P35 TUE - 20

Smart walker for clinical rehabilitation

Table of Contents

1. Milestone 1 feedback
2. Overview
3. Progress and Deviations
4. Technical Updates and Subsystem Design
5. Circuit Diagrams
6. Test Plan and Results
7. Prototyping & Fabrication Plans
8. Thinking Ahead of Milestone

01

Milestone 1 Feedback

Addressing Feedbacks

Feedback

- Focus on Human Machine Interaction

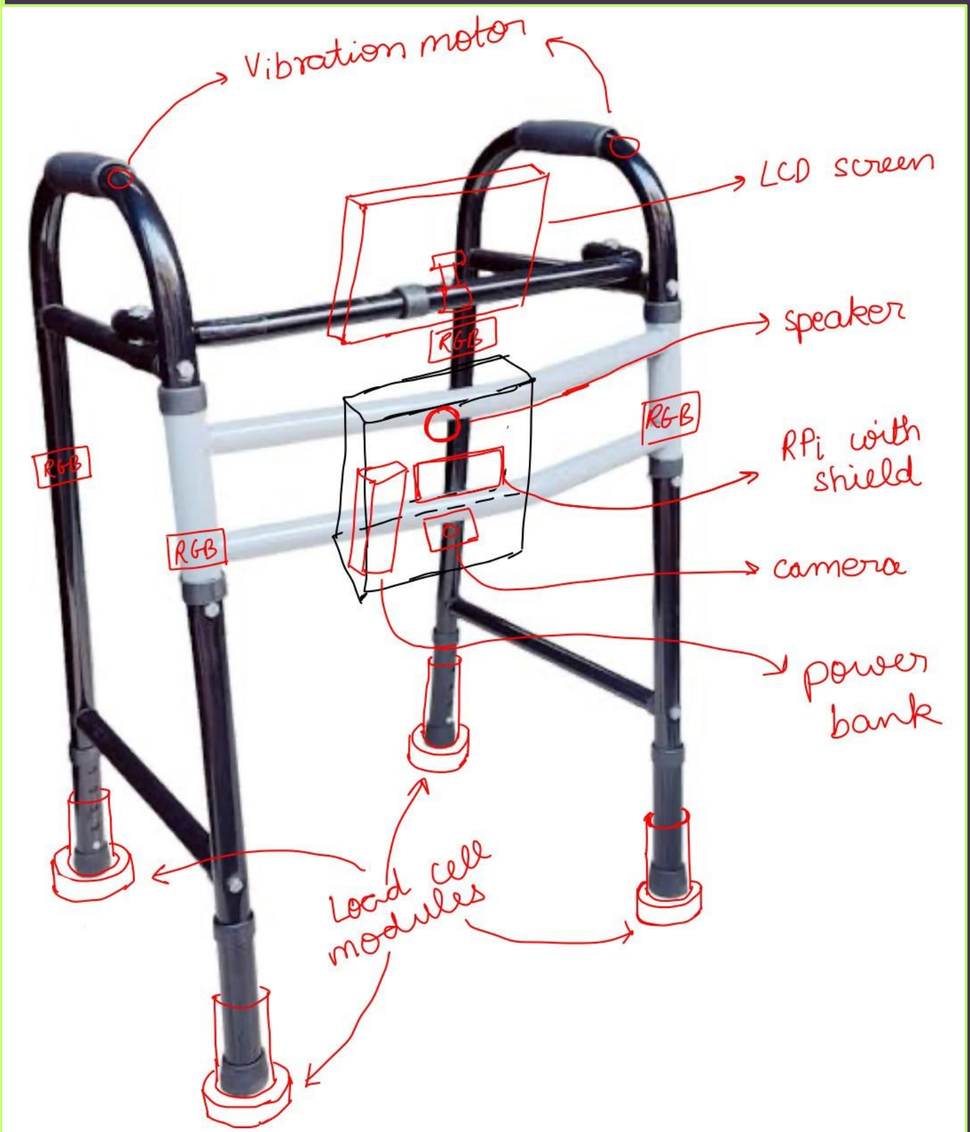
These issue causes :

- Incorporated haptic sensing for when to look at the screens

02

Overview

Overview:



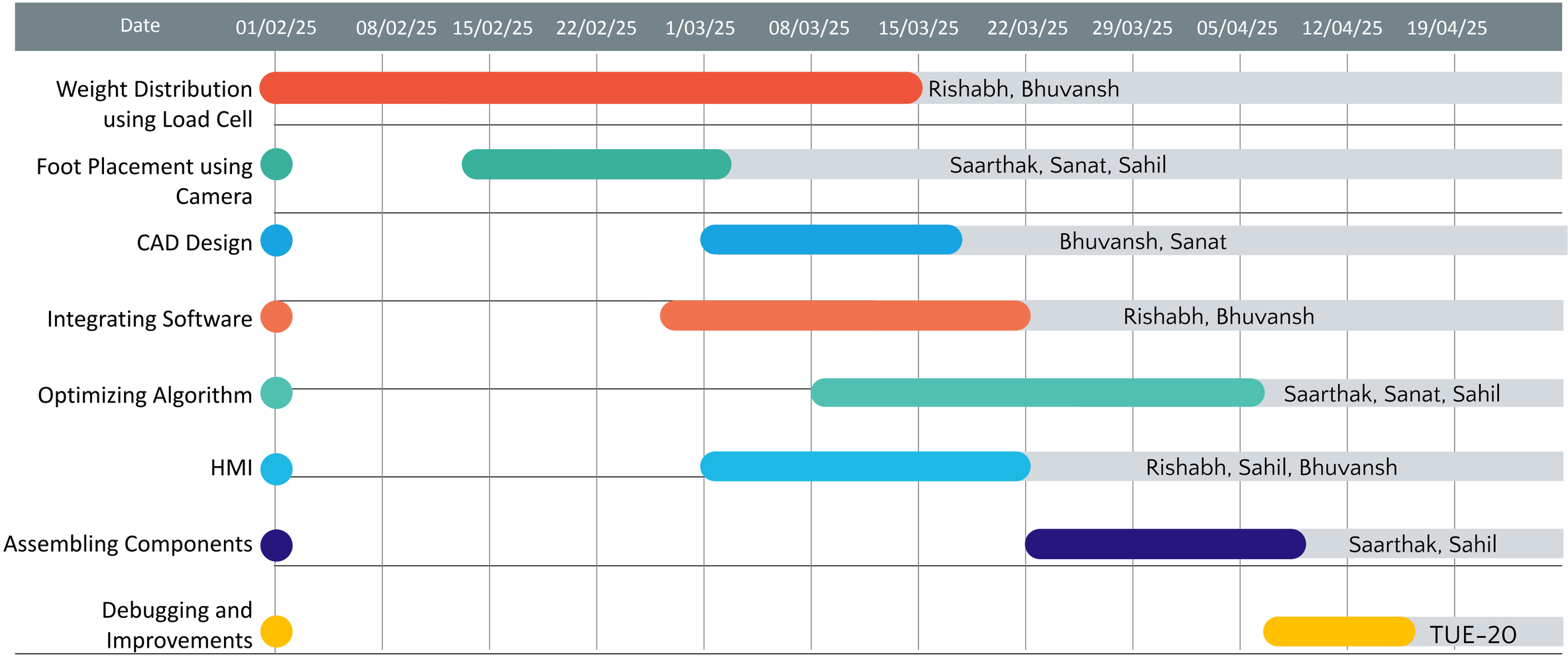
Key Components:

- **Microcontroller** – Raspberry Pi 5 for processing and control.
- **Sensors** – Load cell with HX710A ADC for weight measurement.
- **Stereo camera** - for pose detection.
- **Communication Protocols** – SPI for LED control, PWM for vibration motor, bit-banging for ADC communication.
- **Display & Audio** – Screen with HDMI interface; speaker for audio output.
- **Software & Algorithms** – YOLO-based pose detection, real-time data processing, and calibration techniques.

03

Deviations and Progress

GANTT CHART


Date
01/02/25
08/02/25
15/02/25
22/02/25
1/03/25
08/03/25
15/03/25
22/03/25
29/03/25
05/04/25
12/04/25
19/04/25
Weight Distribution using Load Cell
Rishabh, Bhuvansh
Foot Placement using Camera
Saarthak, Sanat, Sahil
CAD Design
Bhuvansh, Sanat
Integrating Software
Rishabh, Bhuvansh
Optimizing Algorithm
Saarthak, Sanat, Sahil
HMI
Rishabh, Sahil, Bhuvansh
Assembling Components
Saarthak, Sahil
Debugging and Improvements
TUE-20

Deviations:

- **Load Cell Delay:** The load cell hasn’t arrived yet, so we’ve focused on testing other subsystems like pose detection and camera integration.
- **Stereo Module Compatibility:** The stereo module isn’t compatible with Raspberry Pi 5, so for testing purpose we have used the camera module 2 noIR.

