtained from motor parameters or testing manually.

### **3.4 Motor1\_Lq**

Data type: float

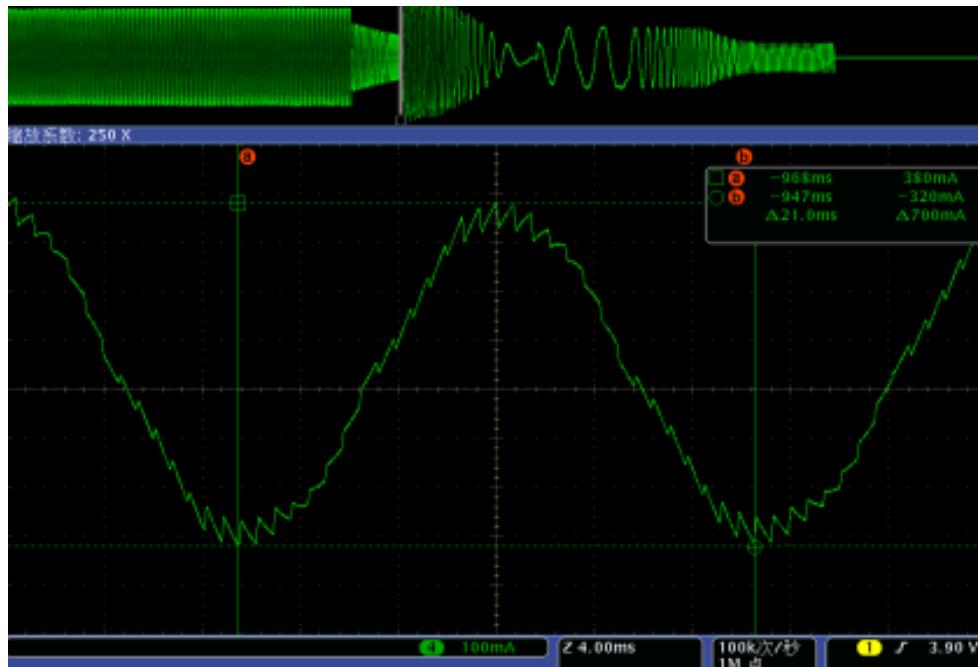
Description: PMSM q axis phase inductance

Unit: mH

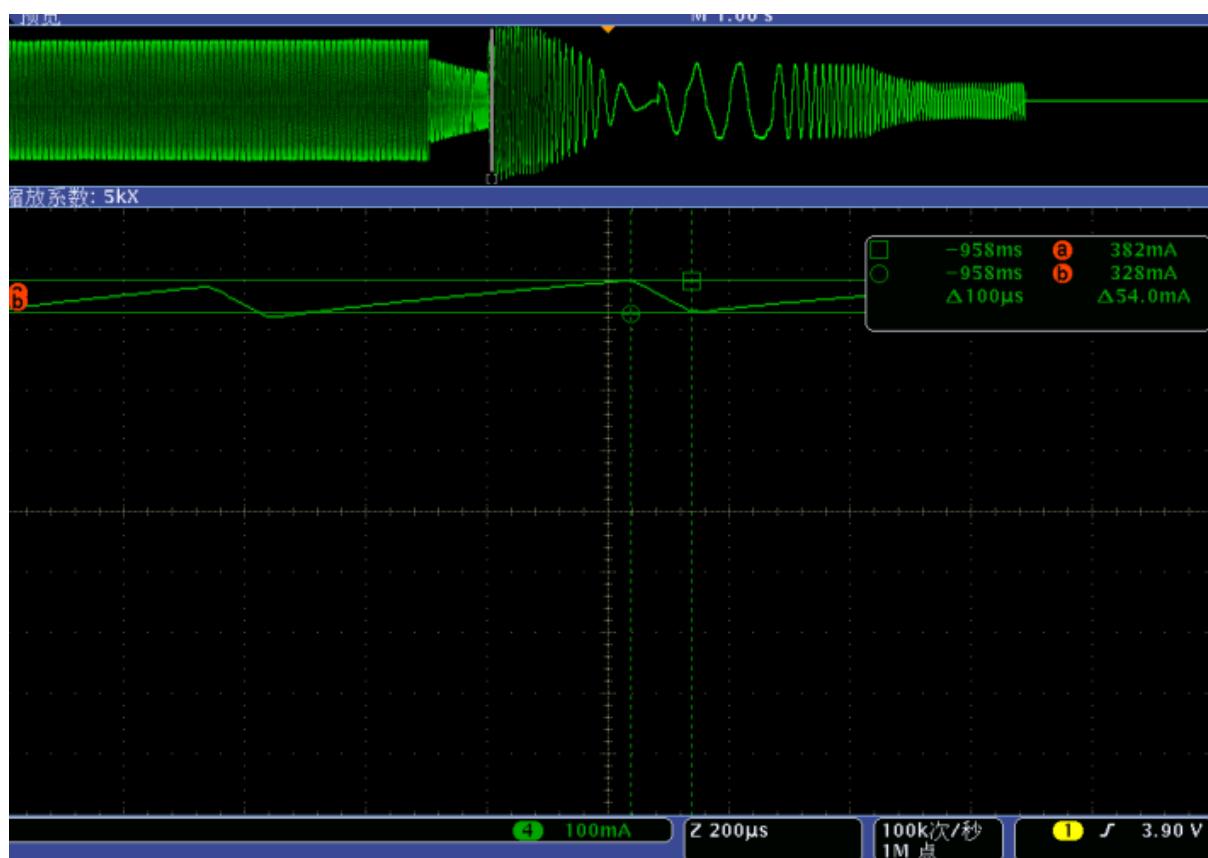
Note: This parameter can be obtained from motor parameters or testing manually.

