Mini-Mapper motor prototype board testing

Ian Ross

December 6, 2020

This document reports testing of the motor prototype board:

1. Basic tests of the electronics by driving the PWM inputs on the board using the Analog Discovery 2, and using the logic analyser and scope functions of the AD2 to monitor the motor encoder and torque measurement outputs.
2. Tests of the software running on the STM32F767ZI Nucleo-144 board, connected to the Analog Discovery 2 to monitor PWM outputs and provide synthetic inputs for the motor encoder and torque measurement code.
3. "Live" testing of the software, connecting the motor board to the Nucleo-144 board.

The software used for these tests is the `shell-motor` application, which provides a command line interface for basic motor operations. (Once this is all working, the emphasis will shift to using a Python test bench GUI on a PC to talk to driver code on the Nucleo board.)

1 Motor board initial testing

Setup

The motor board and AD2 are set up as follows for this series of tests:

- Connect 5V wall wart power supply to motor board.
- Connect 3.3V supply from AD2 to motor board.
- Configure two AD2 digital outputs for `MOTOR_IN1` and `MOTOR_IN2`.
- Configure AD2 digital input for `PULSE`.
- Configure AD2 analog input for `MOTOR_SENSE`.

The initial protocol for testing here was something like this:

1. Remove 5V power from motor board.

2. Disconnect `MOTOR_IN1` and `MOTOR_IN2` from motor board.
3. Configure AD2 digital outputs in “Patterns” panel and check.
4. Connect `MOTOR_IN1` and `MOTOR_IN2` to motor board.
5. Apply 5V power to motor board.
6. Run AD2 digital output waveform generator.
7. Record results: motor motion, `PULSE` and `MOTOR_SENSE` inputs.

The control states I tested with this setup were:

State	`MOTOR_IN1`	`MOTOR_IN2`
Motor halted	HIGH	HIGH
Forward 10%	HIGH	PWM: 10% LOW
Forward 50%	HIGH	PWM: 50% LOW
Forward 100%	HIGH	LOW
Backward 10%	PWM: 10% LOW	HIGH
Backward 50%	PWM: 50% LOW	HIGH
Backward 100%	LOW	HIGH

Results

- Quite shockingly, everything worked first time!
- After the initial setup, I didn’t bother too much with the protocol above, just switching the PWM duty cycle in the AD2 “Patterns” panel. (Which is a more realistic test than I was originally planning to do anyway.)
- The results were pretty much perfect, as far as I can tell. There are only two things that might need some attention.
- Pulse times from the motor encoder might need some smoothing, because the encoder disk isn’t quite straight.
- The signal from the motor coil current sense amplifier that I’m going to use for torque measurements is quite noisy, but that may just be due to seeing the PWM switching frequency coming through into the motor coil current.
- But: PWM speed control works, forwards and backwards; the pulse output from the motor encoder works; and torque measurement “works”, though I need to figure out exactly what I’m seeing there.

A better solution for the wobble in the motor encoder signal is probably to correct the mounting of the encoder disk to the motor shaft. I designed the hole in the encoder disk under-size to ensure a tight fit, but it’s *too* tight because the encoder disks are manufactured as PCBs and have two layers of copper with no relief around the mounting hole. That means that it's not possible to push the encoder disk onto the motor shaft without expanding the hole in the disk. I did that in a not very careful way for the disk I'm currently using, but I have two others, and could try to drill an accurate 2 mm hole in them to fit exactly the diameter of the motor shaft (and then glue them in place).

As for what's going on with the motor coil current signal, I need to investigate that in more detail, but there is at least something reasonable there, and the DC level of the signal does go up and down as the motor drive PWM duty cycle is set to higher and lower values.

2 Software testing

Setup

The Nucleo board and AD2 are set up as follows for this series of tests:

- Connect ground between Nucleo board and AD2.
- Connect `PE5` and `PE6` on Nucleo (`MOTOR_IN1` and `MOTOR_IN2` motor PWM outputs) to logic pins 0 and 1 (inputs) on the AD2.
- Connect `PB10` on Nucleo (`PULSE` motor encoder input) to logic pin 2 (output) on AD2.
- Connect `PA4` on Nucleo (`MOTOR_SENSE` analogue torque input) to AD2 analogue waveform output W1.

The idea here is for the software on the Nucleo board to drive the PWM motor driver signals and for these to be observed by the AD2's logic analyser. The motor encoder input to the Nucleo board will be driven by the AD2's pattern generator, and the analogue torque input to the Nucleo board will be driven by the AD2's signal generator.

The AD2 Waveforms software should be configured as follows:

- Logic analyser with inputs 1 and 2 for motor PWM signals from Nucleo board.
- Pattern generator to logic pin 3 for variable frequency square wave to drive motor encode input on Nucleo board.
- Signal generator to output W1 to drive motor torque measurement input on Nucleo board.

Test command script:

```
# Ensure initial state and command processing OK.

stop


# Test basic motor commanding.

forward 10
stop
reverse 10
stop


# Test torque measurement.

torque on

# Modify DC level of analogue output from AD2 signal generator to test
# different motor coil current levels.

set torque-interval 5000
set torque-interval 500
torque off


# Test motor speed encoder.

encoder on

# Modify frequency of square wave output from AD2 pattern generator to
# test different motor speed measurements.

set encoder-interval 5000
set encoder-interval 500
encoder off
```

Results

- `stop` gives: DIO0 low, DIO1 high. (WRONG: should be both high.)
- `forward 10` gives: DIO0 high, DIO1 high. (WRONG: should be both DIO0 high, DIO1 PWM.)
- `reverse 10` gives: DIO0 PWM, DIO1 high. (WRONG: should be both high.)

2.1 PWM shell regression testing

Register	Old: works	New: broken	Status
RCC			
AHB1ENR	0x30000a	0x30000f	OK
APB1ENR	0x10040400	0x10040402	OK
TIM10			
CR1	0x81	0x80	CEN not set!
SR	0x3	0x1	No interrupt flag
CNT	0x6c3	0x0	Not running!
CCR1	0x2a30	0x1518	OK