SERVOLOCK_TTL

The SERVOLOCK_TTL firmware module allows the user to control the stage position bi-directionally using TTL pulses. When placed in the SERVOLOCK_TTL state, motors are kept on with servo loop active and the target position is adjusted every time a TTL pulse is received. The duration of each TTL pulse determines whether a positive or negative adjustment is applied. The adjustment magnitude is the same as the last relative move and can be changed via serial command even when the SERVOLOCK_TTL state is active. Commanded moves and manual (joystick) moves are ignored when the SERVOLOCK_TTL state is active.

Example applications of the SERVOLOCK_TTL functionality are:

- Making slow constant-speed moves by periodically sending TTL pulses and using a small adjustment amount (e.g. a few encoder counts). It is possible to move slower moves this way than with the normal commanded moves, which by cannot go slower than 1 encoder count every 32 milliseconds. Similarly, SERVOLOCK_TTL function can be used to make moves at speeds between the quantized speed values of normal commanded moves.
- Implementing external feedback control for locking or tracking using a stage (e.g. home-built CRISP-like system). External equipment determines which direction the ASI stage needs to move and sends an appropriate duration pulse depending on the polarity. A similar thing can be done using the VECTOR command, but SERVOLOCK_TTL does not require any serial commands (sometimes a rate-limiter) and can also make arbitrarily small adjustments.
- Creating arbitrary continuous movement profiles. Although the ring buffer can be used, each new position initiates a new discrete move with a ramp up and ramp down which may be unwanted. Furthermore there is no limit on how long the path can be using SERVOLOCK_TTL.

This feature is initially only available on Tiger controllers, but there is nothing preventing the code from being ported to the MS-2000 controller if there is a need. Because TTL triggering is done on a per-card basis, to have truly independent control of X and Y axes they would be need to be on separate 2-axis cards. [1)]

How to use SERVOLOCK_TTL

Engage or disengage the SERVOLOCK_TTL functionality by using the LK command without any arguments (if a CRISP-enabled firmware is present, use the LK F command instead because the LK without arguments applies to CRISP which takes priority).

To provide for bi-directional control, short or long TTL pulses can be sent, and the threshold duration is set by the RT R command (defaults to 0.75 ms; note that there is timing jitter between -0.001 and -0.25 ms so a setting of 0.75 ms will for sure recognize pulses of 0.5 ms as short and 0.75 ms as long but pulses of 0.6 ms may sometimes be recognized a short and sometimes as long). Internally the implementation as follows: a counter is initialized on rising edge of the TTL pulse, and on a separate interrupt the counter is decremented every 0.25 ms until it reaches 0. On reaching 0, the state of the TTL IN0 pin is checked again and a decision is taken depending on whether it is high or low.

During SERVOLOCK_TTL operation, the TTL IN0 mode (TTL X command) is automatically set to value 11. Under normal circumstances the TTL IN0 mode will be restored to its prior value the when exiting SERVOLOCK_TTL operation. Do not change the TTL IN0 mode while SERVOLOCK_TTL is engaged.

As a safety provision, the excursion from the initial position (when first locked) is limited. The maximum distance (in either direction) is set by the LR Z command and defaults to 1 mm.

Position can be freely queried during SERVOLOCK_TTL operation.

Limitations

Experimentally the controller can keep up with TTL pulses at a 1kHz rate. There is no lower bound on the time between pulses.

It seems in testing that the controller catches every pulse, even when it is responding to serial commands. If you absolutely require this feature (e.g. creating extended arbitrary move profiles via TTL pulses) then it is worth verifying yourself.

Interaction with CRISP

The SERVOLOCK_TTL functionality uses many of the same commands as CRISP and indeed the internal implementation is similar to CRISP except that TTL pulses changes the target stage position instead of the CRISP hardware/firmware. When the SERVOLOCK_TTL firmware module is present along with CRISP, CRISP has priority. However, with CRISP present it is still possible to place the controller in the SERVOLOCK_TTL state by using the LK F command, using LK F=84 (ASCII letter T) to turn on SERVOLOCK_TTL control and LK F=90 (ASCII letter Z) to turn it off when done.

Serial Commands

Serial commands related to SERVOLOCK_TTL module:

- Command:LOCK (LK) 2016/02/23 20:02
- Command:LOCKRG (LR) 2016/02/23 19:42
- Command:RBMODE (RM) 2016/03/16 15:18
- Command:RTIME (RT) 2016/02/22 19:30
- Command:TTL 2017/07/26 13:03

advanced feature, tiger
[1)]
otherwise either both axes will both react together or only one will react according to the RM Y setting (but the two axes can have different relative move amounts).