Table of contents

[Introduction 1](#_Toc186040447)

[**Overview.** 1](#_Toc186040448)

[**Structure.** 1](#_Toc186040449)

[**Protocol.** 1](#_Toc186040450)

[API 2](#_Toc186040451)

[**Protocols.** 2](#_Toc186040452)

|  |
| --- |
| Introduction |

**Overview.**

This project aims at enhancing accuracy of a step motor by integrating digital readout (DRO) as the encoder.

The step motor is usually easy to be controlled by sending PWM commands, while the disadvantage comes from the lack of encoder. Thus, the pulses sent to the motor is commonly used to viewed as the travelling distance or further converted to the coordinate. This system combines the motor and the DRO. That is, it relies on the DRO readings as the encoder instead of the accumulative pulses.

Further, this system includes a UI interface providing basic functionalities, such as triggering the motor, emergency stop and data visualization of the relationship between pulses and encoder readings.

**Structure.**

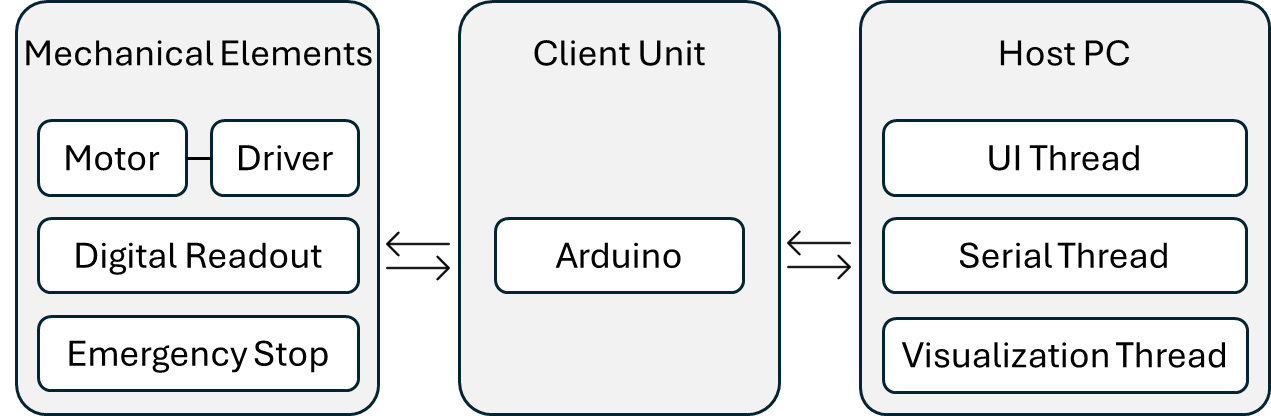


Figure 1.1 Structure Outline

**Protocol.**

From the structure outline, there are four interfaces.

1. Client Unit and Host PC:

Both interfaces on Arduino Mega (Client Unit) and Host PC are USBs. That is, PC would send and receive signals from Arduino Mega via serial communication. In the UI software, there will be a thread that keeps updating bytes coming in and out. For commands sent out to Arduino Mega, e.g. actuating motors, will be monitored by a function connected to button-clicked event. As for status being reported from Arduino Mega, e.g. emergency stop, DRO readings, will be monitored by another function that manage the following actions, including either updating plot, software UI, or showing alert on the message box. The following part of this document will include the format of the protocols and what commands to trigger actions.

|  |
| --- |
| API |

**Protocols.**

In-bound command (command from PC to Arduino)

* Data type: String
* Data length: 7 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. of bytes | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Name | axis | direction | stroke  (10 thousand digit) | stroke  (thousand digit) | stroke  (hundred digit) | stroke  (tens digit) | stroke  (unit digit) | End of command |

* Byte of string
* Details

|  |  |  |
| --- | --- | --- |
| axis | direction | remark |
| L: linear | +: forward (x) | Move probe away from original point |
| -: backward (x) | Move probe closer to original point |

|  |  |  |
| --- | --- | --- |
| axis | pulse | remark |
| L: linear | 0-99999 | Pulses send to motor |

|  |  |  |
| --- | --- | --- |
| name | symbol | remark |
| end of command | # | Arduino will execute command in front of ‘#’ |

Out-bound command (command from Arduino to Arduino)

* Data type: String
* Data length: Flexible
* Byte of string

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. of Segment | 1 | 2 | 3 | 4 | 5 |
| Name | Source Symbol | Pulse | Split symbol | Encoder Reading | End symbol |

* Details

|  |  |  |
| --- | --- | --- |
| name | symbol | remark |
| Source | R | The command comes from the Router |

|  |  |  |
| --- | --- | --- |
| name | symbol | remark |
| Status Symbol | R | Return linear position while the linear stage is moving |
| Status Symbol | F | Return when linear movement is complete |