

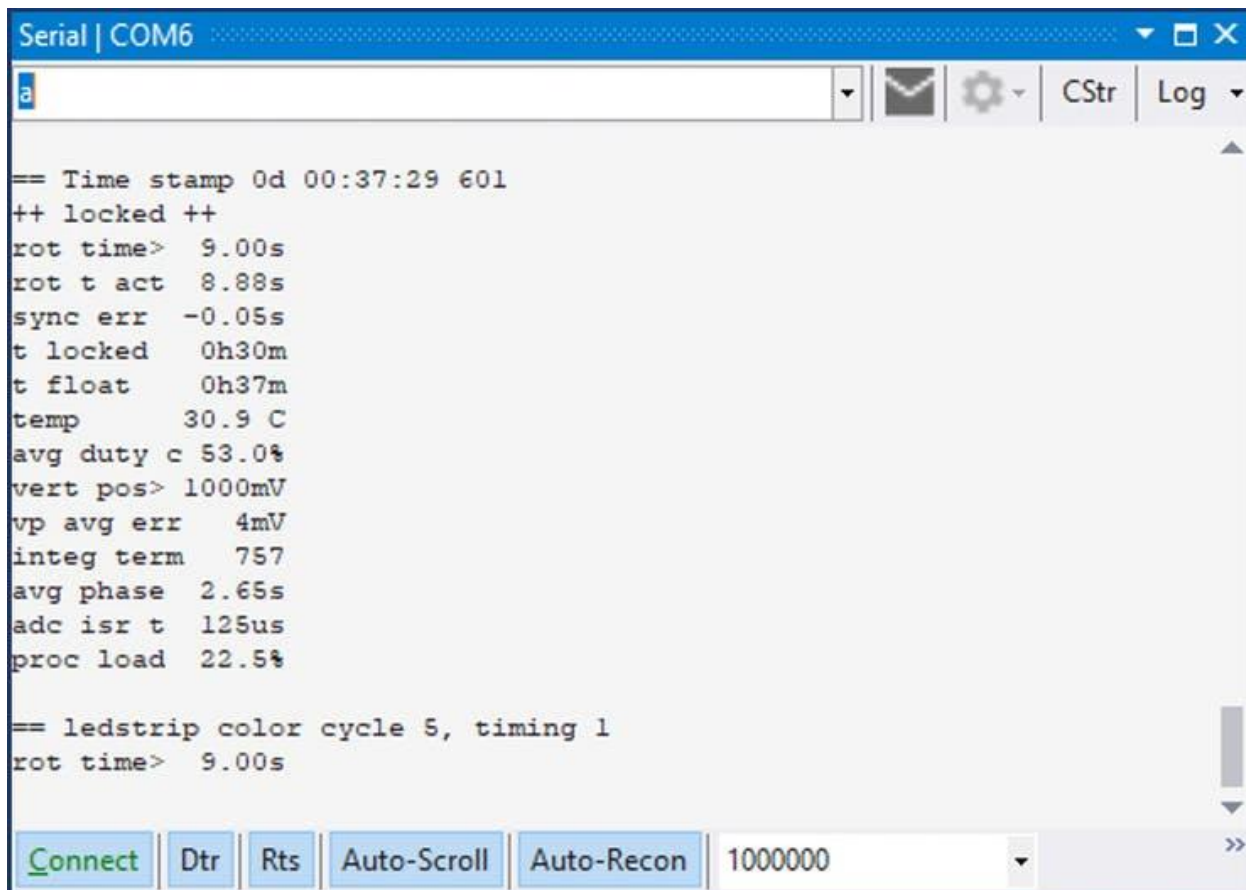
Serial communication

By using the Arduino serial monitor (or other properly configured 'terminal' software) via USB, you can print certain information and change a few settings.

The following information ('parameters') can be printed:

- Currently selected rotation time (**setting**)
- Actual rotation time
- Rotation time sync error: the accumulated globe rotation time error (when locked)
- 'Globe rotation locked' time. Reset when globe rotation is not locked
- 'Globe is floating' time. Reset when globe is not floating
- Temperature (degrees Celsius)
- Lifting magnet average duty cycle (%)
- Globe vertical position: reference (setpoint) in millivolts (**setting**). Note that setpoint values refer to values as read by the ADC, so they are scaled to 10 (or 15 - see PCB section) times the hall-sensor output for the set vertical position
- Smoothed average of the absolute value of the globe vertical position error in millivolts as read by the ADC
- Current value of the integration term, calculated by the globe lifting controller
- Average phase between rotating magnetic field and rotating globe (when locked). Can be used to calibrate the phase between rotating magnetic field and globe, set by the globe rotation controller before locking is obtained
- Average duration of the 'ADC conversion complete' ISR routine in microseconds (smoothed)
- Average processor load (smoothed)

In addition, each status change will be printed (e.g. the selected rotation time is reached).



```
Serial | COM6
a
== Time stamp 0d 00:37:29 601
++ locked ++
rot time> 9.00s
rot t act 8.88s
sync err -0.05s
t locked 0h30m
t float 0h37m
temp 30.9 C
avg duty c 53.0%
vert pos> 1000mV
vp avg err 4mV
integ term 757
avg phase 2.65s
adc isr t 125us
proc load 22.5%

== ledstrip color cycle 5, timing 1
rot time> 9.00s

Connect Dtr Rts Auto-Scroll Auto-Recon 1000000
```

Serial

monitor: 'A' (show all) command output

Use the following commands to print parameters and their values or change settings (sending a 'Carriage Return' is not needed, although you might have to press Enter to send your command to the Arduino, depending on the terminal software you use).

