Application Note

Document No.: AN1097

APM32F035_MOTOR EVAL Senseless Vector Control Scheme

Version: V1.3

Contents

1 General Introduction

1.1 Project Overview

APM32F035 is a specialized chip launched by Geehy Semiconductor Co., Ltd. for motor control. Based on APM32F035, this design provides a dual-resistance sampling vector control scheme and uses the closed-loop sliding-mode observer estimation scheme. The detailed design specifications are shown in the table below:

Table 1 Design Specifications

Control mode	Position Sensorless Field Oriented Control (FOC)
Observer	Sliding-mode observer+PLL
PWM modulation mode	SVPWM
PWM frequency	15KHz
Motor speed	400~3000RPM (2 pairs of poles)
Starting mode	Open-loop starting
Protection function	Overvoltage, undervoltage, overcurrent, locked rotor
Code size	11Kbytes
Development software	Keil MDK (V5.23 version and above)

1.2 APM32F035 Chip Resources

APM32F035 is a high-performance special MCU for motor control which is based on the Arm Cortex-M0+ core, integrates the mathematical operation accelerators (Cordic, SvPWM, hardware divider, etc.) commonly used in FOC algorithms, and integrates such analog peripherals as amplifiers and comparators, as well as CAN controllers.

Table 2 Functions and Peripherals of APM32F035 Series Chip

Product		**APM32F035**	
Model		**C8T7**	**K8T7**
Package		LQFP48	LQFP32
Core and maximum working frequency		Arm® 32-bit Cortex®-M0+@72MHz	
M0CP Co-processor		1	
Flash memory (KB)		64	
SRAM(KB)		10	
Timer	32-bit/1-bit universal	1/2	
	16-bit advanced	1	

Product		APM32F035	
Model		C8T7	K8T7
	16-bit basic	2	
	24-bit counter	1	
	Watchdog (WDT)	2 (1 independent watchdog+1 window watchdog)	
	Real-time clock	1	
Communication interface	USART	2	
	SPI/I2S	1/1	
	I2C	1	
	CAN	1	
12-bit ADC	Unit	1	
	External channel	16	12
	Internal channel	3	
Comparator (COMP)		2	
Operational amplifier (OPA)		4	2
GPIOs		42	27
Operating temperature		Ambient temperature： -40℃ to 105℃ Junction temperature： -40℃ to 125℃	
Working voltage		2.0~3.6V	

2 Hardware Introduction

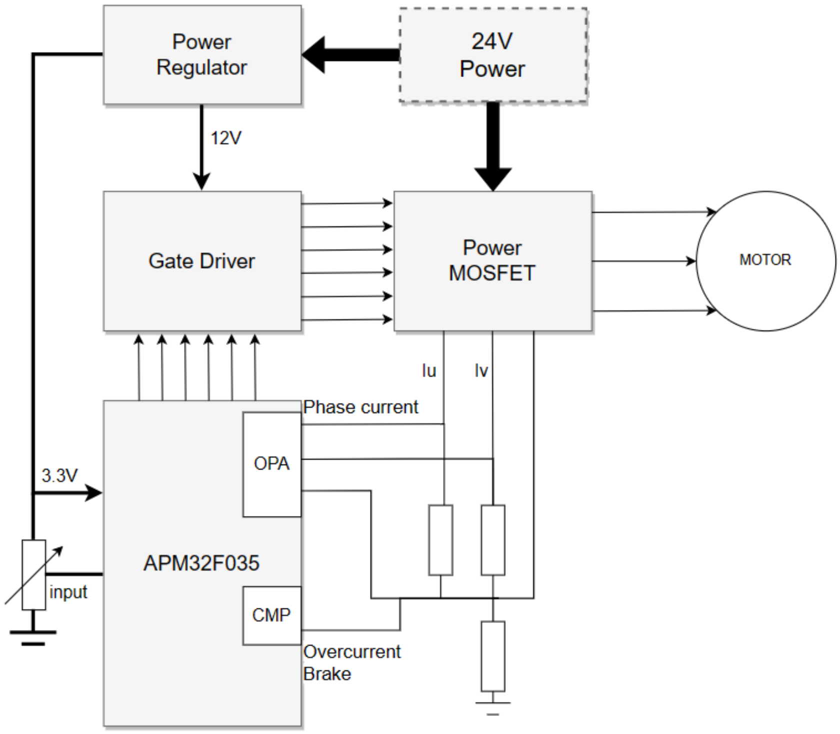
2.1 Overall Hardware Circuit

The overall hardware system is powered by an external 24V power supply and after conversion through the corresponding power step-down circuit, it outputs stable 12V, 5V, and 3.3V voltages. The 12V voltage is output to the Gate driver IC, the 3.3V voltage is output to the APM32F035 series microprocessor, and the power switch tube is directly connected to the 24V power supply. At the same time, this scheme uses a variable resistance knob to adjust the voltage input of 0~3.3V as the input end of the speed command, to adjust the motor speed. Users can directly adjust the input voltage by turning the variable resistor knob in actual use. When the input voltage value exceeds the starting threshold, the motor will start running, and when the voltage value is below the threshold, the motor will stop running.

After the motor is started, the APM32F035 processor can obtain the phase currents Iu, Iv, and Iw of three phases through the built-in operational amplifier and corresponding sampling circuit, and convert this data through the coordinate axis to control the torque current and phase of the motor. After the FOC control calculation link, adjust the TMR1 peripheral to output the corresponding three-way complementary PWM waves to control the switching components of the inverter.

The hardware block diagram is shown in the figure.

