POLITECNICO DI TORINO

III Facoltà di Ingegneria

Corso di Laurea in Ingegneria Elettronica

Tesi di Laurea Specialistica

Integration of an attitude control and power management subsystem for a modular satellite

Supervisor:
Prof. DANTE DEL CORSO
Prof. LEONARDO REYNERI
Prof. CLAUDO SANSOE'

Candidate:
Qiu Longjia

September 2010

Summary

In the recently years, nanosatellite are playing a more and more important role in our scientific research and many university are designing their own nanosatellite. This trend began in 1999, when California Polytechnic State University (Cal Poly) and Stanford University developed the CubeSat specifications to help universities worldwide to perform space science and exploration. Several companies have built CubeSats, including large-satellite-maker Boeing. However, the majority of development comes from academia, with a mixed record of successfully orbited CubeSats and failed missions.

Politecnico di Torino has also developed its own satellite which is called PiCPoT (Piccolo Cubo del Politecnico di Torino). The main goals of PiCPoT is to test commercial components (COTS) in space and collect data in the space environment. It contains several on-board cameras and telemetry system transmitting on either a 437MHz (9600 FSK AX.25) or 2440MHz (10Kbps GFSK, +/-125KHz deviation) link. PiCPoT was launched aboard on 26 July 2006. Unfortunately, according to mission control officials, the engines of the Dnepr rocket shut down 86 seconds into its flight. The rocket crashed down and all satellites on board were destroyed.

After that, Politecnico di Torino starts to design another nanosatellite which is the evolution of PiCPoT. The new project named Aramis and it proposed a new modular architecture. The advantages of this architecture are the following. Firstly, better reusability through modularity. Secondly, overcome the limited CubeSat size (practically limited to 10×10×30 cm^3).

Aramis consist of two main parts. One is power management subsystem and another one is telecommunication subsystem. The main work of this thesis is to implement the attitude control system which is part of power management subsystem. In order to implement a precise attitude control system, the key point is to design a good motor control system.

To design the proper motor control system, this work made the following studies:

1. Analyze different kinds of motors and compare the difference between them. According to the requirements of project Aramis, choose the proper motor type. After that, a motor control system simulation was carried out to find out the details of working procedure. Then, explore two kinds of design strategies and figure out which one is fit for project Aramis.

2. Realize the motor control system in hardware level. To meet the requirements of project Aramis, the following functions have to be designed. Firstly, for the limitation of power bus, a voltage regulator is necessary. Secondly, in order to ensure the motor works in a safe state, the function of current measurement has to be implemented. Thirdly, to handle the speed of motor rotation, a speed control loop is designed. After this, the whole motor control system design is complete, and the system can be built.

3. Implement the motor control system in software level. Since the motor control system is controlled by the MSP430 microcontroller, it is required to study the architecture and functional units of MSP430. According to the needs of project Aramis, the details of timer and ADC12 are introduced. The basic operation data flow for motor control is finally described.

Chapter 1 Introduction

1.1 Nanosatellite

Nanosatellites [1] , also called "nanosats", are a relatively recent term used to describe artificial satellites with a mass between 1 and 10 kg. Larger satellites are often called microsatellites, while smaller satellites are called picosatellites. The term "nanosatellite" appears to have been introduced by NASA some time around 2004. It is still in the process of adoption, as many satellites of this size are simply called "small satellites." Usually, it contains the following five basic functions:

1. Power conversion/storage/management
2. Attitude determination/control
3. Housekeeping
4. On-board spacecraft control
5. Telecommunication

1.2 Project of Aramis

The project of Aramis comes from the evolution of PICPOT [2] . The purpose of this new architecture is to have better reusability through modularity and limited CubeSat size (practically limited to 10×10×30 cm^3)

Aramis modular can be divided into two tiles. Fig. 1 - 1 shows the modular architecture of Aramis:

- Power Management Tile (PMT)
- TT&C (telemetry, tracking and control functions) and AOCS Tile (TTC)

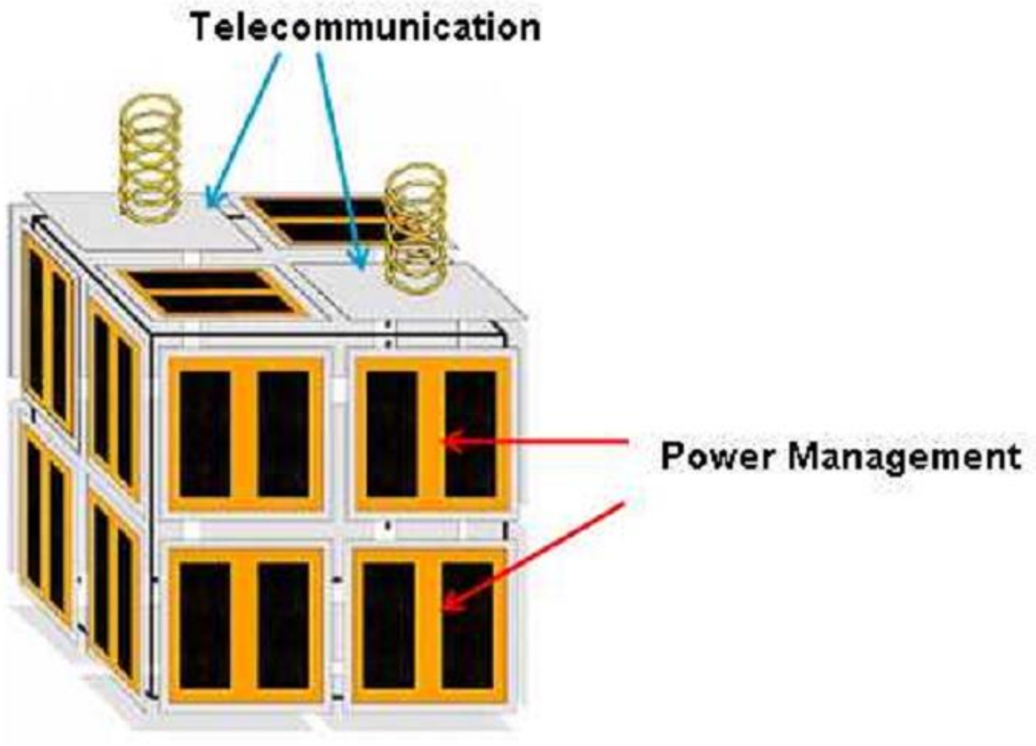


Fig. 1 - 1 : Modular Architecture of Aramis

1.2.1 Power Management Tile

There are six parts inside of power management tile.

- Thermo mechanical subsystem
- Power subsystem
- Attitude determination sensors
- Attitude control actuators
- Standard housekeeping
- Test connector

1.2.2 OBC & TT&C Tile

OBC & TT&C tile is consisted with the following parts:

- Thermo mechanical subsystem
- Dual Telecommunication subsystem

- Dual-redundant on-board computer and on-board data bus
- Central opening for earth observation equipment

1.3 Major Task

While the satellite running in the space, the position of the satellite is very essential for the control system. Therefore, the attitude control system is crucial for the satellite. There are five main components that are used in Aramis attitude control system implementation: Magnetometer, gyroscope, brushless, solenoid and micro controller. In order to control the satellite running in a right position, brushless namely motor control plays an important role.

This paper describes the design of a three phase brushless DC motor control system. It is based on L6235 DMOS driver for three-phase brushless DC motor. In addition, several key functions as protection circuit and high speed PWM current control allow to drastically reduce the external components count to meet requirements for Aramis project.

The motor is the fundamental part of the attitude control subsystem. In order to control the attitude of the satellite in a accurate way, the motor driver has to be designed very well. Firstly, the driver must provide the over current protection and thermal protection, and the motor still keeps working even some errors happened. Moreover, the motor driver has to supply basic functions. In other words, the motor working states can be changed in a easy way. The function of enable, brake, forward, backward and so on have to be offered to users. Lastly, the system must provide current working condition of the motor to the ground station. More specifically, rotating speed of the motor, output current of the driver and working state of the motor, all of these information has to be offered by motor driver circuit.

Chapter 2 Motor Control System

2.1 Motor Type

Motor is a device that creates motion which converts electric energy into mechanical energy. Usually, there are two kinds of motor. One is AC motor and another one is DC motor.

2.1.1 AC Motor

An AC motor is an electric motor which is driven by an AC current. It has two basic parts, there is an external stationary stator having coils supplied with alternating current to produce a rotating magnetic field, and an internal rotor attached to the output shaft that is given a torque by the rotating field.

Depending on the type of rotor used, there are two types of AC motors. The first one is the synchronous motor, which rotates exactly at the supply voltage frequency or a sub multiple of the frequency. In the application of industry, synchronous ac motors have two important functions. First, ac motors provide highly efficiency. It means as much as ac energy converting to mechanical power. Second, ac motors can operate at leading or unity power factor, thereby providing power factor correction.

The second one is the induction motor, which runs slightly slower than the supply frequency. The magnetic field on the rotor of this motor is created by an induced current.

2.1.2 DC Motor

A DC Motor uses direct current - in other words, the direction of current flows in one direction.

Brushless DC motors use a rotating permanent magnet in the rotor, and stationary electrical magnets on the motor housing. A motor controller converts DC to AC. This design is simpler than that of brushed motors because it eliminates the complication of transferring power from outside the motor to the spinning rotor. Advantages of brushless motors include long life span, little or no maintenance, and high efficiency. The disadvantages include high initial cost, and more complicated motor speed controllers.

2.1.3 Motor Comparison

There are three popular motors in the market. They are AC asynchronous motor(ACAM), brushed DC motor(BDC) and brushless DC motor(BLDC). Table 2 - 1 compares BLDC motor with other two motors.

Table 2 - 1 : BLDC Motor Compared with Other Motors

SYSTEM / PROPERTY	ACAM	BDC	BLDC
Mechanical	SOFT	ROBUST	ROBUST
Overload Capacity	SMALL	LARGE	LARGE
Controllability	HARD	EASY	EASY
Stationarity	BAD	GOOD	VERY GOOD
Noise	LARGE	VERY LARGE	LITTLE
Maintainability	EASY	DIFFICULT	EASY
Life Span	LONG	SHORT	LONG
Volume	LARGE	SMALL	VERY SMALL
Efficiency	LOW	HIGH	HIGH
Cost	LOW	HIGH	HIGH

Since our design is based on the project Aramis, the motor is set on satellite. It is impossible to repair the motor, so we need a long life span, no maintenance and also high efficiency motor. From the above table, it is easy to find BLDC motor is the best choice for this project.

2.1.4 BLDC Motor

Fig. 2 - 1 is a three phases brushless DC motor with hall sensor [4] . From Table 2 - 2 we get the general idea for the BLDC motor driver. Through the signal from hall sensor, the motor can be driven very easily. The motor rotates, the hall sensor signal also changes, so the phase of output voltages changes in the same time. The BLDC motor can be controlled in this way.

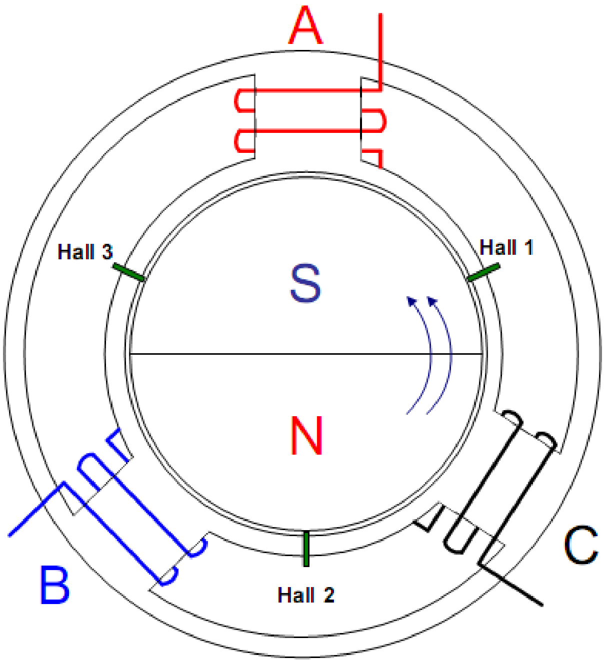


Fig. 2 - 1 : Three Phases Brushless DC Motor with Hall Sensor

Table 2 - 2 : 60 and 120 Electrical Degree Decoding Logic in Forward Direction

Hall 120°	1	2	3a	-	4	5	6a	-
Hall 60°	1	2	-	3b	4	5	-	6b
H_1	H	H	L	H	L	L	H	L
H_2	L	H	H	H	H	L	L	L
H_3	L	L	L	H	H	H	H	L
OUT_1	Vs	High Z	GND	GND	GND	High Z	Vs	Vs
OUT_2	High Z	Vs	Vs	Vs	High Z	GND	GND	GND
OUT_3	GND	GND	High Z	High Z	Vs	Vs	High Z	High Z
Phasing	1->3	2->3	2->1	2->1	3->1	3->2	1->2	1->2

Brushless DC (BLDC) motors have been used in various industrial applications and have increased demand in diverse fields because of its high efficiency, simple control compared with AC motors, low EMI, and high reliability due to the absence of brushes. Most three-phase motors, including BLDC motors need at least six PWM channels for inverter power devices such as IGBTs and MOSFETs. Fig. 2 - 2 and Fig. 2 - 3 show the sequence diagram for the hall sensors in 120 degree and 60 degree. A Hall effect sensor is a transducer that varies its output voltage in response to changes in magnetic field. Hall sensors are used for proximity switching, positioning, speed detection, and current sensing applications. The difference between these two figures is the position of hall sensors. In the first sequence diagram, the angle among the three hall sensors is 120 degree. In the second sequence diagram, the angle changed into 60 degree.

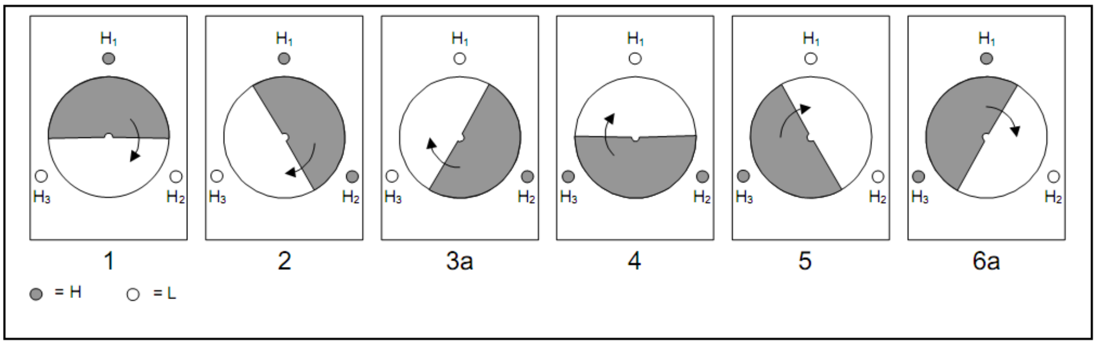


Fig. 2 - 2 : 120°Hall Sensor Sequence

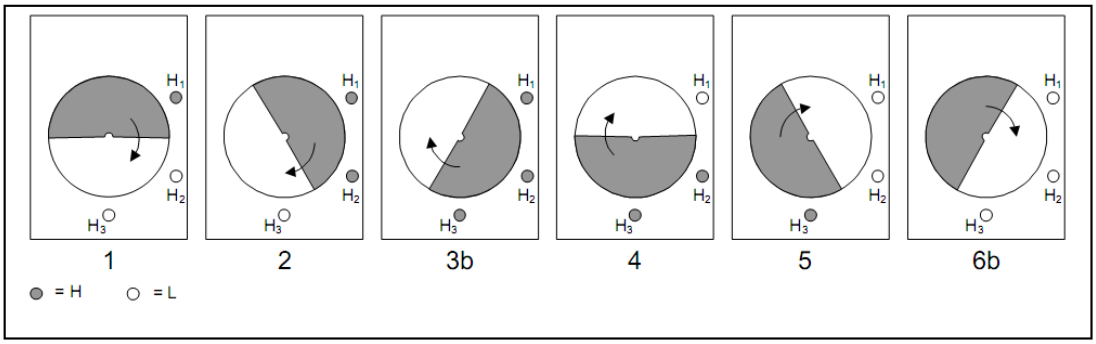


Fig. 2 - 3 : 60°Hall Sensor Sequence

2.2 Motor Control System Simulation

2.2.1 Simulation Strategy

According to the different properties, there are at least two choices to design the inverter circuit. One is using MOSFET and another one is using IGBT. Since IGBT is much more expensive than MOSFET, MOSFET is a good choice for this project.

Fig. 2 - 4 shows the three phases inverter. Since the power supply of satellite is battery, we can just use DC voltage to drive motor. Here we choose EC 32 flat, diameter 32mm, brushless, 6 watt DC motor with hall sensors. Because of hall sensors, we can control speed and position with high accuracy. In order to drive motor by using DC voltage, DC voltage inverter is a necessary one. Here we have three H-bridges circuit for inverting DC voltage:

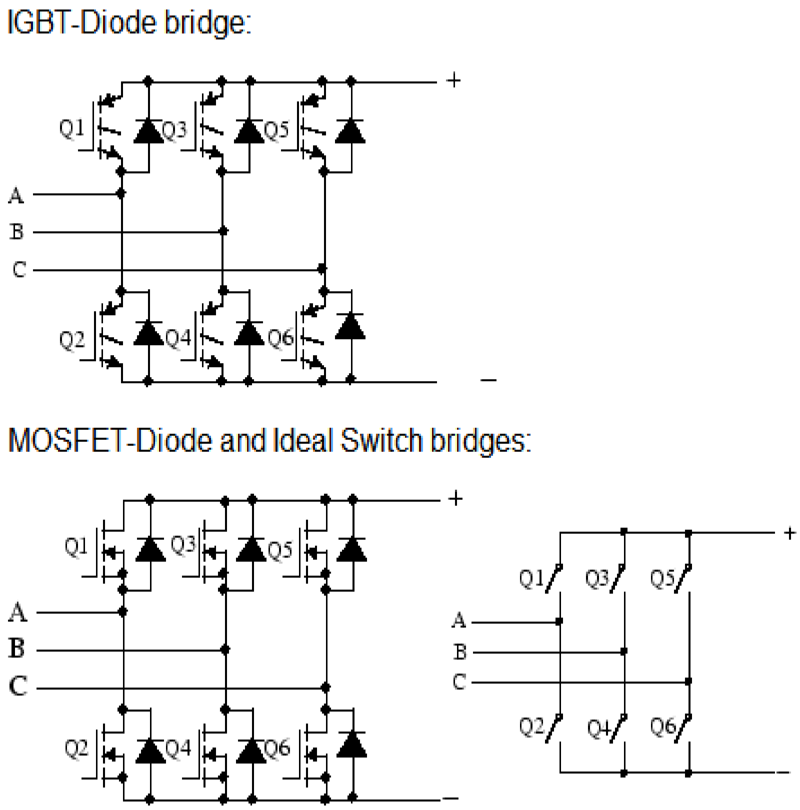


Fig. 2 - 4 : Three Phases DC Voltage Inverter

The following schematic Fig. 2 - 5 is a high-level diagram of motor driver simulation. It consists of four parts: three-phase diode rectifier, three-phase inverter, BLDC motor with hall-sensor, current controller and speed controller. In general, all the motor driver system is almost same with this. The input of this motor driver system is an AC source.