04

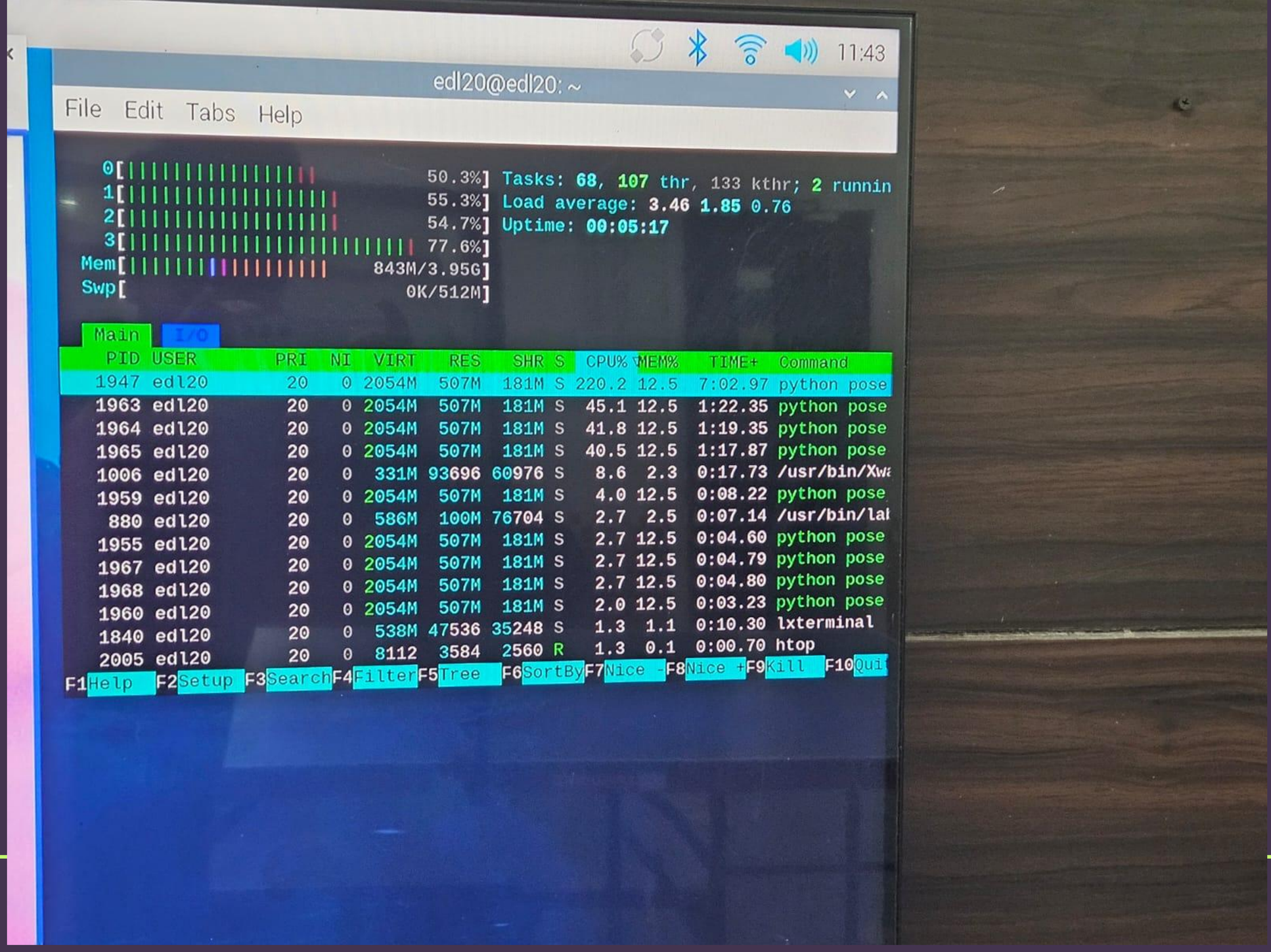
Technical Updates & Subsystem Design

Load Cell & ADC Selection:

- The HX711 and HX710a ADC modules were evaluated for load cell data acquisition.
- The load cell's expected voltage output is ±4.95 mV (max), while the ADC input range is approximately 12 mV for both HX711 and HX710a.
- Sampling Rate Considerations:
  - HX711: 10 SPS / 80 SPS
  - HX710A: 10 SPS / 40 SPS
  - Real-time applications need a minimum of 24 SPS, leaving us with two viable options: 40 SPS and 80 SPS. At higher sampling rates, noise and offset are lower for HX710A.
- Final Selection: While both will be used for prototyping, the HX710A has an integrated temperature sensor, aiding in calibration, making it the preferred choice for the final design. Also HX711 has extra channel which we are not using.

Pose Detection & Processing Load:

- The YOLO model was successfully tested for pose detection.
- CPU utilization was 60-70%, and GPU (HDMI output driving 1080p TV) was 70%.
- Since our final screen will have a lower resolution than 1080p, we expect even better performance.
- The model currently runs at 40-50 FPS, ensuring real-time pose tracking.


