

# Welcome to Arduino\_Slider's documentation!

This library contains an API for interacting with an Arduino microcontroller loaded with firmware to move a rotating slider.

## **sliderlib.get\_datetime\_str()**

returns current datetime as a string with format yyyy-mm-dd-HH-MM-SS

**Returns:** returns current datetime as a string with format yyyy-mm-dd-HH-MM-SS

## **sliderlib.get\_time\_str()**

returns current time as a string with format HH-MM-SS

**Returns:** returns current datetime as a string with format HH-MM-SS

## **sliderlib.load\_mapping(name, directory='mappings')**

returns the mapping 'name' stored in 'directory' just a wrapper around pickle load

**Parameters:**

- **name** –
- **directory** –

**Returns:**

## **sliderlib.store\_mapping(d, name=None, directory='mappings')**

stores mapping as a pickle file just a wrapper around pickle dump

**Parameters:**

- **d** – mapping dictionary
- **name** – name of mapping
- **directory** – directory to store mapping

**Returns:**

*class sliderlib.SliderController(mapping=None, angle\_order=None, dest\_folder=None, animal\_name=None, exp\_tag=None, cell\_id=None, title\_string=None)*

**\_\_init\_\_**(mapping=None, angle\_order=None, dest\_folder=None, animal\_name=None, exp\_tag=None, cell\_id=None, title\_string=None)

create slider object, connect to Arduino

**Parameters:**

- **mapping** – mapping between slider angle and desired speed
- **angle\_order** – order to iterate through angles
- **dest\_folder** – folder to store generated data files
- **animal\_name** – identifier for animal
- **exp\_tag** – tag to identify experiment (eg 'slow')
- **title\_string** – custom title string to replace default naming convention of name\_tag\_dateTime

## **add\_note(note, index=-1)**

add a note to 'notes' column of data, saves changes

**Parameters:**

- **note** – text note to add
- **index** – trial to which note should be added. default is to add to last trial run

**Returns:**

## **analyze\_test\_log()**

print fwd and bwd mean velocity and stdev for each angle/motor\_speed combination that was tested

**Returns:**

## **engage\_stepper()**

send command to Arduino that will engage the stepper motor

**Returns:**

**init()**

move stepper motor to first angle, set slider to move forward next

**Returns:**

**move\_slider**(*speed=None, num\_slides=0, log=True*)

move slider to other end of track, writes new data to file after each time slider is moved

**Parameters:**

- **speed** – (optional) speed with which to move, if None passed in uses mapped speed
- **num\_slides** – number of round trips slider should make (pass in 0 to just move to other end of track)
- **log** – (bool) add slides to log

**Returns:**

**move\_stepper\_to**(*pos*)

move stepper to nearest possible angle to specified angle

**Parameters:** **pos** – angle in degrees

**Returns:**

**print\_slider\_pos**()

use sparingly, high error rate query Arduino for current slider position and print to screen

**Returns:**

**print\_stepper\_pos**()

print full stepper info to screen, commanded position, real position, and current step number

**Returns:**

**run\_all**(*num\_slides*)

iterate through all angles in angle\_order and move slider num\_slides number of times (round trips)

**Parameters:** **num\_slides** – number of round trips slider will make at each angle

**Returns:**

**send\_cmd**(*cmd*)

send string command to Arduino

**Parameters:** **cmd** – (string) command to send

**Returns:**

**set\_angle\_idx**(*idx=0*)

go to angle at idx in angle\_order

**Parameters:** **idx** – index of desired angle in angle order

**Returns:**

**stepper\_next**(*deg=None*)

move stepper to next angle in angle\_order if an angle is passed in, stepper will rotate forward that number of degrees

**Parameters:** **deg** – (optional and not recommended) move stepper forward this many degrees

**Returns:**

**stepper\_prev**(*deg=None*)

move stepper to previous angle in `angle_order` if an angle is passed in, stepper will rotate backward that number of degrees

**Parameters:** `deg` – (optional and not recommended) move stepper backward this many degrees

**Returns:**

**`store_test_log`**(*title=None, dest\_folder=None*)

write test log to file defaults to prepending '[testLog\\_](#)' to standard title and storing in standard data directory just a wrapper around pickle dump

**Parameters:**

- `title` – (optional) name of test\_log file
- `dest_folder` – directory in which to store test\_log

**Returns:**

**`test_all`**(*num\_slides=5*)

runs test\_slider speed for all angles using current speed mapping errors a lot, may be better off running one at a time then resetting everything

**Parameters:** `num_slides` – number of slider round trips to take at each angle

**Returns:**

**`test_slider_speed`**(*motor\_speed=150, num\_trials=1*)

NOTE: this fails a lot after the angle is changed. also fails if slider stalls.

all successful slides were recorded, you may still analyze these. start new session to clear error.

record the average speed of the slider through the middle third of the track in meters per second call `analyze_test_log()` to print average speeds to screen

**Parameters:**

- `motor_speed` – value 0 - 255
- `num_trials` – number of round trips slider should take

**Returns:** `timestamps_fwd, pos_fwd, timestamps_bck, pos_bck, f_speed, b_speed` from last trial

**`write_data_to_csv`**(*file\_path=None*)

write trial log to file (note this is called anytime the slider is moved or a note is added) if no argument is passed, will store using values passed in when object was created:

`dest_folder/animalName_expTag_dateTime.scv`

**Parameters:** `file_path` – full path (including file name) to write data

**Returns:**

## Indices and tables

- [Index](#)
- [Module Index](#)
- [Search Page](#)