AN2267
Application note

Implementation of current regulator for BLDC motor control with ST7FMC

Introduction

A conventional method of controlling BLDC motors is to implement an inner current loop for torque / current control. Reference to this inner loop is provided either by an outer speed loop or by some other means based on application requirement. The linearity of inner current / torque loop is greatly affected by the faithfulness of current feedback. In the first section, an outline to various approaches for obtaining current feedback is presented and analyzed with the limitations of each. In the subsequent sections, a presentation is given of a simple, linear and cost effective approach of implementing the inner current loop by sampling the DC link current at the mid-point of PWM “on time” with ST7FMC. Experimental results are also discussed.

An accompanying software file is available with this application note and can be downloaded from www.st.com/mcu

Contents

1 Outline to various approaches

A BLDC motor driven in a conventional 6-step method greatly resembles a brushed DC motor. Hence, one may choose to regulate the average DC link current. But this actually results in constant power operation for the motor because at constant DC link voltage, if the average link current is regulated at a certain value, it effectively regulates the power at that point for any variation in motor load, and the average load current / motor torque varies inversely with speed depending on the load. Any effort to compensate the average DC link current data with the duty cycle to obtain average phase current will be impaired by a filter time constant, rendering this option ineffective.

Since the DC link current does not reveal winding currents during PWM “off time”, one may choose to monitor all 3 winding currents and build a regulator. But this requires two current sensors to monitor any two phase currents, while the third phase current can be reconstructed from these two. However, the cost of these sensors makes this option expensive.

A third option would then be to regulate the peak current per PWM period. Though it is inexpensive and easy to implement, it is not exactly linear. During PWM on time, at lower duty cycles, when both speed and BEMF are small, the phase current rises much faster than at higher duty cycles when the speed and BEMF are large. The same peak currents per PWM period represent different average currents at different duty cycles. An intuitive geometric approach will reveal this as shown in *Figure 1*. A typical variation in average current vs duty cycle at a given peak current reference is shown in *Figure 2*.

Figure 1. Peak current regulation at different duty cycles with BEMF load

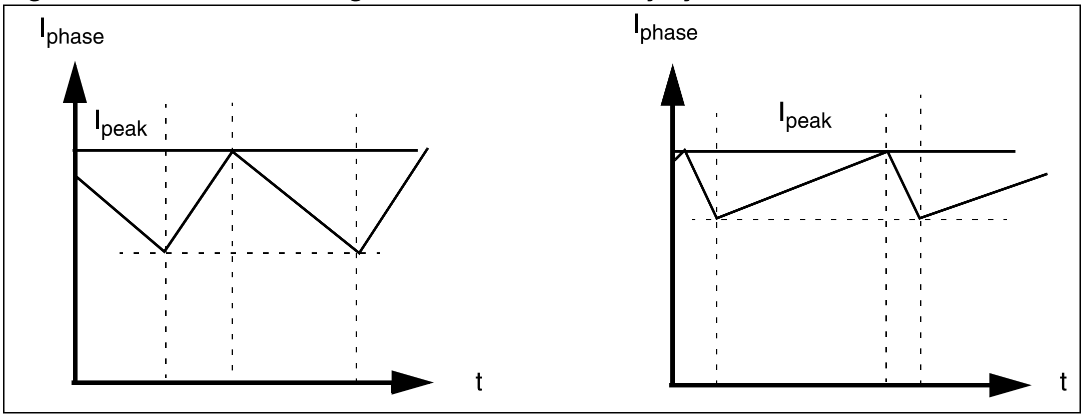
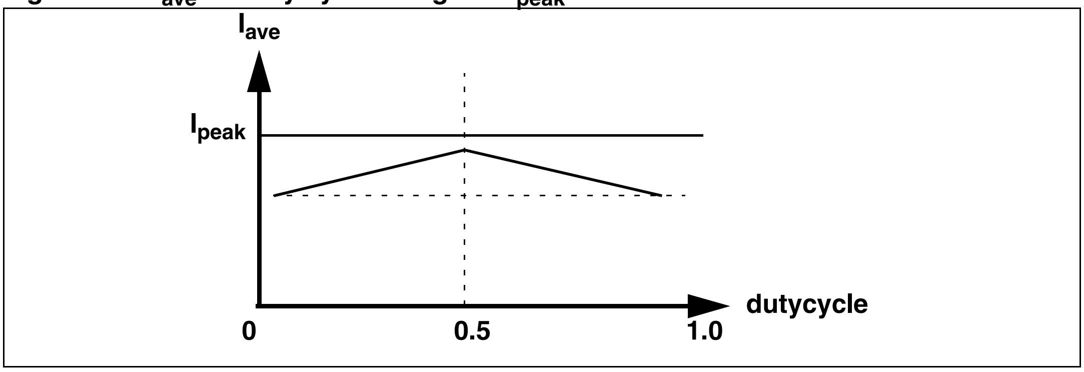


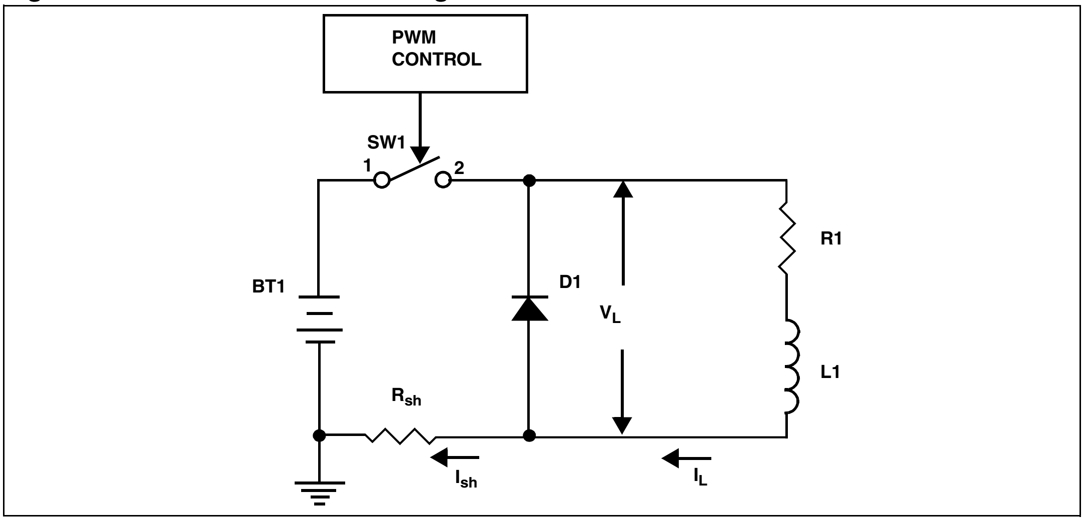
Figure 2. I_{ave} vs duty cycle at a given I_{peak}