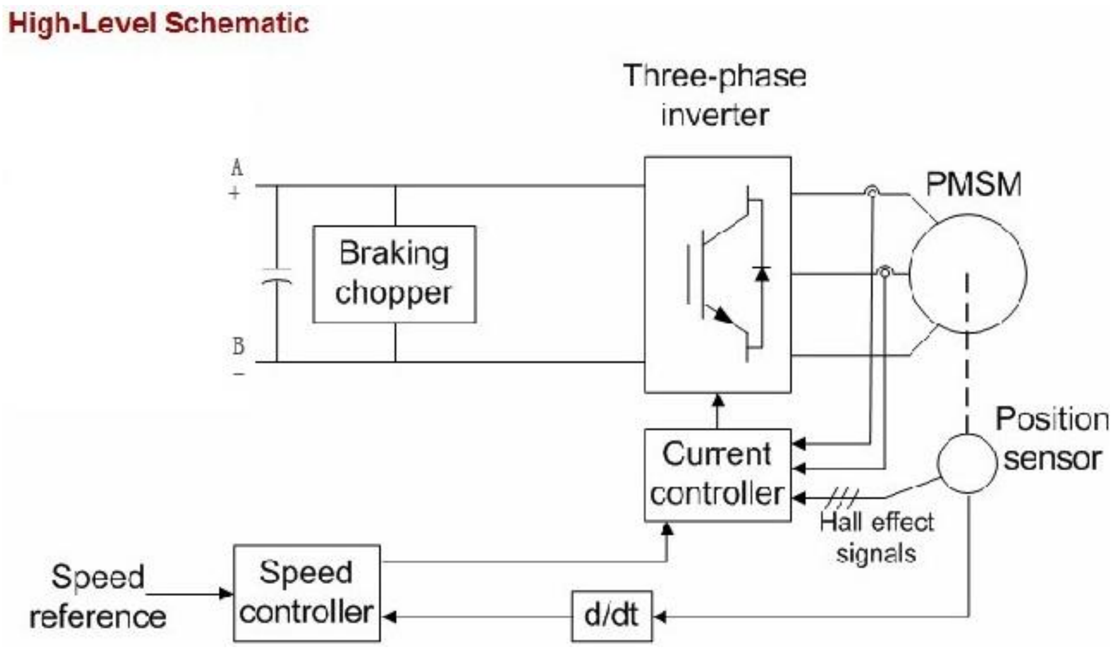


Fig. 2 - 5 : High-Level Schematic of Motor Driver Simulation

Since it is a high-level schematic, in order to understand the details of this system, we implement a simulation of this system in the Matlab. Fig. 2 - 6 is the model of this system. A three-phase motor is fed by a six step voltage inverter. The inverter is a MOSFET bridge. A speed regulator is used to control the DC bus voltage. The inverter gates signals are produced by decoding the Hall effect signlas of the motor. The three-phase output of the inverter are applied to the PMSM block's stator windings.

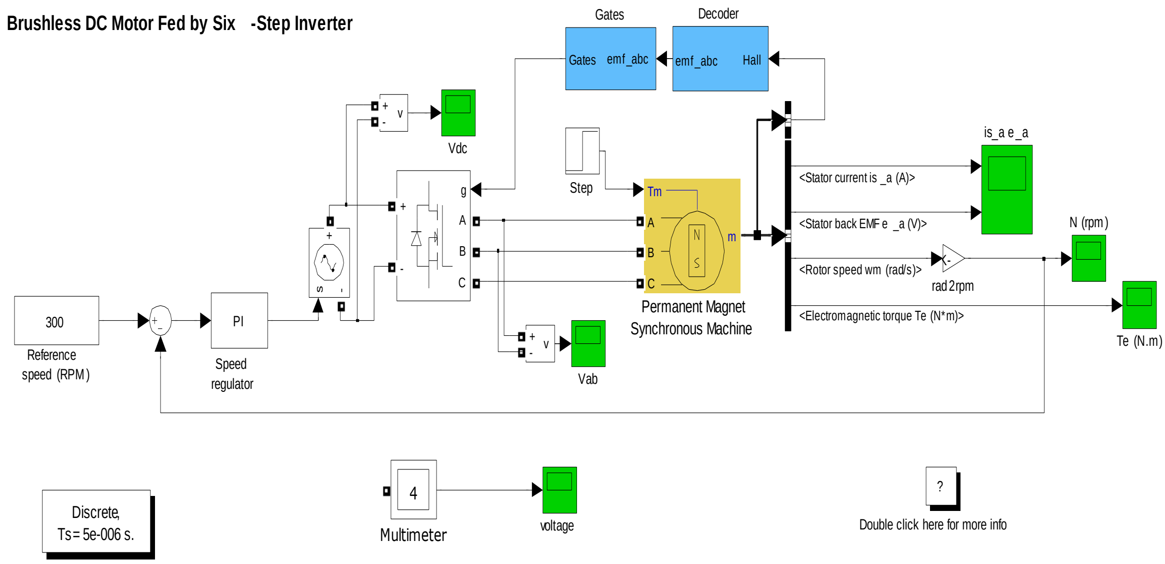


Fig. 2 - 6 : Brushless DC Motor Fed by Six-Step Inverter

The control parts are the main parts of this model, two control loops are used. The inner loop synchronizes the inverter gate signals with the electromotive forces. The outer loop controls the motor's speed by varying the DC bus voltage. Fig. 2 - 7 shows the decode logic circuit of inner loop. The inputs of this block are signals from hall sensor. For the purpose of controlling the three half-bridges, these signals have to be decoded to control the switch of the three half-bridges. Fig. 2 - 8 shows the gate control circuit of inner loop. The inputs are decoded signal from decode block, the outputs are the control signal of six switches. Fig. 2 - 9 are the three half-bridges circuit of control loop, it realizes the voltage inverter function. Through these blocks, the main part of a BLDC motor driver are implemented.

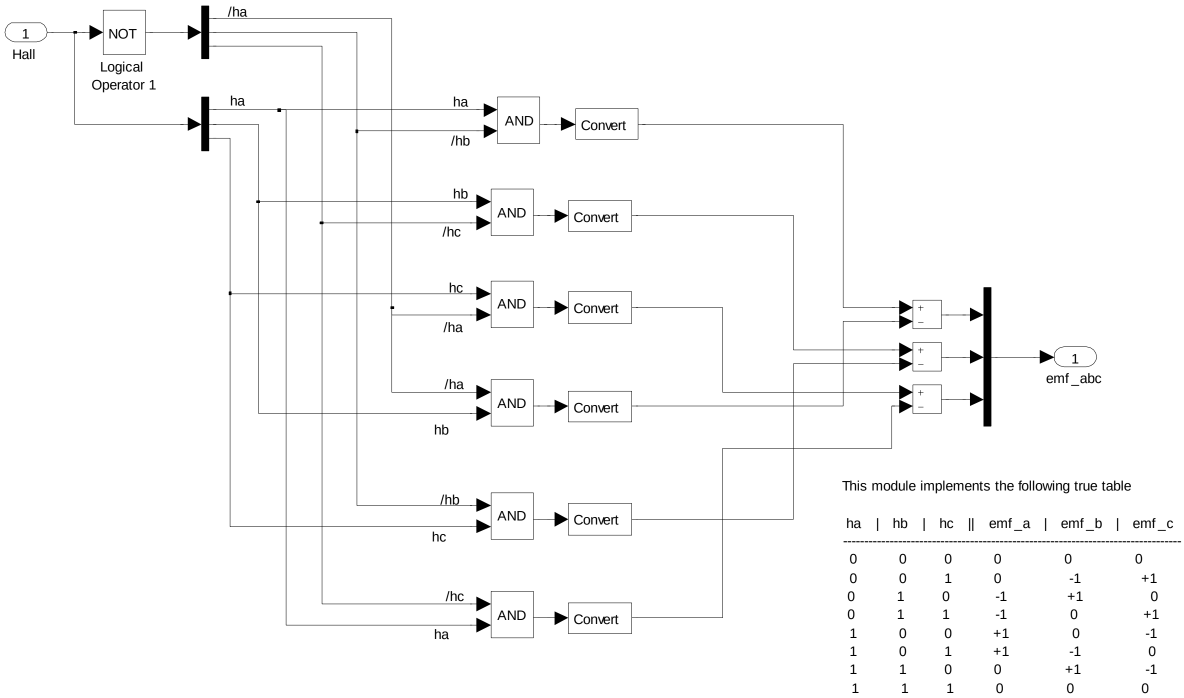
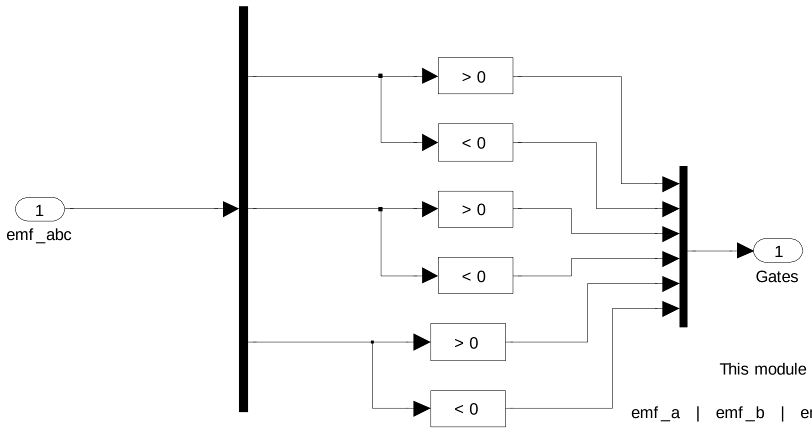


Fig. 2 - 7 : Decode Logic Circuit of Inner Loop



This module implements the following true table

emf_a	emf_b	emf_c	Q1	Q2	Q3	Q4	Q5	Q6
0	0	0	0	0	0	0	0	0
0	-1	+1	0	0	0	1	1	0
-1	+1	0	0	1	1	0	0	0
-1	0	+1	0	1	0	0	1	0
+1	0	-1	1	0	0	0	0	1
+1	-1	0	1	0	0	1	0	0
0	+1	-1	0	0	1	0	0	1
0	0	0	0	0	0	0	0	0

Fig. 2 - 8 : Gate Control Circuit of Inner Loop

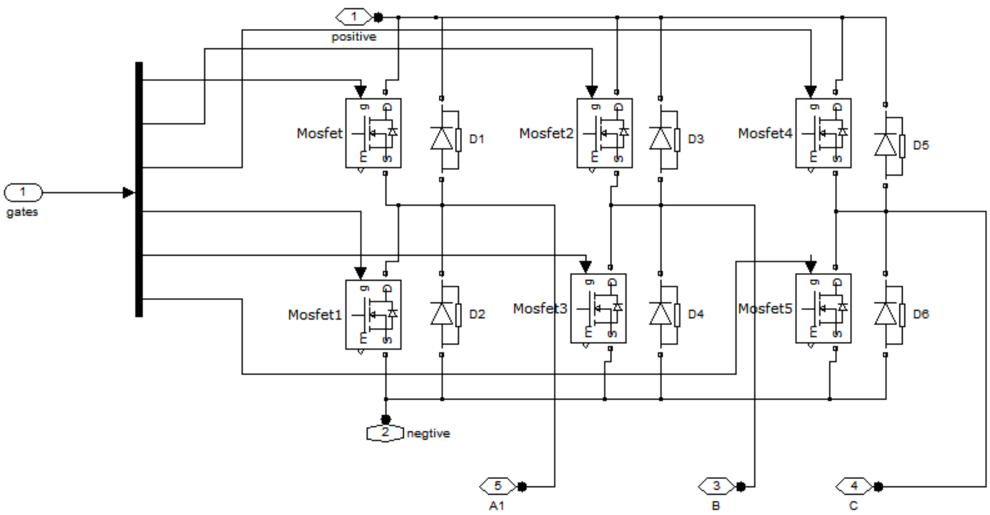


Fig. 2 - 9 : Three Half-bridges Circuit of Control Loop

2.2.2 Simulation Results

A three-phase motor is fed by a six step voltage inverter. The inverter is a MOSFET bridge. A speed regulator is used to control the DC bus voltage. The inverter gates signals are produced by decoding the Hall effect signals of the motor.

The three-phase output of the inverter are applied to the PMSM block's stator windings. The load torque applied to the machine's shaft is first set to 0 and steps to its nominal value at t = 0.1s. Due to the size of figure, simulation stop time is set at 0.4s. It is easy to check the measurements and the initial state of motor.

1. Fig. 2 - 10 shows the differential voltage between outputs of motor driver. It is easily to be found the voltage inverter works well. There is a fixed phase shift between these three voltage, so the motor will keep running.

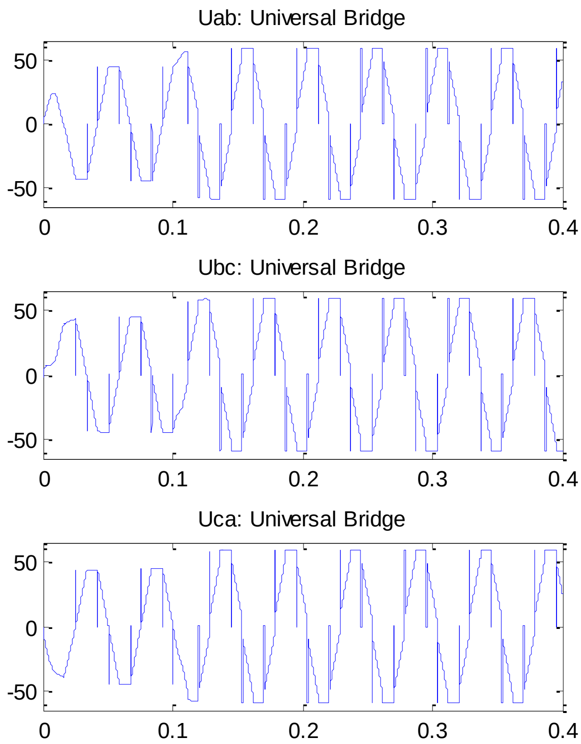


Fig. 2 - 10 : Differential Voltage Between Outputs of Motor Driver

2. Fig. 2 - 11 shows voltage of six switches in the three half-bridge. It indicates the decode block is designed in a right way, so the control signal for the switches are showed in the following figure. The two switches of each half-bridge are always that one is switch on and another one is switch off.

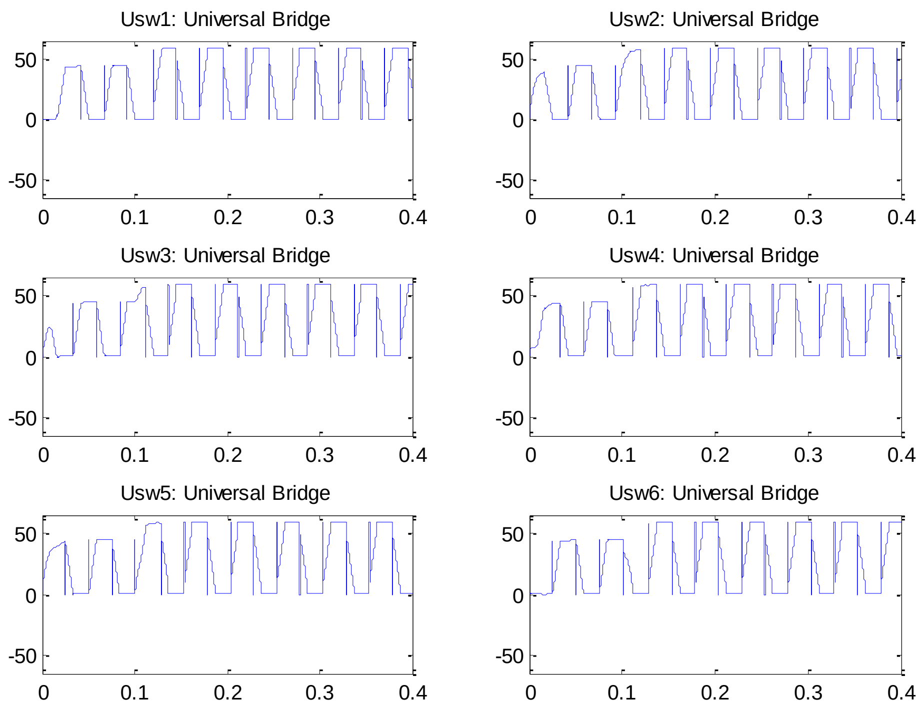


Fig. 2 - 11 : Voltage of Six Switches in The Three Half-bridge

2.3 Design Strategy

There are at least two strategies for the motor controller design. One is using MCU to control the motor directly, another one is to use a IC controller to control the motor. We will talk about the details of these two possible ways, and we have to decide which way is better for the project.

2.3.1 MCU

High efficiency variable speed and variable torque motor control is only possible when we are using microcontrollers [3] . Low cost is still a main factor in designing low-end products using motor control. Digital electronic control of motors offers much higher efficiency and better power usage at competitive cost. Despite all the obvious facts, replacing the cost of inefficient noisy motors and mechanical components with the more efficient brushless motors and digital electronics microcontrollers would still require concentrated technical support and marketing efforts.

In BLDC motor drives, polarity reversal is performed by power transistors switching in synchronization with the rotor position. Therefore, the BLDC motor has to use either an internal or external position sensor to detect the actual rotor position. Also the rotor position can be estimated without the need for a position sensor. In general, a BLDC motor may use either 60 deg or 120 deg commutation intervals. Fig. 2 - 12 shows a schematic of the BLDC motor and the ideal current waveforms versus position. Also, the position sensors outputs are illustrated.

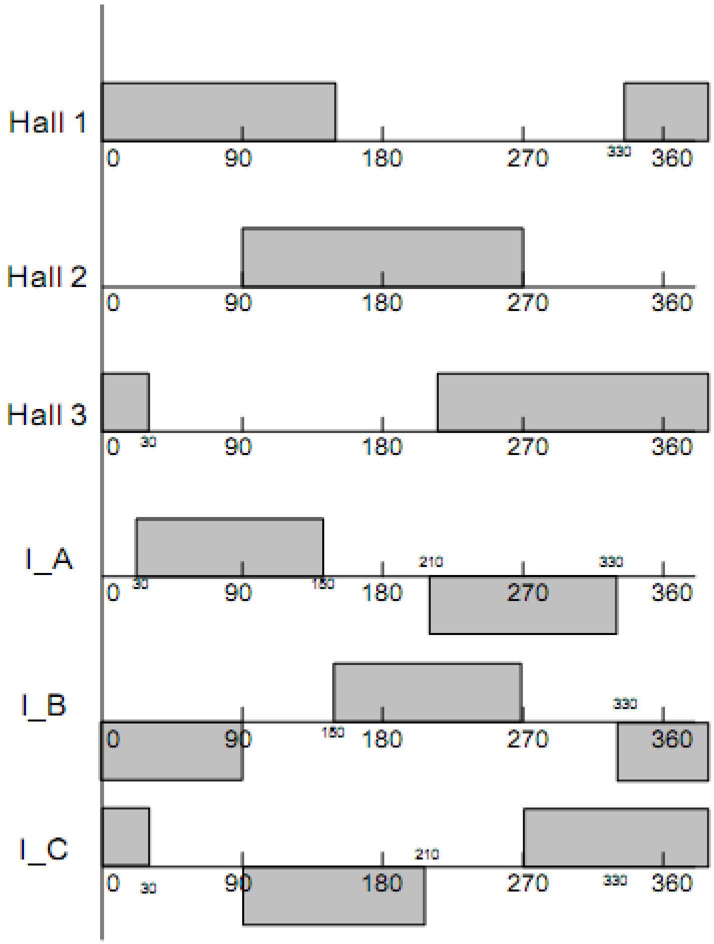


Fig. 2 - 12 : BLDC Motor Signals and Current Waveforms

In order to meet the requirements of high efficiency, simple control compared with AC motors, low EMI, and high reliability, generally a single chip controller, a special-purpose processor, a programmable logic device (PLD) or drive device to generate control signal is necessary. Using a single chip controller provides some good points, such as its small size and cost effective drive. However, if more functions are required this type of drive system will need more devices which will make this drive system more expensive. Adapting a special purpose processor or device for a BLDC motor drive presents several advantages, such as small drive size and less development time. Therefore, these processors are more expensive than general purpose processors. Fig. 2 - 13 indicates the schematic of motor controller system based on MSP430 platform. We can easily extend the function of motor controller by using MSP430.

Although it is very easy to use and control, it still has some problems. Especially in our situation, the resource is very limit, so the available ports from microcontroller are limited. Except these ports for the driver system, we also need some I/O ports to implement the function of ENABLE, FW/BW, BRAKE, SPEED CONTROL, CURRENT MEASURE and so on. From above all, we get to the conclusion if we choose MCU strategy to implement the BLDC motor driver system, we will occupy too much resources from microcontroller. Besides this reason, we also have to consider the power consumption issue and high switching frequency operation problem. Apparently, we decide to look for another solution for the motor driver system.

