Hardware Setup:

The general structure of electronic hardware is illustrated in Fig. 1. The main controller is a Raspberry Pi 3 Model B, that has a real-time operating system achieved via Preempt-RT patch (https://rt.wiki.kernel.org/). A Maxon 70/10 motor drive was employed with Current Control configuration. Analog motor drive commands were sent via an D/A converter (Waveshare 12-bit) which can communicate with the Raspberry Pi unit via SPI (Serial Peripheral Interface). The two 23-bit encoders (Broadcom AS38-H39E-S13S), namely position encoder and torque encoder were utilized to measure motor side angle and spring deflection, respectively. A custom-made PCB was used to transfer encoder data to Raspberry Pi 3 unit via SPI. The circuit schematic and PCB layout were included as the supplementary material. Please see *CustomCircuit_schematic.pdf and CustomPCB_schematic.pdf*

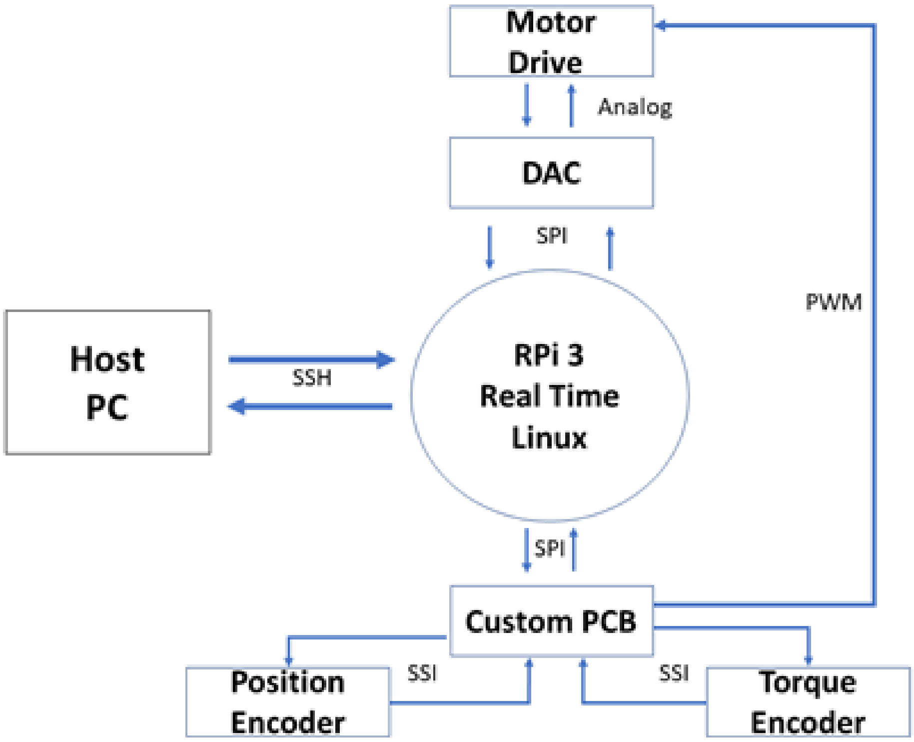


Figure 1: General structure of the electronics hardware.

Software Files:

For each controller, the following C files are provided in a respective folder. When experimentally implementing a controller, make sure that the following files are included in the same folder alongside with makefile.h and myrunner.m. The respective role of these files are as follows.

SEALibgen.c : This script includes the necessary functions that are necessary to use the SEA unit. It is same for all the controllers.

SEALibgen.h : It is the header file of SEALibgen.c.

SEALibex.c : This script includes the controller-specific functions that are necessary to implement the torque controller in question. The duration of the experiment can be set using this script.

SEALibex.h: It is the header file of SEALibex.c.