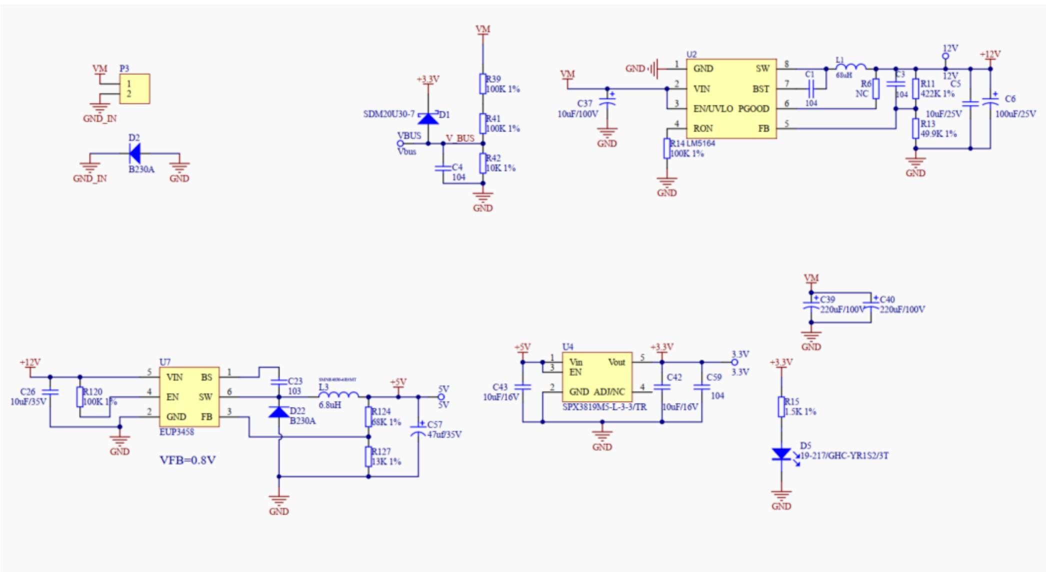
Figure 1 Hardware System Block Diagram



2.2 Interface Circuits and Settings

2.2.1 Power circuit

Figure 2 Power Circuit



As shown in the figure, supply voltage V_BUS =VM/((100K+100K+10K)/10K)=VM/21

A 12-bit ADC is adopted, and the sampling range 0-3.3V corresponds to 0-4096

Then the maximum sampling voltage corresponding to 3.3V is: VM= 3.3 *21 =69.3V

2.2.2 Phase Current Sampling Circuit

Figure 3 MOSFET Circuit

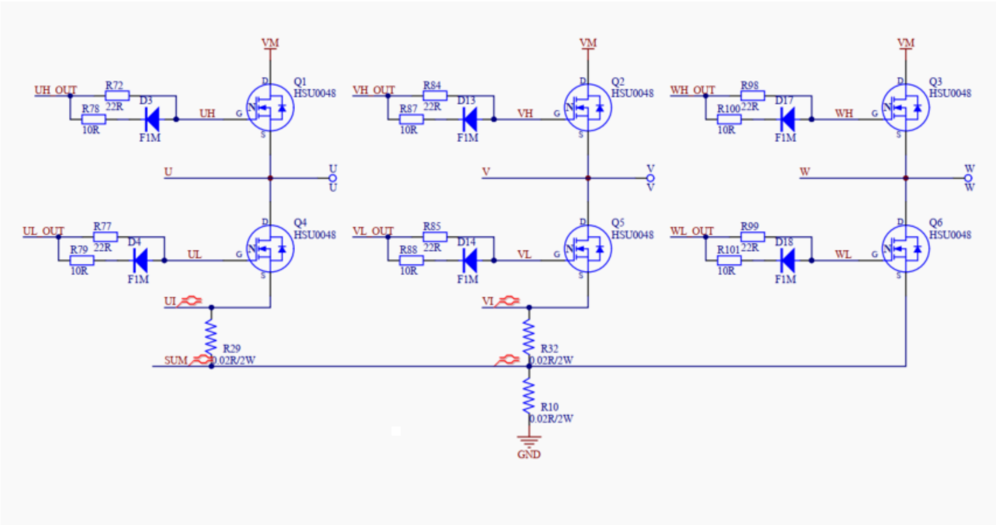
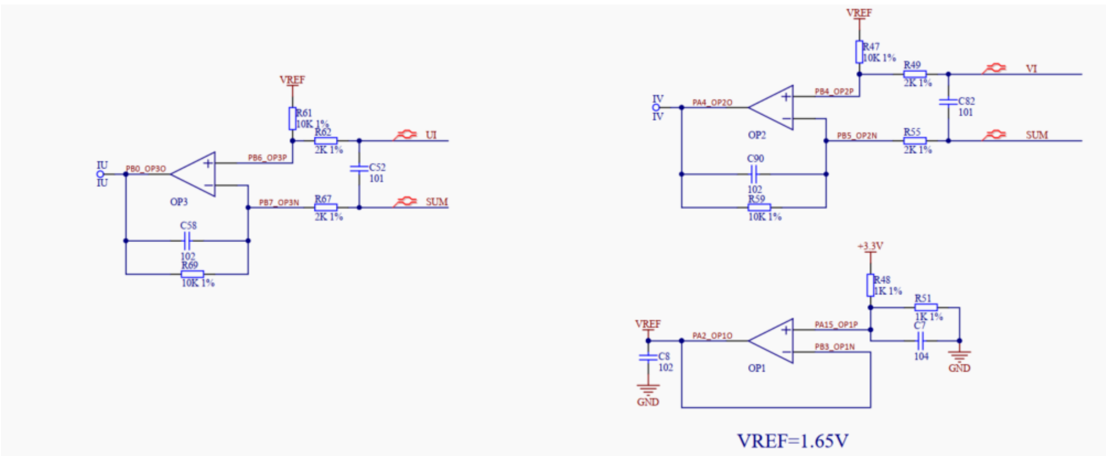


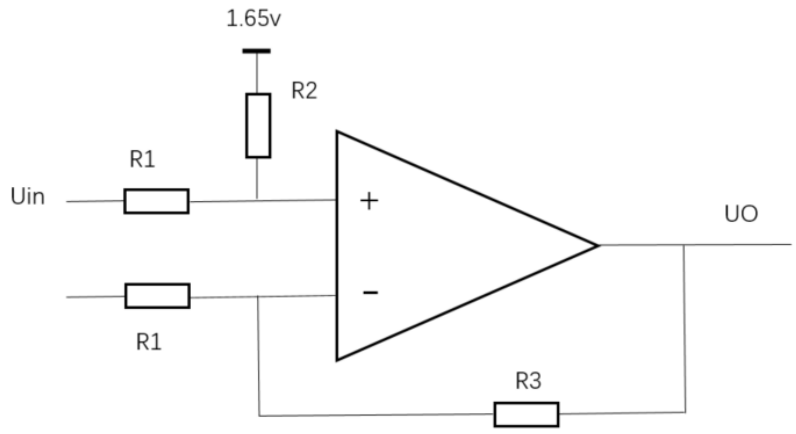
Figure 4 Current Sampling Circuit



As shown in the figure, IU= UI*4.86+1.60

Where 4.86 is the amplification factor of the operational amplifier and 1.6 is the bias voltage. The derivation process is as follows:

Figure 5 Amplifier circuit diagram



R3 is a 10K feedback resistor coupled with an internal 294K resistor, resulting in a combined resistance of 294K.

According to the virtual short concept, the equation for the positive terminal can be written as:

$$\frac{1.65 - U+}{10K} = \frac{U+ - Uin}{2K}$$, which means: $$\frac{1.65 - U+}{R2} = \frac{U+ - Uin}{R1}$$

Similarly, the negative terminal can be formulated as follows:

$$\frac{UO - U-}{9.671K} = \frac{U- - 0}{2K}$$, which means: $$\frac{UO - U-}{R3} = \frac{U- - 0}{R1}$$

Based on the virtual short: U+ = U-, the final equation can be obtained: 1.604 + 4.86Uin = Uo

Where 1.604 is the bias voltage and 4.86 is the amplification factor.

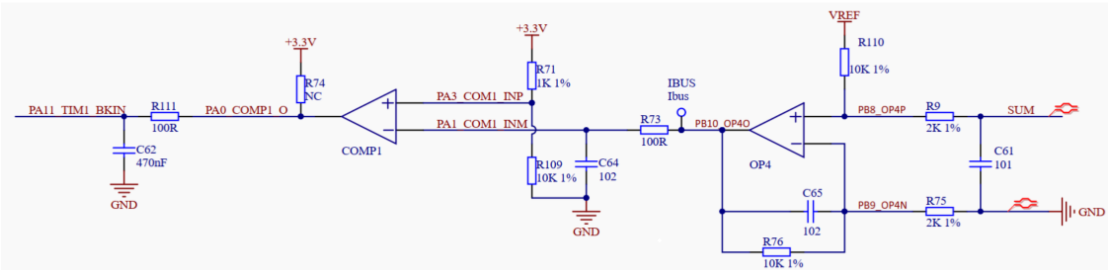
A 12-bit ADC is adopted, and the sampling range 0-3.3V corresponds to 0-4096

As shown in Figure 3, when the sampling resistance is selected as 0.02R,

the maximum peak-to-peak current corresponding to 3.3V is 1.6/4.86/0.02=16.46A.

2.2.3 Overcurrent protection circuit

Figure 6 Overcurrent Protection Circuit



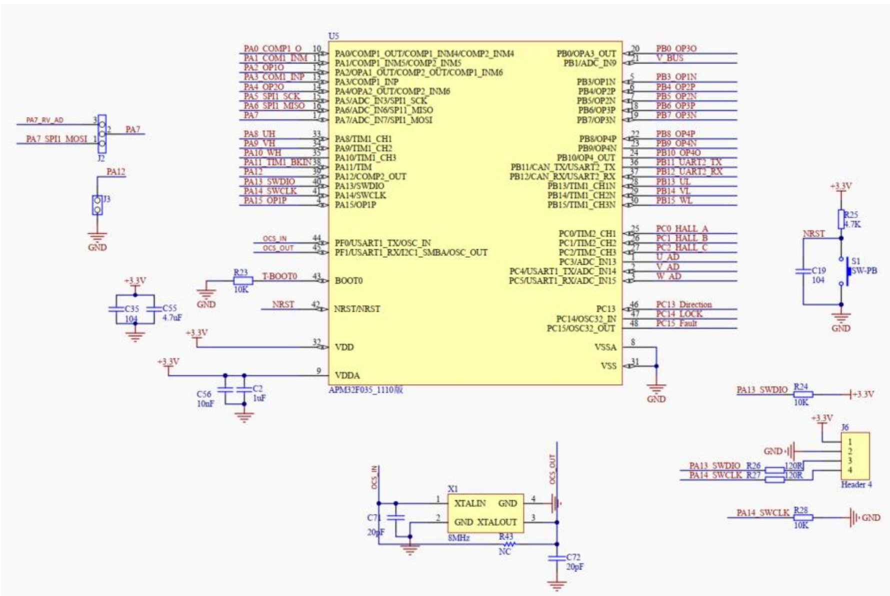
As shown in the figure, a built-in operational amplifier OPA4 is used to sample the bus current. A 12-bit ADC is adopted with a sampling range of 0-3.3V corresponding to 0-4096. From Figure 2-3, it can be seen that the sampling resistance is 0.02R;

The output end of OPA4 is used as the reverse input end of COMP1, and resistance voltage division is adopted at the forward input end. Through simple calculation, it can be concluded that the input is 3V;

Then the maximum current corresponding to 3V is (3-1.65)/5/0.02=13.5A.

