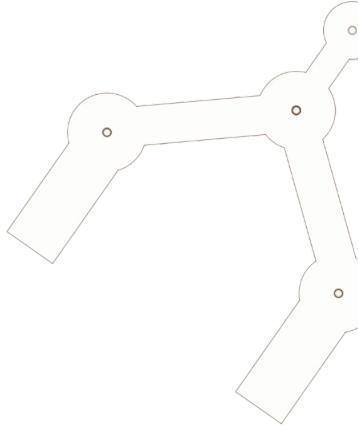


SURG-R  
EE



SURG-R EE



01

## Electronic System

Hardware & Low-level Control

02

## Mechanical Design

Physical System

03

## Kinematics

System Modelling

04

## Motion Control Framework

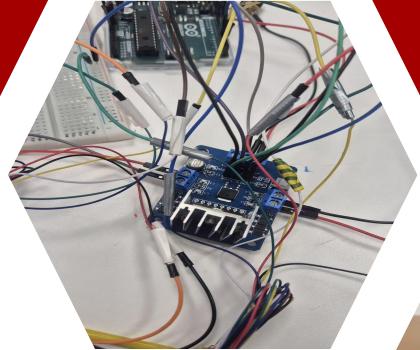
High-level Motion Control

# Dual-Arm Robotic Drawing System



# Hardware & Control Platform

Arduino



Motor  
drivers

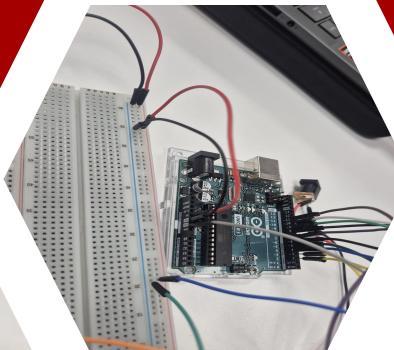


PID-based  
motion control



```
if (positionInDegrees_m2 > 360.0) {  
    positionInDegrees_m2 = 360.0;  
  
} else if (positionInDegrees_m2 < -360.0) {  
    positionInDegrees_m2 = -360.0;  
  
} else {  
    positionInDegrees_m2 = (int)((-360.0 / 360.0) * (PositionInDegrees));  
}  
  
delay(500); // Slow down the print frequency
```

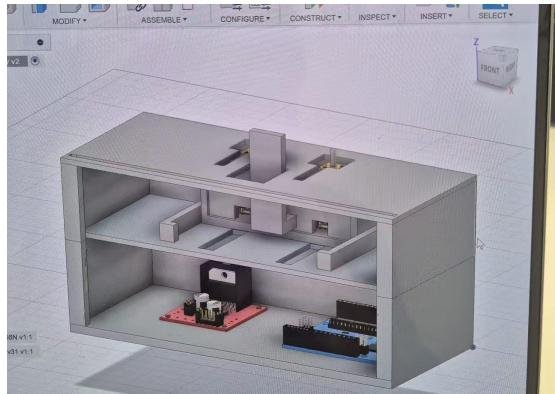
Interrupt Service Routine (ISR) - This function is called every 500ms.  
void updateEncoder() {  
 aState = digitalRead(outputA); // Read current state of channel A  
 bState = digitalRead(outputB); // Read current state of channel B  
  
 if (aState != alastState) {  
 // Check the direction of rotation by comparing the states  
 if (aState != bState) {  
 counter++; // Clockwise rotation  
 } else {  
 counter--;  
 }  
 }  
 alastState = aState;  
}