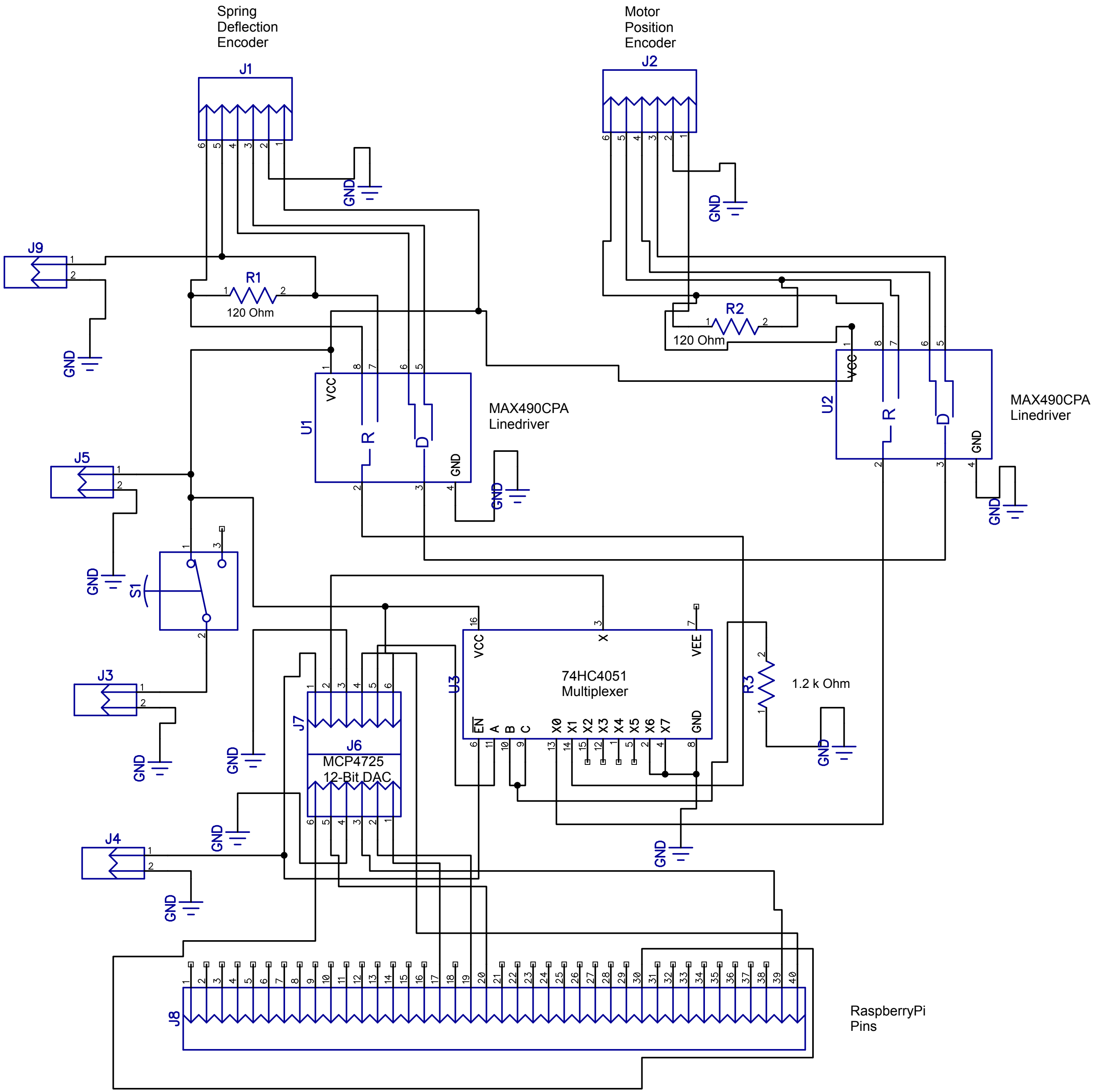
SEACont.c : This script executes the controller in question. In other words, all the necessary operations are handled in this file, such as, rendering the real-time control, reading of encoders, generating the reference signal, sending motor command signals, and performing necessary mathematical operations