Measure method:

When motor is running, brake motor manually and measure the phase current waveform. The current waveform is shown in Figure 3-1. Then Lq is calculated by using braking phase current. This method is only applicable to SPM motor.



When motor braking, the phase current waveform detail is shown in Figure 3-2.



Consider  $K_e$ ,  $R_s$  and `Motor1_pole_pairs` are known, record the following data when motor braking:

The frequency of current waveform: F

The current falling time: T

The current amplitude changes in falling time: A

The peak-peak current:  $A_0$

So  $Lq$  is calculated as follows,

$$\text{Motor1\_Lq} = (F * 60 / \text{Motor1\_pole\_pairs} * \text{Motor1\_Ke} / 1000 - \text{Motor1\_Res} * A_0 / 2) * T / A$$

### **3.5 Motor1\_Res**

Data type: float

Description: PMSM line resistor

Unit:  $\Omega$

Measure method: Test any two phases of resistance using multi-meter, record three test results  $R_{uv}$ ,  $R_{vw}$ ,  $R_{uw}$ , so

$$\text{Motor1\_Res} = (R_{uv} + R_{vw} + R_{uw}) / 3$$

### **3.6 Motor1\_Ke**

Data type: float

Description: PMSM Back Electromotive Force coefficient

Unit: V/Krpm

Measure method: Turn motor rotor rapidly, test motor three-phase terminal line voltage waveform, and choose a relatively stable range of line voltage amplitude, record voltage peak-peak value and frequency in the range.