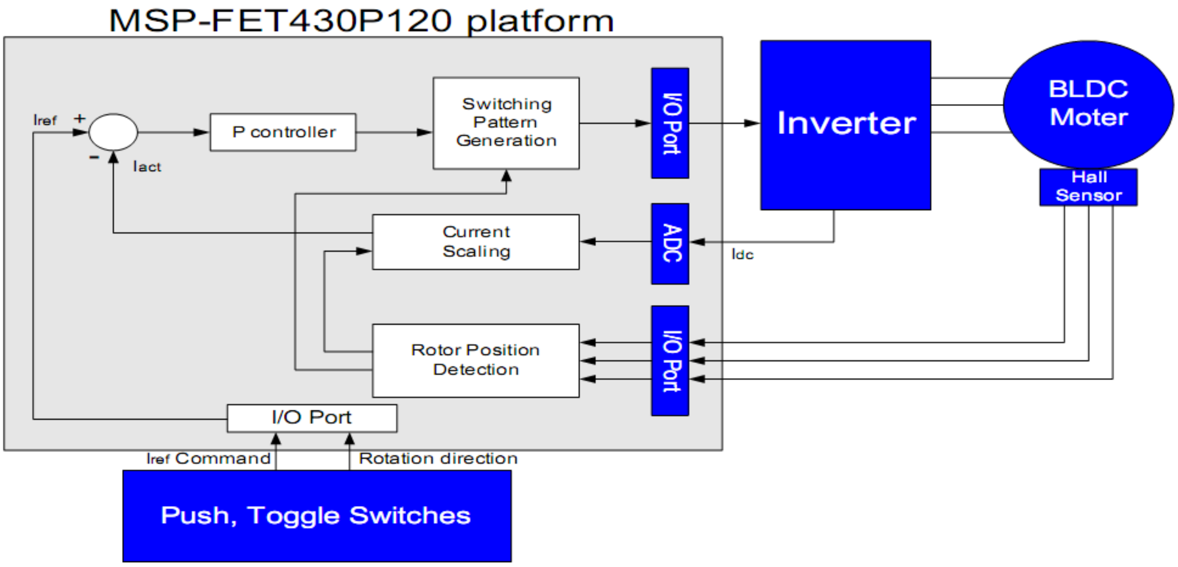


Fig. 2 - 13 : BLDC Motor Driver System in MSP platform

2.3.2 IC Controller

A single IC chip cost much less than a MCU, so use the IC chip to control the motor is a very economic way. And that was not all, the IC chip is much easy to use comparing with MCU. The designer does not have to handle control logic, torque and the speed control, power deliver issues and ensure safe operation in every load condition. Therefore, the designer would not spend much time on the programming issue, all the things have been done in the IC chip. Clearly, it is a good way for this project. The cost is low, the volume is small and it is easy to use.

We analyzed three choices to choose a proper IC controller. First one is TDA5145 brushless DC motor drive circuit. Second one is A8904 3-phase brushless dc motor controller/driver with back EMF sensing. Third one is L6235 DMOS driver for three-phase brushless dc motor.

2.3.2.1 TDA5145

The TDA5145 is a bipolar integrated circuit which is used to drive 3-phase brushless DC motors in full-wave mode. The device has no hall sensor, therefore it uses the back-EMF sensing technique to sense the rotor position. The chip includes bidirectional control, brake function and also has a special circuit built-in to reduce the EMI (soft switching output stages). Features of TDA5145 is in the following.

- Full-wave commutation (using push/pull drivers at the output stages) without position sensors
- Built-in start-up circuitry
- Three push-pull outputs:
  - output current 2.0A (typ.)
  - built-in current limiter
  - soft-switching outputs for low Electromagnetic Interference
- Thermal protection
- Flyback diodes
- Tacho output without extra sensor
- Motor brake facility
- Direction control input
- Reset function
- Transconductance amplifier for an external control transistor

2.3.2.2 A8904

The A8904 is a 3-phase brushless DC motor controller. It is designed for applications where accurate control of high-speed motors is required. The three half-bridge outputs are low on-resistance, N-channel DMOS devices have capable of driving up to 1.2 A. The A8904 provides complete, reliable, self-contained back EMF sensing, motor start-up, and running algorithms. It contains a programmable digital frequency-locked loop speed control circuit and the linear current control circuitry provides accurate motor speed regulation. Features is in the following.

- Pin-for-pin replacement for A8902CLBA
- Start-up commutation circuitry
- Sensorless commutation circuitry

- Option of external sector data tachometer signal
- Option of external speed control
- Oscillator operation up to 20 MHz
- Programmable over current limit
- Transconductance gain options: 500 mA/V or 250 mA/V
- Programmable watchdog timer
- Directional control
- Serial port interface
- TTL-compatible inputs
- System diagnostics data-out ported in real time
- Dynamic braking through serial port or external terminal

2.3.2.3 L6235

The L6235 is a DMOS Fully Integrated Three-Phase Motor Driver [5]. It contains the over current protection. The device is implemented in multipower-BCD technology, it combines isolated DMOS Power Transistors with CMOS and bipolar circuits on the same chip. The device contains a three-phase DMOS Bridge which is used as a voltage inverter, a constant off time PWM Current Controller and the decoding logic for single ended hall sensors that generates the required sequence for the power stage. These things are the main parts to drive a three phase BLDC motor. Features is in the following.

- Operating Supply Voltage From 8 To 52v
- 5.6a Output Peak Current (2.8a Dc)
- $R_{DS}(ON)$ 0.3ω Typ. Value @ T_j = 25° C
- Operating Frequency Up To 100khz
- Non Dissipative Over current Detection and Protection
- Diagnostic Output
- Constant T_{off} Current Controller
- Slow Decay SYNCHR. Rectification

- 60° & 120° Hall Effect Decoding Logic
- Brake Function
- Tacho Output For Speed Loop
- Cross Conduction Protection
- Thermal Shutdown
- Undervoltage Lockout
- Integrated Fast Freewheeling Diodes

2.3.2.4 IC Comparison

According to the features of these three IC components, Table 2 - 3 compares the performance of them.

Table 2 - 3 : IC Performance Comparison

FEATURES \ IC	TDA5145	A8904	L6235
$R_{DS}(ON)$	×	√	√
Over Current Detection	×	×	√
Brake Function	√	√	√
Direction Function	√	√	√
Tacho Output	×	×	√
Cross Conduction Production	×	√	√
Thermal Shutdown	√	√	√
Hall Sensor	×	×	√

As we can see from the table, it is clearly to see that L6235 is better than the other two IC chips. The most important reason for this is that L6235 has three hall sensor. Therefore, it is easy to control the speed of motor and the accuracy is much better than the other two ICs. Except this reason, L6235 has TACHO output as well.

It allows us to control the speed of motor in a dynamic way. Lastly, L6235 is a circuit with over current detection. It will avoid the destroy from high current. Until now, we decide to choose L6235 as our motor controller. The details of L6235 design will be introduced in the next chapter.

Chapter 3 Hardware Design

3.1 Hardware Implementation

3.1.1 L6235 Specification

We choose L6235 as motor driver, so the first thing is to take a look at this IC [7].

The following Fig. 3 - 1 is the pin connection of L6235. There are three kinds of packages of L6235, we choose the following one.

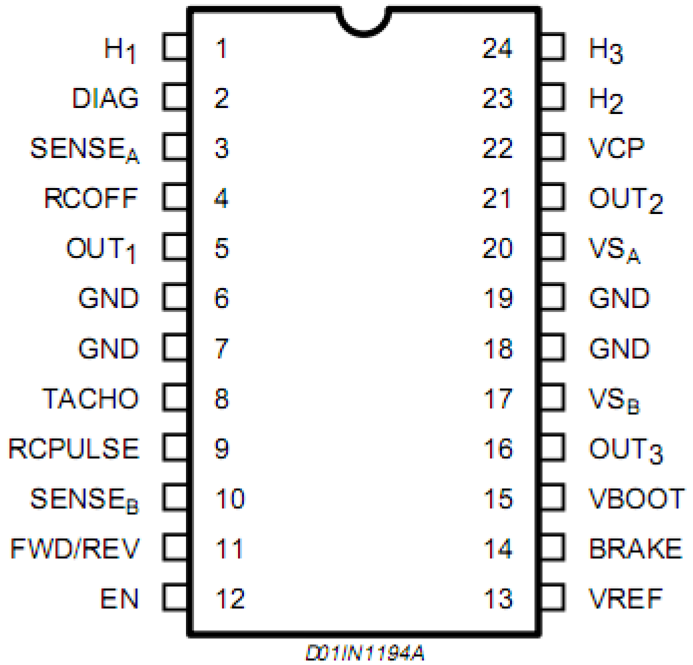


Fig. 3 - 1 : PowerDIP 24/SO24

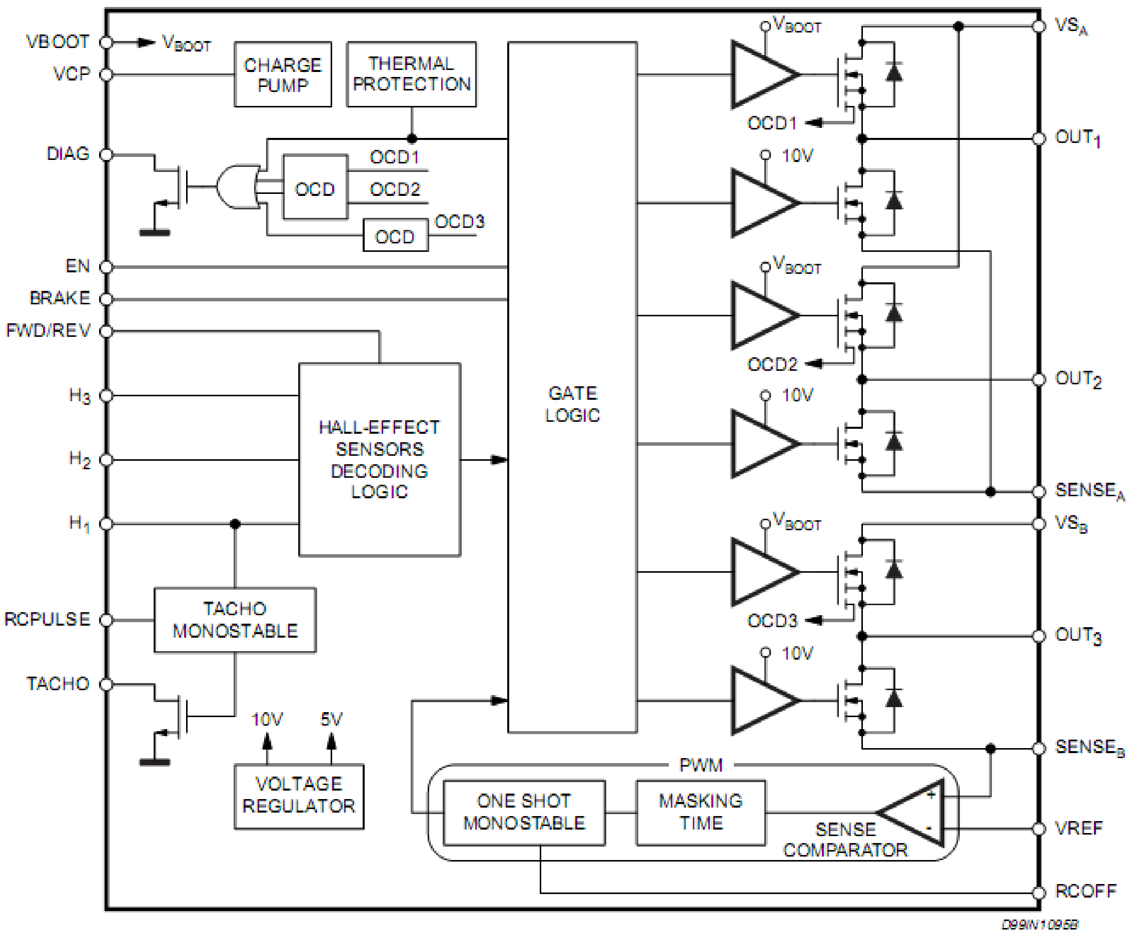


Fig. 3 - 2 : Block Diagram of L6235

The above schematic Fig. 3 - 2 is the block diagram of L6235. From this diagram, the function of L6235 chip is clear to see. It includes the control logic, speed control power management and also voltage inverter. This chip implements most function which can be used in motor control, including the enable, forward, backward and brake function.

3.1.2 Design of Voltage Regulator

3.1.2.1 Design Circuit of Voltage Regulator

The first thing we have to design is a voltage regulator. Because supply voltage of L6235 is more than 8V, the hall sensor supply voltage and OPA supply voltage both are 5V. Since the satellite has limited resource, it is not possible to have 8V and 5V directly from power bus. For the hall sensor power supply, we should use a voltage regulator to supply the voltage. The hall sensor power supply should be larger than 3.5V. The best way to solve this problem is that power bus supplies a voltage higher than 8V and by using a voltage regulator we get a 5V voltage. We plan to use LM317 to implement this [8]. The following are the features of LM317.

- Guaranteed 1% output voltage tolerance (LM317A)
- Guaranteed max. 0.01%/V line regulation (LM317A)
- Guaranteed 1.5A output current
- Adjustable output down to 1.2V
- Current limit constant with temperature
- P+ Product Enhancement tested
- 80 dB ripple rejection
- Output is short-circuit protected

In operation, the LM317 develops a nominal 1.25V reference voltage V_{ref}, between the output and adjustment terminal. In order to design a output voltage around 5V, according to the equation

$$V_{out} = 1.25\left(1+\frac{R_2}{R_1}\right)+I_{ADJ}(R_2)$$

(Eq. 3 - 1)

We choose $R1 = 240\Omega, R2 = 680\Omega,\ C1 = 0.1\mu F, C2 = 1\mu F.$ C1 is used as an input bypass capacitor. 0.1uF is suitable for almost all the applications. C2 is used as an output bypass capacitor. With a 10uF bypass capacitor 80dB ripple rejection is obtainable at any output level. Since our power is from battery, 1uF insures the stability of the circuit. The simulation circuit Fig. 3 - 3 is a 5V voltage regulator.

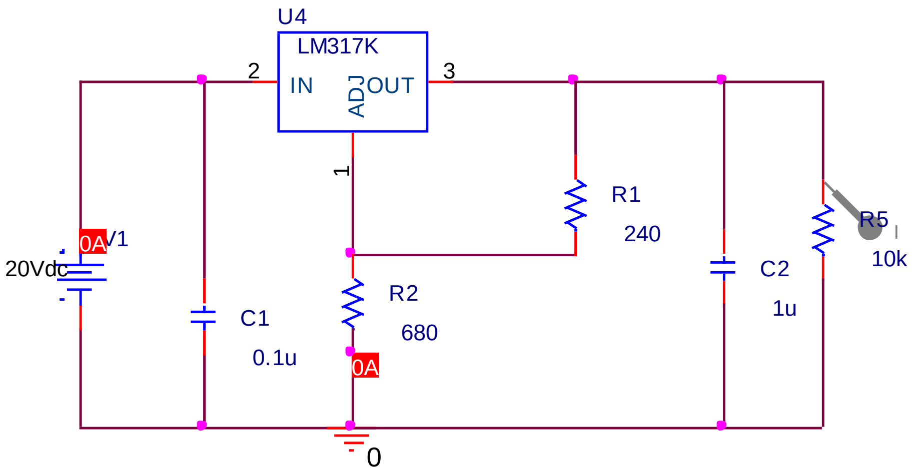


Fig. 3 - 3 : 5V Voltage Regulator

3.1.2.2 Test of Voltage Regulator

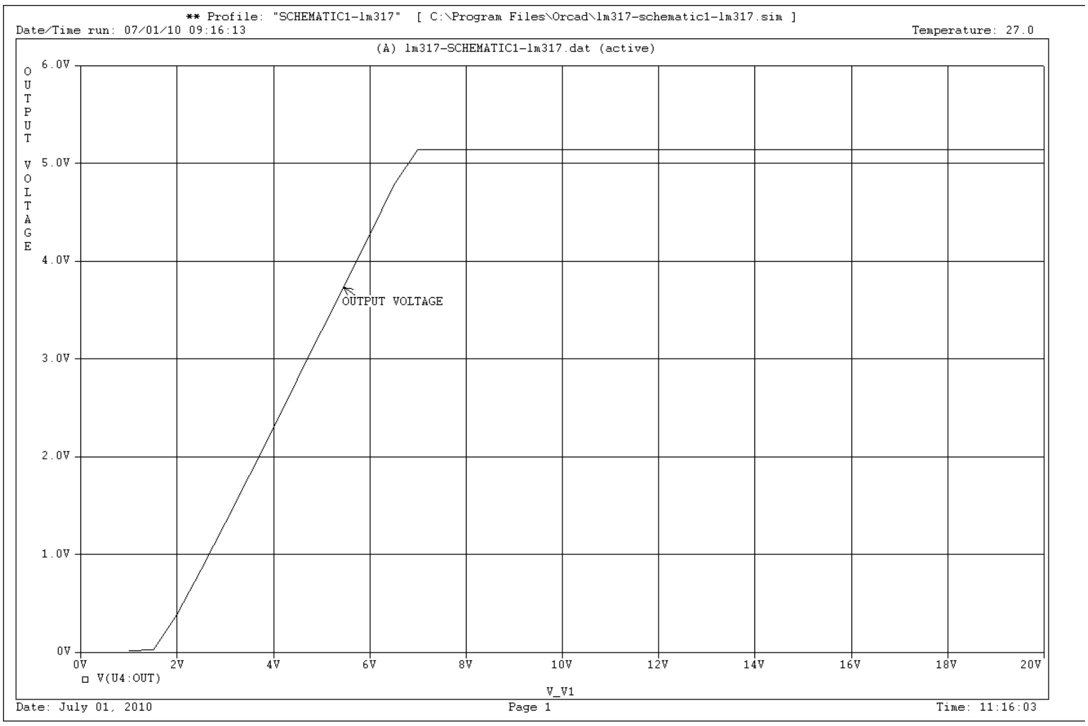


Fig. 3 - 4 : Voltage Regulator Simulation Test

From Fig. 3 - 4, we can see the voltage regulator works very well. When the input voltage is larger than 7V, the output voltage is always 5V. Fig. 3 - 5 is the layout of 5V voltage regulator. Test work is also carried out on the PCB board of 5V voltage regulator. The result is still the same, when the input voltage is larger than 6.8V, the output voltage can be kept at 5V steadily. Until now, the voltage regulator design is finished, we can move to the next step.

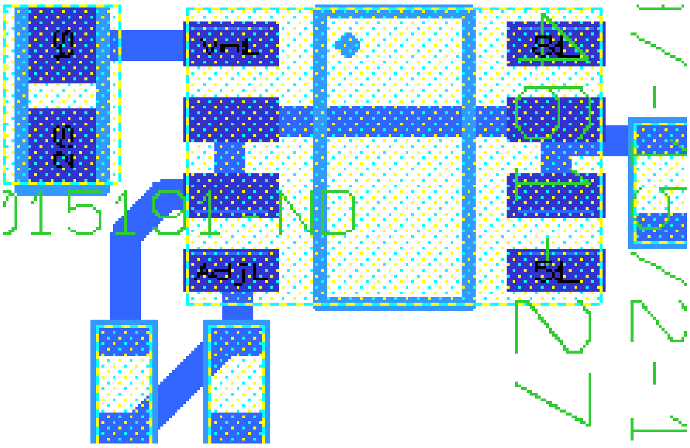


Fig. 3 - 5 : Layout of 5V Voltage Regulator

3.1.3 Design of Current Measurement

3.1.3.1 Current Measurement Theory

Since the current value is very important to the control circuit, it has to keep in a proper value, thus the motor can work in a safe and steady state. Therefore, a current measure circuit is greatly necessary. In general, there are two kinds of methods to measure current. One is low side current measurement and another one is high side current measurement. Fig. 3 - 6 shows the different between high side current measurement and low side current measurement. Low side refers to the return path from the load. High-side refers to the supply path to the load. The low side method is usually at a low voltage to ground, the high-side is usually at a high voltage to ground. The decision to place a current shunt in either position has advantages and disadvantages that must be accounted for and assessed based on the particular application.

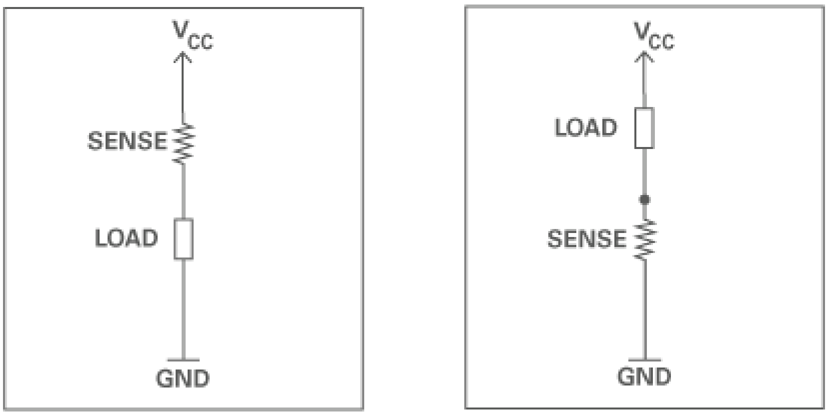


Fig. 3 - 6 : High Side Versus Low Side Current Measurement

Hide Side Current Measurement:

Advantages:

- Current sensor connected directly to the power source and can detect any downstream failure and trigger appropriate corrective action.
- Won't create an extra ground disturbance that comes with a low side current sensing design

Disadvantages:

- A 0.01% deviation in any resistor value lowers the CMRR to 86dB,
- A 0.1% deviation lowers it to 66dB,
- A 1% deviation lowers it to 46dB.
- Must withstand very high, and often dynamic, common-mode voltages (often outside the limits of the supply rails of the amplifiers used).

