Worked Example: Velmex Serial Control

Many instruments are controlled by ASCII (string) commands communicated through the serial port of a computer. Consider the Velmex VP9000, which is a stepper motor controller that can be manually actuated with a joystick, or automatically with the serial commands outlined below. Once all of the configuration details are accounted for, it is a relatively simple matter to exchange these commands by "writing" to, and "reading" from, the port, provided all parameters are properly encoded as strings. Create a basic VI to send speed and distance commands to the controller in order to move the attached linear slide accordingly. Note that:

- All commands sent to the controller MUST be of type string, so all other data types (such as all of the numeric values) must be converted before use.
- The "F" command (which puts the controller in automation mode) need only be sent ONCE per session, and therefore is not included in the basic interface shown here (we don't want to send the command every time we run the VI). Likewise for the release command "Q." In fact, all of the initial setup and close functionality is included in a separate VI.
- All move commands must be followed by the "R" command to initiate. It is also good practice to begin commands with the clear command "C" whenever appropriate.
- Commands can be "piggy-backed" with commas.
- The controller cannot move any motor faster than 8000 steps/sec, nor slower than 2000 steps/sec, which should be accounted for in the controls.
- If we wish to re-use the VI repeatedly, then we must take care to let the motion finish before sending another command. This is particularly important if we build automatic repetition into the program.
- VISA stands for *Virtual Instrument Software Architecture*—it is a standardized approach to controlling compatible instruments connected via serial (RS-232, -422, -485), parallel, USB, etc.

Basic commands (these are only a small sample of all available):

C: clear command buffer
R: run current commands
F: enable RS-232 control (echo off)
Q: release RS-232 control

D: decelerate to stop
N: null (zero) position encoders
X, Y, Z, T: send position of corresponding axis
S*m*M*x*: set speed of motor *m* to *x* steps/sec
I*m*M*x*: increment motor *m* by *x* steps
I*m*M-*x*: decrement motor *m* by *x* steps