$$\text{Motor1\_Ke} = V * \text{Motor1\_pole\_pairs} * 5.8934465 / F$$

### **3.7 Motor1\_Kt**

Data type: float

Description: PMSM torque const

Unit: N·m/Arms

### **3.8 Motor1\_Kj**

Data type: float

Description: PMSM moment of inertia

Unit: Kg·m<sup>2</sup>

### **3.9 Motor1\_dki**

Data type: float

Description: D axis current loop integral parameter

Unit: none

Range: about 0.3

### **3.10 Motor1\_dkp**

Data type: float

Description: D axis current loop proportion parameter

Unit: none

Range: about 0.3

**3.11 Motor1\_qki**

Data type: float

Description: Q axis current loop integral parameter

Unit: none

Range: about 0.3

**3.12 Motor1\_qkp**

Data type: float

Description: Q axis current loop proportion parameter

Unit: none

Range: about 0.3

**3.13 Motor1\_ski**

Data type: float

Description: Speed loop integral parameter

Unit: none

Range: about 0.0005

**3.14 Motor1\_skp**

Data type: float

Description: Speed loop proportion parameter

Unit: none

Range: about 0.3

**3.15 Motor1\_Speed\_Inc\_Acceleration\_Hz**

Data type: float

Description: Acceleration of speed loop

Unit: Hz

**3.16 Motor1\_Speed\_Dec\_Acceleration\_Hz**

Data type: float

Description: Deceleration of speed loop

Unit: Hz

**3.17 Motor1\_OpenLoop\_Inc\_CurrentAPS**

Data type: float

Description: Current vary step in open loop

Unit: A

**3.18 Motor1\_OpenLoop\_CurrentA**

Data type: float

Description: Current value in open loop

Unit: A

### **3.19 Motor1\_Orient\_TimeS**

Data type: float

Description: Orient time

Unit: s

### **3.20 Motor1\_Force\_Inc\_Speed\_Hz**

Data type: float

Description: Open loop acceleration

Unit: Hz

### **3.21 Motor1\_CloseLoop\_Speed\_Hz**

Data type: float

Description: Target speed when switching to closed loop

Unit: Hz

### **3.22 Motor1\_Initial\_Speed\_Pre\_Close\_Hz**

Data type: float

Description: Speed initial value before closed loop

Unit: Hz

### **3.23 Motor1\_Detect\_Direction\_TimeS**

Data type: float

Description: Detection time of rotational direction

Unit: s

### **3.24 Motor1\_Inverse\_Rotor\_Brake\_TimeS**

Data type: float

Description: Brake time of motor reversion

Unit: s

### **3.25 Motor1\_Pre\_CloseLoop\_TimeS**

Data type: float

Description: Time before closed loop

Unit: s

### **3.26 Motor1\_Pre\_CloseLoop\_Current\_MinA**

Data type: float

Description: Minimum current before closed loop

Unit: A

### **3.27 Motor1\_Pre\_CloseLoop\_Current\_MaxA**

Data type: float

Description: Maximum current before closed loop

Unit: A

**3.28 Motor1\_Inc\_SpeedAcc\_TimeS**

Data type: float

Description: Acceleration interval

Unit: s

**3.29 Motor1\_Inc\_Speed\_MAX\_Hz**

Data type: float

Description: Maximum speed acceleration

Unit: Hz

**3.30 Motor1\_PWM\_Brake\_Inc\_TimeS**

Data type: float

Description: the PWM brake acceleration times

Unit: S

**3.31 Motor1\_Over\_CurrentA**

Data type: float

Description: Software over current point

Unit: A

**3.32 Motor1\_max\_dcVoltageV**

Data type: unsigned short

Description: DC bus over voltage point

Unit: V

**3.33 Motor1\_min\_dcVoltageV**

Data type: unsigned short

Description: DC bus under voltage point

Unit: V

**3.34 Motor1\_Error\_keep\_timeS**

Data type: unsigned char

Description: Fault maintain time

Unit: s

**3.35 Motor1\_Field\_Value**

Data type: float

Description: Maximum current of field weaken

Unit: A