11:43
edl20@edl20: ~
File Edit Tabs Help
0[50.3%] Tasks: 68, 107 thr, 133 kthr; 2 runnin
1[55.3%] Load average: 3.46 1.85 0.76
2[54.7%] Uptime: 00:05:17
3[77.6%]
Mem[843M/3.95G]
Swp[0K/512M]
Main I/O
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
1947 edl20 20 0 2054M 507M 181M S 220.2 12.5 7:02.97 python pose
1963 edl20 20 0 2054M 507M 181M S 45.1 12.5 1:22.35 python pose
1964 edl20 20 0 2054M 507M 181M S 41.8 12.5 1:19.35 python pose
1965 edl20 20 0 2054M 507M 181M S 40.5 12.5 1:17.87 python pose
1006 edl20 20 0 331M 93696 60976 S 8.6 2.3 0:17.73 /usr/bin/Xwa
1959 edl20 20 0 2054M 507M 181M S 4.0 12.5 0:08.22 python pose
880 edl20 20 0 586M 100M 76704 S 2.7 2.5 0:07.14 /usr/bin/lal
1955 edl20 20 0 2054M 507M 181M S 2.7 12.5 0:04.60 python pose
1967 edl20 20 0 2054M 507M 181M S 2.7 12.5 0:04.79 python pose
1968 edl20 20 0 2054M 507M 181M S 2.7 12.5 0:04.80 python pose
1960 edl20 20 0 2054M 507M 181M S 2.0 12.5 0:03.23 python pose
1840 edl20 20 0 538M 47536 35248 S 1.3 1.1 0:10.30 lxterminal
2005 edl20 20 0 8112 3584 2560 R 1.3 0.1 0:00.70 htop
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice - F8Nice + F9Kill F10Quit

Communication & Bit-Banging Implementation

- The HX711 and HX710a ADC modules were evaluated for load cell data acquisition.
- On an oscilloscope, the generated pulse period was 8-10 µs, well within the required range (min: 0.2 µs, max: 50 µs).
- CPU usage was minimal (1-2%).

