HiTechnic First Motor Controller for LEGO NXT

1. Summary

This specification defines the motor controller for use with the LEGO NXT Mindstorms system. The motor controller provides two channels each of which can support an FTC motor and two shaft encoders. The motors are driven by PWM power while the encoders are continuously monitored. The firmware supports position, speed and power control modes.

Power is supplied from an external 9v – 15v battery via a screw terminal block.

2. Description

2.1. Function

The motor controller interfaces with the NXT using the 6 wire sensor interface. All of the 6 wires will be used.

The motor controller will communicate with the NXT using the LEGO Company's defined use of the I2C interface.

The motor controller will time-out if no I2C communication is detected for 2½ seconds. If a timeout occurs, both motor channels will be set to float mode.

The motor controller will shut down if a fault condition such as an output short circuit, output over-current or internal overheating is detected. It will automatically restart after a one minute cool down period. If a shut down occurs, both motor channels will be set to float mode.

The motor controller will accept 32 bit target values for the encoder counts for position setting mode.

The motor controller will return the readings for the encoder counts from both channels as 32 bit values.

The first motor controller in the daisy chain will use an I2C address of 02/03. Subsequent controllers will obtain addresses of 04/05, 06/07 and 08/09. Only four controllers may be daisy chained. Sensors and other I2C devices may not be connected to the controller daisy chain.

The motor controller will consume 5mA or less from the NXT "brick" 4.3v unregulated supply.

The motor controller will supply up to 4 amps on each motor channel when running from 9v – 15v.
The motor controller uses separate grounds for the NXT I2C interface and the battery supply to reduce ground loop problems. These grounds are tied together with (33 ohm) resistors in each module. Care should be taken when wiring the battery supply between modules to ensure that the terminals are tight and that an adequate gauge of wire is used for the likely motor current consumption. Inappropriate wiring will result in failure to operate correctly. Care has been taken to ensure that typical wiring errors should not

cause permanent damage. However, care should always be taken to avoid wiring errors. Do not use unfused battery packs.

3. Technical Description

3.1. Operation

The motor controller firmware will support the LEGO company’s sensor memory model.

Address	Type	Contents
00 – 07H	chars	Sensor version number
08 – 0FH	chars	Manufacturer
10 – 17H	chars	Sensor type
18 – 3DH	bytes	Not used
3E, 3FH	chars	Reserved
40H – 43H	s/long	Motor 1 target encoder value, high byte first
44H	byte	Motor 1 mode
45H	s/byte	Motor 1 power
46H	s/byte	Motor 2 power
47H	byte	Motor 2 mode
48 – 4BH	s/long	Motor 2 target encoder value, high byte first
4C – 4FH	s/long	Motor 1 current encoder value, high byte first
50 – 53H	s/long	Motor 2 current encoder value, high byte first
54, 55H	word	Battery voltage 54H high byte, 55H low byte
56H	S/byte	Motor 1 gear ratio
57H	byte	Motor 1 P coefficient*
58H	byte	Motor 1 I coefficient*
59H	byte	Motor 1 D coefficient*
5AH	s/byte	Motor 2 gear ratio
5BH	byte	Motor 2 P coefficient*
5CH	byte	Motor 2 I coefficient*
5DH	byte	Motor 2 D coefficient*

The *Sensor version number* field will report a revision number in the format “Vn.m” where *n* is the major version number and *m* is the revision level. Revision numbers will typically reflect the firmware level. The version number will be used to indicate the hardware level.

The *Manufacturer* field will contain “HiTechnc”.

The *Sensor type* field will contain “MotorCon”.

The *Motor 1/2 mode* fields select the operating mode for each channel.

The *Motor 1/2 power* fields set the operating power level for each channel. This value is a signed byte.

The *Motor 1/2 target encoder value* fields will accept a 32 bit target position setting for use in position setting mode. This value is a signed 32 bit value.

The *Motor 1/2 current encoder value* fields will return the 32 bits current encoder value. This value is a signed 32 bit value.

The *Motor 1/2 gear ratio* fields have not been implemented in the current version of the firmware.

The *Motor 1/2 P coefficient* fields specify the differential control loop P coefficient. This value is an unsigned byte.*

The *Motor 1/2 I coefficient* fields specify the differential control loop I coefficient. This value is an unsigned byte.*

The *Motor 1/2 D coefficient* fields specify the differential control loop D coefficient. This value is an unsigned byte.*

The *Battery voltage* field will return the current battery voltage. The high byte is the upper 8 bits of a 10 bit value. It may be used as an 8 bit representation of the battery voltage in units of 80mV. This provides a measurement range of 0 – 20.4 volts. The low byte has the lower 2 bits at bit locations 0 and 1 in the byte. This increases the measurement resolution to 20mV.

*Note: the preset PID coefficient values in the firmware have been tuned for the motors being used and it is recommended they not be changed. Any changes to these values will be lost at power down.

3.2. Parameter ordering

The motor controller mode, power and target position control parameters have been placed in order to permit single multi-byte writes to be used per channel to control its operation. If just the mode or power needs to be changed, then a single byte write can be used. If power and mode need to be changed, then a two byte write of mode, followed by power, can be used. All six bytes can be written to set up a new mode, power and target position. Since the two channels six byte areas are contiguous, a 12 byte write can be used to command both channels simultaneously.

3.3. Channel mode

The motor controller has one mode control byte per channel to control and monitor its operation.

