Interfacing the new AndyMark NeveRest Motor with the HiTechnic Motor Controller

We have tested the new AndyMark NeveRest Motor and it works well with the TETRIX/HiTechnic Motor Controller. Here's what you should expect and what you need to do to make it work the way you want on an FTC robot. We will use *motorD* as our motor in our examples.

First we need to agree on some terminology. The TETRIX shaft encoder used with the TETRIX motors is a US Digital e4P with a 360 degree encoder wheel. Model E4P-360-236-N-D-H-D-B. To use US Digital's terminology it produces 360 Cycles Per Rotation (CPR) or 1440 Pulses Per Rotation (PPR). It is a quadrature encoder producing 4 Pulses per Cycle. In FTC use, with ROBOTC, it will produce 1440 PPR, ticks, or a count of 1440 in *nMotorEncoder[motorD]* per revolution.

The AndyMark NeveRest motor's built in encoder produces 280 cycles per revolution (CPR). We tested it. As expected with 280 CPR, it outputs 1120 PPR, ticks, or a count of 1120 in *nMot orEncoder[motorD]*
per revolution.

So, here's the only tricky bit. The NeveRest motor’s 1120 PPR is only approximately 78% of the TETRIX Motor’s 1440 PPR. You need to adjust your range when running under PIDControl from 0-100 down to 0-78.

The HiTechnic motor controller expects 1440 PPR and is "tuned" to drive a Tetrix motor at about 150 rpm when using PIDControl and a commanded speed of 100% *motor[motorD]=100*. That is, the motor controller will up the power as necessary to try to achieve 3600 Pulses per second when commanded to run the motor at 100% speed, or at about 1800 Pulses per second when commanded to run the motor at a speed of 50%.

At full speed, 150 rpm, the NeveRest motor encoder will only produce 2800 Pulses per second. Using the 1440 PPR Tetrix motor with shaft encoder will produce 2800 ticks at 78% speed or 117 rpm. Since the motor controller will try to adjust power as necessary using the built in PID to achieve the correct speed, a speed setting of 78%, i.e. *motor[motorD]=78;* will achieve full

speed on the NeveRest Motor. Likewise, a speed setting of 39,
motor[motorD]=39;
(½ of 78) will achieve approximately 50% speed on the NeveRest motor running under PIDControl. Any speed from 78% to 100% will cause the controller to run the motor at 100% speed. Please keep in mind that this only works this way under PIDControl and not under openLoop. Under openLoop mode you are just passing a percentage power to the motor controller and you will need to revert to using 0 through 100 for your power levels if you want to get full range & power out of the NeveRest motor.

Additional usage information

Bill Gardner over at Cheer4FTC is also testing the NeveRest Motor and pointed out that it "natively" spins the opposite direction of a TETRIX DC Motor. So be aware any time you replace a TETRIX motor with a NeveRest 40 or vice-versa you will want to reverse the direction of spin programmatically. If you are not using the built-in encoder you could also just reverse the red and black power wires...