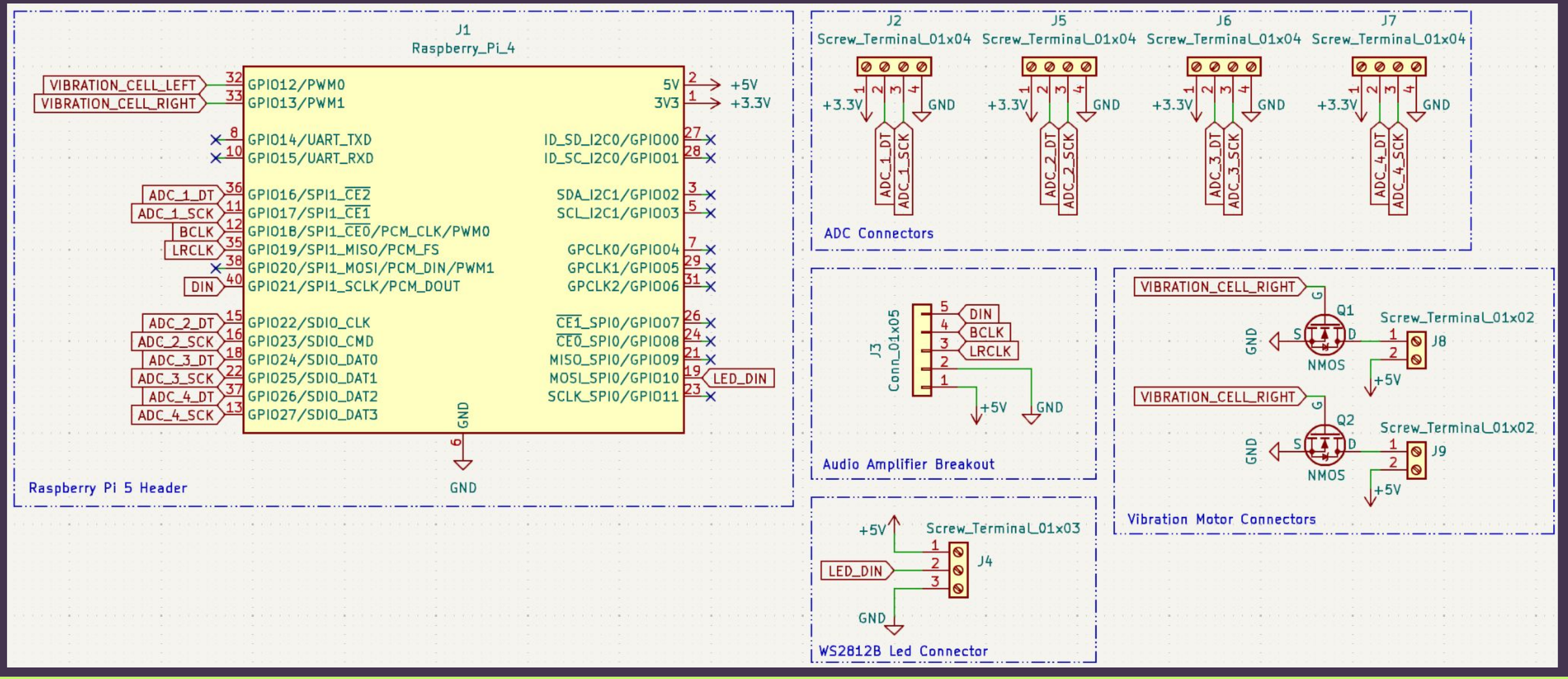
Peripheral Testing & Challenges

- **Vibration Motor**: Yet to be tested, but will use Raspberry Pi's hardware PWM (no CPU overhead).
- **LED Control:** SPI-based control, yet to be tested.
- **Camera Testing:**
  - Noir camera works but has a pinkish tint.
  - The normal camera is not functioning—we suspect hardware issues.
  - Plan: Test the normal camera on a Raspberry Pi 4 in the next lab session.
- **Audio Output Issues:**
  - PWM audio lacks a proper driver.
  - Alternative 1: MAX98357 I^2S audio chip (dedicated breakout board slot in the design).
  - Alternative 2: A screen with built-in audio output.