2 Obtaining the average current

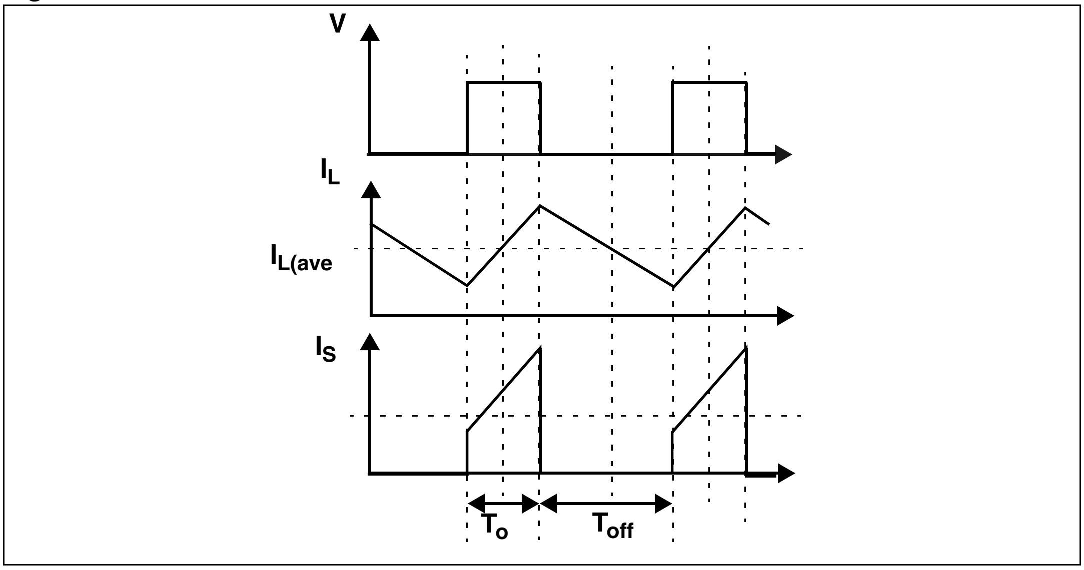
For linear torque control, it is important that we sample the average phase current as feedback to the current regulator. It is best to get this information from the DC link current using only a shunt resistor because of its low cost and simplicity. However, the DC link current is not continuous and is present only during PWM on time. As a simple model for current control, assume a simple buck converter feeding an RL load as shown in *Figure 3*.

Figure 3. Buck converter feeding an RL load



The switching frequency, PWM on time and load inductance are such that the load current is continuous. *Figure 4* shows the load voltage, load current and DC link current waveforms. A close look at the load current waveform reveals that its average value is equal to its instantaneous value during the middle of PWM on time or off time. Since the load current flows through the DC link during PWM on time, sampling the DC link current during the middle of PWM on time gives the average load current.

Figure 4. Buck Converter - Waveforms

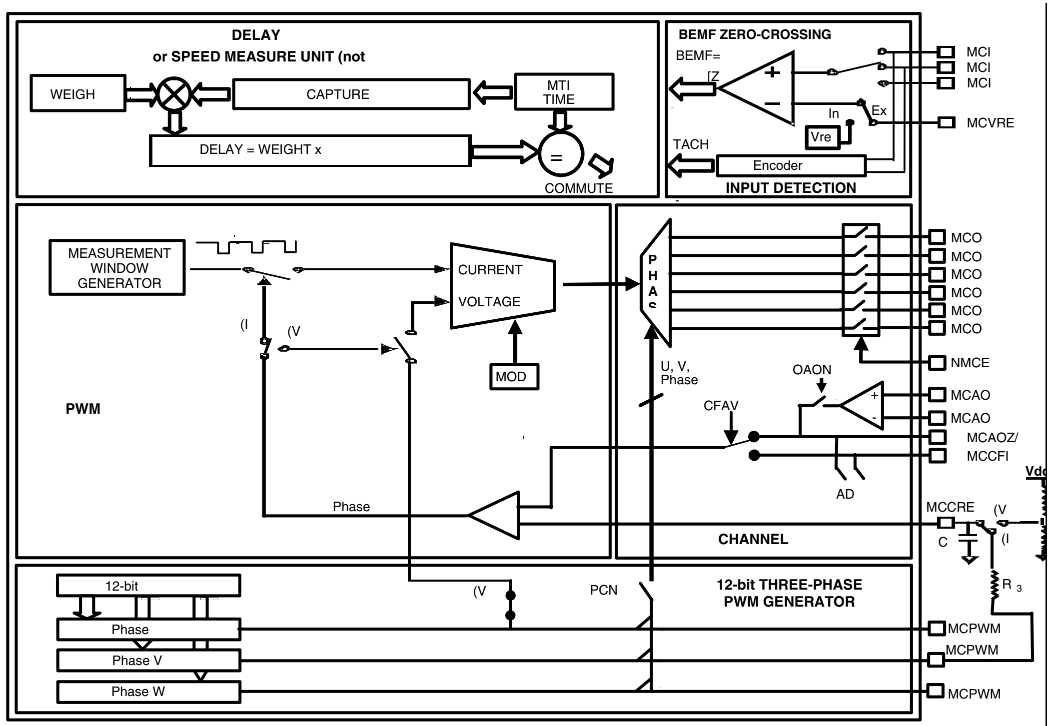


3 BLDC motor control using ST7FMC

The main feature of ST7FMC is its powerful motor control macro cell, capable of generating control signals to drive a sensorless or sensored 3 phase BLDC or AC motor. STMicroelectronics application notes AN1946 [1] and AN2030 [2] explain, in detail, the procedure to control a 3 phase BLDC motor using ST7FMC.