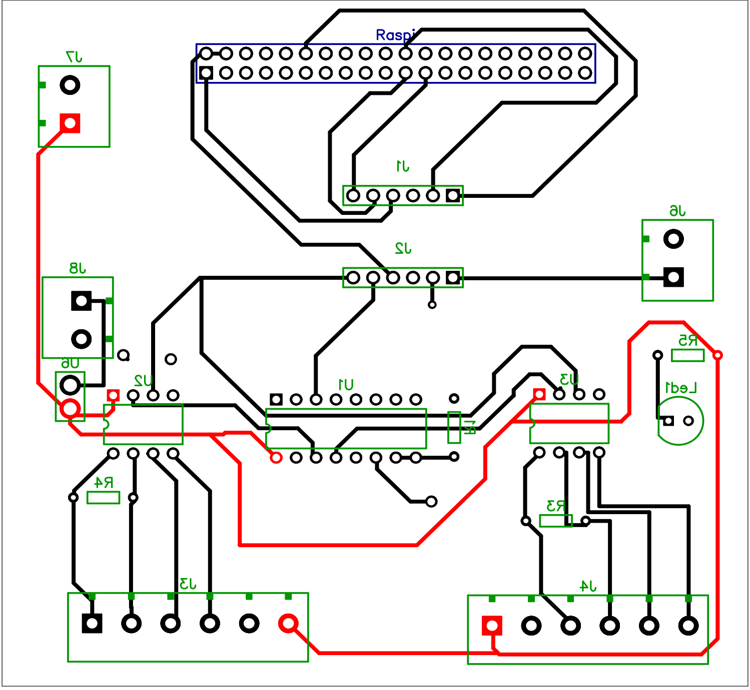
Instructions to Use the Software:

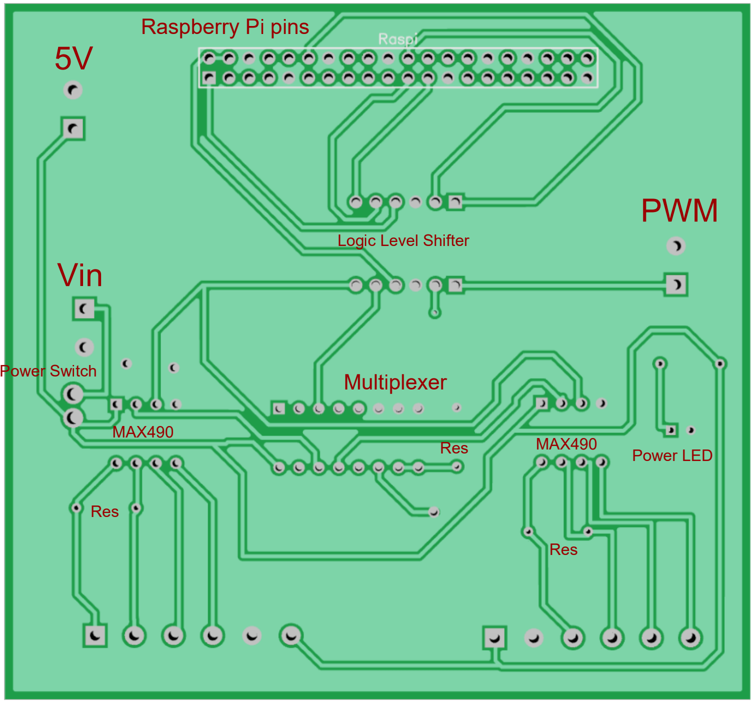
1) Complete the connections provided in Fig. 1. Encoders are connected to the custom PCB board. Encoder – PCB connections are tabulated in Table I. Raspberry Pi 3 is connected to the computer via an ethernet cable. Raspberry Pi-3 compatible DAC connections can be obtained from https://www.waveshare.com/wiki/High-Precision_AD/DA_Board. For Maxon 70/10 connections, refer to its hardware manual
2) Make sure that the libraries, necessary header files and C files are stored in a designated folder, alongside with the Makefile.h header file.
3) The execution of the experiment is initiated via the MATLAB script named myrunner.m
4) When myrunner.m is exectured, it transfers the necessary .c and .h files to the Raspberry Pi 3 unit.
5) Upon the completion of an experiment, all the necessary data are saved to workspace for further investigation.

Table 1: Encoder-Custom PCB Connection

PCB Pin Number	Encoder Cable Color	Cable Function
1	Red	Vcc, Positive Supply (5 V)
2	Black	Ground
3	White	SCL-
4	Brown	SCL+
5	Blue	Dout-
6	Orange	Dout+


Spring
Deflection
Encoder
J1
Motor
Position
Encoder
J2
J9
R1
120 Ohm
R2
120 Ohm
U1
MAX490CPA
Linedriver
U2
MAX490CPA
Linedriver
J5
S1
J3
J7
J6
MCP4725
12-Bit DAC
U3
74HC4051
Multiplexer
R3
1.2 k Ohm
J4
J8
RaspberryPi
Pins
GND





Encoder 1 Encoder 2