SUPPLEMENTARY METHODS: MOTION CONTROL AND DATA ACQUISITION

1. Motion Control via G-code and Python
The custom immersion system's motion is controlled by sending G-code commands to a BIGTREETECH SKR MINI E3 V3.0 control board running Marlin firmware. The control board is powered by a Mean Well LRS-100-24 AC-DC power supply. Communication is handled via a USB serial interface using Python scripts.

Two core functions were used to control X-Y motion:

- • gcode_homing: Aligns the transducer with a reference point.
  - G90: Set absolute positioning mode
  - G0 X1 Y1: Move to an intermediate location
  - G28: Execute homing
  - Time delays are inserted between commands to ensure stability
- • gcode_move_to: Moves the transducer to a specified (X, Y) coordinate using absolute positioning.
  - G0 X{target_x} Y{target_y}
  - Time delay added after each move

2. Signal Acquisition Using SCPI and PyVISA
Waveform data was acquired from a Keysight DSOX3024G oscilloscope using SCPI commands sent via the PyVISA Python library. Communication is over USB-B. A custom Python script sets up the oscilloscope, queries waveform settings, and retrieves the data.

Core SCPI Commands Used:

- • :WAVeform:SOURce CHANnel1 — Select input channel
  • :WAVeform:FORMat ASCii — Set waveform format to ASCII
  • :WAVeform:XINCrement? — Query time increment between samples
  • :WAVeform:DATA? — Retrieve waveform voltage data

The waveform data includes a header that contains metadata (not voltage values). After header parsing, the time axis is reconstructed using the XINCrement value, and voltage values are extracted from the returned ASCII data stream. Each waveform is saved as a .csv file labeled with the corresponding X-Y scan coordinates.

Prepared by: Harshith Kumar Adepu
Affiliation: Purdue University, IMPULSE Research Group