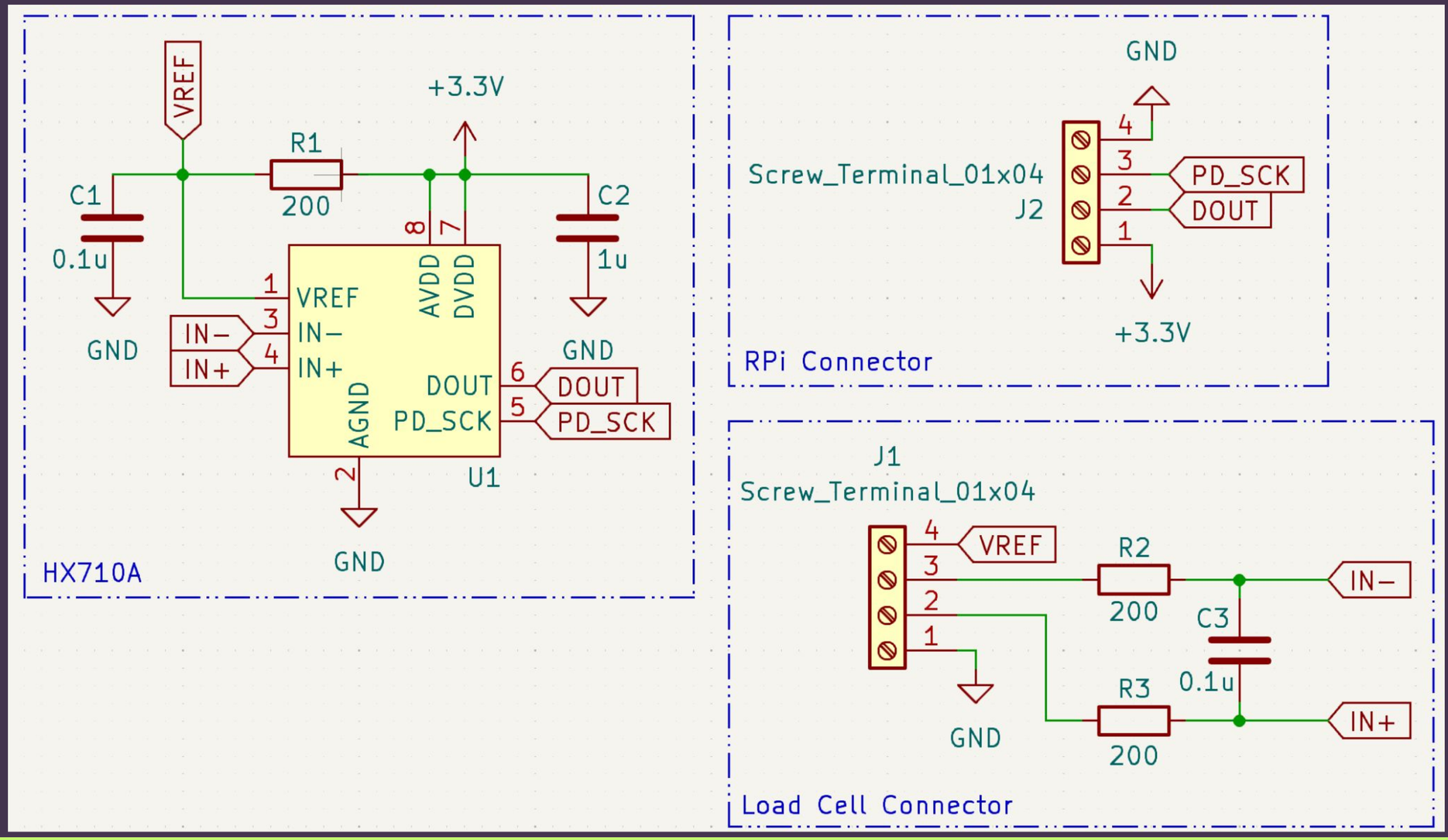
05

Circuit Diagrams

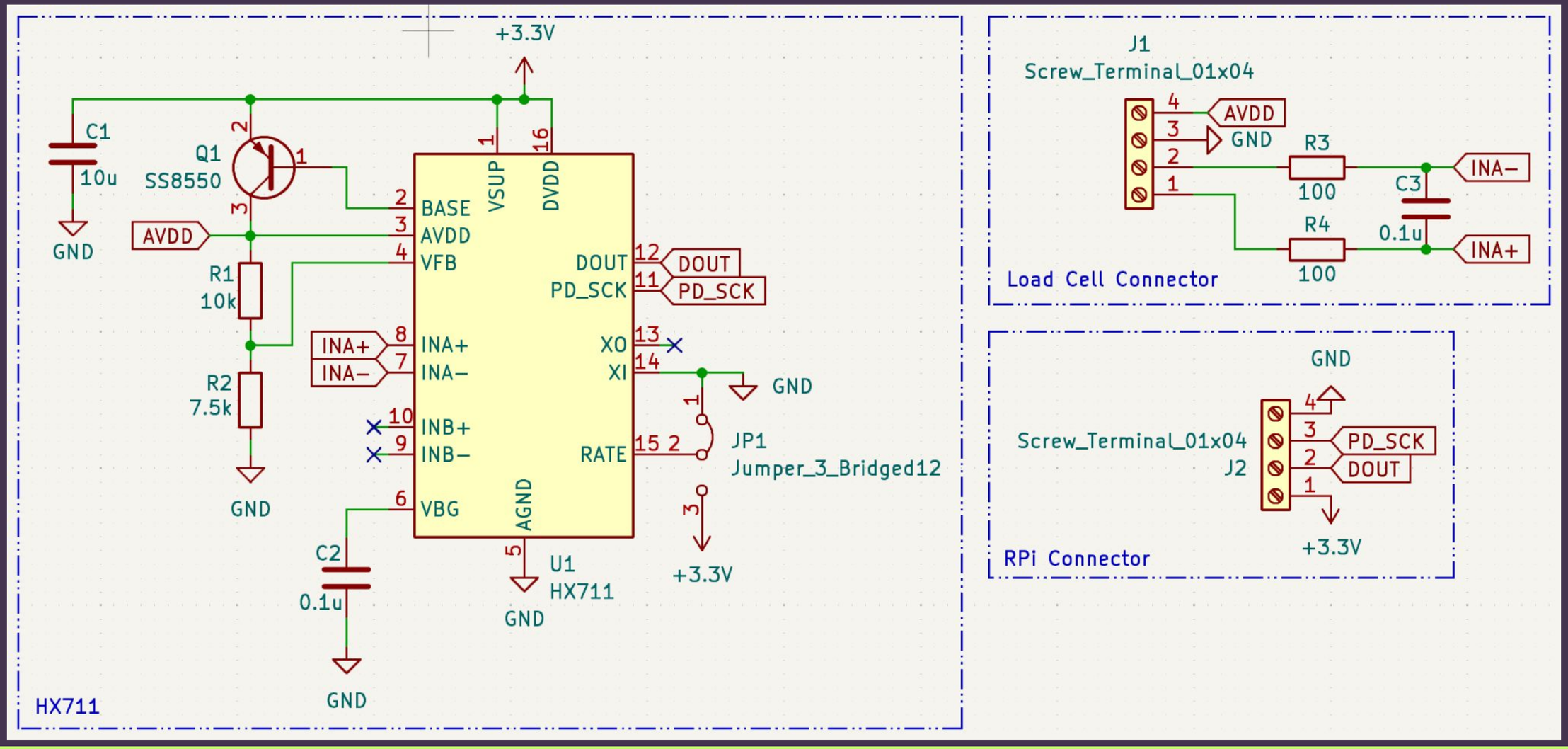
Shield


J1
Raspberry_Pi_4
VIBRATION_CELL_LEFT
VIBRATION_CELL_RIGHT
GPIO12/PWM0
GPIO13/PWM1
GPIO14/UART_TXD
GPIO15/UART_RXD
GPIO16/SPI1_CE2
GPIO17/SPI1_CE1
GPIO18/SPI1_CE0/PCM_CLK/PWM0
GPIO19/SPI1_MISO/PCM_FS
GPIO20/SPI1_MOSI/PCM_DIN/PWM1
GPIO21/SPI1_SCLK/PCM_DOUT
GPIO22/SDIO_CLK
GPIO23/SDIO_CMD
GPIO24/SDIO_DAT0
GPIO25/SDIO_DAT1
GPIO26/SDIO_DAT2
GPIO27/SDIO_DAT3
5V
3V3
ID_SD_I2C0/GPIO00
ID_SC_I2C0/GPIO01
SDA_I2C1/GPIO02
SCL_I2C1/GPIO03
GPCLK0/GPIO04
GPCLK1/GPIO05
GPCLK2/GPIO06
CE1_SPI0/GPIO07
CE0_SPI0/GPIO08
MISO_SPI0/GPIO09
MOSI_SPI0/GPIO10
SCLK_SPI0/GPIO11
LED_DIN
ADC_1_DT
ADC_1_SCK
BCLK
LRCLK
DIN
ADC_2_DT
ADC_2_SCK
ADC_3_DT
ADC_3_SCK
ADC_4_DT
ADC_4_SCK
GND
Raspberry Pi 5 Header
J2
J5
J6
J7
Screw_Terminal_01x04
+3.3V
GND
ADC_1_DT
ADC_1_SCK
ADC_2_DT
ADC_2_SCK
ADC_3_DT
ADC_3_SCK
ADC_4_DT
ADC_4_SCK
ADC Connectors
J3
Conn_01x05
DIN
BCLK
LRCLK
+5V
GND