Low Side Current Measurement:

Advantages:

- Straightforward, easy, and rarely requires more than an op-amp to implement
- Inexpensive and precise

Disadvantages:

- Adds undesirable resistance in the ground path
- May require an additional wire to the load that could otherwise be omitted

Come to the conclusion, the better way is to choose high side current measurement in the project, we can detect any downstream failure and trigger appropriate corrective action. Due to this, we choose INA138 high-side measurement current shunt monitor [9]. The below are the features of this IC.

- COMPLETE UNIPOLAR HIGH-SIDE
- CURRENT MEASUREMENT CIRCUIT
- WIDE SUPPLY AND COMMON-MODE RANGE
- INA138: 2.7V to 36V
- INDEPENDENT SUPPLY AND INPUT COMMON MODE VOLTAGES
- SINGLE RESISTOR GAIN SET
- LOW QUIESCENT CURRENT (25µA typical)
- WIDE TEMPERATURE RANGE: –40°C to +125°C
- SOT23-5 PACKAGE

3.1.3.2 Design Circuit of Current Measurement

Firstly, let us check the inside of INA138 and figure out the operating principle. Fig. 3 - 7 shows the inside of INA138. It consists of two resisters, an operating amplifier and a triode. This device converts a differential input voltage which is between Pin3 and Pin4 to a current output. Then, the current is converted back to a voltage with an external load resistor which is set at Pin1. By using this device, the current can be measured in a easy and precise way.

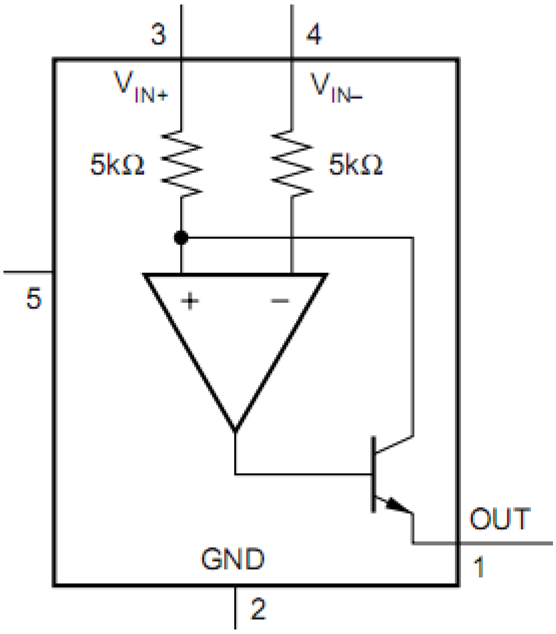


Fig. 3 - 7 : Inside of INA138

As usual, we design the circuit in Pspice and verify the behavior of the circuit. The typical schematic is in the following, as Fig. 3 - 8 showing. According to the equation

$$V_{out} = (I_S)(R_S)(1000\mu A/V)(R_L) \quad \text{(Eq. 3 - 2)}$$

we choose $R_L = 120K\Omega, R_S = 1\Omega$ and the DC voltage sweeps from 0V to 20V.

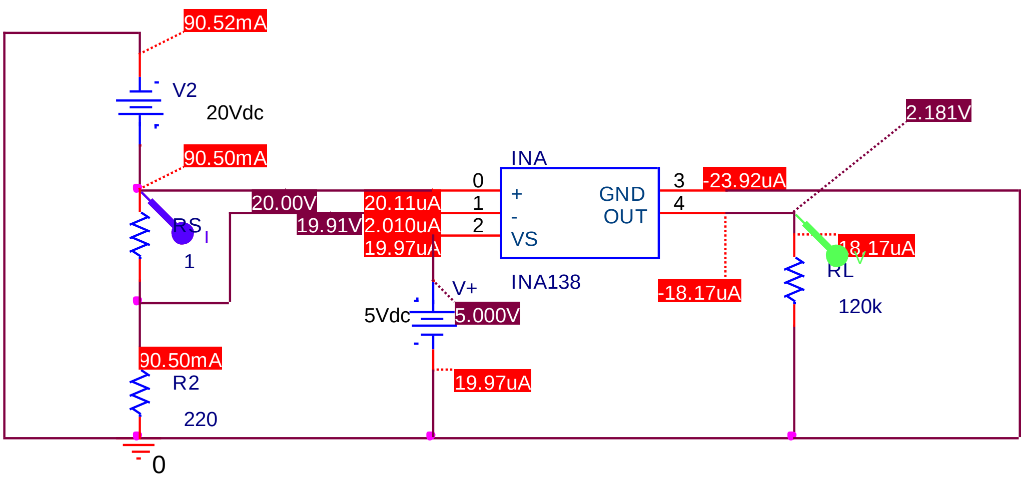


Fig. 3 - 8 : INA138 High Side Current Measurement

3.1.3.3 Test of Current Measurement

The test results are the following. Fig. 3 - 9 shows the relationship between input voltage and input current. We assume $R_L = 120K\Omega$, the power supply for the IC is from 0V to 20V. So the important thing for us is to know the input current of the IC. Here we can easily get the current value from Pspice.

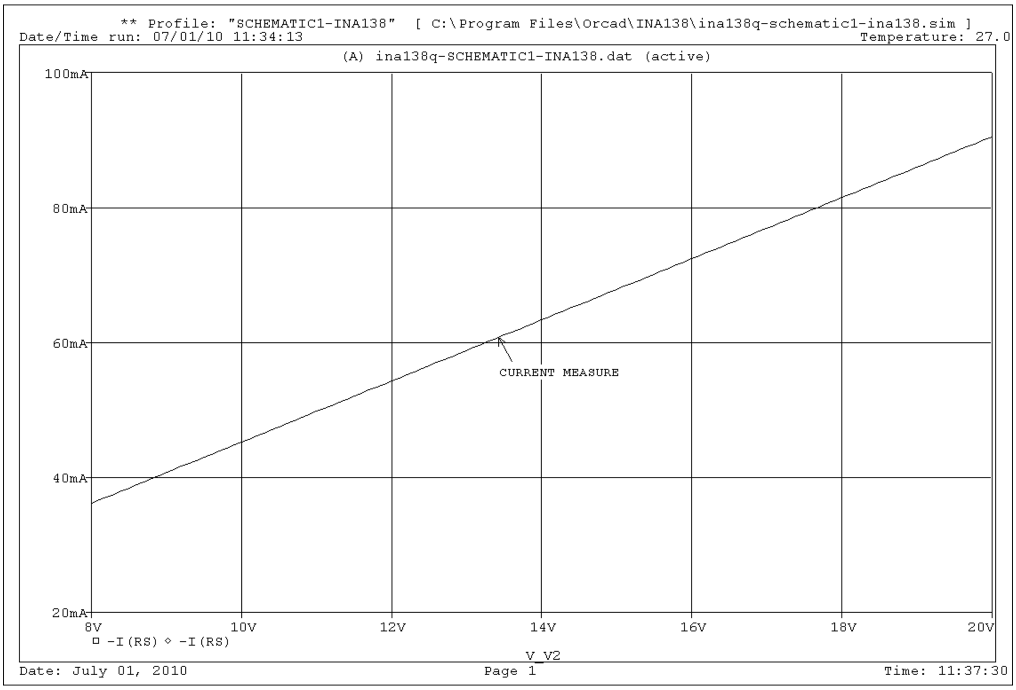


Fig. 3 - 9 : Input Voltage Versus Input Current

As shown in Fig. 3 - 10, we get the value of input voltages and voltage measurement. Then according the test results to verify (Eq. 3 - 2). It is clearly to see the test results are matching with the equation.

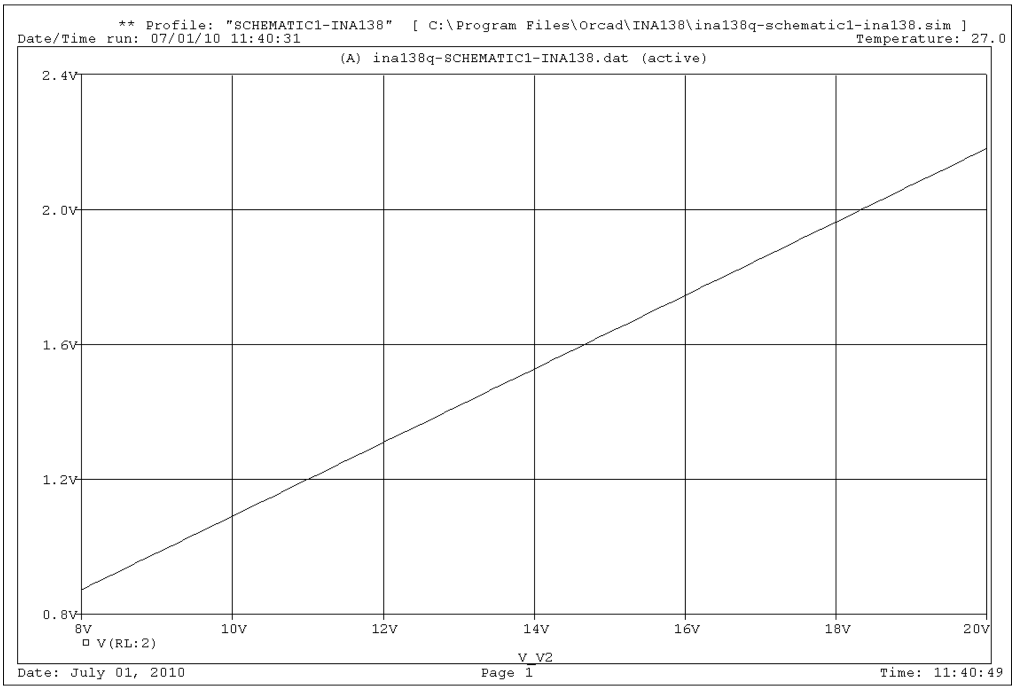


Fig. 3 - 10 : Input Voltage Versus Voltage

The simulations results is not enough, Fig. 3 - 11 is the layout of INA138 high side current measurement. Test results are also shown in the Table 3 - 1. From this table, we verify the function of this circuit. The work of current measurement design is finish. The next procedure is to design the circuit for the L6235.

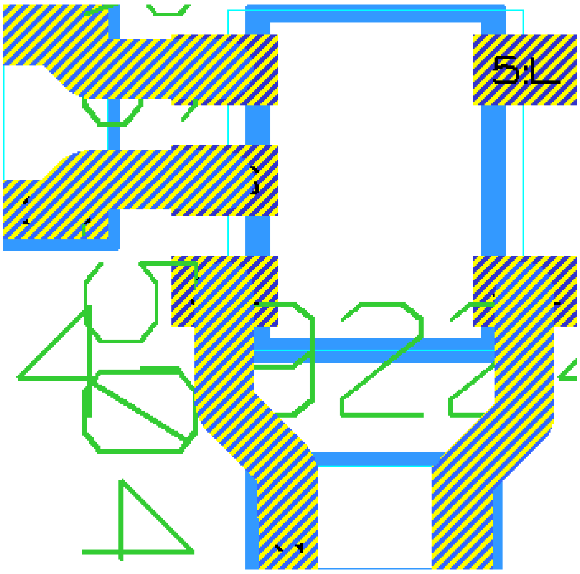


Fig. 3 - 11 : Layout of INA138 High Side Current Measurement

Table 3 - 1 : Simulation Results Versus Measurements Results

Simulation Results			**Measurements Results**		
V_{IN+}	I_S	V_{OUT}	V_{IN+}	I_S	V_{OUT}
3.4V	15.39mA	0.3864V	3.4V	15mA	0.38V
9V	40.73mA	1.022V	9V	40mA	1V
13.4V	60.64mA	1.522V	13.4V	60mA	1.5V
17.4V	78.74mA	1.976V	17.4V	80mA	2V

3.1.4 Design of Speed Control Loop

3.1.4.1 Tacho Design

Tacho is an instrument that measures the rotation speed of a shaft or disk, as in a motor or other machine. The device usually displays the revolutions per minute (RPM) on a calibrated analogue dial. In the IC of L6235, pin8 is called TACHO, it is used as Frequency-to-Voltage open drain output. Every pulse from pin H1 is shaped as a fixed and adjustable length pulse. In order to control the speed of motor in a precise way, design of Tacho control loop is an important work.

A tachometer function consists of a monostable, with constant off time (t_{PULSE}), whose input is H1 which is from Hall Sensor. The monostable output drives an open drain output pin (TACHO). At each rising edge of the H1 signal, the monostable is triggered and at the same time, the MOSFET connected to pin TACHO is turned off for a constant time t_{PULSE}. The off time t_{PULSE} can be set using an external RC network which connected to the pin RCPULSE.

In the purpose to design a speed control loop, the first thing we have to do is to choose a proper value of t_{PULSE}. According to the following equation (Eq. 3 - 3):

$$t_{PULSE} = 0.6 * R_{PUL} * C_{PUL} \quad \text{(Eq. 3 - 3)}$$

We could get the relationships among C_{PUL}, R_{PUL} and t_{PULSE} .

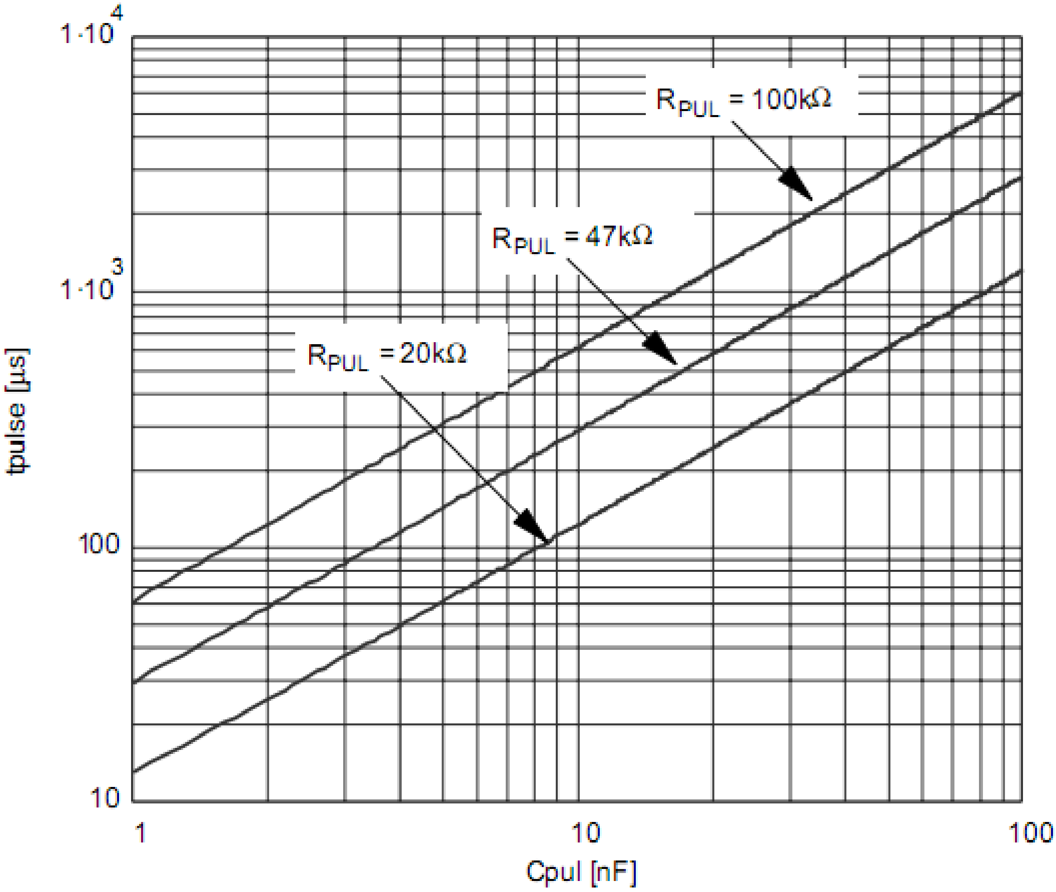


Fig. 3 - 12 : t_{PULSE} Versus C_{PUL} and R_{PUL}

Fig. 3 - 12 indicates that with the increasing of C_{PUL} and R_{PUL}, t_{PULSE} also increases. There are two factors that determine how we choose the value of t_{PULSE}. One is the maximum speed of the motor. If the t_{PULSE} is too large, this will limit the maximum speed. The other reason is limitation of the pulse counter. In the final project, we would design a pulse counter to count the number of t_{PULSE}, it means the number of turns which is very important for us. Since the pulse counter needs some time to hold and sample the signal. The width of the pulse can not be too narrow. Otherwise, the counting error would be risen. On the basis of these two factors, we decide to choose $C_{PUL} = 10nF$ and $R_{PUL} = 47k\Omega$. As a result, we can get $t_{PULSE} = 0.6 * 47k\Omega * 10nF = 282\mu s$. which is suite for our control

system. The following is the Tacho operation waveforms.

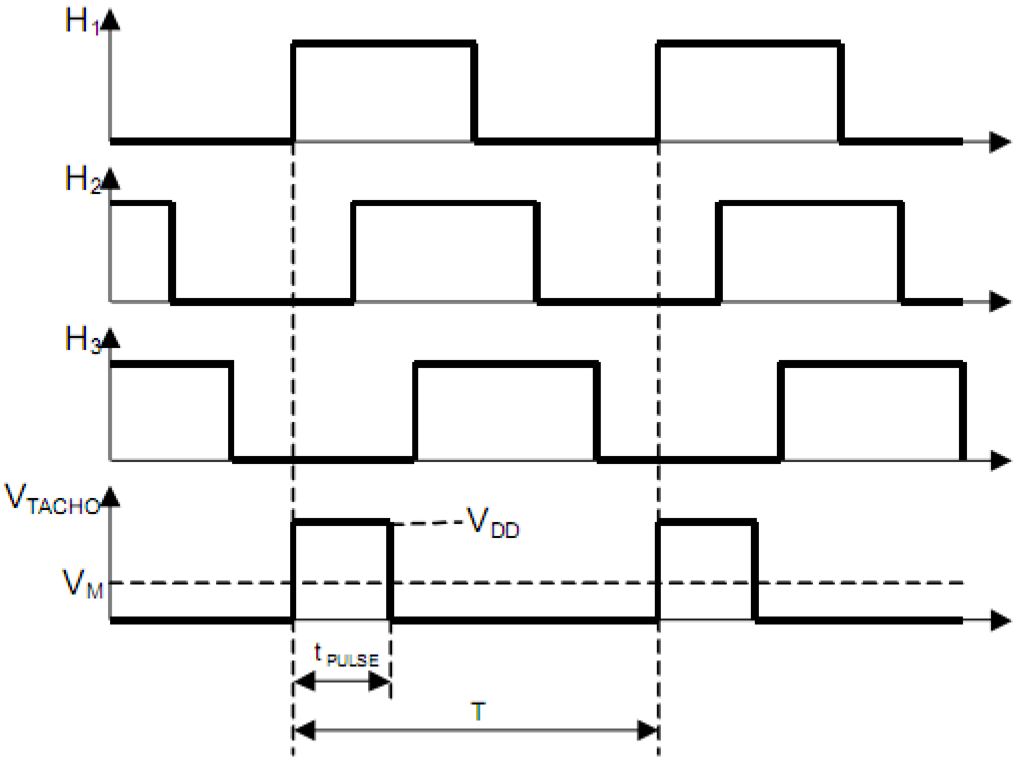


Fig. 3 - 13 : TACHO Operation Waveforms

From Fig. 3 - 13 we can clearly see that on the each rising edge of H1 signal, there will be a rising edge of TACHO and after the duration of t_{PULSE}, it will create a trailing edge of the pulse. Until the next rising edge of H1, this is a whole cycle. It means the motor rotates one turn. Making use of this TACHO signal, we can create a counter to count the number of turns of the motor. It is a quiet easy and accurate way to monitor the working state of the motor.

3.1.4.2 Control Loop Design

We have already designed the Tacho part, the proper pulse signal will come out from Pin TACHO. The next goal is to average this pulse signal and compares it with a reference voltage. By changing the value of reference voltage, we can control the speed of motor in a quick way. The details of this method is in the below.