D7	D6	D5	D4	D3	D2	D1	D0
Busy	Error	-	NTO	Rev	Lock	Sel 1	Sel 0

The Sel 1 and 0 bits encode the operation thus;

Sel	Action
00	Run with power control only
01	Run with constant speed
10	Run to position
11	Reset current encoder

The *Run with constant speed* command will cause the firmware to adjust the motor power to compensate for changing loads in order to maintain a constant motor speed.

The *Run to position* command will cause the firmware to run the motor to make the *current encoder value* to become equal to *the target encoder value*. It will do this using a maximum rotation rate as defined by the motor power byte. It will hold this position in a servo like mode until the *Run to position* command is changed or *the target encoder value* is changed. While the *Run to position* command is executing, the Busy bit will be set. Once the target position is achieved, the Busy bit will be cleared. There may be a delay of up to 50mS after a *Run to position* command is initiated before the Busy bit will be set.

The *Reset current encoder* will reset the current encoder value to 0 and then clear the command resulting in *Off-braked* state to be engaged.

The *Lock* bit is reserved and not implemented at this time.

The Rev bit may be used to alter the forward/reverse direction mapping for both the motor output and the encoder inputs. This function is primarily intended to harmonize the forward and reverse directions for motors on opposite sides of a skid-steer chassis.

The Error bit will be set if the motor controller has shut down due to a fault condition such as an output short circuit, output over-current or internal overheating has been detected. It will be cleared automatically after a one minute cool down period followed by a controller restart. The shut down function includes placing both motor channels into float mode.

The *NTO* or *No Time Out* bit may be set to stop the controller timing out if no I2C communication is received within 2½ seconds. In order to set the No Time Out state, the NTO bit must be set in both motor channel mode registers.

3.4. Channel power

The motor controller electronics is capable of operating each channel independently in PWM mode. This field may be set in the range of -100 to +100. 0 is used to set Off/Brake. Any value above 100 will be treated as 100, except –128 which is used to set Off/Float. Negative values run the motor in the reverse direction. The power should be set positive when being used as a parameter for the *Run to position* command. This field sets the rotation rate in *Run with constant speed* and maximum rotation rate in *Run to position* in the range of -1000 – +1000 degrees per second represented by the range -100 – +100. As with PWM mode, 0 is used to set Off/Brake while –128 is used to set Off/Float.

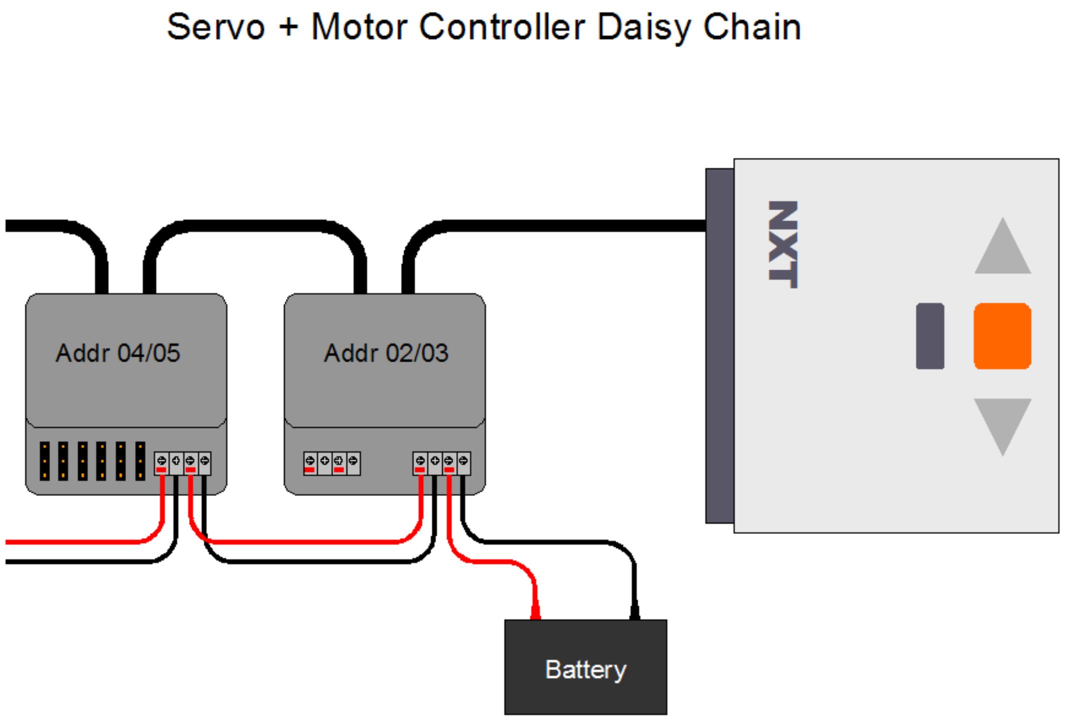
3.5. Channel current encoder value

The motor controller maintains a current encoder value for each motor channel. It is initialized to zero at power up and tracks all subsequent motor movement. It may be reset to zero using the *Reset current encoder* function. The encoder has a resolution of 1440 counts per revolution of the motor shaft or ¼ degrees per count.

3.6. Channel target encoder value

The motor controller uses this value to drive the motor to the target position when executing a *Run to position* command.

4. Appendix 1. Electrical Connections


Servo + Motor Controller Daisy Chain
Addr 04/05
Addr 02/03
NXT
Battery