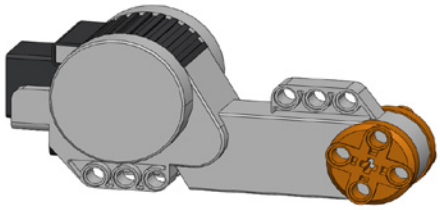
Reference

Motor Encoders

Each LEGO Smart Motor includes a built in encoder that can be used to program the robot to move around in a very precise manner. The built-in encoder has a resolution of 360 counts per revolution of the axle or one tick per degree. Therefore every time the robot wheel spins one wheel rotation the robot counts 360 encoder counts.



LEGO Mindstorms Smart Motor

What is an encoder?

A encoder is a measurement device which converts mechanical motion into electronic signals. The encoder is connected to the motors and outputs digital pulses which are converted by the NXT into usable information for programming our robots.

ROBOTC Reserve words for Encoders

nMotorEncoder[ ] is a reserve word in ROBOTC that allow a programmer to access the current value of the motor encoder. To access the value of the encoder attached to motorB use:

```
nMotorEncoder[motorB]
```

It is good practice to reset encoders to zero before you use them. To reset the value of motorC use the following code:

```
nMotorEncoder[motorC] = 0;
```

Typical Code for using encoders

```
1  task main()
2  {
3      nMotorEncoder[motorB] = 0;//reset the value of encoder B to zero
4      nMotorEncoder[motorC] = 0;//reset the value of encoder C to zero
5
6      while(nMotorEncoder[motorB] < 720)//while encoderB is less than 720
7      {
8        motor[motorB] = 50;//turn on motorB at 50% power
9        motor[motorC] = 50;//turn on motorC at 50% power
10     }
11
12     motor[motorB] = 0; //Turn off motorB
13     motor[motorC] = 0; //Turn off motorC
14 }
```

The code shown pictured above shows a typical use of an encoder.

Lines 3 & 4 clear the value of the encoders in motors B & C.

Line 6, "`while(nMotorEncoder[motorB] < 720)`" is a while loop with a condition. It says - while motorB's encoder value is less than 720.

Lines 7 & 10 make up the structure controlled by the while loop. Lines 8 & 9 turns on motors B & C at half power.

Lines 12 & 13 turn motors B & C off.

Reference

nMotorEncoderTarget

The nMotorEncoderTarget function uses the value of nMotorEncoder to move exactly to a specific position. Imagine if you were driving a car and came to a place where you needed to stop and then slammed on the brakes. That is how the nMotorEncoder function works; it slams on the brakes.
The code "`while(nMotorEncoder[motorB] < 720)`" gets to the encoder value of 720 and slams on the breaks, then the momentum of the robot causes the robot to slide past the stopping point.

"`nMotorEncoderTarget[motorB] = 720`" sets a target value of 720 and ROBOTC monitors, with the help of nMotorRunState, watches for 720 counts and begins to slow down as the "target value" is reached. If you need to be very accurate with movements, then you should use the nMotorEncoderTarget function.

Four steps to use nMotorEncoderTarget

There are four steps to using the nMotorEncoderTarget function.

1. Clear the Encoders — `nMotorEncoder [motorC] = 0;`
2. Set the Encoder Target — `nMotorEncoderTarget [motorC] = 720;`
3. Turn on the Motors — `motor[motorC] = 50; motor[motorB] = 50;`
4. Create a monitor to watch the motors — `while(nMotorRunState[motorC] != runStateIdle) { }`

What is nMotorRunState?

nMotorRunState is a function in ROBOTC that monitors the motor's "state"; Is the motor on, about to stop, or is it off? There are three possible conditions of the motors run state:

```
nMotorRunState[motorB] = runStateRunning; //the robot is moving
nMotorRunState[motorB] = runStateHoldPosition; //the robot is approaching the target and slowing down
nMotorRunState[motorB] = runStateIdle; //the robot has stopped moving
```

The "nMotorEncoderTarget" function monitors the "nMotorRunState" function until it reaches its target.

Example code for implementing nMotorEncoderTarget

```
1  task main()
2  {
3      nMotorEncoder[motorB] = 0;//reset the value of encoder B to zero
4      nMotorEncoder[motorC] = 0;//reset the value of encoder C to zero
5
6      nMotorEncoderTarget[motorB] = 720;//set the encoder target to 720
7      nMotorEncoderTarget[motorC] = 720;//set the encoder target to 720
8
9      motor[motorB] = 50;//turn on motorB at 50% power
10     motor[motorC] = 50;//turn on motorC at 50% power
11
12     while[nMotorRunState[motorC] != runStateIdle)
13     { /* This is an idle loop. The program waits until the condition is satisfied*/}
14
15     motor[motorB] = 0; //Turn off motorB
16     motor[motorC] = 0; //Turn off motorC
17 }
```