**3.36 Motor1\_Comp\_MaxA**

Data type: float

Description: the dead time compensation value of current

Unit: A

### **3.37 Motor1\_is\_MaxA**

Data type: float

Description: Maximum output current

Unit: A

### **3.38 Motor1\_dead\_timeUS**

Data type: float

Description: PWM dead time

Unit: us

### **3.39 Motor1\_PWM\_Carry\_Frequency**

Data type: unsigned short

Description: PWM carrier frequency

Unit: Hz

### **3.40 Motor1\_Current\_Carry\_Frequency**

Data type: unsigned short

Description: Motor control algorithm frequency

Unit: Hz

### **3.41 Motor1\_speed\_Max\_RPM**

Data type: unsigned short

Description: Maximum speed

Unit: RPM

### **3.42 Motor1\_speed\_Min\_RPM**

Data type: unsigned short

Description: Minimum speed

Unit: RPM

### **3.43 Motor1\_Rotor\_direction**

Data type: unsigned char

Description: Motor rotate direction

Unit: none

Range: 0->clockwise, 1->anticlockwise

### **3.44 Motor1\_Running\_Level**

Data type: unsigned char

Description: Motor running level

Unit: none

Range: 0->Brake, 1->Orient, 2->Open loop, 3->Closed loop, 4->Set running speed

### **3.45 Motor1\_Dynamic\_PWM\_Enable**

Data type: unsigned char

Description: switch of the Dynamic PWM function

Unit: none

Range: 0->disable, 1->enable

### **3.46 Motor1\_PWM\_Table**

Data type: unsigned short

Description: the carry frequency of per frequency

Unit: KHZ

## 4 Hardware Configuration

### 4.1 AD\_OFFSET\_NUM

Data type: macro define

Description: AD middle voltage calculator value 2<sup>AD\_OFFSET\_NUM</sup>

Unit: None

### 4.2 AD\_OFFSET\_MAX\_VALUE

Data type: macro define

Description: AD middle voltage maximum offset

Unit: None

### 4.3 Motor1\_Control\_mode

Data type: macro define

Description: Motor1 control mode switch

Range: 0: Speed 1:Torque 2:VF 3: VF\_Diagnose

### 4.4 MOTOR1\_PWM\_POLAR\_AVAILABLE

Data type: macro define

Description: Motor1 control IMP polar

Range: 0: LOW\_POLAR 1: HIGH\_POLAR

### 4.5 SYS\_CLOK

Data type: macro define

Description: MCU main frequency

Unit: MHz

Range: 0—20MHz

### 4.6 MOTOR1\_SHUNT\_NUMBER

Data type: macro define

Description: The number of sampling resistor

Unit: none

Range: 0, 1, 2

### 4.7 SPEED\_FROM\_OUT

Data type: macro define

Description: modify the speed whether the out AD

Unit: none

Range: 0: from the UART or debug set 1: from the out AD

### 4.8 MOTOR1\_U\_PIN

Data type: macro define

Description: AD channel of phase U current

Unit: MCU pin

Range: AN00—AN05

#### **4.9 MOTOR1\_V\_PIN**

Data type: macro define

Description: AD channel of phase V current

Unit: MCU pin

Range: AN00—AN05

#### **4.10 MOTOR1\_W\_PIN**

Data type: macro define

Description: AD channel of phase W current

Unit: MCU pin

Range: AN00—AN05

#### **4.11 DC\_V\_PIN**

Data type: macro define

Description: AD channel of DC bus voltage

Unit: MCU pin

Range: AN00—AN05

#### **4.12 MOTOR1\_SPEED\_PIN**

Data type: macro define

Description: AD channel of speed

Unit: MCU pin

Range: AN00—AN05

#### **4.13 ADC\_Digit**

Data type: macro define

Description: AD converter resolution

Unit: bit

Range: 10-bit, 12-bit

#### **4.14 ADC\_REF**

Data type: macro define

Description: ADC reference voltage

Unit: V

Range: 1.8V/3.3V/5V

#### **4.15 VDC\_Amplifier\_Multiple**

Data type: macro define

Description: DC bus voltage amplifying circuit coefficient

Unit: V

Calculation process:

$$VDC\_Amplifier\_Multiple = (R56+R55+R53)/R53$$

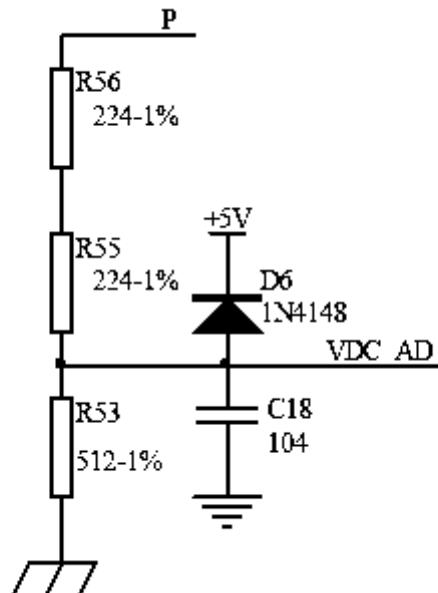


Figure 4-1: DC Bus Voltage Amplifying Circuit

#### 4.16 Motor1\_Current\_Rs

Data type: macro define

Description: Current sampling resistance of Motor1

Unit:  $\Omega$

Note: The current sampling circuit is shown in Figure 4-2, the sampling resistance is  $0.47\Omega$ .

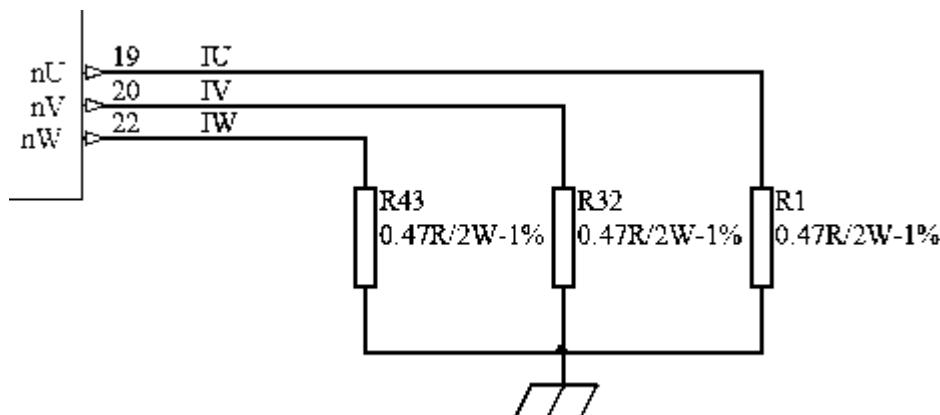


Figure 4-2: Current Sampling Circuit

#### 4.17 Motor1\_Current\_Amplifier\_Multiple

Data type: macro define

Description: Current amplifying circuit coefficient

Unit: none

Calculation process:

$$\text{Motor1_Current_Amplifier_Multiple} = R27/R24$$

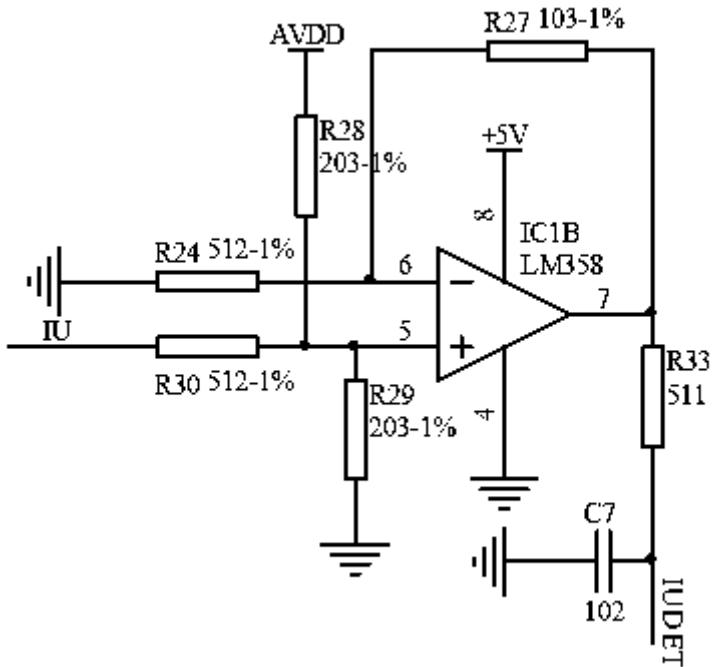


Figure 4-3: Current Amplifying Circuit

#### 4.18 MOTOR1\_BEMF\_LPFK\_MIN

Data type: macro define

Description: the minimum of motor1 estimate filters K value

Unit: None

#### 4.19 MOTOR1\_BEMF\_LPFK\_MAX

Data type: macro define

Description: the maximum of motor1 estimate filters K value

Unit: None

#### 4.20 MOTOR1\_BEMF\_LPFK\_MIN\_HZ

Data type: macro define

Description: the minimum of motor1 estimate cut-off frequency

Unit: None

#### 4.21 MOTOR1\_BEMF\_LPFK\_MAX\_HZ

Data type: macro define

Description: the maximum of motor1 estimate cut-off frequency

Unit: None

## 5 System error code define

Chart 5-1: error code

Error code (Error Type)	Error type	remark
0x00	Initial state	
0x01	Over DC voltage	
0x02	Under DC voltage	
0x04	Soft over current	
0x08	Hardware over current	
0x10	Loss phase	
0x20	No connect the motor	
0x40	AD offset error	
0x80	Software watch dog	
0x100	Lock the motor	
0x200	Handle the error	
0x400	Hardware watch dog	

## 6 Additional Information

For more Information on cypress semiconductor products, visit the following websites:

website address:

<http://www.cypress.com/>

## 7 Appendix

### 7.1 List of Figures and Tables

Chart 5-1: error code.....	37
Figure 1-1: define the variables format.....	7
Figure 1-2: DC Fan Motor Files Structure.....	7
Figure 1-3: DC Fan Motor User H files Structure.....	8
Figure 1-4: DC Fan Motor User S files Structure .....	8
Figure 1-5: DC Fan Motor Driver H files Structure .....	8
Figure 2-1: Appearance of EVB .....	9
Figure 2-2: IAR Embedded Workbench project open follow .....	10
Figure 2-3: IAR Embedded Workbench project Interface .....	10
Figure 2-4: Project Files Structure.....	11
Figure 2-5: Define Motor Name.....	12
Figure 2-6: Define Motor Parameters .....	12
Figure 2-7: PI Regulate Parameters.....	13
Figure 2-8: Motor Start-up Parameters.....	13
Figure 2-9: Motor Protection Parameters .....	14
Figure 2-10: Motor Running Parameters .....	14
Figure 2-11: Code Build (1).....	15
Figure 2-12: Code Build (2).....	15
Figure 2-13: Download Code (1) .....	16
Figure 2-14: Download Code (2) .....	16
Figure 2-15: Debug Interface .....	17
Figure 2-16: Program Control Button .....	17
Figure 2-17: Variables Monitored Interface .....	18
Figure 2-18: Prepare Close Loop Start-up Mode .....	19
Figure 2-19: Prepare Close Loop Start-up Mode (100mA/1s/scale) .....	19
Figure 2-20: No Step Orient and Open Loop Start-up Mode.....	19
Figure 2-21: No Step Orient and Open Loop Start-up Mode (100mA/1s/scale) .....	20
Figure 2-22: Step Orient and Open Loop Start-up Mode .....	20
Figure 2-23: Step Orient and Open Loop Start-up Mode (100mA/1s/scale).....	20
Figure 2-24: Speed Loop Parameters .....	21
Figure 2-25: Current Waveform (100mA/1s/scale) .....	21
Figure 2-26: Speed Loop Parameters .....	21
Figure 2-27: Current Waveform (100mA/1s/scale) .....	22
Figure 2-28: Current Loop Parameters.....	22

Figure 2-29: Current Waveform (100mA/1s/scale) .....	22
Figure 2-30: Current Loop Parameters.....	23
Figure 2-31: Current Waveform (100mA/1s/scale) .....	23
Figure 2-32: Current Loop Parameters.....	23
Figure 2-33: Current Waveform (100mA/1s/scale) .....	23
Figure 3-1: Motor Current Waveform when Braking .....	26
Figure 3-2: Motor Current Waveform Detail.....	26
Figure 4-1: DC Bus Voltage Amplifying Circuit .....	35
Figure 4-2: Current Sampling Circuit .....	35
Figure 4-3: Current Amplifying Circuit .....	36