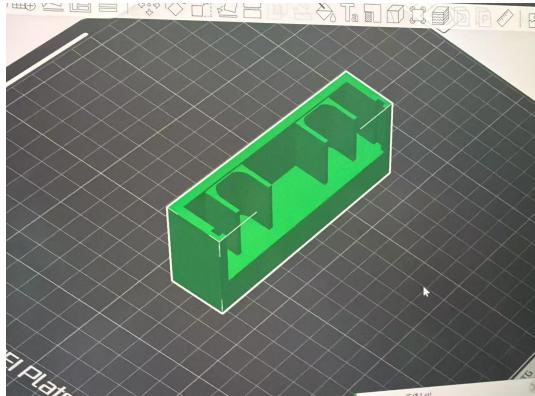
communication  
between PC



# Design and Prototyping Process



Initial arm and  
enclosure  
designed in CAD



Components  
fabricated using  
3D printing and  
assembled



Mechanical issues  
identified during  
assembly and  
testing



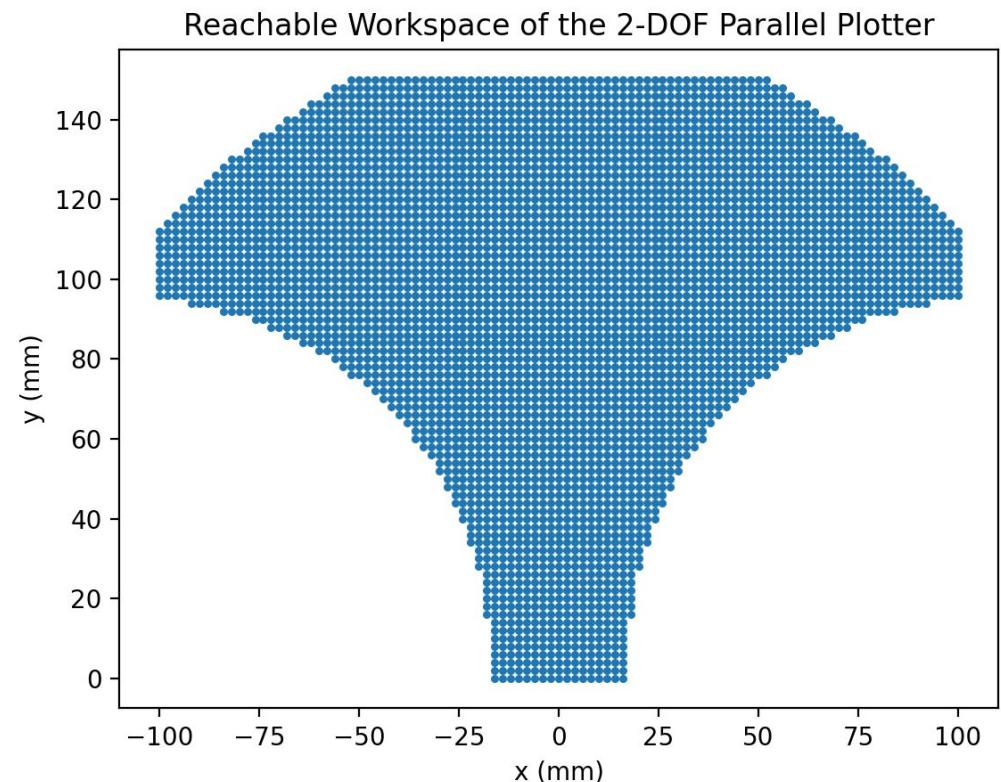
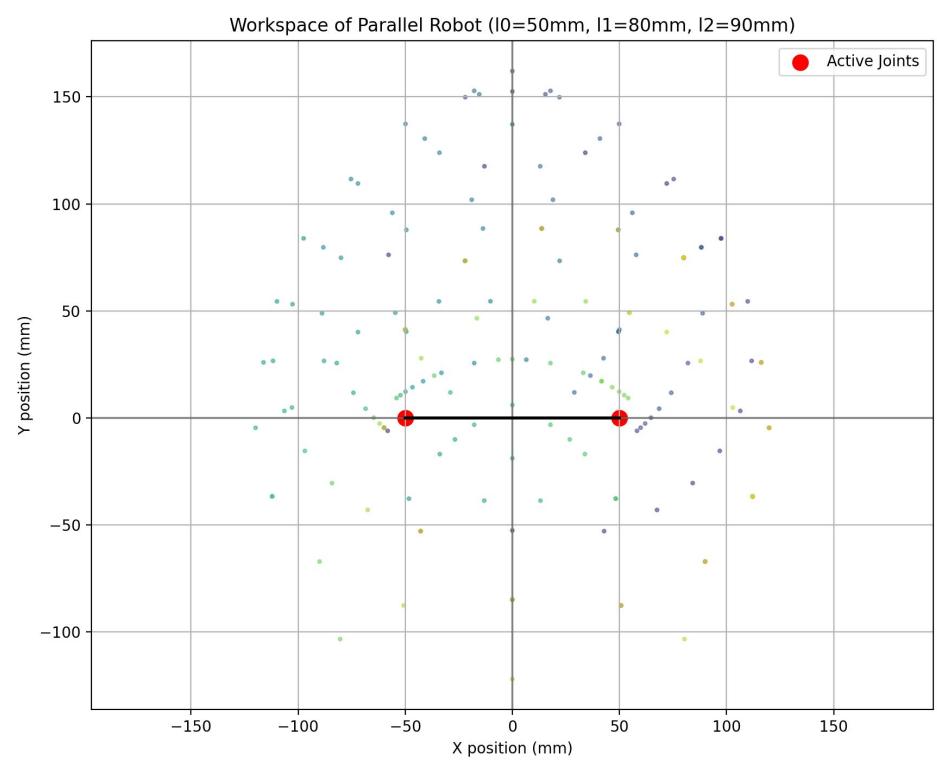
Design iteratively  
refined until stable  
and smooth motion  
achieved

# Kinematics

## Forward/Backward kinematic



## Drawing Area



# ▶ Progressive System Integration