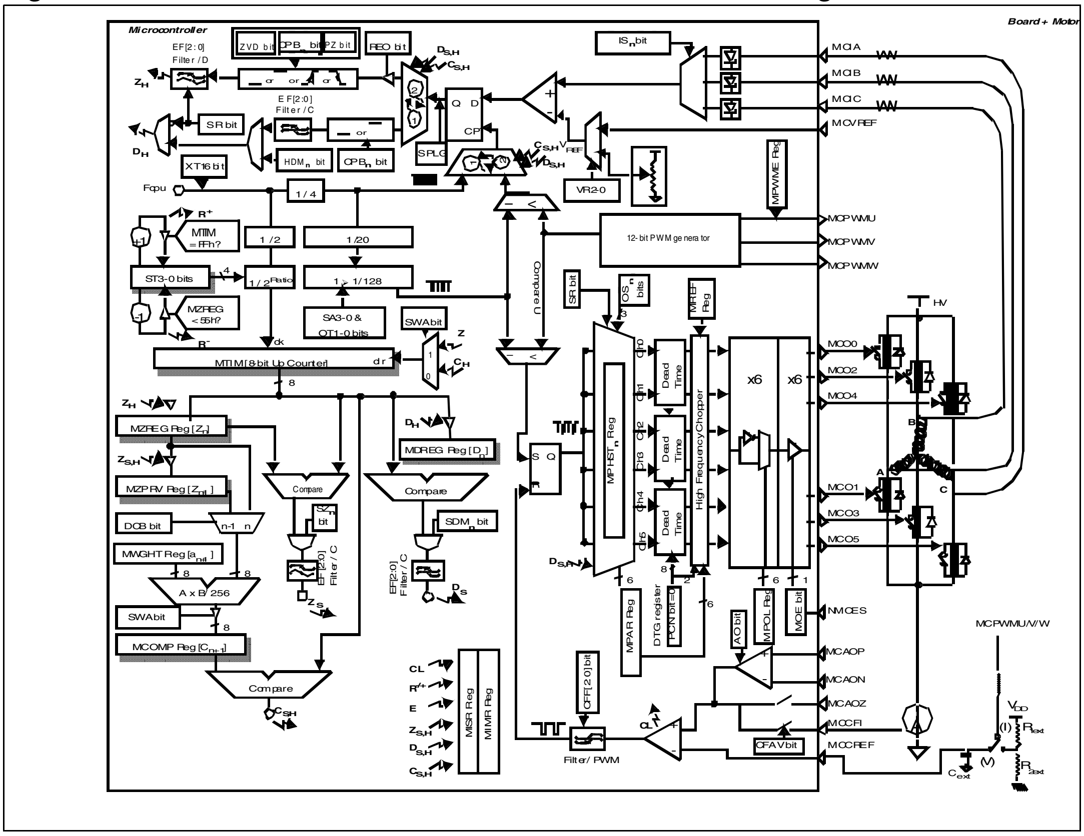
Figure 5 shows the simplified block diagram of the hardware motor control macro cell. The macrocell has multiple timers performing various functions in parallel to generate control pulses for the motor. An auto scalable 8-bit timer (MTIM) monitors the time difference between successive phase back EMF zero crossings (Z events) of the motor. When a Z event occurs, the timer value is captured into MZREG and the timer restarts counting from zero, and, the previous content of MZREG is transferred to MZPRV. This timer is a part of what is called DELAY MANAGER that, based on this time difference and a delay coefficient (MWGHT), identifies the timing for next phase commutation instant (C events). All in parallel, a 12-bit free running counter generates the PWM carrier for inverter switching.

Figure 5. Simplified block diagram of Motor control Macro cell for BLDC motors



[Z] : Back EMF Zero-crossing
Z_n : Time elapsed between two consecutive Z
[C] : Commutation
C_n : Time delayed after Z event to generate C
(I): Current
(V): Voltage

Figure 6. Motor Control Macro cell - BLDC motor control configuration



A PWM output is generated as a result of comparison between this carrier and a compare register (MCPUH:MCPUL) that carries pulse width (duty cycle) information. This PWM signal is directed to one of the six inverter switches by a CHANNEL MANAGER that acts as a traffic diverter on the PWM output. The channel manager also selects a complementary switch, as programmed by the user, which together with the switch receiving PWM will force current into the motor windings. Based on the motor terminal voltages or Hall sensor outputs, an analog block identifies the motor phase BEMF Z events and captures the contents of MTIM timer into MZREG and the previous value of MZREG into MZPRV and this cycle repeats all over again.

4 Implementation using ST7FMC

A typical schematic block diagram of ST7FMC based sensorless control of BLDC motor [2] is shown in *Figure 7*. Refer to *Appendix C on page 16* for a complete schematic of the experimental hardware. This schematic resembles the motor control starter kit schematic from Softec Microsystems, with matching I/O assignments wherever possible.

Figure 7. Schematic block diagram of ST7FMC based sensorless control of BLDC motor

