Technological Arts

APPLICATION NOTE AN101

Using the MicroCore-11 Motor Driver Module

INTRODUCTION

The MicroCore-11 Motor Driver Module is a simple driver board intended to drive two small DC motors or two phases of a small stepper motor under MCU control. It is based on the popular L293D dual H-bridge chip, handling up to 600 mA per driver when adequately heat-sinked. A higher-current version (1 Amp per driver) based on the SN754410NE is also offered. *Caution! Exceeding the ratings may destroy the chip.*

Port assignments are jumper-selectable on the board so that they may be readily changed to suit the the application. A second module can be used to double the number of channels (ie. drive four DC motors or two stepper motors), by jumpering wires to a different set of port pins (typically PORTD). A Quad Motor Driver Module is offered, which supports this configuration in one module.

Screw-terminal blocks are provided for motor connections, and LEDs indicate the direction of current through each motor coil. Digital control is achieved by means of two port lines for each channel. One controls direction of current through the coil, and the other switches the current on and off.

Four spare logic inverters are available for the user's application, simply by removing the jumpers tying unused inputs low. Prototyping area, in the form of plated-through holes on a 0.1" grid, is provided for the addition of circuitry (such as oscillators, interfaces for IR sensors, and position encoders) typically used in a small robot or motion-control system. The Quad version of this module has only one spare logic inverter, and no prototyping area. However, it includes a 1-Amp logic MOSFET for driving a relay, solenoid, or small DC motor (on/off control only-- not reverse).

DC MOTOR CONTROL

The simplest form of motor control is to select the desired current direction with the DIR control line and turn the motor current on with the EN line. To implement speed control, the ratio of ON to OFF time of the EN line can be controlled. This is achieved by controlling the EN line with a pulse-width modulated (PWM) signal. PWM signals are easily generated using the 68HC11's Output Compare function. Refer to the 68HC11 Reference Manual for details and example code. The minimum practical frequency of the PWM signal will depend on the physical characteristics of the motor being used.

STEPPER MOTOR CONTROL

A bipolar stepper motor can be controlled by attaching one phase to each of the two motor control channels. By alternately enabling the coils, and reversing current direction, the motor shaft is made to step through rotation. The maximum speed at which this step sequence may practically be repeated is governed by the physical characteristics of the motor being used.

POWER SUPPLY CONSIDERATIONS

The Motor Driver Module has an EXT PWR connector (J1) which provides the driver chip with the motor supply voltage. The voltage you use will depend on the voltage rating of the motors being driven. The voltage applied to this connector is also fed to a low-dropout regulator which supplies 5V to the logic circuits on the module. This same 5V can also supply the MicroCore-11 Module if jumper W1 is used. The advantage of this is that only one power connection to the system will be required (ie. do not connect external power to MicroCore-11 J1 connector).

Alternatively, power to the Motor Driver Module logic circuits can come from the 5V regulator that is on MicroCore-11. To do this, remove U1 (or L1) on the Motor Driver Module. Then power MicroCore-11 as usual and apply a separate Motor Voltage supply to the Motor Driver Module.

SOFTWARE

The program **mc11mdsm.asm** (look for it at http://support.technologicalarts.ca/docs/MicroCore-11/) is one example of how stepper motor control can be accomplished. Written in 68HC11 assembly language, it uses the Real-time Interrupt (RTI) feature of the MCU as a 4-millisecond timebase to create the step waveforms needed to control current through the two phases of a bipolar stepper motor. The program simply steps the motor through continuous rotation. A slower step speed can be achieved simply by increasing the step-timer variable (each increment corresponds to 4 ms per step). For a far slower step rate, the RTI can be set up to interrupt at a much longer interval. Refer to the 68HC11 Reference Manual for details.

To achieve a higher step rate, a timebase other than the RTI should be used Some possibilities are: the MCU hardware timer system (using an Output Compare interrupt, for example), or a software delay loop.

If a second motor driver module is used, jumpers W6 - W9 should be removed on the second module, and wires should be jumpered to the PORTD lines (PD2 - PD5). Making the software control the motor through PORTD lines requires initializing the PD2 - PD5 lines to be outputs (via the DDRD register), and changing the BSET and BCLR instructions to act on the appropriate PORTD bits instead of PORTA bits.

Check the support webpage mentioned above for additional software examples or applications information.

©2008 Technological Arts