Audio Amplifier Breakout
VIBRATION_CELL_RIGHT
Q1
Q2
NMOS
Screw_Terminal_01x02
J8
J9
Vibration Motor Connectors
J4
Screw_Terminal_01x03
LED_DIN
WS2812B Led Connector

HX710A Load Sensor



HX711 Load Sensor



06

Test Plan & Results

Preliminary Testing

- Pose detection validated with real-time performance.
- Bit-banging for ADC communication tested successfully.
- Pending tests: vibration motor, LED, load cell integration, normal camera verification.

Intermediate Testing (Upcoming)

- Load cell integration once the component arrives.
- Further ADC performance validation.
- Camera compatibility testing on Raspberry Pi 4.
- LED and vibration motor tests.

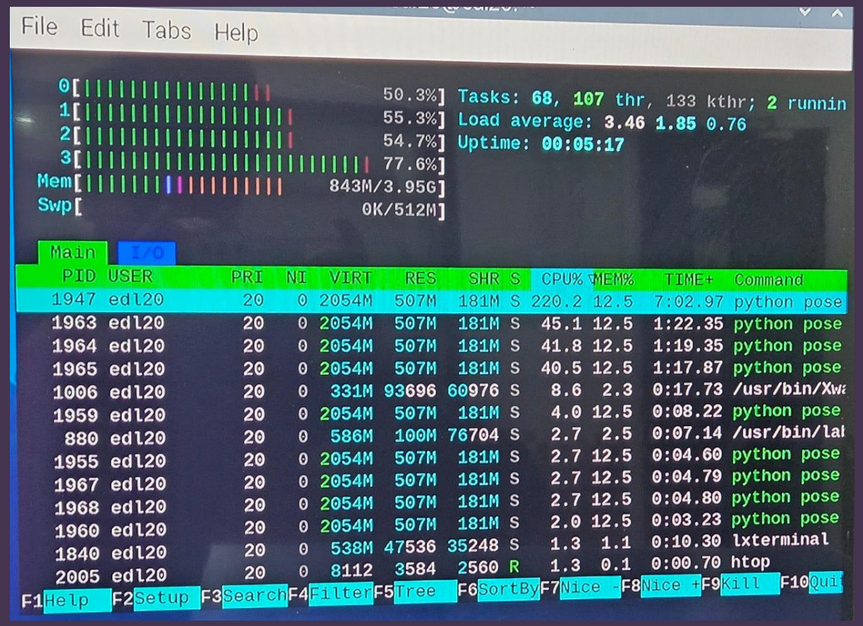
Final Testing & Prototyping Considerations

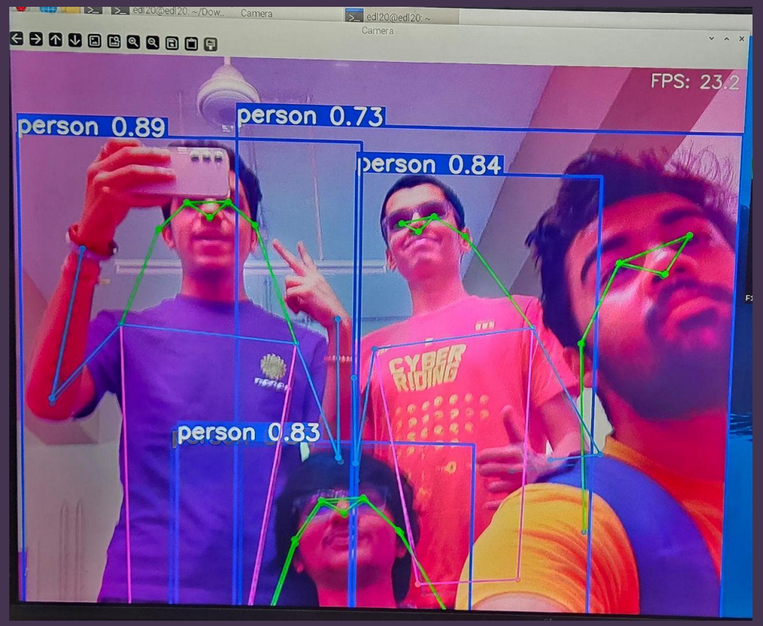
- Assembling the complete system with optimized ADC selection.
- Ensuring real-time data acquisition and minimal CPU load.
- Implementing I^2S audio if PWM remains problematic.

