**Documentation for XYZMotion\_Station**

Program written by: Matt Bisbee at NARS Lab Ohio State University under Dr. Raymond Cao

Instructions Date: December 18, 2021

Overview:

This code is written using serial commands sent using the RS232 connections. The Velmex stages came with a list of commands that were written for encoding and being sent to the serial inputs. Each command starts off as a string and then is encoded in its own line, the port is flushed of previous inputs, and the port is written to the controller. The get response command uses the readline function to get the response the controller sends and decodes it then flushes the output port. Often commands such as movements will take time and if multiple commands are sent (either to the XYZ stage or to other stages), they must wait until the stage is done with the previous command.

Common response from the stage are S which is start, B which is busy, R which is ready as well as numerical responses for certain commands. Numerical responses come from asking the stage for position of each stage and speed for each stage. The responses for stage positions include 1 symbol and 7 digits, the symbol is either + or – in reference to the zero point. The 7 digits are actual number of steps for the position. The conversion of steps to mm is 0.0050 and that conversion is automatically made for both mm and inches. There is also a command to get all three positions for stages. There are commands for a relative movement and an absolute movement; the format for these commands are E,C,I1M\_\_\_\_\_,R and E,C,IA1M\_\_\_\_\_,R respectively. The E stands for connecting to the stage (unnecessary as already connected), C clearing previous commands, I for indexing the motor and the number (1 for X, 2 for Y, 3 for Z), M for move and the blank is filled with a number of steps to move, finally R to run the command. The A in the absolute move string is to move to the specified steps not move an amount of steps as in the relative movement.

The home command sends all three stages to the zero point in the order specific for our setup which is Z then Y, then X. This must be done in this way as the imaging station poses as an obstacle to the motion of the stage due to the protrusion of rotation stage and sample holder. There is a command in the script only to be used while the stages are uninhibited; it resets the zero positions of the stages in the case of an error, or turning off the stages when not zero’ed. An alternative method to reset the correct zero positions is to turn on the controllers and then manually jog the stages to the zero positions. This would be all the way down for Z, as far away from edge of box in Y, and all the way back for X (see graphic). Once jogged to those positions, turn off the controllers and when they turn on again that will be set as zero. It is imperative to have the stages in the correct zero positions as all movements are checked to ensure that it will not hit the box or wall, if the zeros are incorrect, the referenced limits are incorrect.

Note: The conventions shown below are what is coded into the movements. In actuality, the Y and Z directions have flipped positive directions from what is shown; for this setup, it made sense for the conventions to be made the way that they are listed.

Graphical user interface

Description automatically generated

Figure : Schematic of Cart for Referencing XYZ Positions

A picture containing indoor, computer, desk, cable

Description automatically generated

Figure : XYZ Stage Positive Axes Labeled X (stage 1) Y (stage 2) Z (stage 3)

A picture containing text, indoor

Description automatically generated

Figure : Controller with manual jog buttons green circle X, blue circle Y, red circle Z

Graphical user interface, text, application

Description automatically generated

A picture containing qr code

Description automatically generated