

## Documentation for ELL14 Rotational Mount Control Code

### Overview

The ell14 class is designed to communicate with the ELL14 rotational mount via a serial port connection. This documentation explains the functionality of the class, how to access the rotational mount, and provides examples of how to use the class in MATLAB.

### Connecting the ELL14 Rotational Mount

Before using the ell14 class, the rotational mount must be connected to your system via a serial communication port. To determine which COM port the device is connected to, follow these steps:

1. Connect the rotational mount to your computer.
2. Open **Device Manager**.
3. Look under the **Ports (COM & LPT)** section for a device labeled as a serial connection (e.g., "USB Serial Device" or the name of the manufacturer). You will see a "COM#" next to the device name. This is the port number you will use to communicate with the rotational mount.

Once you've identified the correct port (e.g., COM3), use it to instantiate the ell14 class in MATLAB.

### Class Definition and Properties

```

1. classdef ell14 < handle
2.     properties
3.         device % Serial device object
4.         addr  % Address of the rotational mount
5.         Prop  % Placeholder for additional properties
6.     end
7.     properties (Constant)
8.         baudrate = 9600 % Communication baud rate
9.     end
10. end
11.
```

- **device:** A serialport object that establishes a connection with the rotational mount.
- **addr:** Holds the address of the rotational mount.
- **baudrate:** The baud rate is set to 9600, which is the standard communication speed for this device.

## Methods

The following methods are implemented in the ell14 class:

### Constructor: Initialize the Serial Port

```
1. function obj = ell14(port)
2.   obj.device = serialport(port, obj.baudrate);
3.   obj.device.DataBits = 8;           % 8 data bits
4.   obj.device.StopBits = 1;          % 1 stop bit
5.   obj.device.Parity = "none";       % No parity
6.   configureTerminator(obj.device, "CR/LF");
7. end
```

- **port:** The COM port to which the device is connected (e.g., 'COM3').
- This constructor initializes the serial connection to the rotational mount, configuring data bits, stop bits, parity, and the communication terminator.

### Set Device Address

```
1. function setaddress(current_address, set_address)
2.   command = ["TX " address 'ca A'];
3.   writeline(obj.device, command);
4.   obj.addr = address;
5. end
```

- **current\_address:** The current address of the rotational mount.
- **set\_address:** The new address to set for the mount.
- This function sends a command to set the address of the mount.

### Rotate Forward

```
1. function message = rotate_foward()
2.   command = ["TX " num2str(obj.addr) 'fw'];
3.   writeline(obj.device, command);
4.   message = readline(obj.device);
5. end
```

- Rotates the mount forward.
- Returns a message indicating the result of the command.

### Rotate Backward

```
1. function message = rotate_Backward()
2.   command = ["TX " num2str(obj.addr) 'bw'];
3.   writeline(obj.device, command);
4.   message = readline(obj.device);
5. end
```

- Rotates the mount backward.
- Returns a message indicating the result of the command.

## Go to Home

```
1. function message = Home()
2.     command = ['TX "' num2str(obj.addr) 'ho1"']
3.     writeline(obj.device,command);
4.     message = readline(obj.device);
5.     end
6.
```

## Change Step Size (Precision)

```
1. function step = change_step(degree_of_precision)
2.     degtorad = degree_of_precision * (pi/180); % Convert degree to radians
3.     step = round((143360/(2*pi)) * degtorad); % Calculate step size
4.     hex = dec2hex(step); % Convert step size to hex
5.     L = strlength(hex);
6.     steps = [];
7.     for i = 1:L-L
8.         steps = [steps '0'];
9.     end
10.    steps = [steps hex];
11.    command = ['TX "' num2str(obj.addr) 'sj' num2str(steps) '"'];
12.    writeline(obj.device, command);
13.    readline(obj.device);
14. end
```

- **degree\_of\_precision:** The angle in degrees that you want the mount to step by.
- This method converts the desired step angle into radians and sends a command to adjust the mount's precision. The command is formatted and sent as a hex string to the rotational mount.

## Example Usage

Here are examples of how to use the ell14 class in MATLAB:

### Connecting to the Device

```
1. % Specify the correct COM port
2. port = 'COM3';
3.
4. % Create an instance of the ell14 class
5. mount = ell14(port);
```

### Setting the Device Address

```
1. % Set the current address of the device to 1 and change it to 2
2. mount.setaddress(1, 2);
3.
```

### Rotating the Mount Forward and Backward

```
1. % Rotate the mount forward
2. message = mount.rotate_foward();
3.
4. % Rotate the mount backward
5. message = mount.rotate_Backward();
```

### Changing the Step Precision

```
1. % Set the step size to 0.5 degrees
2. mount.change_step(0.5);
```

## **Conclusion**

This ell14 class provides a simple interface for controlling the ELL14 rotational mount via serial communication. Make sure to check the COM port in Device Manager before initializing the class, and follow the examples provided to control the rotational mount's movements and precision.

Email: [rlameche@terpmail.umd.edu](mailto:rlameche@terpmail.umd.edu)