By connecting the pin TACHO to an external pull-up resistor, the output signal average value VM is proportional to the frequency of the Hall Effect signal and, therefore, to the motor speed. This realizes a simple Frequency-to-Voltage Converter. An op amp which is configured as an integrator filters the signal and compares it with a reference voltage Vspeed, which sets the speed of the motor.

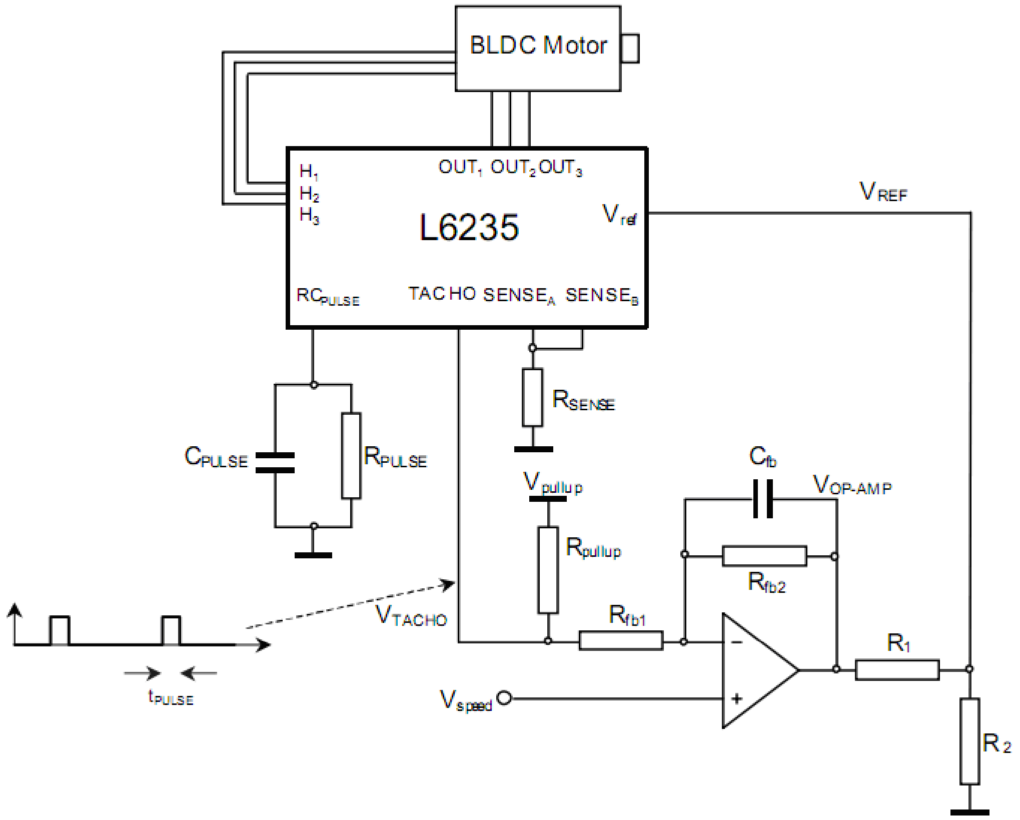


Fig. 3 - 14 : Tacho Speed Control Loop

Fig. 3 - 14 shows Tacho speed control loop. We can see from the above diagram that after comparing with the reference voltage V_{SPEED}, we get a voltage V_{REF}, this is the feedback voltage of the control loop, it will change the speed of the motor.

3.1.4.3 Variable Voltage Design

In the design of control loop, by changing the value of V_{SPEED}, we can control the rotating speed of motor. The only problem for us is how we can get a expect voltage V_{SPEED}. At present, we have two solutions for obtaining a variable voltage.

First one, we can obtain a variable voltage directly from a microcontroller (MCU). Through a digital to analog converter (DAC), a digital signal can be converted into a analog signal, so we can get a fixed voltage. But the problem is not all the series of MCU has DAC device, it means we have to choose an advanced MCU. As a result, the complexity will be increased as well. Since the cost factor is quite important in our project, we decide to give up this solution.

Second method, we can built a low pass filter to obtain a variable voltage. The schematic is shown below.

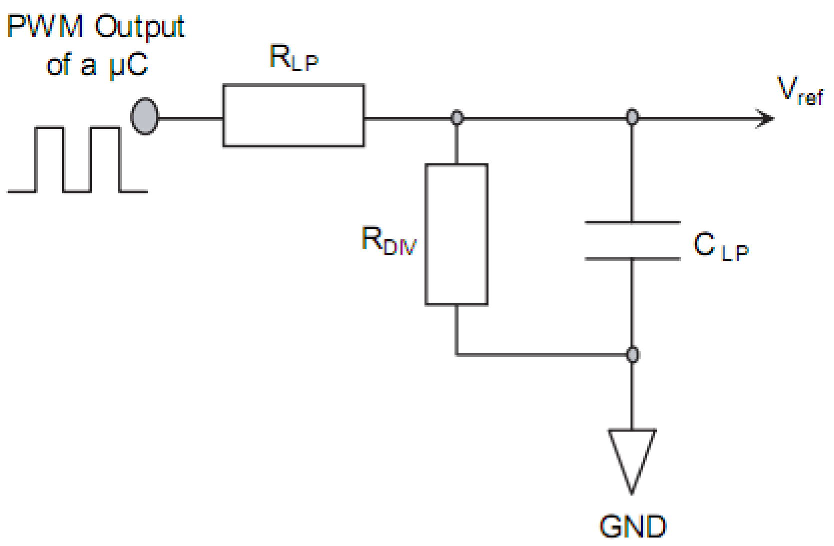


Fig. 3 - 15 : PWM to DC Voltage

Fig. 3 - 15 indicates that any pulse width modulation (PWM) output from a microcontroller can be converted into a DC voltage V_{ref}. By using this circuit, we could get a variable voltage as we need. Obviously, this solution will cost less than the previous one, the only problem is the structure of this circuit is a little complex than using a DAC directly. So we have to find out the relationship between PWM signal and V_{ref}.

Generally, the PWM output voltage ranges is from 0V to 3V, the average voltage V_{ref} will be:

$$V_{ref} = \frac{3V * D_{\mu C} * R_{DIV}}{R_{LP} + R_{DIV}} \qquad \text{(Eq. 3 - 4)}$$

where $D_{\mu C}$ is the duty-cycle of the PWM output from the MCU.

We choose R_{LP} = 56kΩ, R_{DIV} = 45kΩ, C_{LP} = 10nF and a PWM signal changing from 0 to 3V at 100kHz, the low pass filter time constant is about 0.12 ms and the remaining ripple on the V_{ref} voltage will be about 20 mV. If we use higher values for R_{LP}, R_{DIV} and C_{LP}, the ripple on the V_{ref} will reduce, but the reference voltage will take more time to vary after changing the duty-cycle of the MCU PWM, so the response time of the system will be too high. Not only this reason, high values of R_{LP} will also increase the impedance of the V_{ref} net at low frequencies which will cause a poor noise immunity.

Fig. 3 - 16 shows the simulation circuit of PWM to DC voltage. Here we choose a PWM signal which amplitude is 3V, Time Delay = 0, rise time = 0.01us, fall time = 0.01us, pulse width = {dutycycle*10} (set as a global parameters), period = 10us.

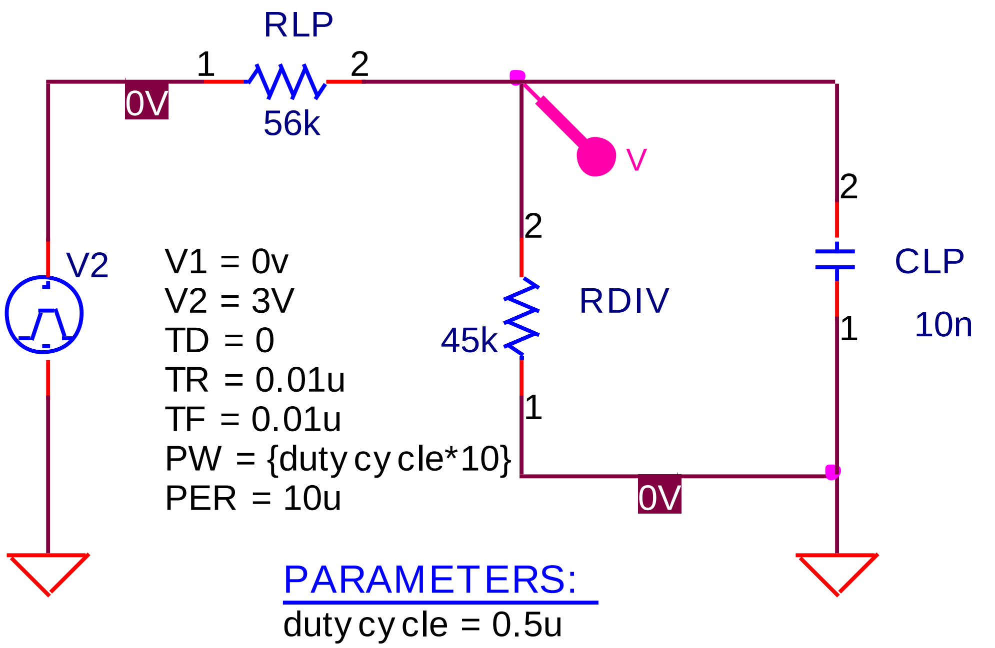


Fig. 3 - 16 : PWM to DC Voltage Simulation

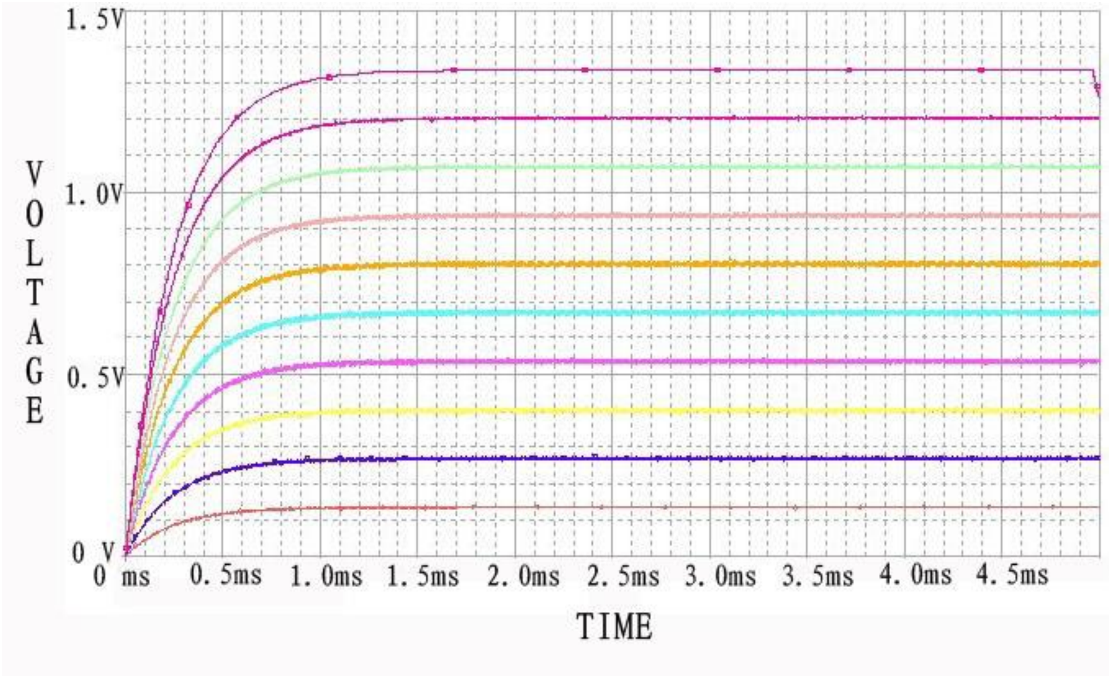


Fig. 3 - 17 : Variable Voltage Simulation By Changing Dutycycle

From the above Fig. 3 - 17 in which X axis represents time and Y axis represents voltage value measured at output, we can get the conclusion that by changing the duty cycle from 10% to 100%, the output DC voltage keeps increasing, so we can get a variable voltage which range is from 0V to 1.35V. This voltage range is fit for speed control loop. So the second way of generating variable voltage is better. Until now, we finish the design of speed control loop.

3.1.5 Final Design

3.1.5.1 Circuit Diagram

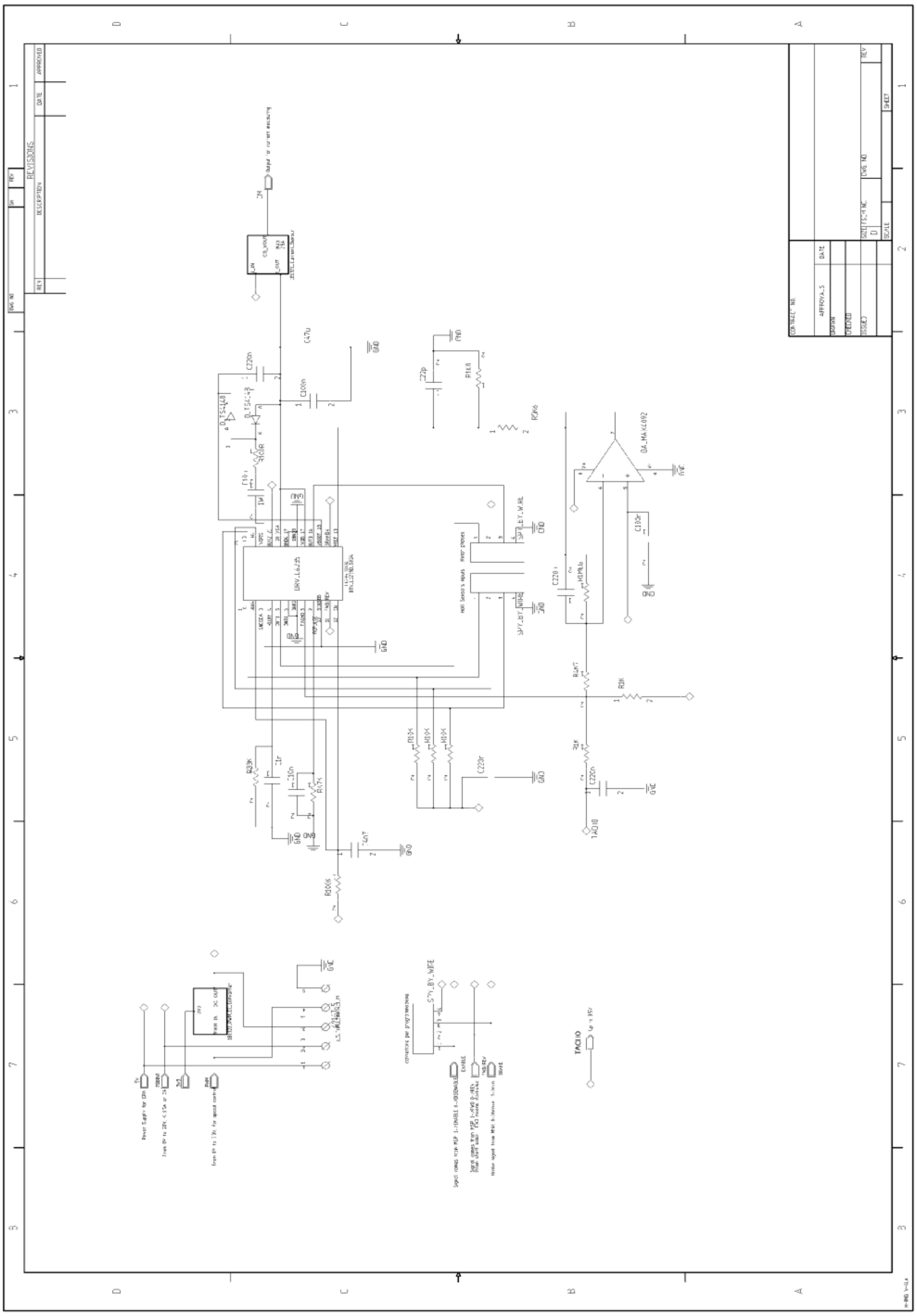


Fig. 3 - 18 : Circuit Diagram of Motor Driver

Fig. 3 - 18 shows the motor driver circuit which is made by Mentor Graphics.

The whole circuit is consist of four parts, voltage regulator module, current measurement module, speed control loop module and L6235 IC control module.

3.1.5.2 PCB Layout

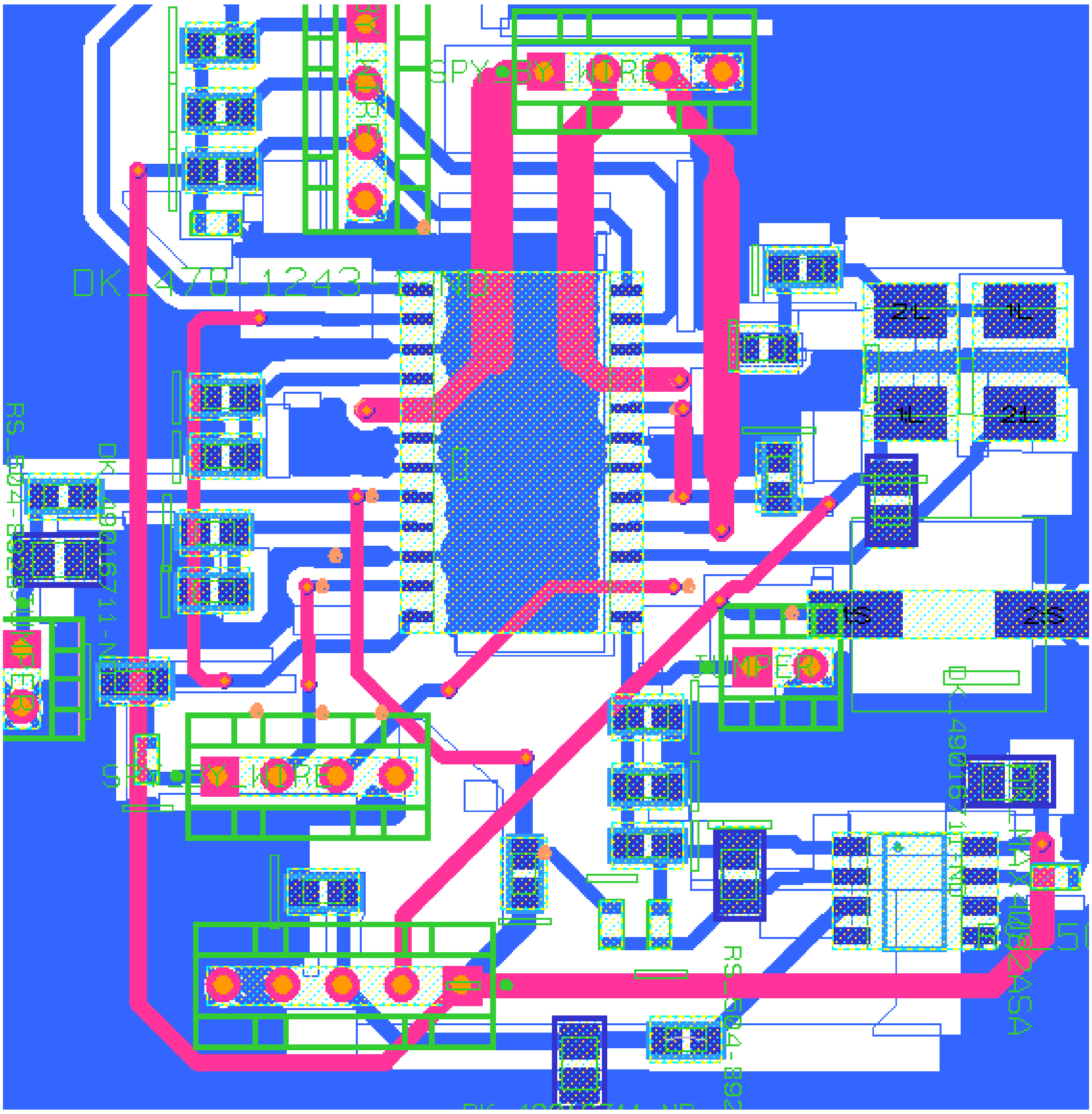


Fig. 3 - 19 : PCB Layout of Motor Driver

Fig. 3 - 19 is the PCB layout of motor driver circuit, which is a double layer PCB. The blue layer is the top side and pink layer is the bottom side. In order to dissipate the heat and reduce electromagnetic interference, a ground layer has been

created on the top which will decrease the noise as well. Due to the low fabrication technology, all the wires on the PCB board are larger than 0.45mm. Since it is a test board, all the test points use a socket which is convenient for the test.

3.2 Hardware Test

3.2.1 Tacho Output Test

Goal :

Verify the hall sensor signal and Tacho output signal.