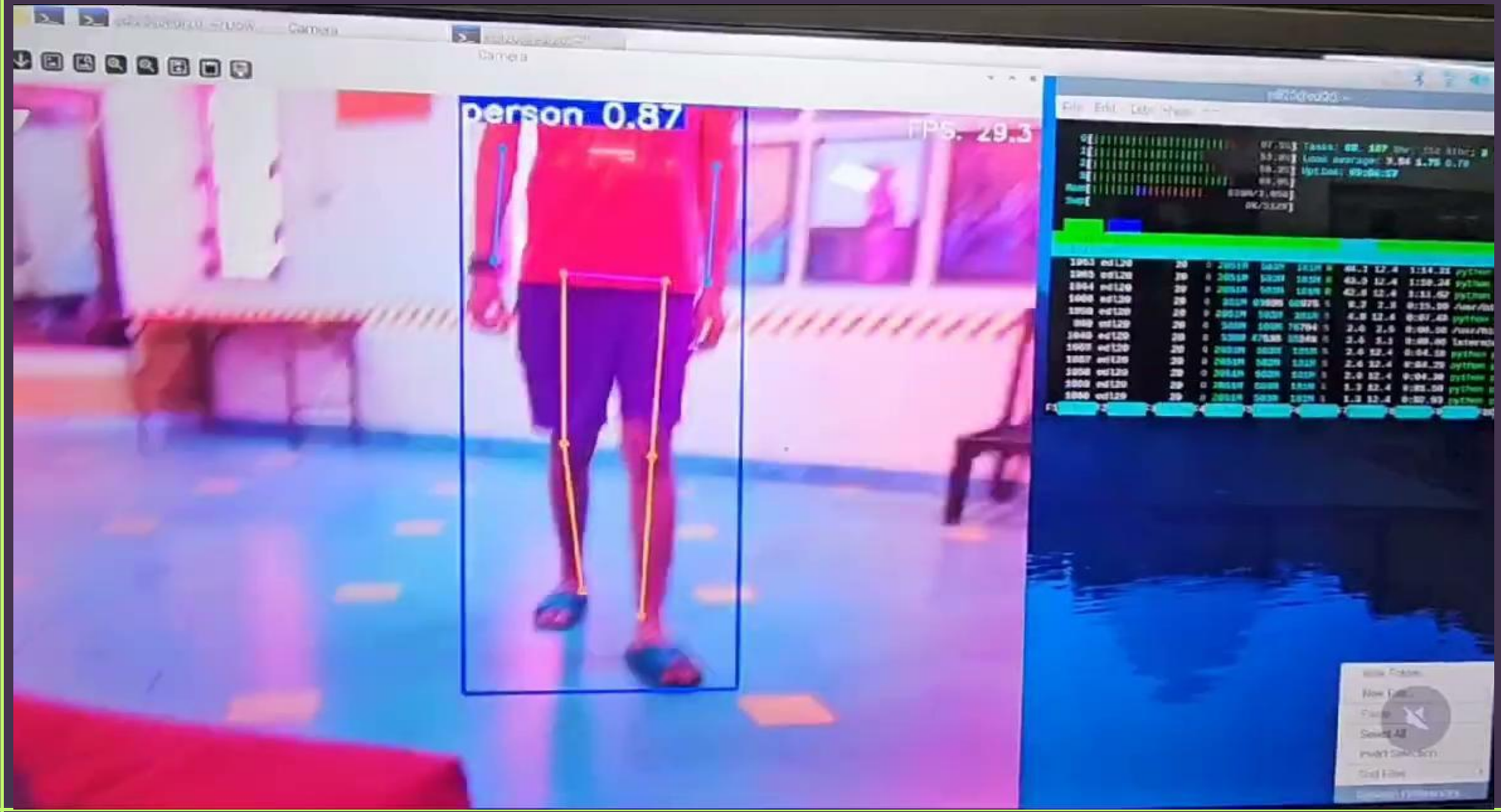
07

Preliminary testing results

- Achieved 20-22 FPS for pose detection at the default clock speed of 2.4 GHz
- uses 50 percent of 3 cores on CPU.
- Boosted performance to 40-45 FPS by overclocking to 2.7GHZ






Camera
person 0.87
FPS: 29.3

08

Thinking Ahead of Milestone

Tue-20

- Initiated CAD Designing, starting with Camera case for stable visual output
- Explored depth estimation using a single camera as an alternative to traditional stereo vision, aiming for a more efficient and cost-effective solution