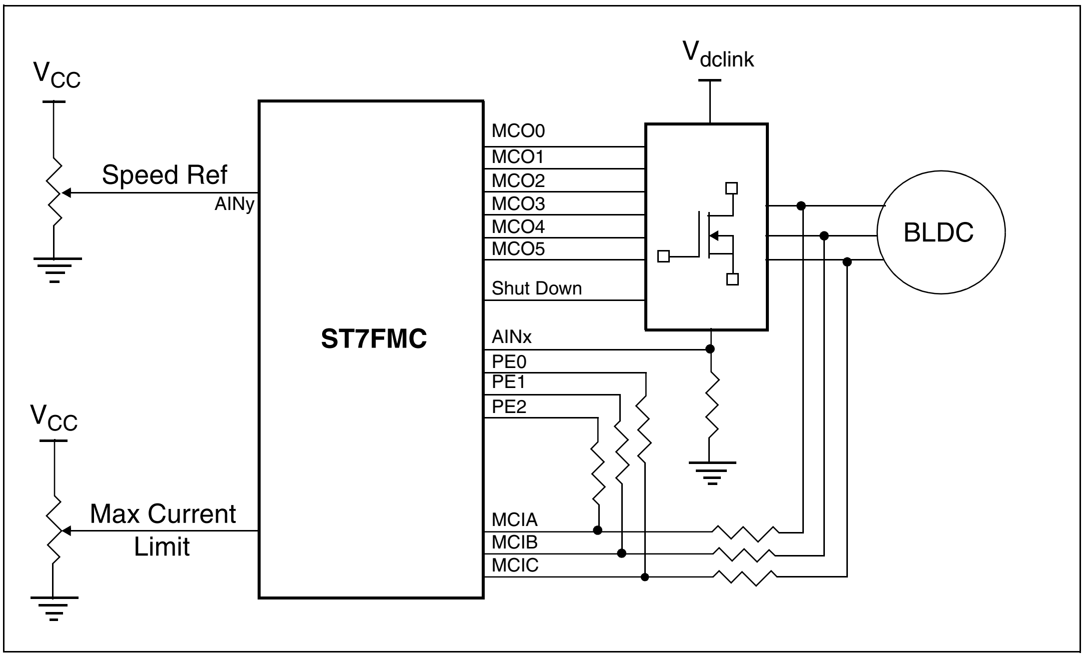
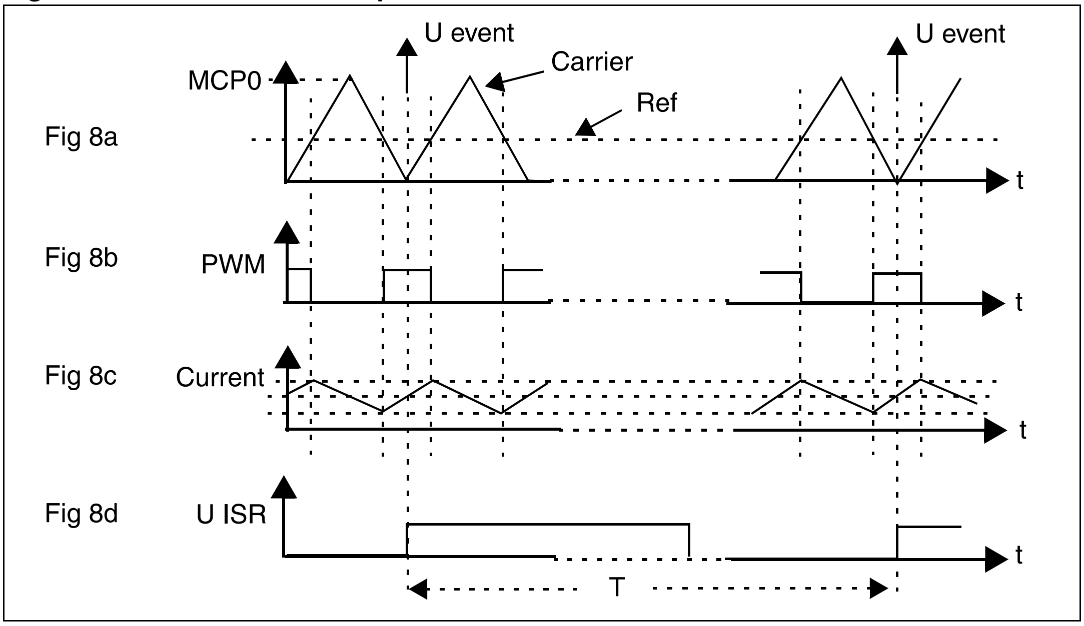


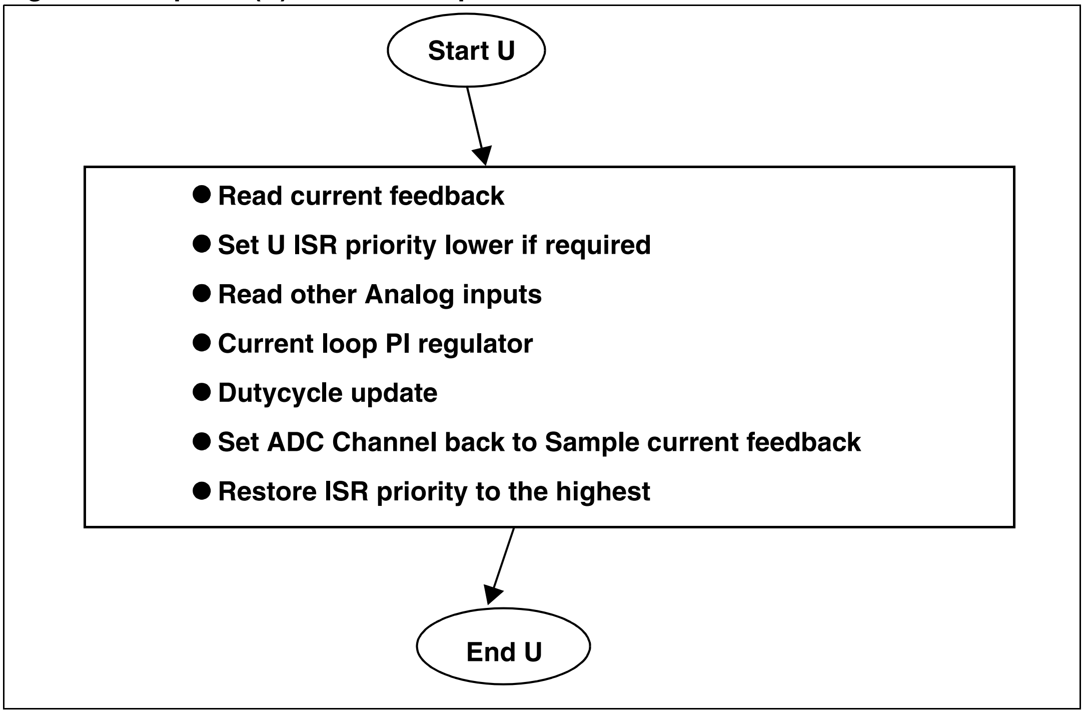
Figure 8a shows the PWM carrier configured in center aligned mode, where the counter counts up to a maximum value (as defined by MCP0) and starts counting down to zero and repeats this cycle again. (See *Appendix A* for information on setting the PWM frequency). The PWM generator is set to generate a duty cycle update interrupt (U event) upon completion of every N carrier cycles as specified by MREP register. (See *Appendix A* for information on setting the periodicity of this interrupt). The timing of the U event or interrupt is positioned as shown in *Figure 8a*. The carrier is compared with MCPU and PWM pulses are generated as shown in *Figure 8b*. Due to the application of PWM voltage on motor windings, a current flows in its windings as shown in *Figure 8c*.

Figure 8. PWM on time midpoint identification and control



From *Figure 8a* and *Figure 8b*, it is clear that the U event takes place at the center of PWM on time. Based on the previous discussions, this is the right instant to read the instantaneous DC link current in order to get the average phase current value. Hence the interrupt associated with U event should be set to the highest priority and the very first instruction in this Interrupt Service Routine (ISR) should read the DC link current value. In any case, there is an interrupt latency time of approximately 3-4µs, which is also the typical conversion time of on-chip Analog to Digital Converter (ADC). If the current feedback analog input channel was previously selected and set for sampling continuously, then, when the first instruction in U event interrupt subroutine reads the ADC data register, it will aptly hold the DC link current value fairly close to that during the middle of PWM on time.

Figure 9. Update (U) event interrupt subroutine flow chart



The flowchart in *Figure 9* shows the actions within the U event interrupt service routine. To coordinate the reading of any other analog inputs to the ADC, it is recommended that they are all read within this U event subroutine after the DC link current read. However, before returning from the interrupt, it is important to restore the ADC to sample the DC link current channel again so that on re-entry in the next U event, the DC link current value can be read from ADC right away. If required, interrupt priority of this routine can be lowered after reading the current value upon entry, but should be restored to the highest value before returning for obvious reasons.

Refer to the accompanying file for a complete listing of the code and experimental workspace.

5 Results

Experimental implementation of this scheme yielded satisfactory results. A closed loop regulator for BLDC motor control with inner current and outer speed loops as shown in *Figure 10* was implemented. Current loop sampling time of 500µs and speed loop sampling time of 2ms was chosen. The amount of computing time required within a 2ms time window to execute through a full cycle of control loop and all motor control ISRs at an electrical frequency of 200Hz is less than 1ms. The important waveforms obtained are shown in figures *11* and *12*. *Figure 11* shows the convergence of reference and actual phase current values at the instant of occurrence of U event which is the feedback sampling instant. Notice that the U event occurs during the middle of PWM ON time. *Figure 12* shows the tight control of motor average phase current for a given current reference.

Figure 10. Closed loop current and speed control - block diagram

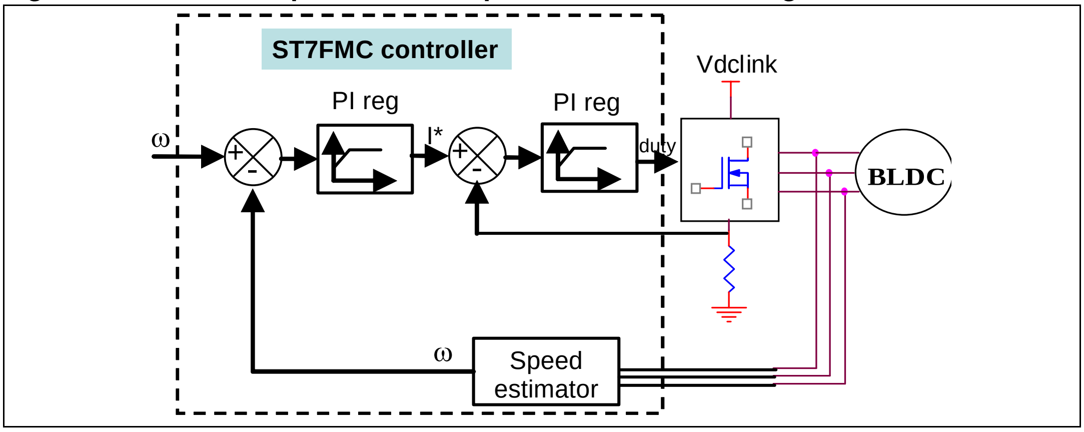


Figure 11. DC link current sampling at U event and closed loop convergence

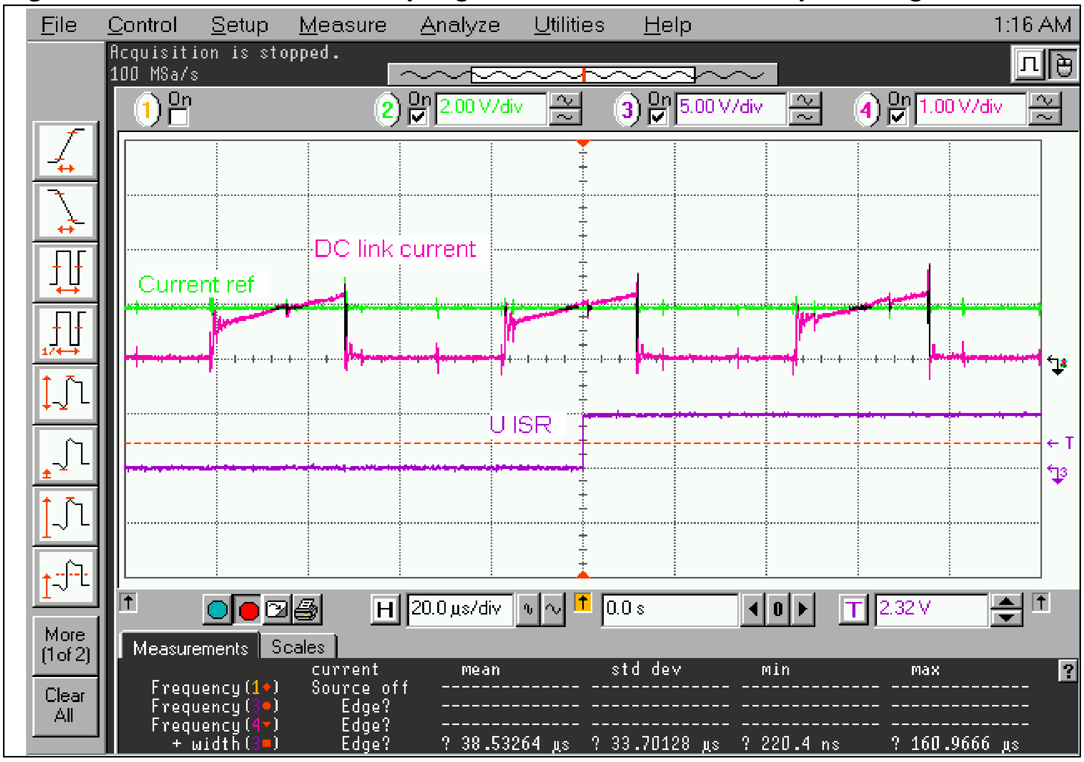
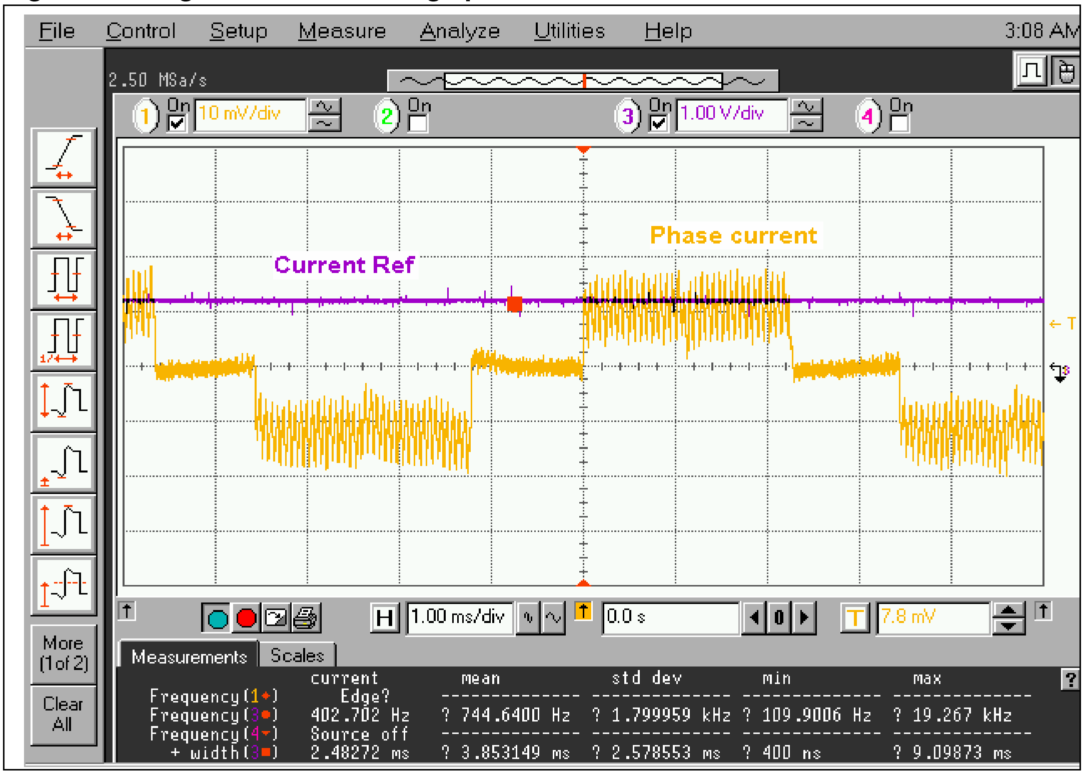


Figure 12. Tight control of average phase current vs reference



6 Conclusion

The experiments performed based on the described method gave fairly linear current control. One limitation of this sampling method is when the motor current becomes discontinuous, in which case the actual average current is less than the instantaneous value at the mid point of PWM on time, and correcting this error is quite cumbersome.

7 References

[1]. STMicroelectronics AN1946 - Sensorless BLDC motor control and BEMF sampling methods with ST7MC

[2]. STMicroelectronics AN2030 - Back EMF detection during PWM on time by ST7MC

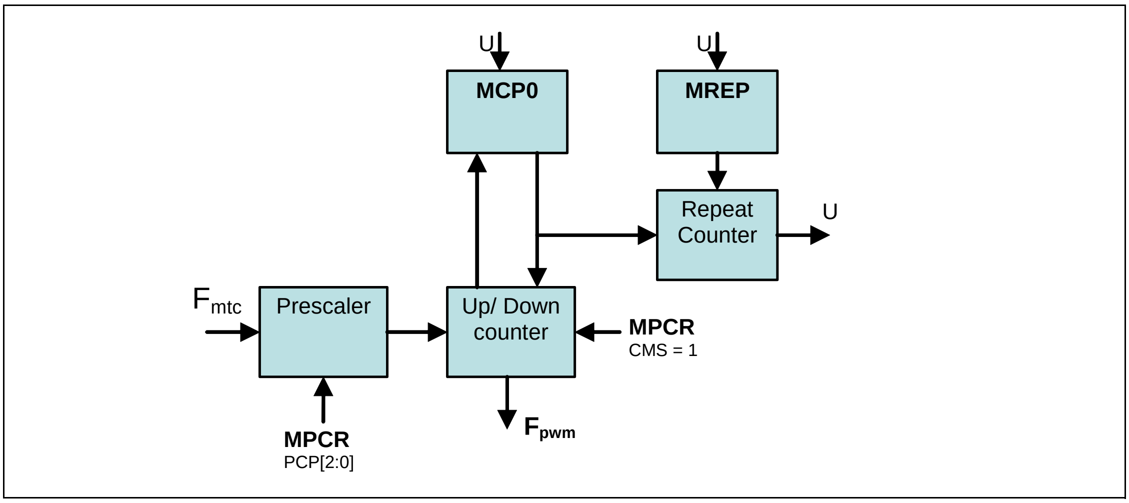
Appendix A Sampling inner current loop procedure

Procedure to set carrier frequency (Fpwm) and periodicity of U event (TU) for sampling inner current loop:

Chosen F_{pwm} = 16KHz

where,

$F_{pwm} = F_{mtc} / (Prescaler . 2 . MCP0)$



Given F_{mtc} = 16MHz,

and choosing Prescaler = 1,

then,

MCP0 = 500

Choosing TU = 500µS

where,

TU = Tpwm . (MREP + 1) / 2

Substituting for TU and T_{pwm},

MREP = 15

Appendix B Event U interrupt service routine

```
/***************************************************
          Motor control - Event U interrupt service routine
**************************************************/
@interrupt @nosvf  void mtcU_CL_SO_ISR(void)
{
    if (bitTest_TRUE(MISR, PUI) )     // check for U event presence
    {
       /* ===   Current loop PI Controller begins here   === */
       currentFb = (ADCDRMSB << 2) + ADCDRLSB;  // get new value of currentFb
       piconCur();                                   // call current loop PI regulator
to get new dutycycle
       MCPUL = PIconCur.byte.b2;    // update MCPUH :MCPUL with new dutycycle
       MCPUH = PIconCur.byte.b3;
       /* ===   Current Loop PI controller ends here   === */

       // Read potentiometer to get latest speed reference
       getADC_10bit (speedRef , SPEED_REF_CHNL);
       if (speedRef > SPEED_REF_MAX)
           speedRef = SPEED_REF_MAX;

       // Current Feedback measurement setup for next cycle
       ADCCSR = ADON + CURRENT_FDBK_CHNL;
       ADCDRMSB;  // to clear EOC of prev conv
       MISR = 0xff - PUI;            //reset IT flag
    }
    return;
}
```

Appendix C ST7MC 3-phase motor control schematics

Figure 13. Schematic 1 of 3

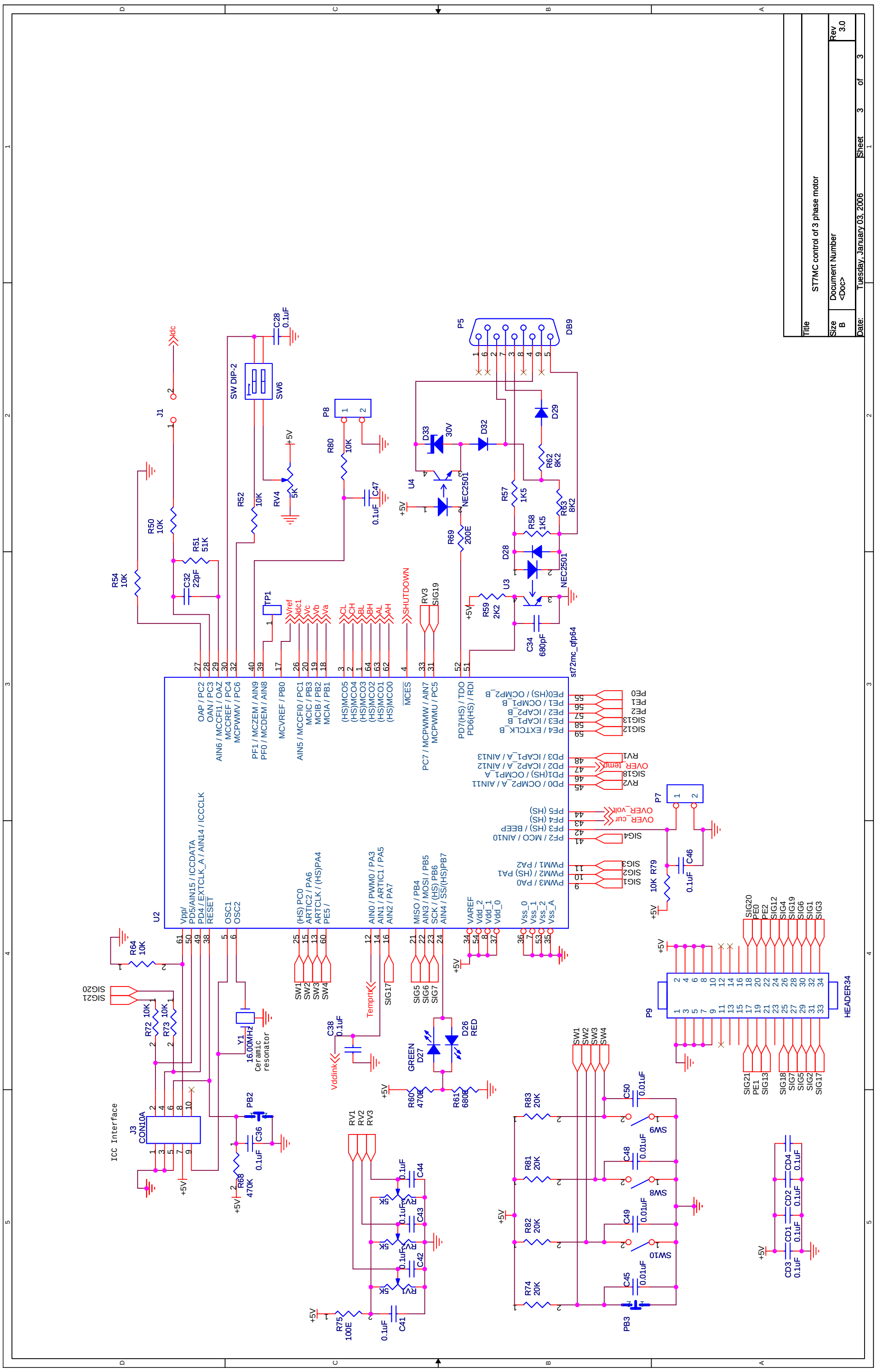


Figure 14. Schematic 2 of 3

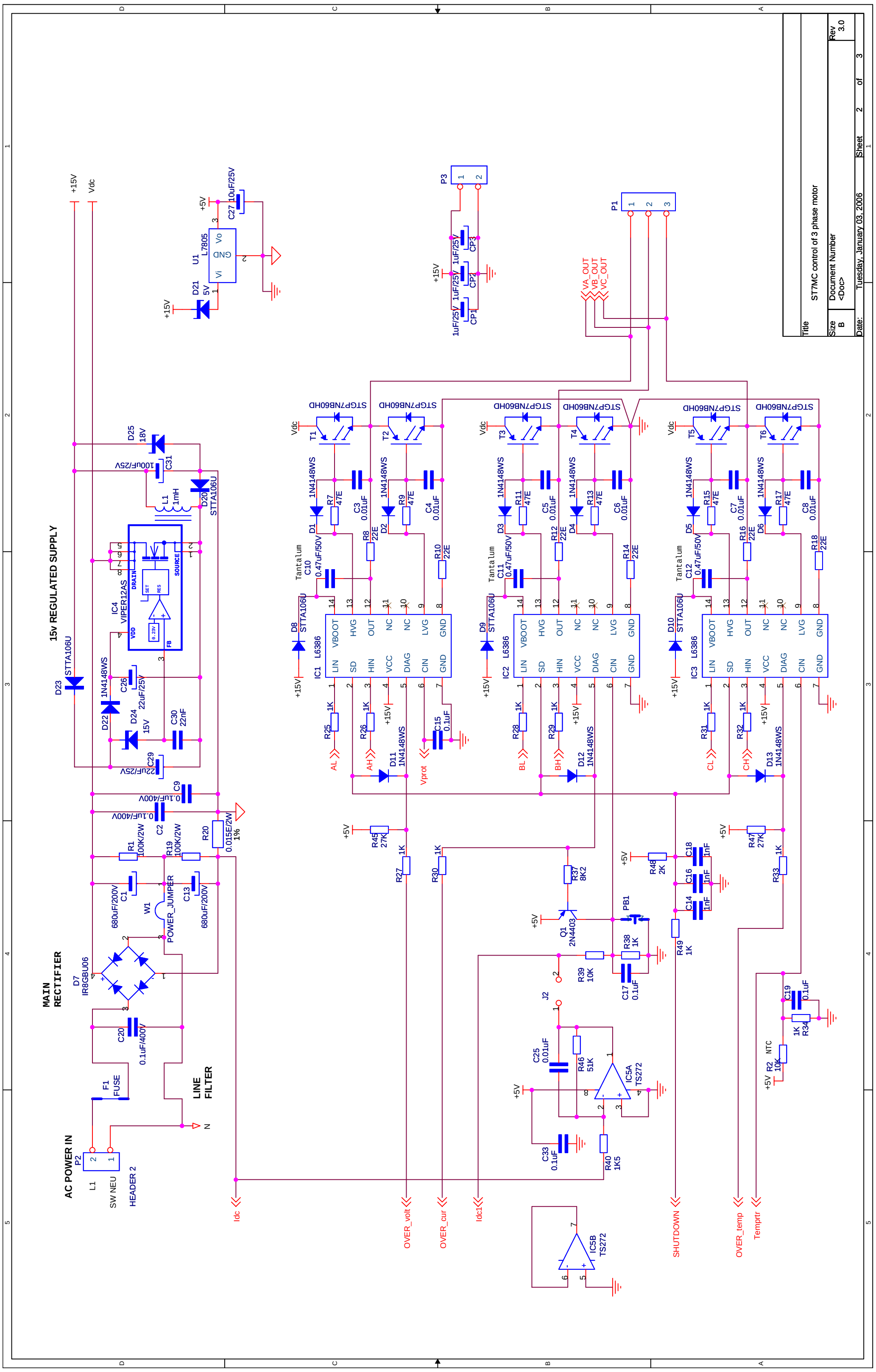
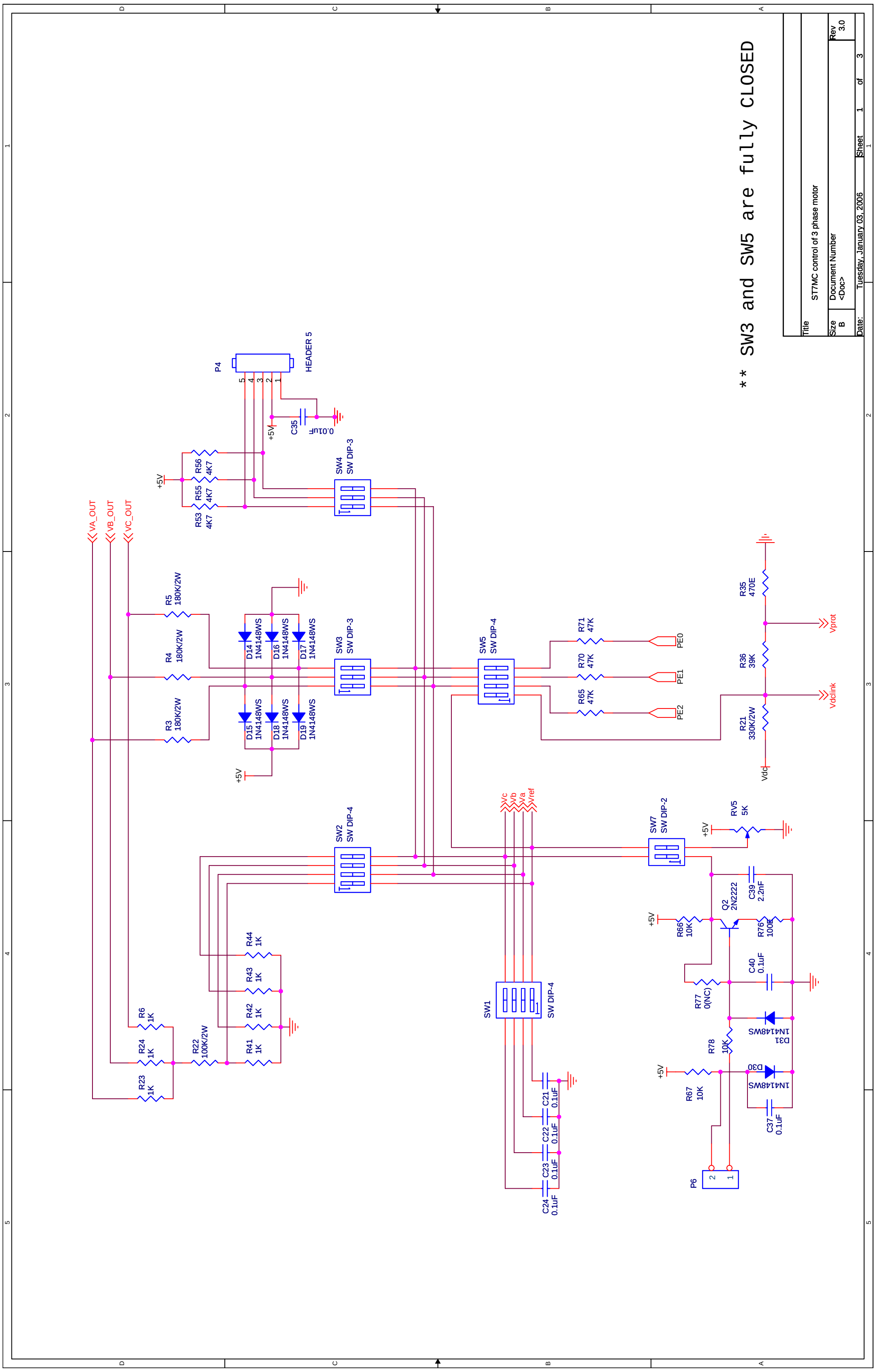


Figure 15. Schematic 3 of 3



STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com