Test condition:

Vs = 10V; Vspeed= 0.5V;

Test Instruments:

DC power supply : GW DUAL TRACKING WITH 5V FIXED, MODEL: GPC-3030D

Oscilloscope : Lecroy wavesurfer 44Xs

Multimeter : vellman DVM9912

Layout Consideration:

Due to the current of the output is very large, the interference between hall sensor signal and output signal is quite large. The good way is to separate these two parts on the PCB board. So we mount hall sensor socket on the top side and output socket on the bottom side. Moreover, we put hall sensor socket close to the IC, the error for decoding will be small, this is greatly important for the decoding logic. About the output socket, since the current of the output is quite large, it is better to increase the width of the wire. Lastly, due to the reason of Electromagnetic Interference (EMI), we put hall sensor wires and output wires in cross shaped. That would decrease EMI as much as possible.

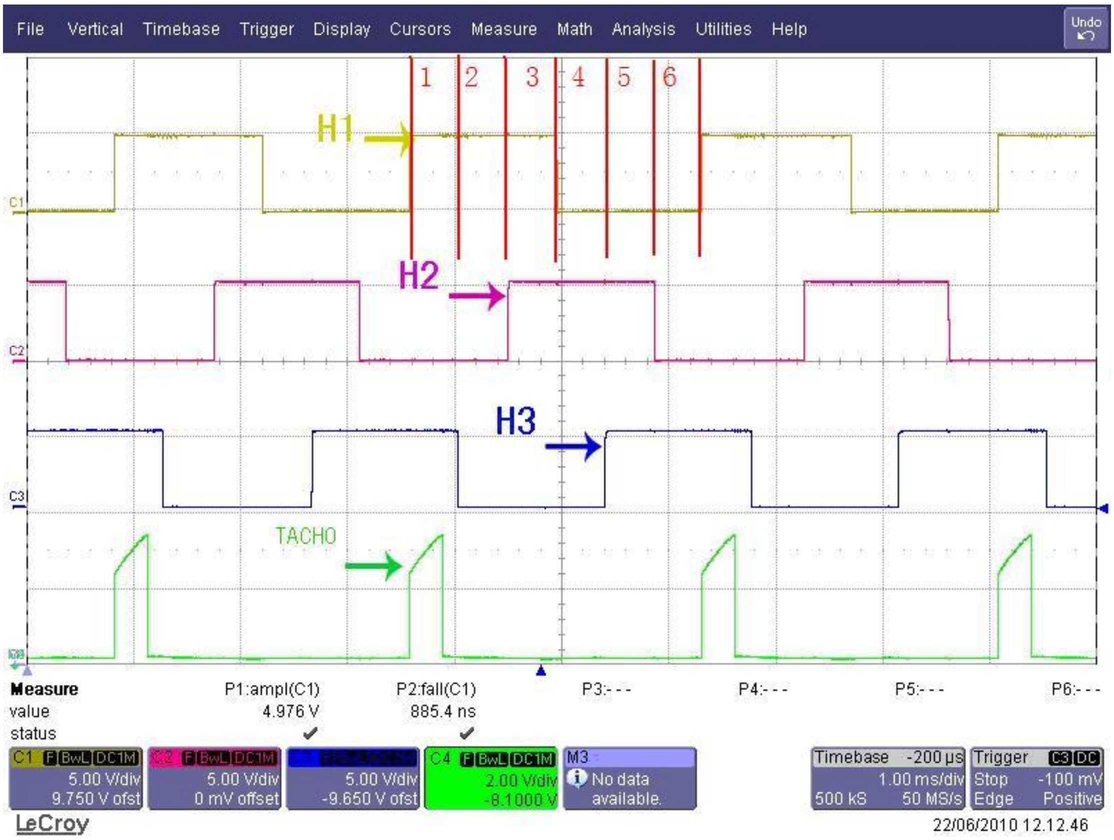


Fig. 3 - 20 : Tacho operation waveforms

Fig. 3 - 20 indicates the decoding logic and Tacho operation waveforms. As we can see from the figure, at the beginning of stage 1, the Tacho output signal changes from 0 to 1, then it goes to 0 again after t_{PULSE}. The reason why Tacho waveform is not a rectangle is we add a low pass filter at the output port of Tacho. This low pass filter is used to get rid of high frequency components from Tacho waveforms.

Table 3 - 2 : Hall Sensor Signal Decoding

Stage	H1	H2	H3
1	*High*	*Low*	*High*
2	*High*	*Low*	*Low*
3	*High*	*High*	*Low*
4	*Low*	*High*	*Low*
5	*Low*	*High*	*High*
6	*Low*	*Low*	*High*

Table 3 - 2 shows the decoding logic. As the above figure showing, we split one cycle into six stages, the hall sensor logic can be shown in the following table.

3.2.2 RC Pulse Test

Goal :

Verify the synchronization between RC pulse and Tacho output.

Test condition:

Vs = 10V; Vspeed = 0.5V;

Test Instruments:

Same as before

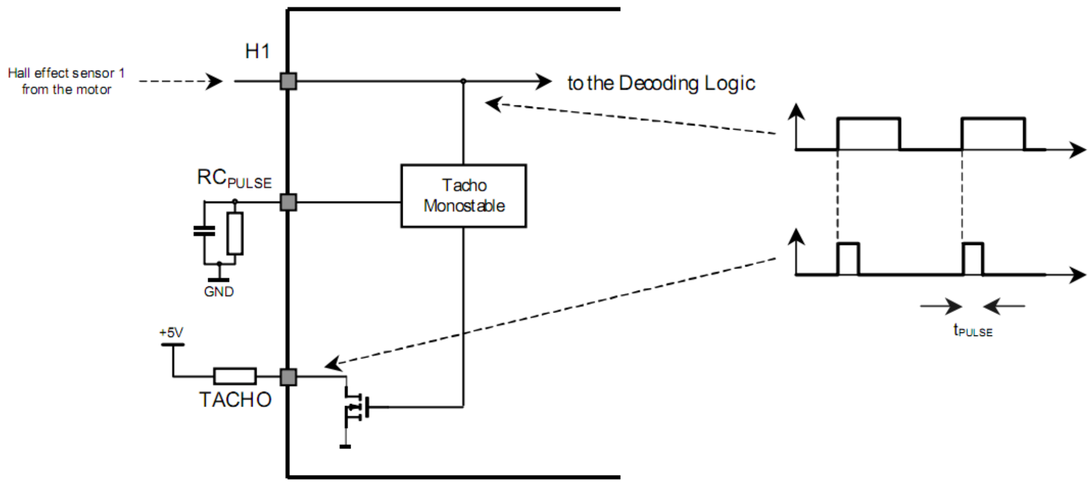


Fig. 3 - 21 : Tacho Monostable

H1 input is internally connected to a monostable, then through an open drain MOSFET, it will generate a fixed width pulse on the TACHO output (see Fig. 3 - 21). By using this output realizing a speed control loop is very easy and inexpensive.

Using an external pull-up resistor on the open drain output, the resulting waveform at the TACHO pin will be a pulse whose frequency is proportional to the motor rotation speed, and also with a fixed on-time (t_{PULSE}) set by an external RC network connected at the RCPULSE pin.

Fig. 3 - 22 illustrates the synchronization between RC pulse and Tacho output. Each RC pulse will generate a Tacho output. The width of these two signals are same, so the proper width of t_{PULSE} can be chosen by changing the value of R_{PULSE} and C_{PULSE}.

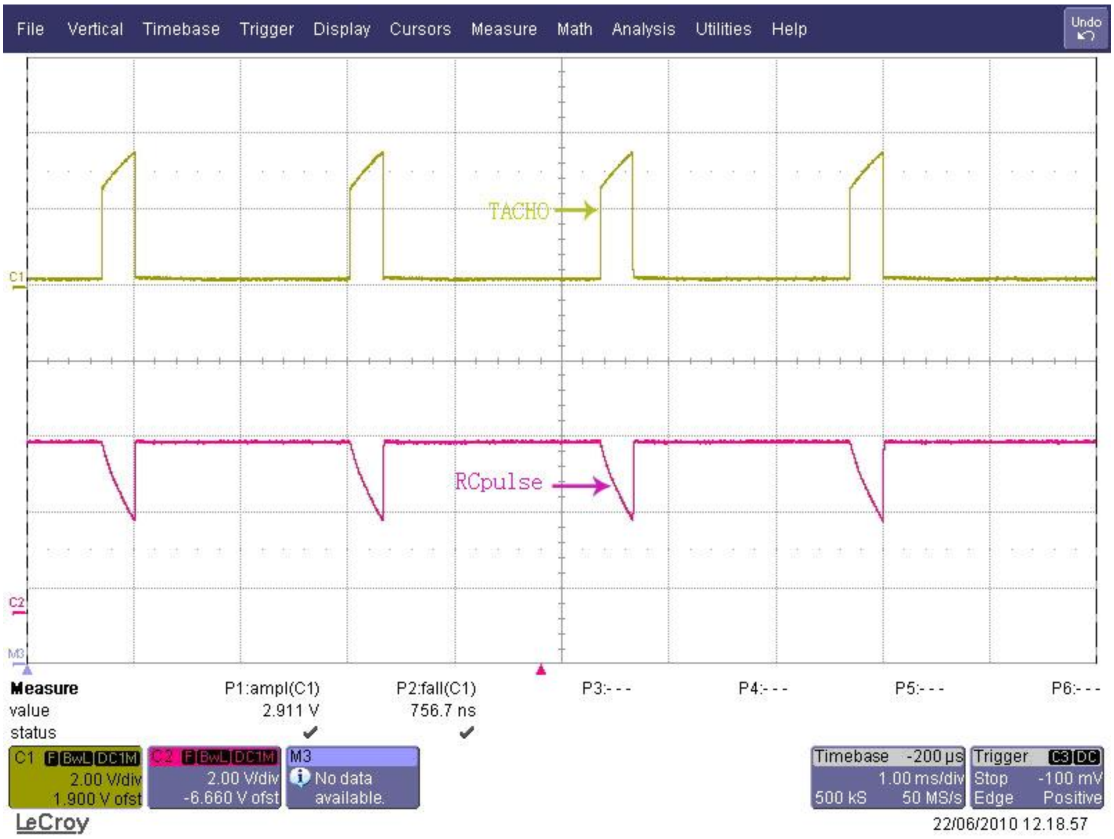


Fig. 3 - 22 : RC pulse Versus Tacho output

3.2.3 Speed Control Loop Test

3.2.3.1 Stop and Running Situation Test

Goal :

Test the motor in stop and running situation and verify the relationship between Vspeed and Tacho output.

Test condition:

Vs = 10V; Vspeed = 0V (in stop situation); Vspeed = 0.4V(in running situation).

Test Instruments:

Same as before

Layout Consideration:

Due to the distance between Tacho and amplifier is very long, EMI is quite large. The noise from amplifier is also large, so the worst situation is when the Vspeed is set at 0V, the motor does not stop. It will keep running in a low speed. The solution to solve this problem is to increase the width of Vspeed wire and also increase the distance between L6235 and MAX4092 which will decrease the EMI. Furthermore, by changing the value of R2 which is the resistor used in speed control loop, this problem can be solved as well.

Fig. 3 - 23 : Stop Situation Test

Set Vspeed at 0V, the motor will be stopped, so there will be no Tacho output pulse (see at Fig. 3 - 23). It is obviously that Vref equals to 0V when Vspeed is set at 0V. Therefore, the motor will be stopped and there is no pulse coming from

Tacho output. Actually, Vref is not equal to 0 all the time. Due to the noise, the value of Vref is a little different from 0. Since it is too small, the motor is always in stop situation. It proves that our solution for layout is right.

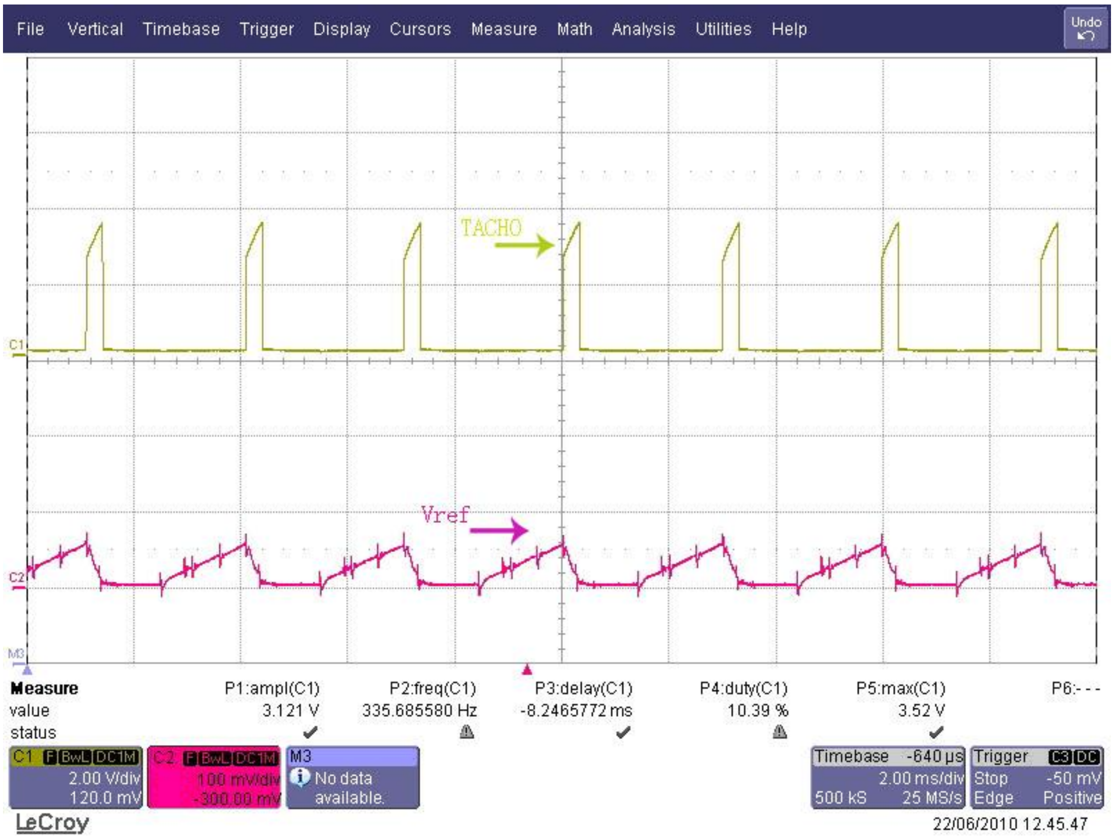


Fig. 3 - 24 : Running Situation Test

Fig. 3 - 24 indicates the running situation. In this test, Vs = 10V, Vspeed = 0.4V. From the above figure, we can get that the frequency of Tacho pulse is 335Hz. Since the Vref is not equal to 0, by using a comparator, the running speed of the motor can be controlled.

3.2.3.2 Speed Control Loop Test With Varying Vspeed

Goal :

Test speed control loop and find out the relationship between speed and Vspeed.

Test condition:

Vs = 10V; Vspeed= 0V – 3V;

Test Instruments:

Same as before

Table 3 - 3 : Speed Test with Varying Vspeed

Vspeed	Amp	Tacho Freq.	Duty	Max
0V	61mV	0	0	0
0.2V	3.091V	135Hz	4.19%	3.48V
0.4V	3.121V	336Hz	10.39%	3.52V
0.6V	3.031V	551Hz	17.06%	3.53V
0.8V	3.317V	573Hz	17.73%	3.54V
1.0V	3.119V	568Hz	17.56%	3.55V
1.6V	2.999V	573Hz	17.71%	3.59V
2.0V	3.190V	573Hz	17.71%	3.62V
2.4V	3.046V	574Hz	17.69%	3.64V
3.0V	3.191V	573Hz	17.78%	3.69V

In Table 3 - 3 we can get the conclusion that with the increasing of Vspeed, the motor speed and t_{PULSE} duty cycle are increased in the meantime. The amplitude of Tacho pulse almost keeps at a constant level while the maximum value of Tacho pulse is gradually increasing. Until Vspeed equals to 0.8V, the speed of the motor does not increase anymore (see at Fig. 3 - 25). While Vspeed is smaller than 0.8V, the relationship between Vspeed and Tacho output pulse frequency is linearity. As we have designed in variable voltage design, using a PWM generated from MCU, we can get a variable voltage Vspeed which range is from 0V to 1.35V. In this test, it is obviously that Vspeed range from 0V to 1.35V is enough for this situation.

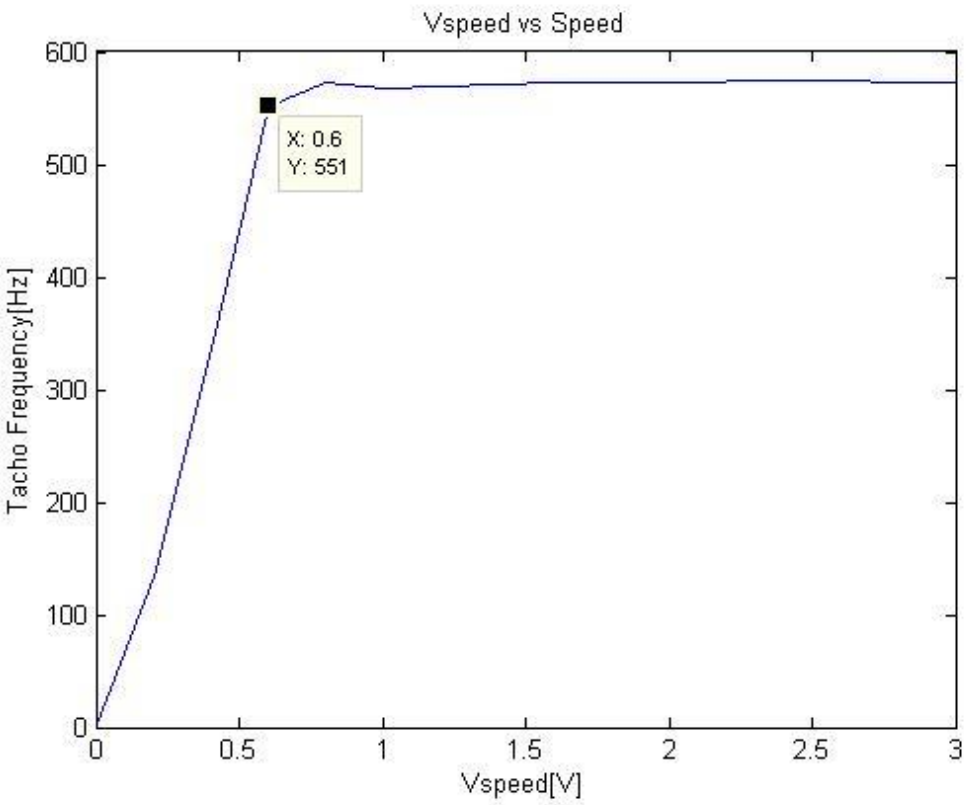


Fig. 3 - 25 : Vspeed Versus Speed

3.2.3.3 Speed Control Loop Test With Varying Vs

Goal :

Test speed control loop and find out the relationship between speed and Vs.

Test condition:

Vs = 6V - 10V; Vspeed=0.43V;

Test Instruments:

Same as before

Table 3 - 4 shows that when Vs is equal to 6V, the motor does not work. Until Vs up to 6.5V, the motor is startup. With the rising of Vs, the frequency of Tacho output pulse and duty cycle of Tacho output pulse are gradually increased as well. The relationship between Vs and Speed is almost linear (see at Fig. 3 - 26).