2.2.4 Minimum system circuit

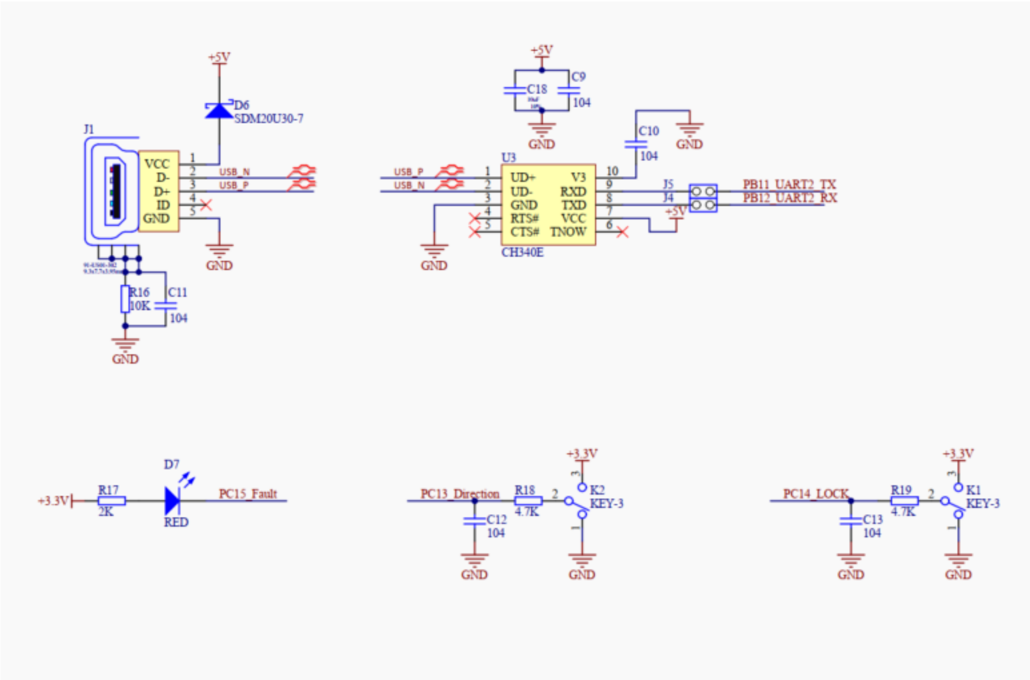
Figure 7 Minimum System Circuit



As shown in the figure, the utilization of APM32F035 MOTOR EVAL V1.0 board hardware interface resources is described in the above figure. The external crystal oscillator input of HSE is 8MHz, and the SWD burning interface is adopted for burning.

2.2.5 Communication Interface and Button Circuit

Figure 8 Communication Interface and Button Circuit



As shown in the figure, a USB-to-serial port and a fault indicator light are reserved in the APM32F035 MOTOR EVAL V1.0 board hardware for debugging by developers; the two buttons are responsible for implementing the functions of controlling the running direction of the motor and locking.

2.3 Physical System Hardware

The picture of the system is shown in the figure, and it mainly includes the following six interfaces:

(1) Power input interface (connect to 24V; pay attention to positive and negative poles)

(2) Three-phase motor interface (phase sequence only affects the direction of rotation)

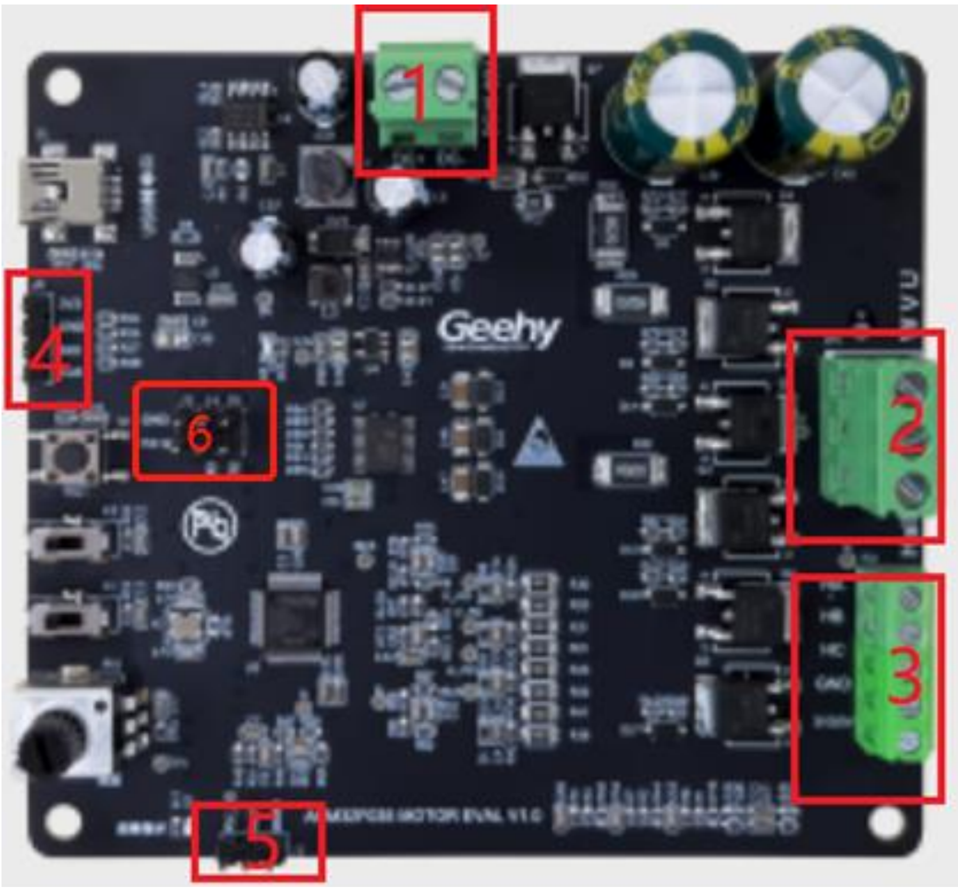
(3) HALL input interface

(4) SWD debugging interface

(5) The jumper cap on the J2 port should be connected to PA7 (connected to PA7_RV_AD) to read the AD value of the potentiometer.

(6) The jumper caps on the J4 and J5 ports should be connected for serial communication, linking to the host computer.

Figure 9 Hardware Picture



3 Software Introduction

3.1 Overall Program Architecture

The overall code architecture of this project can be divided into four layers: user layer, peripheral driver layer, motor control driver layer, and motor algorithm layer. The specific functional descriptions are as follows:

3.1.1 USER Layer

main.c: The main function entry is responsible for switching motor initialization parameters, underlying peripherals, interrupt priority, while cycle, and low-speed state machine loop;

apm32f035_int.c: All interrupt handling functions, mainly including TMR1 interrupt function and ADC interrupt handler function;

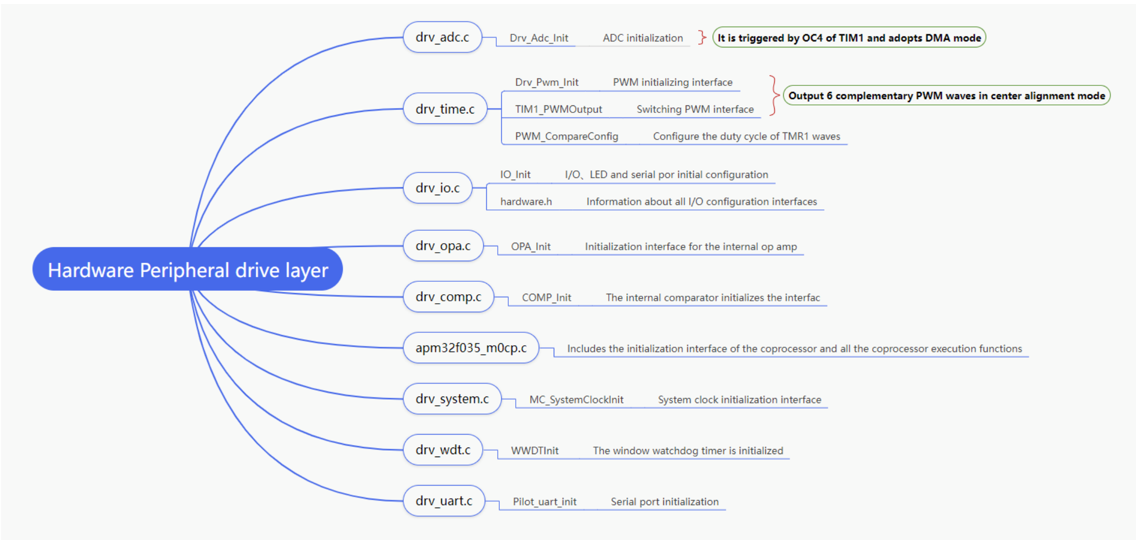
user_function.c: Includes initialization configuration, parameter reset, and other handler functions of motor parameters;

parameter.h: Includes all required configuration parameter information;

3.1.2 Peripheral Driver Layer (HARDWARE Layer)

The peripheral driver layer is mainly responsible for the peripheral driver functions and configuration of the APM32F035 chip, mainly including GPIO, PWM, ADC, OPA, COMP, and M0CP coprocessors, as shown in the following figure.

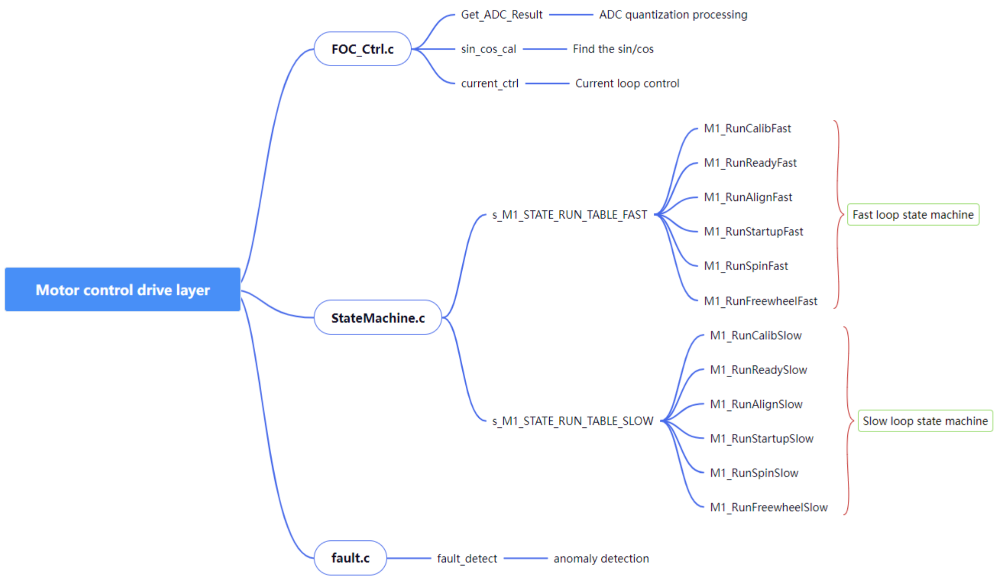
Figure 10 Peripheral Driver Layer



3.1.3 Motor Control Drive Layer (MOTOR_CONTROL Layer)

The motor control driver layer is mainly responsible for the control run logic and core processing algorithm call of the motor, as shown in the following figure.

Figure 11 Motor Control Driver Layer



3.1.4 Geehy Motor Algorithm Layer (Geehy_MCLIB Layer)

The motor algorithm layer includes coordinate transformation, vector control, and other related functions, as well as math libraries, sliding-mode observers, and other library functions.

3.2 Introduction to State Machine

In this case, the structure of embedding the sub-state machine into the main state machine is adopted, as shown below:

Four main states: INIT, STOP, FAIL, and RUN;

The six RUN sub-states of the main state are **run calib, run-ready, run-align, run-startup, run-spin, and run-freewheel**.

The main state machine is described below:

Fault: When an error occurs in the system, it will remain in this state until the error flag bit is cleared;

Then after delay for a while, it will jump from the Fault state to the STOP state and wait for the start command.

Init: This main state executes variable initialization.

Stop: The system waits for the speed command after completing initialization. In this state, the PWM output is turned off.

Run: In the running state, if a Stop command is issued, the system will stop running.

When the system is running in the Run state, its sub-states will be called and executed.

Run-Calib: The current biased ADC self-calibration function can be executed. After this state is executed, the system will switch to the Ready state and disable the PWM output.

Ready: Enable PWM output, synchronously sample the current, and conduct abnormal state inspection.

Align: Execute sampling current, call the pre-positioning algorithm, and update the PWM. Execute within the specified time, and the system will switch to the Startup sub-state and sample the DC bus voltage for filtering.

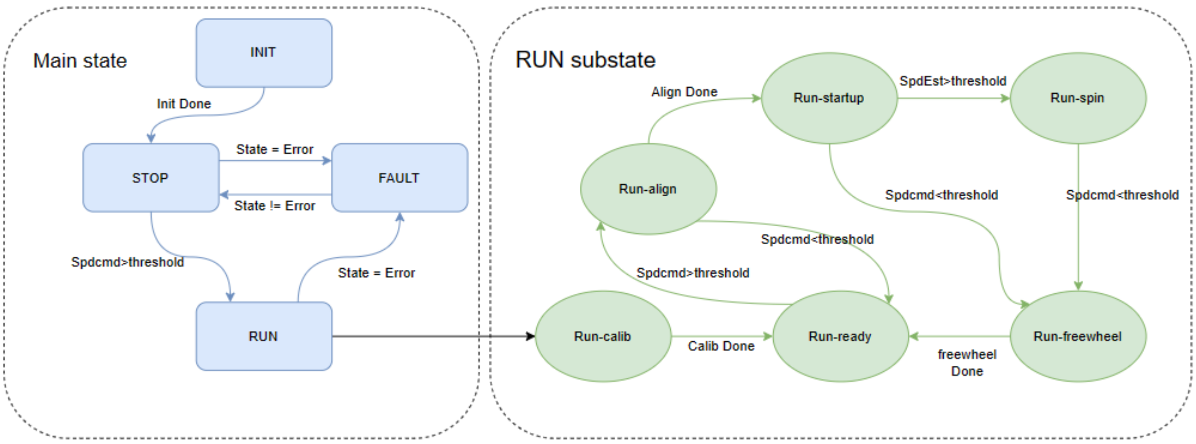
Startup: Sample the current, use an open-loop starting motor, and call the observer to estimate the rotor speed and position, call the corresponding algorithm, and update the PWM. If the motor is started successfully, the system will enter the spin sub-state and sample the DC bus voltage for filtering.

Spin: Sample the current, call the observer to estimate the rotor speed and position, call the corresponding algorithm, update the PWM, and the motor starts to switch to closed-loop operation.

Freewheel: Enable PWM output and stop the machine by shortening the brake. Due to rotor inertia, the state can be switched only after the motor stops running and is further switched to the Ready state. If an error occurs, the system will enter the Fault state.

To sum up, the state machine flowchart of the system is shown in the figure below.

Figure 12 State Machine Flowchart



3.3 Top-layer Peripheral Configuration

3.3.1 PWM Output Configuration

void Drv_Pwm_Init(uint16_t u16_Period,uint16_t u16_DeadTime)

(1) The general configuration of PWM is as follows:

Set the PWM clock frequency division to 1, select the center-aligned mode 2, and set the repeat counter to 1, as shown in the figure below.

Figure 13 General Configuration of PWM

```
    /* Time Base configuration ,init timel freq*/
    TIM_TimeBaseInitStructure.period            = u16_Period;
    TIM_TimeBaseInitStructure.div               = 0;
    TIM_TimeBaseInitStructure.counterMode       = TMR_COUNTER_MODE_CENTERALIGNED2;
    TIM_TimeBaseInitStructure.clockDivision     = TMR_CKD_DIV1;
    TIM_TimeBaseInitStructure.repetitionCounter = 1;
    TMR_ConfigTimeBase(TMR1, &TIM_TimeBaseInitStructure);
```

Figure 14 Center-aligned Mode Selection

Center Aligned Mode Select

In the Center-aligned mode, the counter counts up and down alternately; otherwise, it will only count up or down. Different Center-aligned modes affect the timing of setting the output comparison interrupt flag bit of the output channel to 1; when the counter is disabled (CNTEN=0), select the Center-aligned mode.

00: Edge alignment mode

01: Center-aligned mode 1 (the output comparison interrupt flag bit of output channel is set to 1 when counting down)

10: Center-aligned mode 2 (the output comparison interrupt flag bit of output channel is set to 1 when counting up)

11: Center-aligned mode 3 (the output comparison interrupt flag bit of output channel is set to 1 when counting up/down)

(2) PWM Output Status Configuration

Set the output status of the upper and lower tubes of PWM and enable the configuration of PWM output of the upper and lower tubes to be effective,

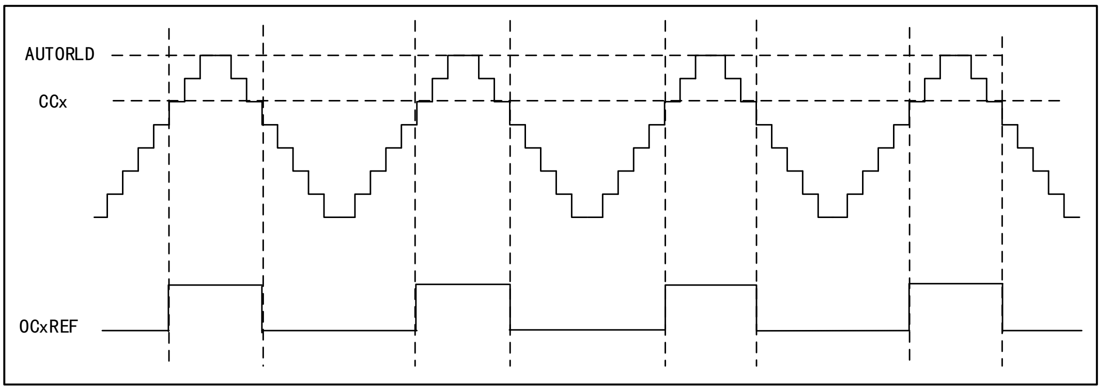
configure the enabled brakes, configure the brake polarity, disable the automatic output, and prevent automatic PWM output in the next update event.

Figure 15 PWM Output Status Configuration

```
   /* Automatic Output enable, Break, dead time and lock configuration*/
   TIM_BDTRInitStructure.RMOS_State       = TMR_RMOS_STATE_ENABLE;//--------
   TIM_BDTRInitStructure.IMOS_State       = TMR_IMOS_STATE_ENABLE;//--------
   TIM_BDTRInitStructure.lockLevel        = TMR_LOCK_LEVEL_OFF;//00: Lock off, register no write protection; 01: Lock level 1, can not write TIMx_BDTR register DTG, BKE, BKP, AOE bit and TIMx_CR2 register OISx/OISxN bit;
   TIM_BDTRInitStructure.deadTime         = u16_DeadTime;//dead-time
   /**
    * Brake configuration: enable brake
    * Brake input polarity: active in low level
    * Auto output enable configuration: Disable MOE bit hardware control
    */
   TIM_BDTRInitStructure.breakState       = TMR_BREAK_STATE_ENABLE;//TMR_BREAK_STATE_ENABLE TMR_BREAK_STATE_DISABLE;
   TIM_BDTRInitStructure.breakPolarity    = TMR_BREAK_POLARITY_HIGH;
   TIM_BDTRInitStructure.automaticOutput  = TMR_AUTOMATIC_OUTPUT_DISABLE;
   TMR_ConfigBDT(TMR1, &TIM_BDTRInitStructure);

   /*pwm driver set, channel 1,2,3,4set pwm mode*/
   TIM_OCInitStructure.OC_Mode         = TMR_OC_MODE_PWM2;
   TIM_OCInitStructure.OC_OutputState  = TMR_OUTPUT_STATE_ENABLE;  //TMR_OUTPUT_STATE_DISABLE;
   TIM_OCInitStructure.OC_OutputNState = TMR_OUTPUT_NSTATE_ENABLE; //TMR_OUTPUT_NSTATE_DISABLE;//------------
   TIM_OCInitStructure.Pulse           = u16_Period;
   TIM_OCInitStructure.OC_Polarity     = TMR_OC_POLARITY_HIGH;
   TIM_OCInitStructure.OC_NPolarity    = TMR_OC_NPOLARITY_HIGH; //Complementary output polarity-------
   TIM_OCInitStructure.OC_Idlestate    = TMR_OCIDLESTATE_RESET;  // TMR_OCIDLESTATE_SET; //
   TIM_OCInitStructure.OC_NIdlestate   = TMR_OCNIDLESTATE_RESET; // TMR_OCNIDLESTATE_SET;//
```

Figure 16 Timing Diagram of PWM2 Center-aligned Mode



In count-up mode, when TMR1_CNT<TMR1_CCR1, Channel 1 is invalid level; otherwise it is valid level;

In count-down mode, when TMR1_CNT>TMR1_CCR1, Channel 1 is a valid level; otherwise it is an invalid level.

3.3.2 ADC Configuration

void Drv_Adc_Init(void)

1. ADC underlying configuration

DMA mode is adopted, and the quantized data of ADC is directly transported to the ADC_ConvertedValue array for storage. The ADC trigger condition uses CC4 of TMR1 as the trigger source, to enable ADC and configure ADC interrupt priority and its enable. Details are shown below:

Figure 17 ADC Underlying Configuration

```
void Drv_Adc_Init(void)
{
    ADC_Config_T   ADC_InitStructure;
    DMA_Config_T   DMA_InitStructure;
    DMA_InitStructure.peripheralAddress  = (uint32_t)&(ADC->DATA);//ADC address
    DMA_InitStructure.memoryAddress      = (uint32_t)&ADC_ConvertedValue[0]; //memory address
    DMA_InitStructure.direction          = DMA_DIR_PERIPHERAL; //Direction (from peripherals to memory)
    DMA_InitStructure.bufferSize         = TOTAL_CHANNEL;//TOTAL_CHANNEL; //The size of the transferred content -- the number of transfers
    DMA_InitStructure.peripheralInc      = DMA_PERIPHERAL_INC_DISABLE; //The peripheral address is fixed
    DMA_InitStructure.memoryInc          = DMA_MEMORY_INC_ENABLE;//DMA_MEMORY_INC_ENABLE;
    DMA_InitStructure.peripheralDataSize = DMA_PERIPHERAL_DATASIZE_HALFWORD ; //Peripheral data unit
    DMA_InitStructure.memoryDataSize     = DMA_MEMORY_DATASIZE_HALFWORD ;    //Memory data unit
    DMA_InitStructure.circular           = DMA_CIRCULAR_ENABLE  ; //DMA mode: cyclic transmission
    DMA_InitStructure.priority           = DMA_PRIORITY_LEVEL_VERYHIGH ; //Priority: High
    DMA_InitStructure.memoryTomemory     = DMA_M2M_DISABLE;   //Memory to memory transmission is disabled
    ADC_Reset();
    DMA_Config(DMA_CHANNEL_1, &DMA_InitStructure);  //Configure channel 1 for DMA
    DMA_Enable(DMA_CHANNEL_1);
    ADC_ClockMode(ADC_CLOCK_MODE_ASYNCLK);//48M/4=12mADC_CLOCK_MODE_SYNCLKDIV4
    ADC_ConfigStructInit(&ADC_InitStructure);
    ADC_InitStructure.convMode    = ADC_CONVERSION_SINGLE;
    ADC_InitStructure.scanDir     = ADC_SCAN_DIR_UPWARD;
    ADC_InitStructure.extTrigConv1 = ADC_EXT_TRIG_CONV_TRG1;  //  timer1 CC4
    ADC_InitStructure.extTrigEdge1 = ADC_EXT_TRIG_EDGE_RISING;
    ADC_InitStructure.dataAlign   = ADC_DATA_ALIGN_RIGHT;
    ADC_InitStructure.resolution  = ADC_RESOLUTION_12B;
    ADC_Config(&ADC_InitStructure);
    ADC_ConfigChannel(ADC_CHANNEL_2 | ADC_CHANNEL_8 | ADC_CHANNEL_9 | ADC_CHANNEL_7 ,ADC_SAMPLE_TIME_1_5);
    ADC->CFG1_B.OVRMAG = 1;
    ADC_EnableInterrupt(ADC_INT_CS);
//========================ADC interrupt=============================================
    NVIC_EnableIRQ(ADC_COMP_IRQn);
    NVIC_SetPriority(ADC_COMP_IRQn,0);
    ADC_DMARequestMode(ADC_DMA_MODE_CIRCULAR);
    ADC_EnableDMA();
    ADC_Enable();
    ADC_StartConversion();//Gotta start it up
```

3.3.3 OPA and COMP Underlying Configuration

1. OPA underlying configuration

To configure the underlying configuration of OPA, first configure the OPA pin, DISABLE the operational amplifier OPA, configure to use an external resistor network, and then ENABLE it, as shown in the figure below;

Figure 18 OPA Underlying Configuration

```
void OPA_Init(void)
{
    OPA_Disable(OPA1);
    OPA_Disable(OPA2);
    OPA_Disable(OPA3);
    OPA_Disable(OPA4);
    OPA_SelectGainFactor(OPA1,OPA_GAIN_FACTOR_0);
    OPA_SelectGainFactor(OPA2,OPA_GAIN_FACTOR_0);
    OPA_SelectGainFactor(OPA3,OPA_GAIN_FACTOR_0);
    OPA_SelectGainFactor(OPA4,OPA_GAIN_FACTOR_0);
    OPA_Enable(OPA1);
    OPA_Enable(OPA2);
    OPA_Enable(OPA3);
    OPA_Enable(OPA4);
}
```

1. COMP underlying configuration

COMP is used for overcurrent anomaly detection. To configure the underlying configuration of COMP, first configure the COMP pin. The external connection method used on the board to connect COMP is the output and TMR1's BKIN. Set the output reverse, and trigger the BKIN of

TMR1 at a low level, as shown in the following figure;

Figure 19 COMP Underlying Configuration

```
void COMP_Init(void)
{
    COMP_Config_T   compConfig;
    /*  Configure COMP1 */
    COMP_ConfigStructInit(&compConfig);
    compConfig.invertingInput = COMP_INVERTING_INPUT_PA1;
    compConfig.output         = COMP_OUTPUT_NONE;
    compConfig.outputPol      = COMP_OUTPUTPOL_INVERTED;
    compConfig.hysterrsis     = COMP_HYSTERRSIS_NO;
    compConfig.mode           = COMP_MODE_HIGHSPEED;
    COMP_Config(COMP_SELECT_COMP1,&compConfig);
    /* Enable COMP2 */
    COMP_Enable(COMP_SELECT_COMP1);
```

3.4 Calibration Standardization

3.4.1 Concept of Per Unit

There are typically two methods to define the magnitude of a variable:

Firstly, the nominal value: the value obtained from measuring the variable using instruments and meters. This employs the International System of Units (SI), voltage (V), current (A), and rotational speed (r/min).

Secondly, the per unit value: the nominal value divided by the reference value. This uses the per-unit system (p.u).

Advantages of the per-unit system include compatibility with fixed-point MCU operations and prevention of data overflow, among others.

3.4.2 Per Unitization of the Entire System Software

All are calibrated to the Q15 format.

Firstly, voltage calibration: based on the actual maximum measurable bus voltage as the reference value, it is mapped to a voltage range of 0-3.3V at the actual ADC pin. The corresponding mapping in the program is 0-32767, as shown in the figure below:

Figure 20

```
/*Voltage Sampling*/
#define UDC_MAX                         (69.0f)      //  unit:V       Max DC Voltage of Hardware
#define DCBUS_OVER                      Q15(48.0f/UDC_MAX)      //  unit:V
#define DCBUS_UNDER                     Q15(20.0f/UDC_MAX)      //  unit:V
```

Secondly, current calibration: based on the actual maximum measurable bus current as the reference value, the voltage mapped to the actual ADC pin is 0-3.3V, and the corresponding mapping in the program is 0-32767

Figure 21

```
/*Current Sampling*/
#define ADC_REFV                SYS_REFV
#define R_SHUNT                 (0.02f)    //  unit:ohm
#define CURRENT_OPA_GAIN        (4.86f)    //  unit:ohm Using the combination of 2K/10K resistors, combined with the internal coupling resistance calculation
#define I_MAX                   (16.46f)   //  Offset 1.6V, I MAX=1.6/Gain/R
/*  I/Ibase *Rs*OP_Gain = (V-Voffset)/32768 * 3.3     On the right side is the actual voltage
    IGAIN = 3.3/OP_Gain/Rs/Ibase                      The bias has been removed
*/
#define IGAIN_Q10                               (2113)
```

Thirdly, angle calibration: 0~ Π= 0- 32767

Fourthly, speed calibration: based on the actual rated speed, select the appropriate speed reference value, and the corresponding mapping in the program is 0-32767

Figure 22

```
#define SPEED_CALIBRATION             (5000.0f)    // unit:rpm    rated speed of motor
#define MAX_SPEED                     (MAX_RPM)    // unit:rpm     max speed of motor
#define OVER_SPEED_VALUE              Q15(3500.0f/SPEED_CALIBRATION)
```

3.5 Settings of Key Parameters

All parameters in this system are configured in parameter.h of the user layer, mainly including system parameters, baseboard-related parameters, state machine-related parameters, motor-related parameters, etc, as follows:

3.5.1 System Parameters

Table 3 System Parameters

Parameter name	Parameter description	Set value
SYS_REFV	Supply voltage of the system	3.3 (V)
SYSCLK_HSE_72MHz	Main frequency of the system	72000000 (Hz)
PWMFREQ	PWM frequency	15000 (Hz)
DEAD_TIME	PWM dead time	1.0 (μs)
SLOWLOOP_FREQ	Control frequency of slow loop	1000 (Hz)

（1） The reference voltage of F035 is set to 3.3V, and the main frequency is 72M. According to the parameter configuration of the F035 chip, the SYS_REFV and SYSCLK_HSE_72MHz parameters should remain at their default values.

（2） PWMFREQ and PWM_PERIOD: The default program uses a frequency of 15K, which is sufficient.

（3） DEAD_TIME: The dead time is set to a default value of 1μs.

（4） SLOWLOOP_FREQ: The default value for the slow loop frequency is 1KHz.

3.5.2 Backplane Hardware Parameters

Table 4 Parameters of Backplane Hardware

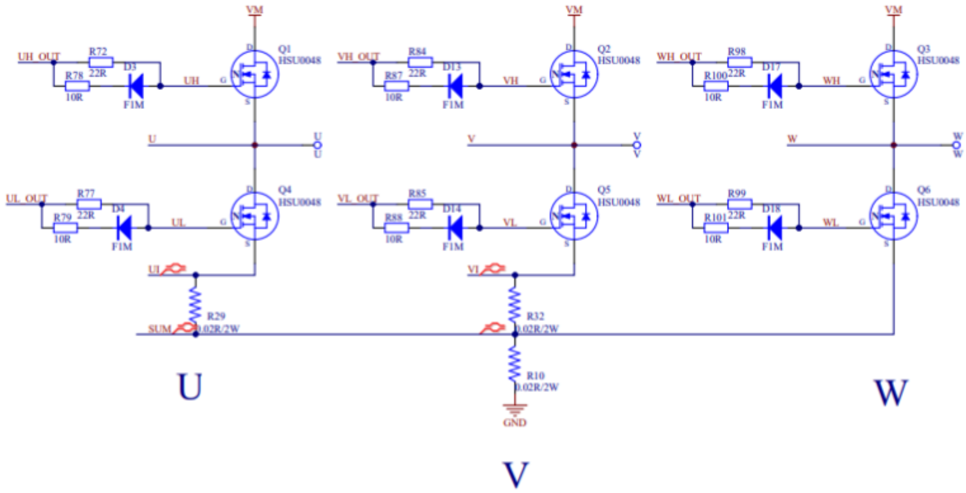
Parameter name	Parameter description	Set value
ADC_REFV	ADC reference voltage	3.3 (V)
R_SHUNT	Sampling resistance value	0.02 (Ω)
CURRENT_OPA_GAIN	Amplification factor of operational amplifier	4.86
I_MAX	Current Per Unit Base Value	16.46 (A)
IGAIN_Q10	Conversion factor of current	2113
UDC_MAX	Voltage Per Unit Base Value	69.0 (V)
DCBUS_OVER	Overvoltage threshold	48.0 (V)
DCBUS_UNDER	Undervoltage threshold	20.0 (V)

If these parameters do not match the bottom plate, it will affect the collection of phase current and bus voltage. It may cause faults such as overvoltage, low voltage, and overcurrent to occur during program operation.

（1） ADC_ REFV: reference voltage for ADC sampling, maintain the default

（2） R_ SHUNT: sampling resistance, fill in the sampling resistance value measured on the actual board; The sampling resistor of the dual resistance sampling scheme is located below the Mos of the lower bridge arm, and the sampling resistor of the F035 development board is shown in the figure:

Figure 23



（1） Current_ OPA_ GAIN: amplification factor of the operational amplifier. The actual amplification factor of the F035 internal operational amplifier after coupling is 4.86. (Refer to 2.2.2 Phase Current Sampling Circuit for a detailed calculation process)

（2） I_ MAX and IGAIN_ Q10: I_ MAX is the reference value for system current, I_ MAX=1.6/Gain/Rs, 1.6 is the bias voltage. RS and Gain calculate the reference value of the current based on the measurements above. (Please refer to the description of the 2.2. b phase current sampling circuit). IGAIN_ Q10 is a coefficient that converts the collected voltage into current, which is related to the sampling resistance, operational amplifier multiple, reference voltage, and reference current. It is represented in Q10 format. IGAIN=3.3/OP_ Gain/Rs/Ibase, shift IGAIN ten places to the left, multiply by 1024 to obtain IGAIN_ Q10.

（3） UDC_ MAX: the maximum voltage value that the system can collect, and the bus voltage is collected by dividing the voltage by the bus resistance. (The specific calculation process can be found in 2.2.1 Power Circuit)

（4） DCBUS_ Over and DCBUS_ INDER: the threshold for overvoltage and undervoltage is modified based on the actual operating voltage of the motor.

3.5.3 Parameters of State Machine

Table 5 Parameters of State Machine

Parameter name	Parameter description	Set value
PRECHARGE_TIME	Bootstrap capacitor pre-charging time	40(pwm cycles)
STOP_TO_RUN_SPEED	Speed command threshold for jumping from Stop to Run state	450(rpm)

Parameter name	Parameter description	Set value
STARTUP_TO_SPIN_SPEED	Jumping from Startup to Spin State Actual Speed Threshold	600(rpm)
FREEWHEEL_SPEED	Stop the machine after the speed command is below the threshold	400 (rpm)
ID_ALIGN	Predetermined current size	1.0 (A)
CUR_DEC	The slope value of current decrease	0.5(A/s)
CUR_INC	The slope value of the current rise	0.5(A/s)
STARTUP_SPD_INC	Slope value of speed increase under an open loop	300 (rpm/s)
SPEED_TO_THETA	Integral coefficient for converting speed to angle	Refer to the program for detailed values
SPIN_SPD_INC	Slope value of speed increase under closed-loop	1000(rpm/s)
SPD_DEC	Slope value of speed increase under closed-loop	1000(rpm/s)
FREEWHEEL_TIME	Freewheel time	1.01(s)
FAULTRELEASE_TIME	The time for resetting after a malfunction occurs	3.01(s)

（1） PRECHARGE_ TIME: the precharge time for the bootstrap capacitor; it can be kept at the default setting.

（2） STOP_ TO_ RUN_ Speed: the speed threshold from the stop state to the run state

（3） STARTUP_ TO_ SPIN_ Speed: the speed threshold for switching from startup to spin state, that is, the speed threshold for switching to closed-loop state.

（4） FREEWHEEL_ Speed: the speed threshold for switching from run to Freewheel

（5） ID_ ALIGN: set the current size to ensure that the motor can be dragged normally. If the startup fails or steps out, the current value can be increased appropriately.

（6） CUR_ DEC and CUR_ INC: the slope value of the increase or decrease in current value, in units of A/S

（7） STARTUP_ SPD_ INC: open loop acceleration. For heavy loads, increase the open loop current and reduce the open loop acceleration to ensure smooth and smooth starting. For light loads, increase the open-loop acceleration and switch to closed-loop operation as soon as possible.

（8） SPEED_ TO_ THETA: the integration coefficient for converting speed to angle, just keep it as default.

（9） SPIN_ SPD_ INC and SPD_ DEC: closed loop acceleration and deceleration, affecting the speed response speed

（10） FREEWHEEL_ TIME: time of freewheel status

（11） FaulTRELEASE_ TIME: the time to reset after a fault occurs

3.5.4 Motor Related Parameters

Table 6 Motor-Related Parameters

Parameter name	Parameter description	Set value
Rs	Motor phase resistance	0.15 (ohm)
Ls	Motor phase inductance	0.00037 (H)
POLEPAIRS	Number of motor poles	2 (unit)
SPEED_CALIBRATION	Reference value of speed	5000 (rpm)
MAX_SPEED	Given maximum speed value	3000(rpm)
OVER_SPEED_VALUE	Overspeed threshold	3500(rpm)
MAX_DUTY	Maximum duty cycle	0.95(unit)
SPD_PI_LIMIT	Output limit value of speed PI	4.6(A)

（1） RS: the measurement of phase resistance can be roughly measured using a multimeter or using a bridge. The resistance value obtained by connecting any two phases of the motor to both ends of the multimeter is half of the phase resistance.

（2） LS: the phase inductance is often measured using a bridge, and the measured frequency is based on the frequency used by PWM. If any two phases of the motor are connected at either end of the bridge, half of the inductance value obtained is the phase inductance; You can rotate one angle at a time and take multiple measurements to calculate the average value.

（3） POLEPAIR: the measurement of the number of poles can be done by connecting the oscilloscope terminals and ground to any two phases of a three-phase motor, rotating the motor by hand once, and producing several sine waves, which are several pairs of stages.

（4） SPEED_ CALIBRITION: the reference value of speed, measured in RPM.

（5） MAX_ Speed: the maximum speed of the motor, set according to the motor specifications.

（6） Over_ SPEED_ Value: software overcurrent threshold setting. When the sampled phase current exceeds the set value, the system reports a software overcurrent fault.

（7） MAX_ DUTY: maximum duty cycle, keep default.

（8） SPD_ PI_ Limit: the output limit of the speed PI.

3.5.5 Other parameters

Table 7 PI Parameters

Parameter name	Parameter description	Set value
M1_IQ_KP_Q15	Q-axis current loop KP parameter Q15 format	4000
M1_IQ_KP_Q10	Q-axis current loop KP parameter Q10 format	0
M1_IQ_KI_Q15	Q-axis current loop KI parameter Q15 format	400
M1_IQ_KI_Q10	Q-axis current loop KI parameter Q10 format	0
M1_ID_KP_Q15	D-axis current loop KP parameter is in Q15 format	4000
M1_ID_KP_Q10	D-axis current loop KP parameter is in Q10 format	0
M1_ID_KI_Q15	D-axis current loop KI parameter Q15 format	400
M1_ID_KI_Q10	D-axis current loop KI parameter Q10 format	0
M1_SPEED_KP_Q15	The velocity ring KP parameter is in Q15 format	16384
M1_SPEED_KP_Q10	The velocity ring KP parameter is in Q10 format	512
M1_SPEED_KI_Q15	Speed loop KI parameter Q15 format	163
M1_SPEED_KI_Q10	Speed loop KI parameter Q10 format	9
SMO_ERR_MAX	Observer error maximum Q15	0.5
SLIDE_GIAN	Observer gain value	0.5
M1_PLL_KP_Q15	The KP parameter of PLL is in Q15 format	0
M1_PLL_KP_Q10	The KP parameter of PLL is in Q10 format	2560
M1_PLL_KI_Q15	The PLL KI parameter is in Q15 format	0
M1_PLL_KI_Q10	The PLL KI parameter is in Q10 format	20

The PI parameters of the current loop, position loop, observer, and PLL have been adjusted in the program and do not need to be modified.

Table 8 Potentiometer-Related Parameters

Parameter name	Parameter description	Set value
VSP_MIN_VOL	Minimum voltage collected by potentiometer	0.5
VSP_MAX_VOL	Maximum voltage collected by potentiometer	2.5
VSP_MIN_AD	Minimum AD value collected by potentiometer	Refer to the program for detailed values
VSP_MAX_AD	Maximum AD value collected by potentiometer	Refer to the program for detailed values
MIN_RPM	Minimum speed of potentiometer regulation	50
MAX_RPM	Maximum speed of potentiometer regulation	3000

MAX_RPM and MIN_RPM are configured based on the motor speed. Retain the default values for other parameters.

Table 9 Potentiometer-Related Parameters

Parameter name	Parameter description	Set value
OC_TIME	Overcurrent fault filtering	10
UV_TIME	Undervoltage fault filtering	10
OV_TIME	Overvoltage fault filtering	10
OS_TIME	Overspeed fault filtering	10
PLLERR_TIME	Startup failure filter	10

When a fault is detected continuously, the system enters the corresponding fault state.

Table 10 Potentiometer-Related Parameters

Parameter name	Parameter description	Set value
SPD_SOFT_OR_KNOB	Speed control mode	0 upper computer, 1 knob

Table 11 Potentiometer-Related Parameters

Parameter name	Parameter description	Set value
BIAS_OVER	Maximum bias voltage	1.7(V)
BIAS_UNDER	Minimum bias voltage	1.55 (V)

4 Debugging Steps

（1） Measure the relevant parameters of the motor, including phase resistance, inductance, and pole number, and modify the relevant parameters of the motor

（2） Determine the working voltage and rated speed of the motor

（3） Connect the motor and check whether the motor can rotate

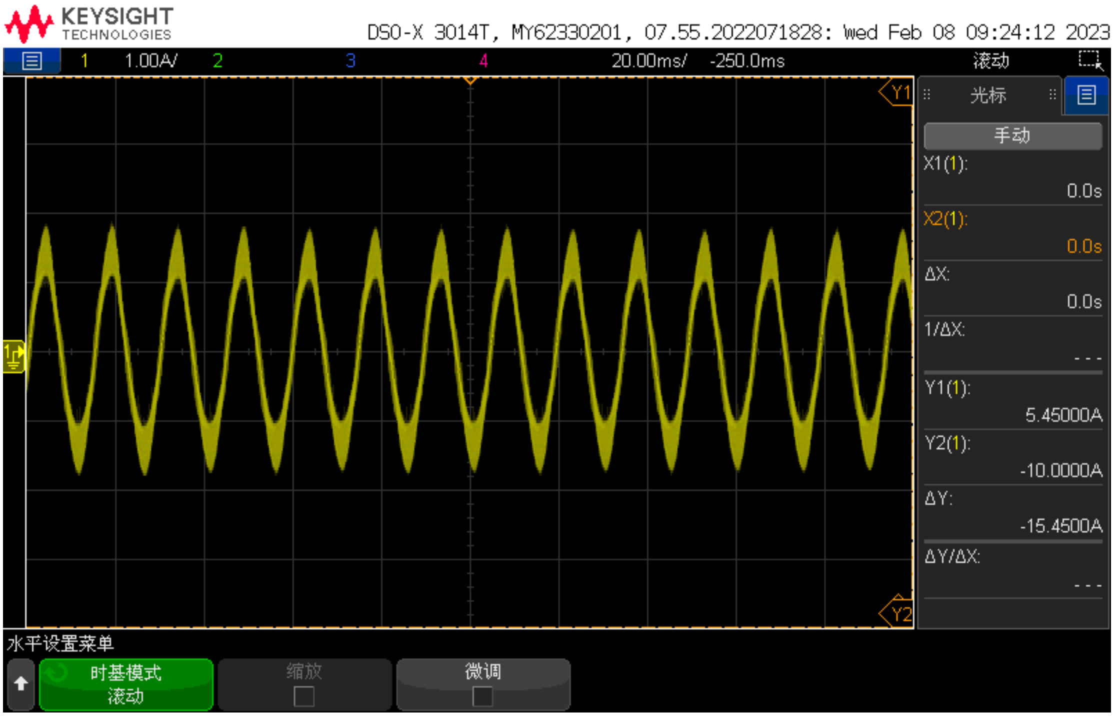
The motor cannot rotate. Check the fault code M1FaultID_Record through debug to see whether a fault is reported:

- Overvoltage or undervoltage: Check the supply voltage and set overvoltage and undervoltage thresholds. Check whether the MCU supply voltage is 3.3V;
- Software overcurrent: check whether the size of sampling resistance, operation amplifier multiple, and other parameters are consistent with the development board;
- Startup failure: check whether the cable is properly connected. Or increase the preset bit current appropriately

If it can rotate, but there is a problem and no fault is reported, it can be adjusted according to the above parameter description.

5 Actual test waveform

Figure 24 Actual Test Waveform



6 Revision History

Table 12 Document Revision History

Date	Revision	Revision History
July 26, 2023	1.0	New
August 14, 2023	1.1	(1) Modified the production information form (2) Modified the format
October 24, 2023	1.2	Modified the content to match the program
January 16, 2024	1.3	(1) Modified the amplification derivation process of the operational amplifier. (2) Made some revisions to the wording and details in the description.

Statement

This document is formulated and published by Geehy Semiconductor Co., Ltd. (hereinafter referred to as "Geehy"). The contents in this document are protected by laws and regulations of trademark, copyright and software copyright. Geehy reserves the right to make corrections and modifications to this document at any time. Please read this document carefully before using Geehy products. Once you use the Geehy product, it means that you (hereinafter referred to as the "users") have known and accepted all the contents of this document. Users shall use the Geehy product in accordance with relevant laws and regulations and the requirements of this document.

1. Ownership

This document can only be used in connection with the corresponding chip products or software products provided by Geehy. Without the prior permission of Geehy, no unit or individual may copy, transcribe, modify, edit or disseminate all or part of the contents of this document for any reason or in any form.

The "极海" or "Geehy" words or graphics with "®" or "™" in this document are trademarks of Geehy. Other product or service names displayed on Geehy products are the property of their respective owners.

2. No Intellectual Property License

Geehy owns all rights, ownership and intellectual property rights involved in this document.

Geehy shall not be deemed to grant the license or right of any intellectual property to users explicitly or implicitly due to the sale or distribution of Geehy products or this document.

If any third party's products, services or intellectual property are involved in this document, it shall not be deemed that Geehy authorizes users to use the aforesaid third party's products, services or intellectual property, unless otherwise agreed in sales order or sales contract.

3. Version Update

Users can obtain the latest document of the corresponding models when ordering Geehy products.

If the contents in this document are inconsistent with Geehy products, the agreement in thesales order or the sales contract shall prevail.

4. Information Reliability

The relevant data in this document are obtained from batch test by Geehy Laboratory or cooperative third-party testing organization. However, clerical errors in correction or errors caused by differences in testing environment may occur inevitably. Therefore, users should understand that Geehy does not bear any responsibility for such errors that may occur in this document. The relevant data in this document are only used to guide users as performance parameter reference and do not constitute Geehy's guarantee for any product performance.

Users shall select appropriate Geehy products according to their own needs, and effectively verify and test the applicability of Geehy products to confirm that Geehy products meet their own needs, corresponding standards, safety or other reliability requirements. If loses are caused to users due to the user's failure to fully verify and test Geehy products, Geehy will not bear any responsibility.

5. Legality

USERS SHALL ABIDE BY ALL APPLICABLE LOCAL LAWS AND REGULATIONS WHEN USING THIS DOCUMENT AND THE MATCHING GEEHY PRODUCTS. USERS SHALL UNDERSTAND THAT THE PRODUCTS MAY BE RESTRICTED BY THE EXPORT, RE-EXPORT OR OTHER LAWS OF THE COUNTIRIES OF THE PRODUCTS SUPPLIERS, GEEHY, GEEHY DISTRIBUTORS AND USERS. USERS (ON BEHALF OR ITSELF, SUBSIDIARIES AND AFFILIATED ENTERPRISES) SHALL AGREE AND PROMISE TO ABIDE BY ALL APPLICABLE LAWS AND REGULATIONS ON THE EXPORT AND RE-EXPORT OF GEEHY PRODUCTS AND/OR TECHNOLOGIES AND DIRECT PRODUCTS.

6. Disclaimer of Warranty

THIS DOCUMENT IS PROVIDED BY GEEHY "AS IS" AND THERE IS NO WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, TO THE

EXTENT PERMITTED BY APPLICABLE LAW.

GEEHY WILL BEAR NO RESPONSIBILITY FOR ANY DISPUTES ARISING FROM THE SUBSEQUENT DESIGN OR USE BY USERS.

7. Limitation of Liability

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL GEEHY OR ANY OTHER PARTY WHO PROVIDE THE DOCUMENT "AS IS", BE LIABLE FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, DIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE DOCUMENT (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY USERS OR THIRD PARTIES).

8. Scope of Application

The information in this document replaces the information provided in all previous versions of the document.

Geehy Semiconductor Co.,Ltd.

Bldg.1, No.83 Guangwan Street, Zhuhai, Guangdong, China   +86 0756 6299999   www.geehy.com