- '+' or '-' : print next or previous parameter. When in edit mode: select next or previous parameter value from list.
- 'E': edit current parameter. When in edit mode: end editing and save currently selected parameter value in EEPROM memory. Only available for changing rotation time and fine-tuning globe vertical position, no function for other parameters.
- 'C': cancel editing without saving. No function when not in editing mode.

- 'S' to start or stop showing live values. This will start or stop printing the actual value of the currently selected parameter regularly to the serial monitor (for instance the time it took to complete the last globe rotation). Does not apply to settings.
- 'A': print all parameters at once, as well as current led strip settings.

```

Serial | COM6
e+e

++ not locked (0d 00:04:31 854) ++
rot time> 12.00s

++ locked (0d 00:05:17 115) ++
rot time> 12.00s
rot time> 12.00s << +, - to change value, E to end edit, C to cancel
rot time> 9.00s << +, - to change value, E to end edit, C to cancel
rot time> 9.00s

++ free running (0d 00:05:50 784) ++
rot time> 9.00s

++ measuring (0d 00:05:53 095) ++
rot time> 9.00s

++ not locked (0d 00:06:04 961) ++
rot time> 9.00s

++ locked (0d 00:06:42 188) ++
rot time> 9.00s

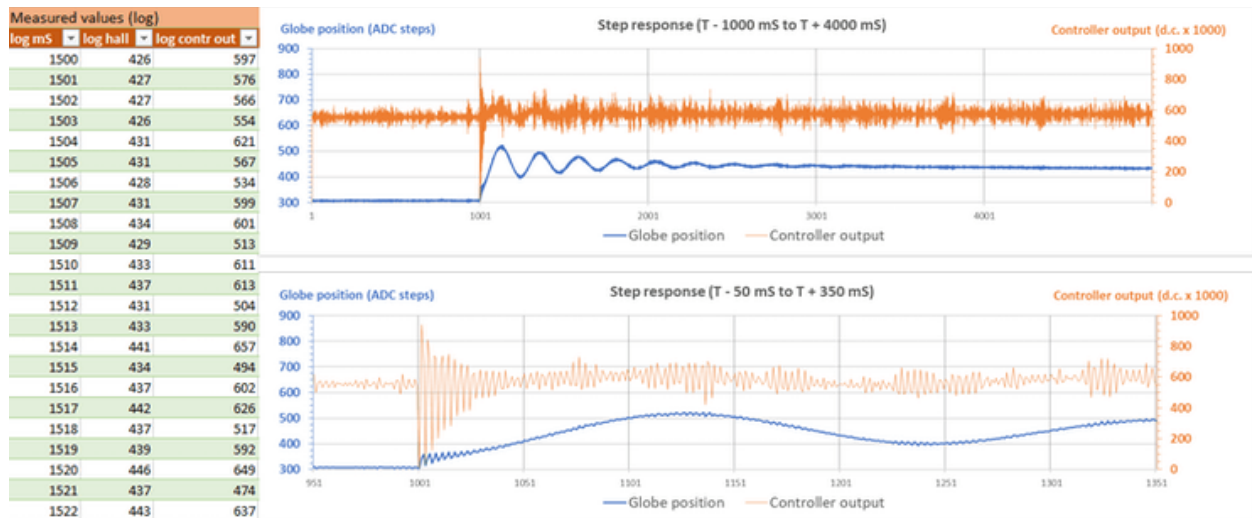
```

Serial monitor: status and commands example

Additional commands:

- '?': print available commands (help).
- 'T': print time stamp (time in milliseconds since last reset - a real time clock is not available).
- 'LC0' to 'LC5': select a led strip cycle. 0 = Off, 1 = constant brightness white, 2 = constant brightness magenta, 3 = constant brightness blue, 4 = fade between white and blue, 5 = smoothly cycle between all colors (from blue to cyan, green, yellow, red, magenta, and blue again,...). Selection is stored in EEPROM memory.

- 'LT1' to 'LT4': change led strip cycle time. 1 = fastest (minutes), 4 = slowest (hours). Note that the timing is not relevant for all led strip cycle types. Selection is stored in EEPROM memory.
- 'R0' and 'R1': (step) response. Applies a step change ('R1') or **no** step ('R0') to the globe vertical position reference (setpoint) and then sends the measured response (actual globe vertical position) and controller output (PWM duty cycle) to the serial monitor where it can be logged for evaluation in excel etc. The step applied ('R1' command only) increases the vertical position setpoint with 100 ADC steps. Expressed in millivolts at the hall sensor output: $100 \text{ ADC steps} * (5000 \text{ millivolts} / 1023 \text{ ADC steps}) / 10 = 49 \text{ millivolts}$ (if the analog gain is set to 10 - see section 'PCB'). The Arduino starts sending data to the serial monitor, 1 second before the step is applied ('R1' command) and continues sending data for 20 seconds. Every millisecond, a line containing three values is sent, the values being separated by a semicolon. Value 1: millisecond (1 to 20000) Value 2: lifting hall-effect sensor value read (in ADC steps) Value 3: controller output (PWM duty cycle as a value from 0 to 999 = 100%)



Logged step response (extract) and excel step response charts - analog gain is 15

A few notes on the use of the 'R0' and 'R1' commands:

- The 'Rn' command requires a high baud rate: the baud rate set by the program is 1000000, so this shouldn't be a problem.

- The value of the integration term (see section 'Globe lifting control system') including extra accuracy digits (see section ('Arduino Nano code')) at the start of the 'Rn' output (first millisecond) is printed to the serial monitor as well. This allows 'replaying' controller calculations in excel based on logged hall-effect sensor values (in ADC steps), verifying that the calculated controller output in excel is identical to the logged controller output.
- When sending step response data has ended, the globe vertical position is gently brought back to its original setting.
- Do not use this command while you are showing Live values (see 'S' command)