Table 3 - 4 : Speed Test with Varying Vs

Vs	Amp	Freq	Duty	Max
6V	0	0	0	0
6.5V	2.851V	340Hz	10.52%	3.29V
7V	2.928V	332Hz	10.26%	3.52V
7.5V	2.933V	341Hz	10.55%	3.54V
8V	3.301V	351Hz	10.87%	3.52V
8.5V	3.094V	364Hz	11.27%	3.54V
9V	3.074V	377Hz	11.67%	3.53V
9.5V	2.983V	382Hz	11.83%	3.52V
10V	2.970V	394Hz	12.21%	3.53V

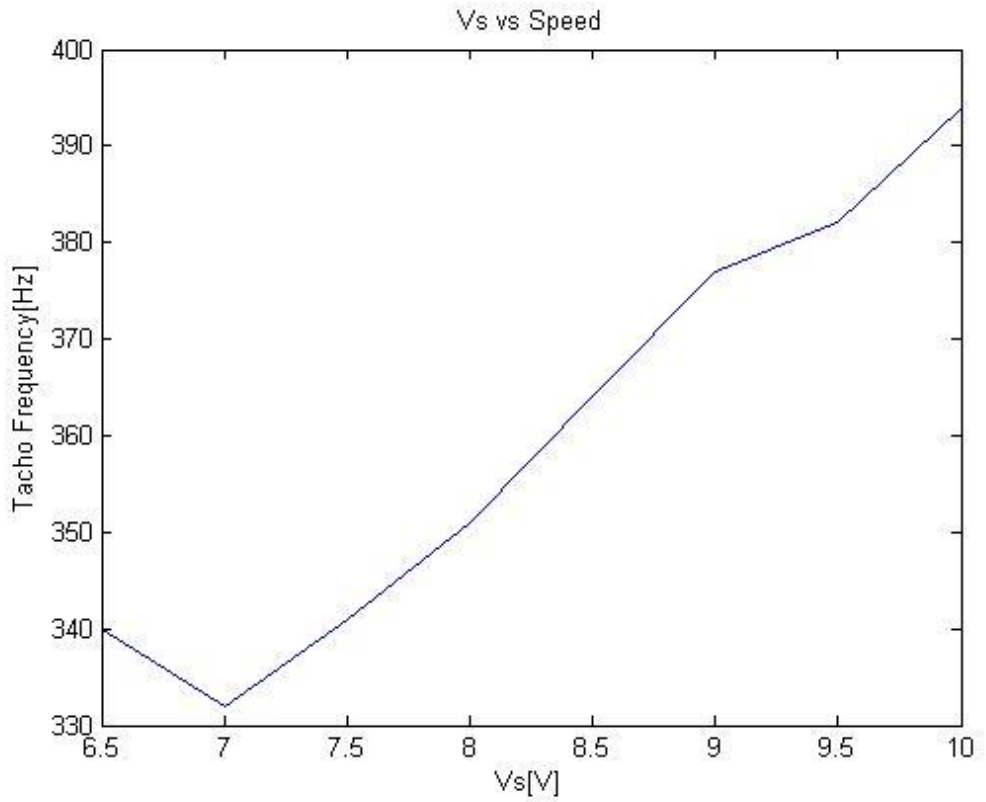


Fig. 3 - 26 : Vs Versus Speed

Chapter 4 Software Design

4.1 Software Implementation

Before coming into the details of software implementation, the basic functions provided by motor control system have to be presented at first. Since the control system is separated into two parts, one is ground control system and another one is outer space control system. These two systems can communicate with each other through antennas, so the whole process would be like the following. When people send a command from ground station and it is received by antenna which is set at outer space, then the control system will decode the command and execute the command. For the motor control system, there are al least three functions have to be implemented.

1. Motor Timer
2. Motor Counter
3. Current Measure

4.1.1 Motor Timer

4.1.1.1 Timer_A

The first requirement for the software design is to keep the motor running during a specified time. In order to implement this function, Timer_A of MSP430 has to be introduced [10],[11].

Timer_A is a 16-bit timer/counter with multiple capture/compare registers. It supports multiple capture/compares, PWM outputs, and interval timing. Timer_A also has extensive interrupt capabilities. Interrupts may be generated from the counter on overflow conditions and from each of the capture/compare registers. The features of Timer_A are the following:

- A 16-bit counter
- 4 modes of operation –Stop, Up, Continuous, Up/Down
- 3 capture/compare registers
- 2 interrupt vectors –TACCR0 and TAIV

Table 4 - 1 descries timer modes of MSP430. There are four different modes for the timer, due to the different application, a proper modes can be chosen.

Table 4 - 1 : Timer Modes

MCx	Mode	Description
00	Stop	The timer is halted.
01	Up	The timer repeatedly counts from zero to the value of TACCR0.
10	Continuous	The timer repeatedly counts from zero to 0FFFFh.
11	Up/down	The timer repeatedly counts from zero up to the value of TACCR0 and back down to zero.

In this project we need is a count down timer, so up mode is suit for our purpose. Fig. 4 - 1 shows up mode diagram. A fixed time interval is assigned at first, then the MCU start to count. Every time when the count number equals to TACCR0, Timer_A interrupt will be triggered.

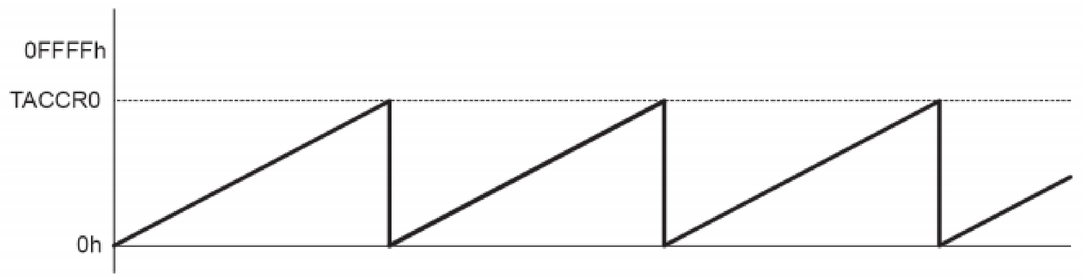


Fig. 4 - 1 : Up Mode of Timer_A

The next thing is to set timer. With the purpose of choosing an expected time interval, the function of timer register has to be brought in.

15	14	13	12	11	10	9	8
Unused						TASSELx	
rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)

7	6	5	4	3	2	1	0
IDx		MCx		Unused	TACLR	TAIE	TAIFG
rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	w–(0)	rw–(0)	rw–(0)

Fig. 4 - 2 : Timer_A Control Register

Fig. 4 - 2 describes the Timer_A control register. There are 16 bits in this register, each of them has different meaning. The important bits are the following:

TASSELx Bits 9-8 Timer_A clock source select

00 TACLK

01 ACLK

10 SMCLK

11 Inverted TACLK

MCx Bits 5-4 Mode control

00 Stop mode

01 Up mode

10 Continuous mode

11 Up/down mode

According to the above data, a proper timer can be selected.

4.1.1.2 Motor Timer Implementation

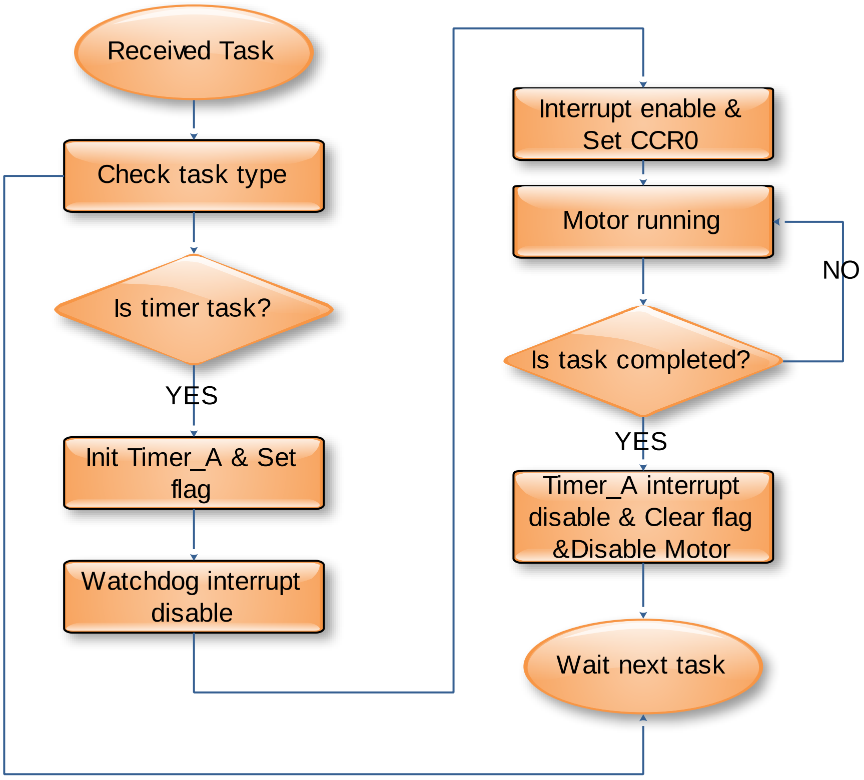


Fig. 4 - 3 : Motor Timer Flow Chart

Fig. 4 - 3 indicates that while MCU received a task and check the task type, if it is a timer task, the MCU will start to initialize Timer_A and set flag as 1 which means the motor is running and during this period, MCU will not accept any task anymore. After that, watchdog timer is set to disable and interrupt is set to enable. The important step is to choose a proper interrupt time interval. In the purpose of saving power, the longer of time interval, the more power saving. After testing, one second is a suitable time interval for this task. Since ACLK frequency of MCU is 32768Hz, and the timer works in up mode. If CCR0 equals to 16384, the timer interval will be set as one second. Every second MCU will wake up and check the task has been completed or not. If the task has been finished, MCU will set flag to 0 and send a stop signal to motor. In other words, the enable port of motor driver will be set as 0. Therefore, the motor will be stopped and MCU wait for a next task.

4.1.1.3 Motor Timer Test

Table 4 - 2 : Motor Timer Test Results

Set Time	Actual Time	Error
10s	11.8s	*18%*
50s	54.6s	*9.2%*
100s	103.7s	*3.7%*
500s	504.5s	*0.9%*

Table 4 - 2 shows the results of motor timer test. From this table, we can get the conclusion that the longer time we set, the smaller error we got. It was easy to understand that since the delay of circuit and EMI among the signals, the error must exist. While the set time is larger, the error rate will be smaller. In order to reduce the error, the efficient way is to decrease the EMI of the board.

4.1.2 Motor Counter

The second requirement for the software design is to keep the motor running for a specified number of turns. In order to implement this function, the Tacho output pin is connected to MCU. Firstly, MCU receives a task and check task type. If it is a counter task, the initial process will be started and the flag will be set as 1 which means the motor is running and it does not accept any task until the flag is set as 0. the next step is to stop the interrupt of watchdog. After that, the wire connecting with pin Tacho should be selected as a input pin. The pulse from pin Tacho is 0V to 3V (see at Fig. 4 - 4). When the input voltage equals to 3V, the input value of MCU is 1, otherwise, it is 0. So a state flag can be used to mark the state of input value. If the input values changes, it also means that there is either a rising edge or a falling edge. By counting the number of edge changed, the number of turns can be known. The next step is to check whether the task has been finished. If the task is finished, the MCU will set flag to 0 and stop the motor. If not, the motor will keep running and counting the number of turns until the task is finished. The whole process can be seen from Fig. 4 - 5.

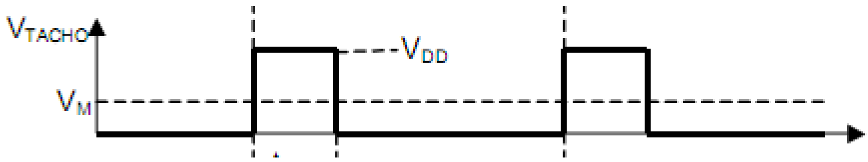


Fig. 4 - 4 : Tacho Waveforms

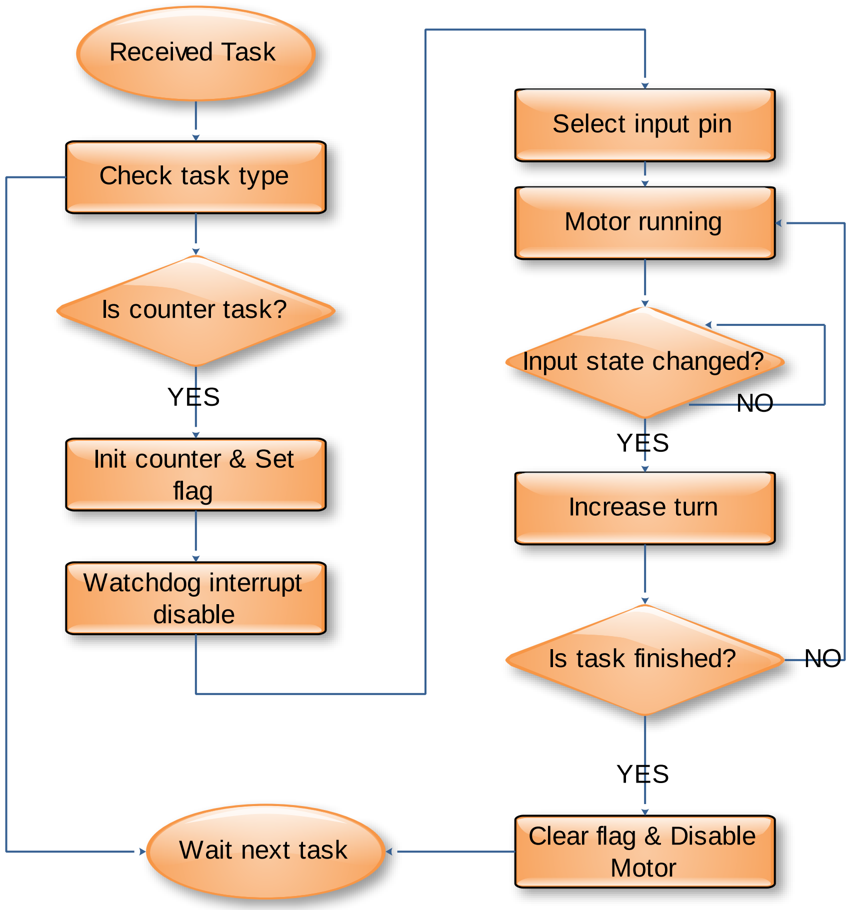


Fig. 4 - 5 : Motor Counter Flow Chart

Table 4 - 3 is the results of motor counter test. The error keeps around 5% which is suit for Aramis project. The mainly reason for the error is inertia of the motor. To solve this problem, the easiest way is when we set the turns

of the motor, it is better to consider the error in the setting. In this way, the error can be reduced aggressively.

Table 4 - 3 : Motor Counter Test Results

Set Turns	Actual Turns	Error
100	107	*7%*
200	211	*5.5%*
500	538	*7.6%*
1000	1067	*6.7%*

4.1.3 Current Measure

4.1.3.1 ADC12

The ADC12 module is a high-performance 12-bit analog to digital converter (ADC12). The ADC12 is implemented in the MSP430x43x MSP430x44x, and MSP430FG461x devices [12]. This module implements a 12-bit SAR core, sample select control, reference generator and a 16 word conversion-and-control buffer. The conversion-and-control buffer allows up to 16 independent ADC samples to be converted and stored without any CPU intervention. Features of ADC12 are the following:

- Greater than 200-ksps maximum conversion rate
- Monotonic 12-bit converter with no missing codes
- Sample-and-hold with programmable sampling periods controlled by software or timers.
- Conversion initiation by software, Timer_A, or Timer_B
- Software selectable on-chip reference voltage generation
- Software selectable internal or external reference
- Eight individually configurable external input channels

- Conversion channels for internal temperature sensor, AVCC, and external references
- Independent channel-selectable reference sources for both positive and negative references
- Selectable conversion clock source
- Single-channel, repeat-single-channel, sequence, and repeat-sequence conversion modes
- ADC core and reference voltage can be powered down separately
- Interrupt vector register for fast decoding of 18 ADC interrupts
- 16 conversion-result storage registers

In these features, the first important thing is conversion modes. The ADC12 module has four conversion modes.

1. Single channel single-conversion
   - A single channel is converted once
2. Sequence-of-channels
   - A sequence of channels is converted once
3. Repeat-single-channel
   - A single channel is converted repeatedly
4. Repeat-sequence-of-channels
   - A sequence of channels is converted repeatedly

Due to the different conversion modes, the ADC12 can be used in different applications. Since our goal is counting the pulse number from Tacho, we can choose the first conversion mode. For the purpose of choosing a proper sampling rate, Fig. 4 - 6 is greatly important.

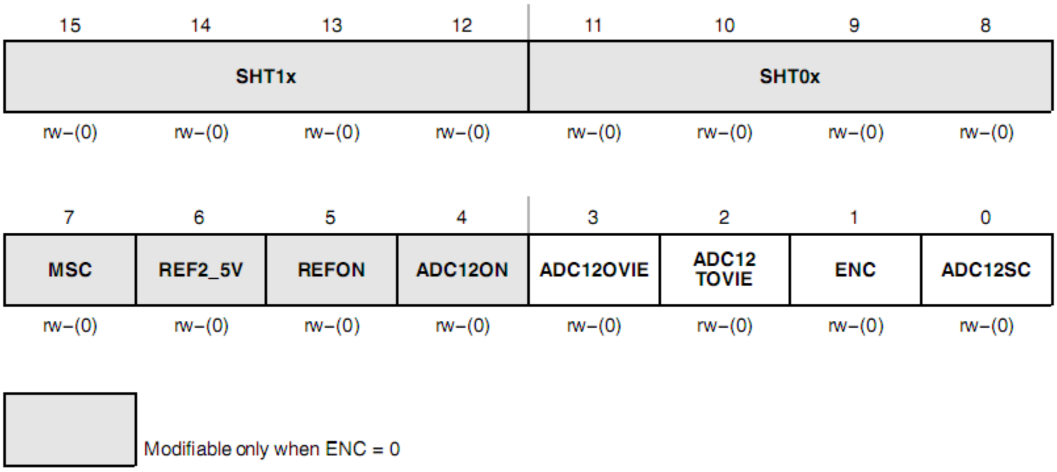


Fig. 4 - 6 : ADC12 Control Register 0

The first section is SHT1x whose range is from bits 15 to 12. The function of this section is to set sample-and-hold time. These bits define the number of ADC12CLK cycles in the sampling period for registers ADC12MEM8 to ADC12MEM15.

The second section is SHT0x which is from 11 bit to 8bit. It has the same function with SHT1x. The different thing is these bits define the number of ADC12CLK cycles in the sampling period for registers ADC12MEM0 to ADC12MEM7.

Table 4 - 4 : SHTx Bits vs. CLK cycles

SHTx Bits	CLK cycles	SHTx Bits	CLK cycles
0000	*4*	*1000*	*256*
0001	*8*	*1001*	*384*
0010	*16*	*1010*	*512*
0011	*32*	*1011*	*768*
0100	*64*	*1100*	*1024*
0101	*96*	*1101*	*1024*
0110	*128*	*1110*	*1024*
0111	*192*	*1111*	*1024*

From Table 4 - 4 an appropriate sample and hold time can be selected. For example, if ADC internal oscillator (ADC12OSC) is used as ADC12 clock source, frequency of ADC12OSC is about 5MHz, if set SHT0x to 0011, then the sample and hold time is 32*(1/5000000) second = 6.4 μ s.

The other bits such as ADC12ON which is used to turn on or turn off ADC12, ENC which is used to enable or disable ADC12 and REFON which is used to turn on or turn off reference generator.

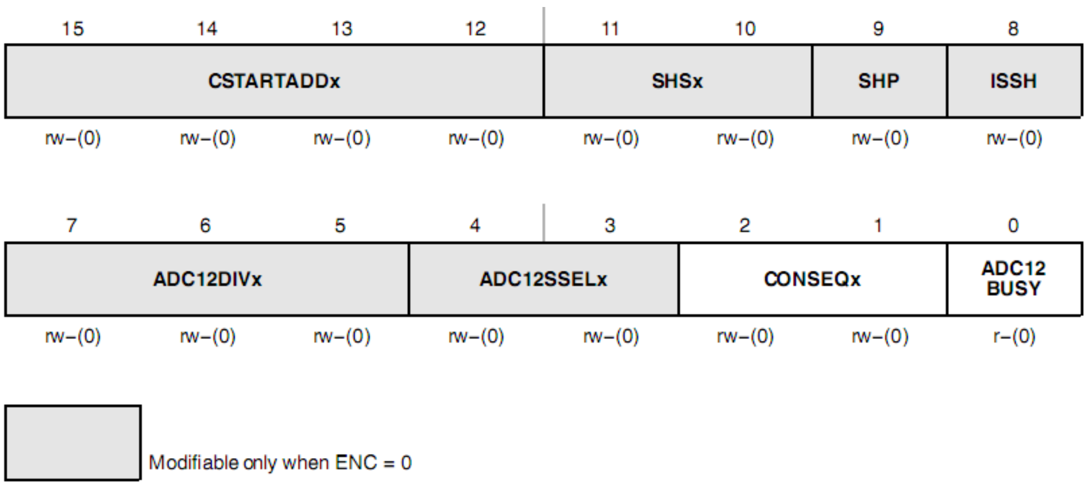


Fig. 4 - 7 : ADC12 control register 1

Since ADC12OSC has to be used as ADC12 clock source, and also single channel, single conversion mode has to be selected, Fig. 4 - 7 ADC12 control register 1 must be introduced as well.

