Software

To build a high speed accurate motor controller we were limited to the ARM based AT91SAM3X8E microcontroller because of the speed high speed quadrature output on the back of the motors.

The AT91SAM3X8E ARM microcontroller is fitted to the Arduino DUE development PCB’s, this gave us something off the shelf to build a prototype with.

As we going to use the Arduino DUE in the prototype the software was written in the Arduino IDE.

The Arduino IDE has lots of great library files, but Arduino DUE microcontroller is the development PCB in the Arduino range with quadrature input so no library files were available. After some research we found some information on enabling the quadrature input and which memory locations to read the current value from.

The first program was written to just test the quadrature output of the motor with an Arduino DUE. The motor had to be turned by hand and 1 revolution, no matter how fast or slow gave 4000 pulses.

For motor testing we were able to power a spare 75 volt motor from a 20 volt power supply and control the motor using an H Bridge designed for low power low voltage motors. The H Bridge was controlled using a direction signal and PWM to control the speed. We were able to exercise the motor using feedback from the quadrature input, making the motor do one revolution forward then one revolution backwards.

As there was a fair bit of over-shoot, the software was modified to ramp down the PWM as it got closer to the set-point with very little over shoot and very good accuracy.

The USB serial port on the Arduino DUE development PCB was used for programming and debugging, a second serial port was enabled and a serial interface connected so that set-points could be sent to the Arduino DUE.

During testing it was discovered that there was very little torque from the motor when it was close to the set-point, a decision was made that we would need to use PID control to get more torque from the motor when it was close to the set-point.

Arduino provide a PID library for the Arduino development PCB’s, this took a lot of development to get working correctly, the main problem with the motor was not having a load, so it would spin up very fast and oscillate wildly.

We were able to demonstrate the speed, torque and accuracy of the motor controller on the work bench.