SHSx is used for selecting sample and hold source. Table 4 - 5 indicates how to select sample and hold source which is used to trigger a conversion.

Table 4 - 5 : Sample and Hold Source Select

BITs	**SOURCE**
00	ADC12SC bit
01	Timer_A.OUT1
10	Timer_B.OUT0
11	Timer_B.OUT1

The next important part is ADC12SSELx which controls the selecting of ADC12 clock source. If it equals to 00, ADC12OSC is selected as ADC12 clock source.

CONSEQx is significant as well. It was clearly to see that by changing this part, different sequence mode can be selected.

Table 4 - 6 shows different sequence mode by setting different bits. It was clearly to see that by changing this part, different sequence mode can be selected.

Table 4 - 6 : ADC12 clock source select

BITs	**CLOCK SOURCE**
00	ADC12OSC
01	ACLK
10	MCLK
11	S MCLK

4.1.3.2 Current Measure Implementation

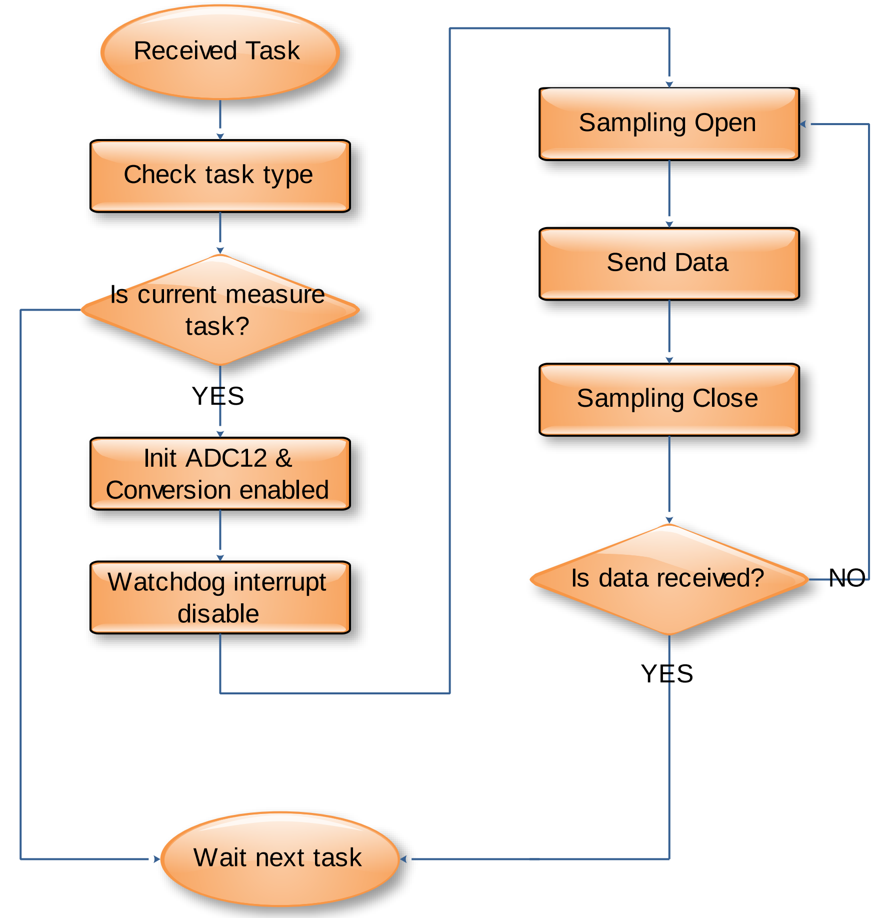


Fig. 4 - 8 : Current Measure Flow Chart

Fig. 4 - 8 indicates the current measure procedure. While the control system received a command from ground station, it decodes the command and check the task type. If the task is to measure the current, ADC12 will be initialed at once and the conversion will be enabled, otherwise the control system change into waiting mode again. After the initial process, watchdog interrupt is set to disable and ADC12 is set to sampling open. The sampling data will be send to ground station subsequently and the sampling is closed. If the data is received by ground station, the control system will be into waiting mode, if not, the sampling process will be started once again.

Chapter 5 Conclusion

With the development of miniaturization technology, nanosatellite are paid great attention and developed rapidly. Many universities, including Politecnico di Torino have developed their own nanosatellite. Aramis is the second nanosatellite made by Politecnico di Torino, with a design focused on a new modular architecture .

In this thesis, the design of motor control system for nanosatellite attitude control is described. In order to get the proper attitude in the outer space, a precise motor control system is necessary. Meanwhile, since the special situation of outer space, lifespan and energy consumption and many other factors must be taken into account.

In chapter 2, the motor type and the motor control system are introduced at beginning. After that, the simulation of motor control system is shown and then two design strategies are presented. Considering all the requirements, the proper IC controller is adopted.

In chapter 3, the details of hardware implementation are presented. To meet the requirements of attitude control system, current measurement and speed control loop are designed, and then the main part of motor control circuit is presented as well. During the test, we verified that the current through the output is quite large, and can cause EMI towards hall sensor signals. Due to these interference, when reference voltage is set at 0V, the motor still kept running. Problem was solved by changing the PCB layout. The hall sensor signals crossed with output signals instead of running parallel. Tacho output, RC pulse and speed control loop, all these three test results only have 3% errors.

In chapter 4, software design is described. Due to the requirements of attitude control system, some concepts of micro controller are introduced and the details of the function are also implemented. During the test, all the functions were correctly carried out and the errors are kept around 5% which are acceptable for Aramis project.

For the future work, since the software design can be improved. The current situation is mainly focused on the realization of specific functions. These functions must be integrated with the whole system. To achieve this goal, the attitude control software must use the same approach as the other parts of Aramis, that is UML-based description.

References

[1] D. Del Corso, C.Passerone, L. M.Reyneri, C.Sansoè, M.Borri, S.Speretta, M.Tranchero , "Architecture of a small low-cost satellite," in Proceedings of the 10th Euromicro Conference on Digital System Design, Lubeck, Germany, August 2007,pp.428-431.

[2]Stefano Speretta, Leonardo M. Reyneri, C.Sansoè, Maurizio Tranchero, Claudio Passerone, Dante Del Corso, “ MODULAR ARCHITECTURE FOR SATELLITES ” in Proceedings of the 58th international Astronautical Congress, Hyderabad, India, September 2007. IAC-07-B4.7.09

[3] Texas Instruments Corp. Online:<http:// www.ti.com />

[4] Chee W.Lu.Torque Controller for Brushless DC motor. IEEE Transaction on Industrial Electronics. 1999,46(2): 471-473.

[5] Neyram Hemati, MingC.Leu. A complete Model Characterization of Brushless dc motors. IEEE Transaction on Industrial Electronics. 1992,28(1): 172-180.

[6] Jong Sun Ko. A Robust Digital Position Control of Brushless DC Motor with Dead Beat Load Torque Observer. IEEE Transaction on Industrial Electronics. 1993,40: 512-520.

[7] L6235 DMOS Driver For Three-Phase Brushless Dc Motor
http://www.st.com/stonline/products/literature/ds/7618/.pdf

[8] LM117/LM317A/LM317 3-Terminal Adjustable Regulator
http://www.st.com/stonline/books/pdf/docs/2154.pdf

[9] INA138 High-Side Measurement Current Shunt Monitor
http://focus.ti.com/lit/ds/symlink/ina138.pdf

[10] MSP430F438/439 Mixed Signal Microcontroller
http://www.ti.com/litv/pdf/slas713

[11] IAR Embedded Workbench Version 3+ for MSP430 User's Guide (Rev. S)
http://www.ti.com/litv/pdf/slau138s

[12] MSP430x1xx Family User's Guide (Rev. F)
http://www.ti.com/litv/pdf/slau049f

[13] PiCPoT. Online:< http://www.delen.polito.it/en/content/view/full/353/>

[14] Chandler, G.D.; McClure, D.T.; Hishmeh, S.F.; Lumpp, J.E.; Carter, J.B.; Malphrus , B.K.; Erb, D.M.; Hutchison, W.C.; Strickler, G.R.; Cutler, J.W.; Twiggs, R.J, "Development of an Off-the-Shelf Bus for Small Satellites," Aerospace Conference, 2007 IEE E 3-10 March 2007 Page(s):1 – 16

[15] visual paradigm. Corp. Online:< http://www.visual-paradigm.com/>

Appendix A　Code

Motor timer:

```
#include   <msp430xG43x.h>
static int num;   // number for count the seconds in Timer mode , count the turns in TACHO
mode
static int setTime=0;       // setTime , unit is in second
static int setTurns=0;      // setTurns
void motorTimer();
void TACHO();
static int flag;
static int sel;
static int flag_g;
static int flag_l;
void main(void)
{
   WDTCTL = WDTPW + WDTHOLD;
   //motorTimer(10);      //for example, we let motor running 10 seconds.
   TACHO(100);      // set turns for motor running, the problem is TACHO output from L6235
                    // maybe is too large for MSP430, and the tpulse maybe is too short.
                    // we have to check it later.
 }
void motorTimer(int time)
{
   if (flag==1)
      return;
   flag=1;
   sel=0;                // 0 means choose motorTimer, in order to let Timer_A know
   setTime=time*2;
   num=0;
   WDTCTL = WDTPW + WDTHOLD;                        // Stop WDT
   P5DIR |= 0x02;                               // we assume p5.1 is the enable output for
motor driver
   CCTL0 = CCIE;                                // CCR0 interrupt enabled
   CCR0 = 16384;                                // 16384 is equal to 1 sec. timer
   TACTL = TASSEL_1 + MC_1;                     // SMCLK, up mode
  _BIS_SR(LPM0_bits + GIE);                 // Enter LPM0 w/ interrupt

}
```

Motor Counter:

```
#include <msp430xG43x.h>
static int num; // number for count the seconds in Timer mode , count the turns in TACHO
mode
static int setTime=0; // setTime , unit is in second
static int setTurns=0; // setTurns
void motorTimer();
void TACHO();
static int flag;
static int sel;

static int flag_g;
static int flag_l;
void main(void)
{
   WDTCTL = WDTPW + WDTHOLD;
   //motorTimer(10);          //for example, we let motor running 10 seconds.
   TACHO(100);   // set turns for motor running, the problem is TACHO output from L6235
                 // maybe is too large for MSP430, and the tpulse maybe is too short.
                   // we have to check it later.
  }

void TACHO(int setturns)
{
//   if (flag==1)
//       return;
//   flag=1;


WDTCTL = WDTPW + WDTHOLD;
unsigned int tmp,turns,flag;
turns=0;
flag=0;
tmp=0;
P5DIR |= BIT1;
P5SEL = 0x00;
P4SEL = 0x00;
P4DIR&=~BIT1; //P1.0 is used as Input Pin

while(1)
{
if ( (P4IN & BIT1) == 0x02)
{
```

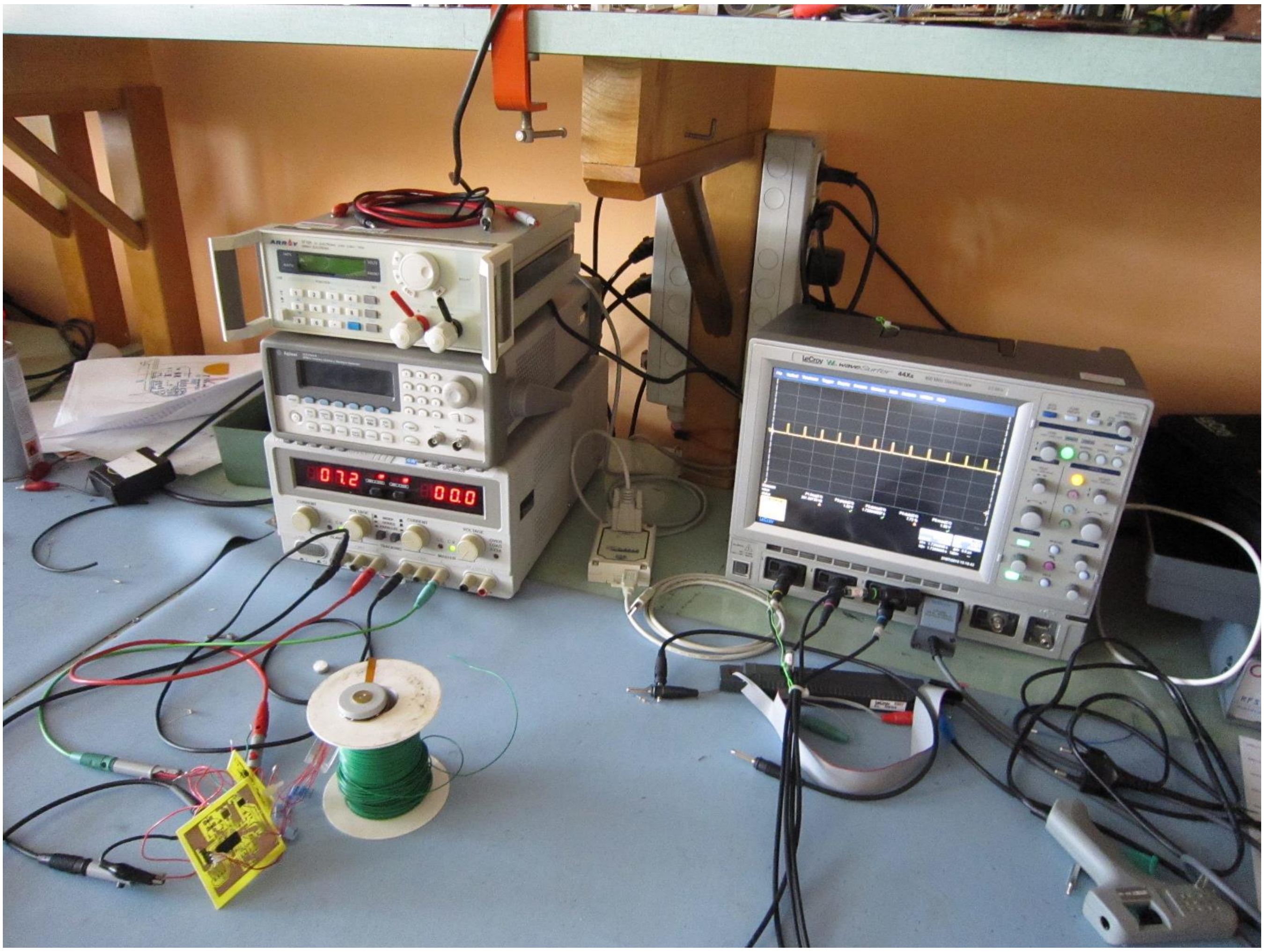
```
P5OUT= P4IN ; // P5.0 High
flag=1;
if (tmp!=flag)
{turns++;}
tmp=flag;

}
else
{
P5OUT= P4IN ; // P5.0 High
flag=0;
if (tmp!=flag)
{turns++;}
tmp=flag;
}

if (setturns*2<turns)
        {
             P5OUT=0x00;
             return;
        }
    }
}

// Timer A0 interrupt service routine
#pragma vector=TIMERA0_VECTOR
__interrupt void Timer_A (void)
{
  if (sel==0)
    {
  P5OUT = 0x02;                    // Toggle P5.1 using exclusive-OR, it is easy for test.
                                   //CCR0 +=16400 ;
num++;
if (num>setTime)    // check if we achieve the Timer goal
{
 P5OUT=0x0;       // set enable 0
 CCTL0 ^= CCIE;      // disable the interrupt
 flag=0;
}
    }
}
```

Current Measure :

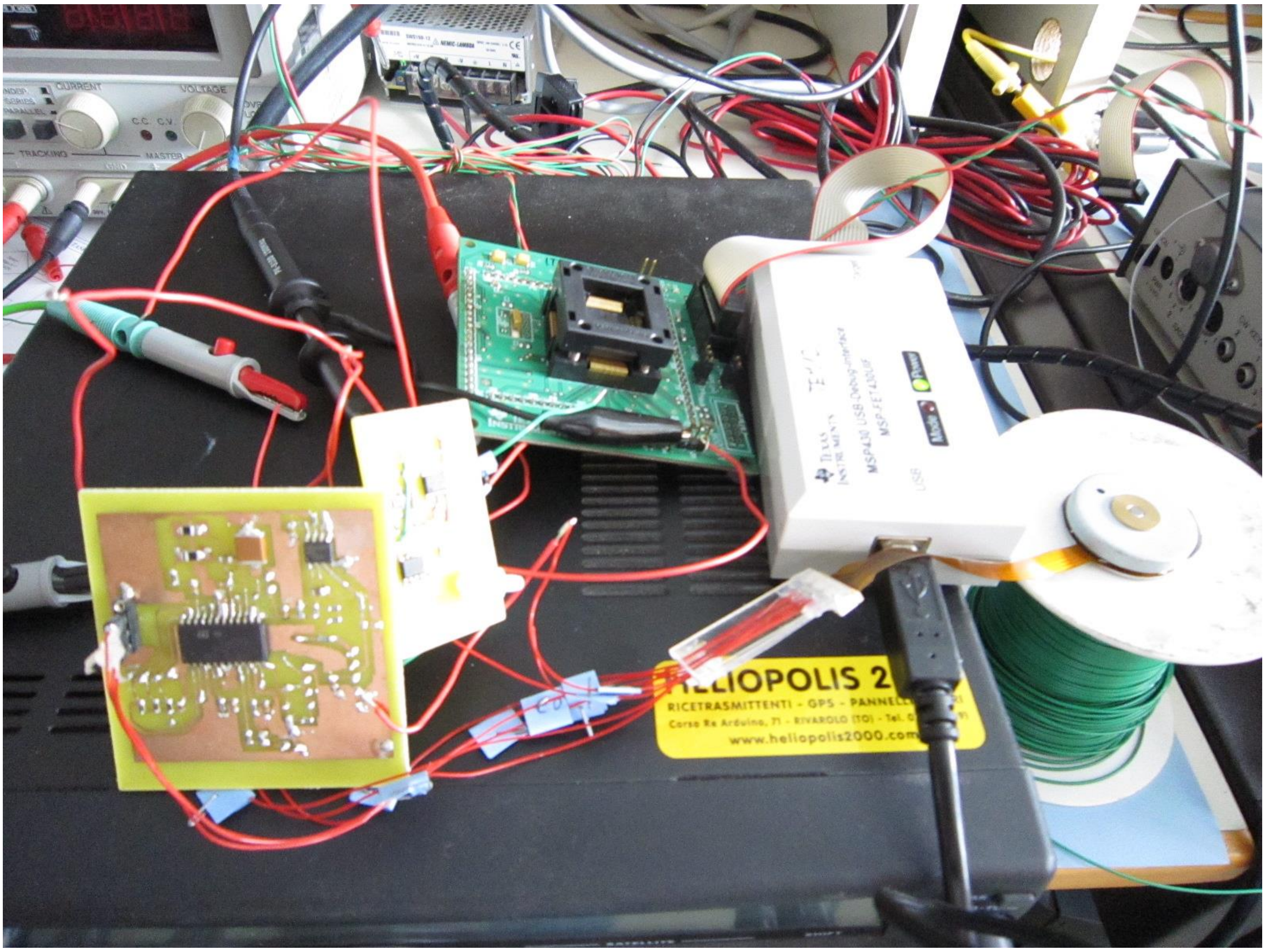
```
#include    <msp430xG43x.h>

void main(void)
{
   WDTCTL = WDTPW + WDTHOLD;                         // Stop WDT
   int VS=0;
   ADC12CTL0 = SHT0_2 + ADC12ON;                     // Set sampling time, turn on ADC12
   ADC12CTL1 = SHP;                                  // Use sampling timer
   ADC12IE = 0x01;                                   // Enable interrupt
   ADC12CTL0 |= ENC;                                 // Conversion enabled
   P6SEL |= 0x01;                                    // P6.0 ADC option select
   P5DIR |= 0x02;                                    // P5.1 output

   for (;;)
   {
      ADC12CTL0 |= ADC12SC;                          // Sampling open
      _BIS_SR(CPUOFF + GIE);                         // LPM0, ADC12_ISR will force exit
   }
}

// ADC12 interrupt service routine
#pragma vector=ADC12_VECTOR
__interrupt void ADC12_ISR (void)
{
      VS=ADC12MEM0;
      _BIC_SR_IRQ(CPUOFF);                           // Clear CPUOFF bit from 0(SR)
}
```

Appendix B Test Environment

Hardware Test:



System Test:


CURRENT
VOLTAGE
TRACKING
MASTER
MSP430 USB-Debug-Interface
MSP-FET430UIF
Power
Mode
USB
HELIOPOLIS 2
www.heliopolis2000.com

PCB Top View: