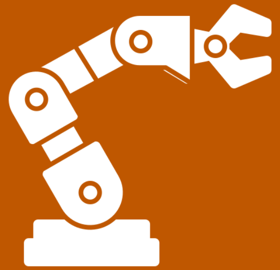
The University of Texas at Austin
Cockrell School of Engineering

April 2025

# Slicer for Multi-Axis 3D-Printing Manipulator

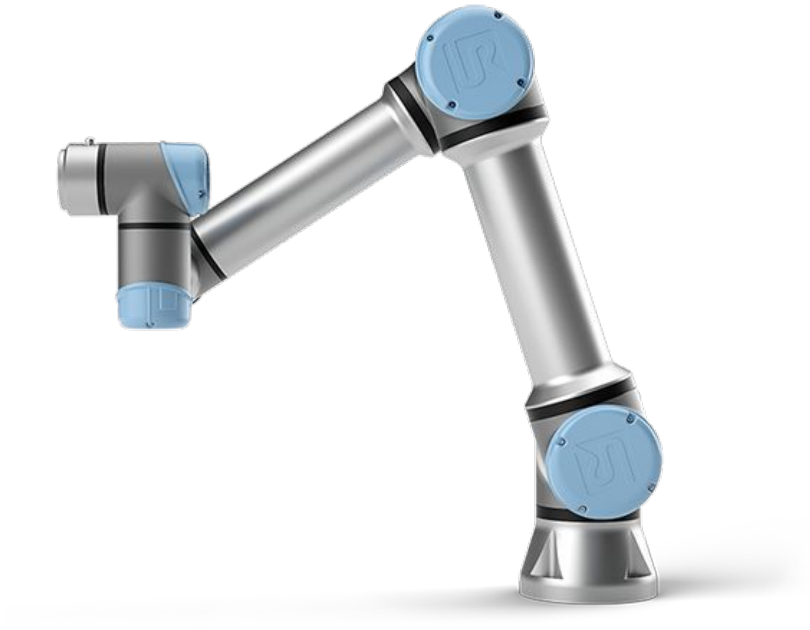
Final Design Presentation
Sponsored by Dr. Zhengui Sha and the SiDi Lab



**Team A: John Lyle, Eddie Flores, Elizabeth Cazes, Aditya Rao**
ME 366K, The University of Texas at Austin
Dr. Chris Rylander

# Agenda

1. Background
2. Project Requirements
3. Design
4. Testing
5. Conclusion



# Problem Statement

Develop an integrated system for a 6-DOF robot arm equipped with an FDM extruder to allow for enhanced cooperative printing

The system should be capable of

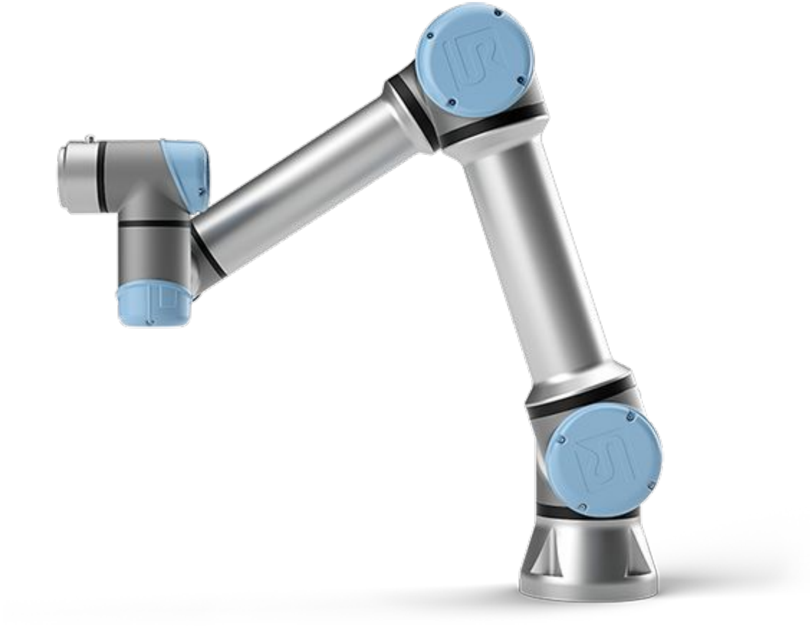
- 3D model slicing
- Robot motion planning
- 3D printing capabilities

***DOF-** Degrees of freedom

***FDM -** Fused deposition modeling, most common 3D printing technique

# Agenda

1. **Background**
   a. *Project Background*
   b. *Current Practices*
   c. *Background Research*
2. **Project Requirements**
3. **Design**
4. **Testing**
5. **Conclusion**



*DOF - Degrees-of-freedom



# Project Background

- Cooperative 3D printing is an emerging necessity for many manufacturing industries
- Has focused on printers with a 3-axis gantry system
- Incorporating multiple of these printers is difficult due to limited flexibility and size constraints
- Part of a multi-team project working with a 6-DOF arm in Dr. Sha's lab, where we collaborated with a FIRE team and other research students

# Current Practices

## 3-Axis Gantry System


MakerBot. "MakerBot Sketch Large 3D Printer." *MakerBot*, https://www.makerbot.com/3d-printers/sketch-large/.

- Difficult to integrate multiple gantry-based systems
- Size constraints
- Cannot print complex geometries

## 3-DOF Robot Arm

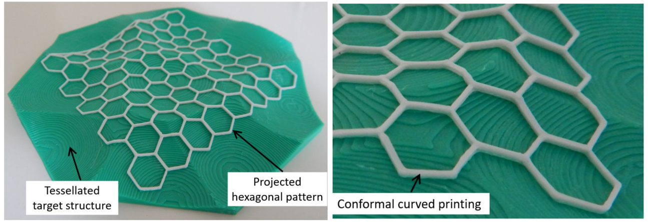

Epson. (n.d.). *Epson Synthis T3 All-in-One SCARA Robots* [Product image]. Epson America, Inc. https://epson.com/For-Work/Robots/SCARA/Epson-Synthis-T3-All-in-One-SCARA-Robots/p/RT3-401SS

- Low flexibility
- Causes collisions
- Size constraints

# Background Research

Conformal 3D printing is where material is deposited along non-planar surfaces, conforming to the geometry of the underlying substrate

- Conformal 3D printing techniques
  - Trajectory Planning
    - Creating a path that conforms to an object's surface, allowing the printing of complex geometries with multiple curvatures
  - Algorithmic biomedical approach
    - Utilizes planar projection by taking a slice in the XY plane and projecting it on a non-planar surface




Rodriguez-Padilla, C., Cuan-Urquizo, E., Roman-Flores, A., Gordillo, J. L., & Vázquez-Hurtado, C. (2021). In *Algorithm for the Conformal 3D Printing on Non-Planar Tessellated Surfaces: Applicability in Patterns and Lattices. Applied Sciences*, 11(16), 7509. https://doi.org/10.3390/app11167509

*STL - Standard Tesselation Language or Stereolithography is 3D file format for 3D printing



# Background Research Cont.

## Slicing Algorithms

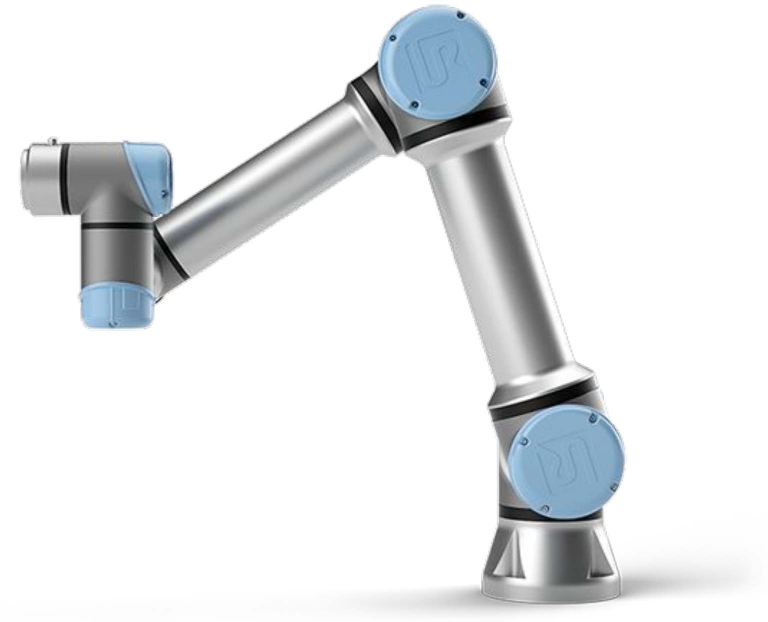
- General methodology
  - Simple software pipeline for slicing algorithm
  - Common slicing practices and current standards

- Intersection Calculations
  - Looked into methodologies to calculate and find the points where the STL model meets each slicing plane

TEXAS ENGINEERING

# Agenda

1. Background
2. **Project Requirements**
   a. *Key Objectives*
   b. *Functionalities*
3. Design
4. Testing
5. Conclusion



# Key Objectives

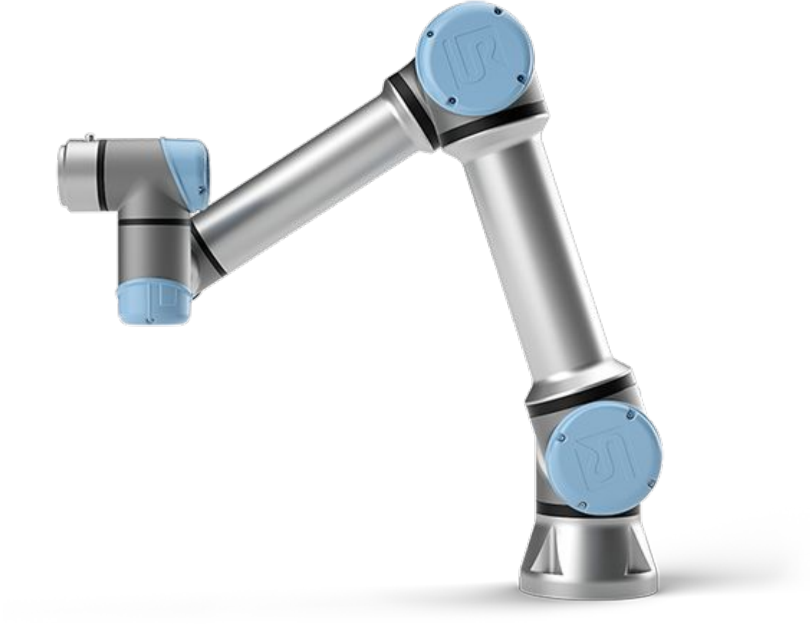
- Take in a standard 3D file format input (.STL)
- Slice the object and output “g-code” for robot arm
- Plan the robot’s motion path
- Integrate the extruder hardware into UR5e system
- Utilize the robot arm and extruder to 3D print
  - Try to print on conformal surfaces

# Functionalities of Design

| Functional Requirements | Need or Want | Means for Meeting Requirement | Testing/Verification |
|---|---|---|---|
| Take in 3D file format input | N | Integrate open source packages into slicing algorithm | Ensure slicing data is correct |
| Slicing of object | N | Slicing algorithm | Visualization + prototype testing |
| Motion planning of robot arm | N | ROS2 system | Simulation + testing |
| Integration of extruder and robot arm | N | Test and run both systems through one interface | Testing and calibration |
| Enable planar FDM printing | N | Integration of all systems | Simulation + prototype testing |
| Enable conformal FDM printing | W | Integration of all systems | Simulation + prototype testing |
| Minimize volume occupied by robot arm | W | Optimize motion planning through ROS2 | Simulation |

# Agenda

1. Background
2. Project Requirements
3. **Design**
   a. *Selected Design*
   b. *Software Architecture*
   c. *Hardware Integration*
   d. *Slicing Algorithm*
   e. *User Interface*
   f. *Motion Planning*
4. Testing
5. Conclusion



*ROS - Node based middleware service commonly used in robotics



# Design Alternatives

## **Robot Software** Design Alternatives:

- ROS1: End of life software support
- **ROS2: Widely used in research**
- Python Scripts: Inefficient for scaled system and integration

## **Slicing Software** Design Alternatives

- Open Source Software: Incorrect outputs for UR5e robot arm
- **Custom Slicer: Autonomy in decision making and output format**

## **Hardware** Design Alternatives:

- DuetBoard: Requires low level motor control
- **Arduino: Direct communication with UR5e**

***ROS2 -** Node based middleware service commonly used in robotics



# Selected Design

## Hardware Integration

- Arduino based custom circuitry
  - Direct communication with UR5e arm
  - Easy for prototyping
  - Allows for control of all hardware seamlessly

## Slicing Software

- Custom Python based slicing algorithm
  - Customized functions, user inputs, and outputs
  - Integrates easily into ROS2 architecture

***RViz2 -** Visualization tool designed for robotics applications

***Moveit 2 -** Open source motion planning software



# Selected Design

## Software Architecture

- ROS2 node based system
  - Easy integration with Python and C++
  - Enables multiple action servers
  - Built-in robot control packages

## Motion Planning

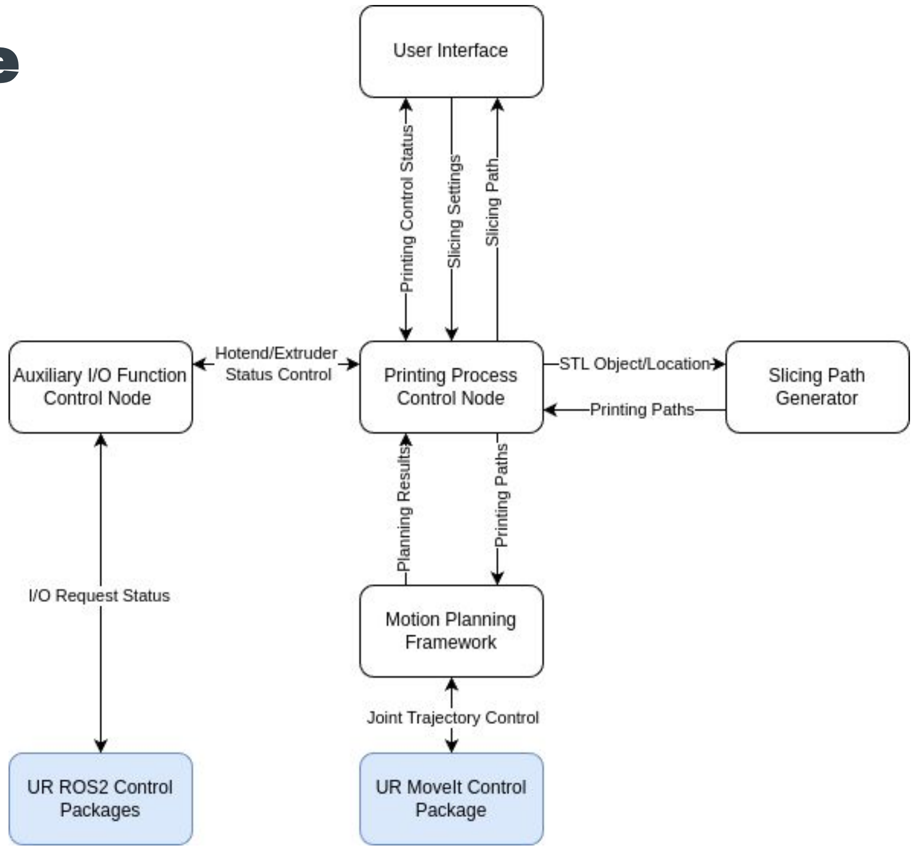
- Custom MoveIt2 Package
  - Built into ROS2
  - Research standard for motion planning

## User Interface

- Custom RViz2 Panel
  - Built into ROS2
  - Allows for changes in user inputs
  - Interactive marker interface

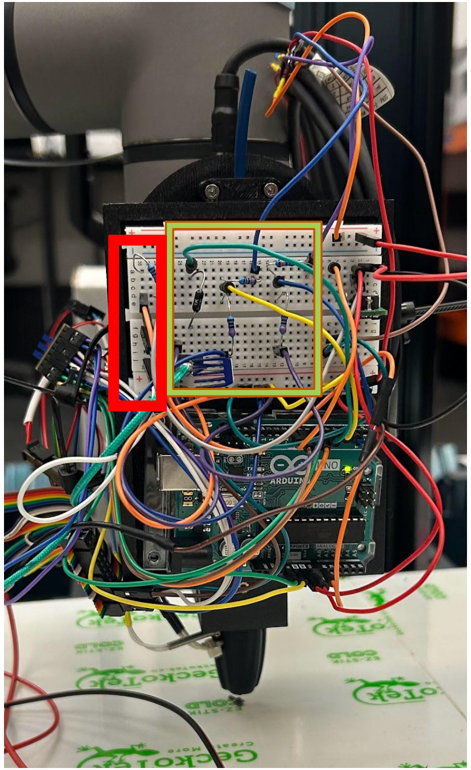
# Software Architecture

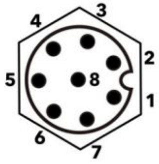
- ROS2 node based system
- 9 custom ROS2 packages
  - 3 MoveIt related packages
  - 3 UI/printing related packages
  - 1 slicing package
  - 1 test utilities package
  - 1 hardware integration package
- Action-server/publisher-subscribers



# Extruder/Hardware Integration

- Controlling On/Off of motor and heating element -
  - Digital Outputs from Tool Head of UR5e are read by Arduino
- Temperature monitoring -
  - Arduino reads thermistor at Tool Head and sends to UR5e
- Two external power supplies
  - Heating element requires more Amps than UR5e can supply
  - Motor driver uses separate supply for better monitoring






| Pin # | Signal | Description |
|---|---|---|
| 1 | AI2 / RS485+ | Analog in 2 or RS485+ |
| 2 | AI3 / RS485- | Analog in 3 or RS485- |
| 3 | TI1 | Digital Inputs 1 |
| 4 | TI0 | Digital Inputs 0 |
| 5 | POWER | 0V/12V/24V |
| 6 | TO1/GND | Digital Outputs 1 or Ground |
| 7 | TO0/PWR | Digital Outputs 0 or 0V/12V/24V |
| 8 | GND | Ground |

# Slicing Algorithm

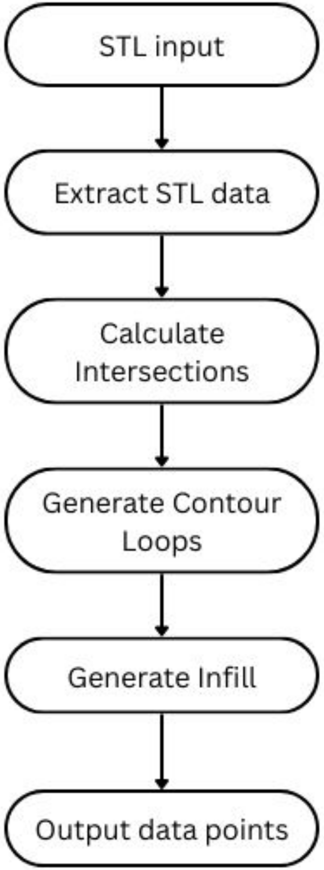
Followed a common, simple slicing algorithm

Resources

- Open source slicing software
- Academic research articles

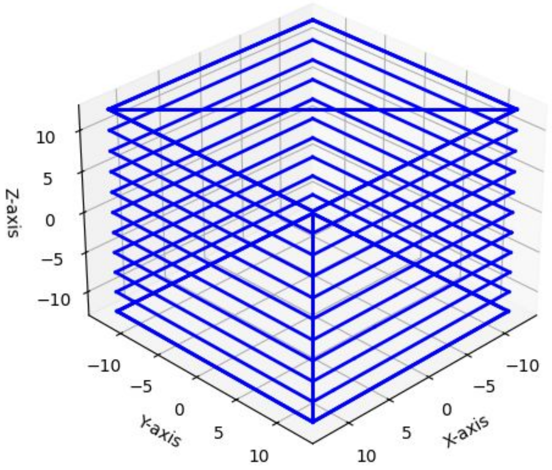
Goals

- Capacity to slice simple geometries
- Include solid infill
- Output as points (x,y,z) for the UR5e robot arm
- Include inputs for user to change



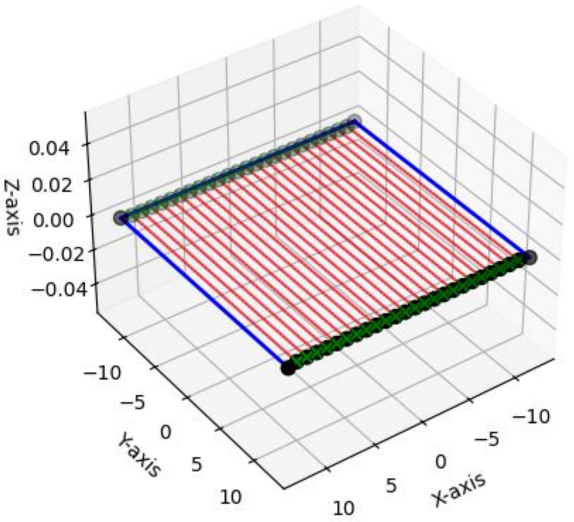
# Slicing Algorithm

## Calculation of Intersections

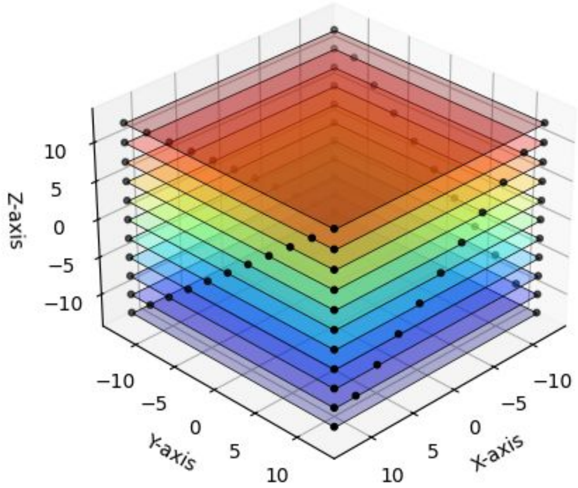



## Contours and Infill for Layer

**z height = 0 mm**




## Closed Contour Loops



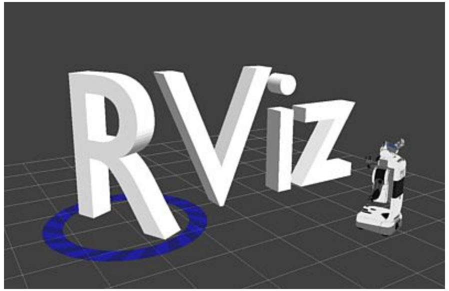
# User Interface

Interface panel for the user to set/alter settings and visualize/move the 3D object

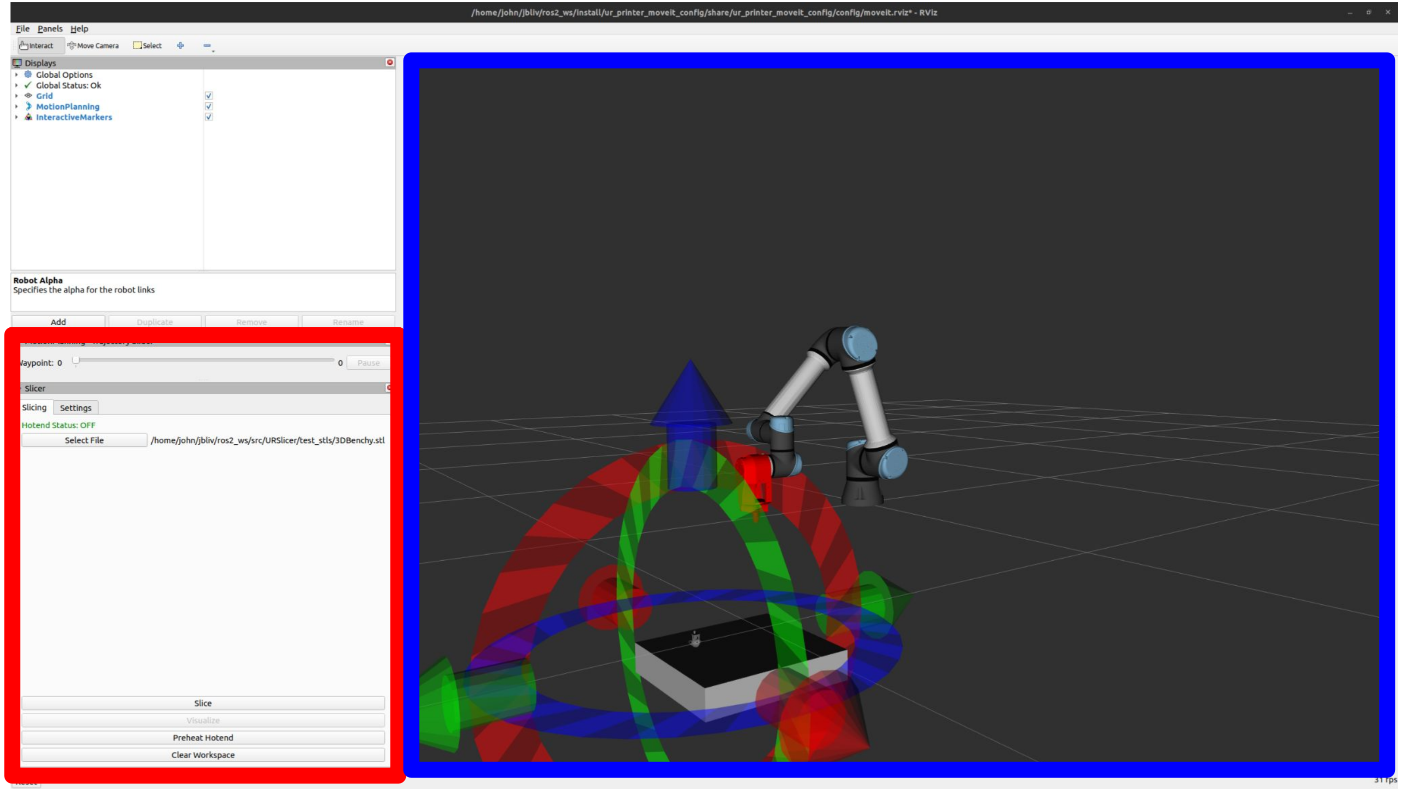
Features and Capabilities:

- Standard slicing settings
- Interactive object placement
- Robot and bed visualization
- Integrated printer control

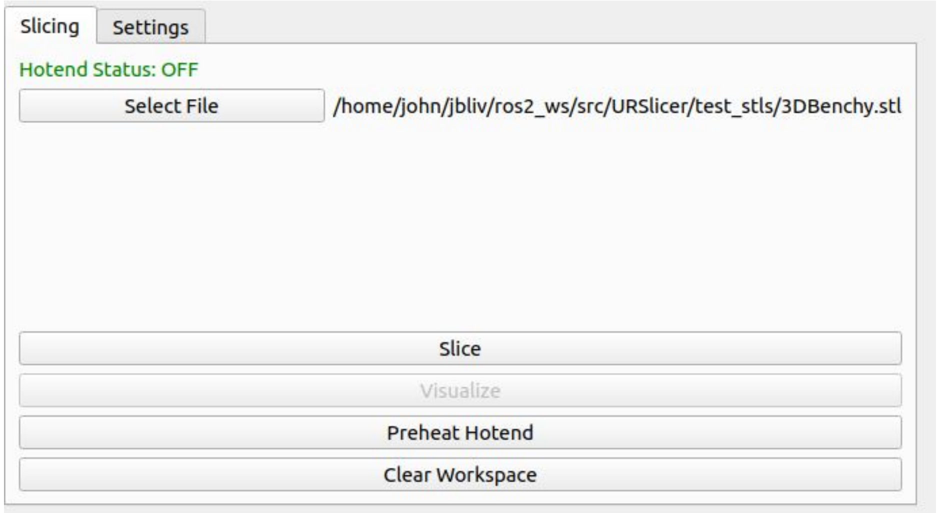
Key Dependencies:



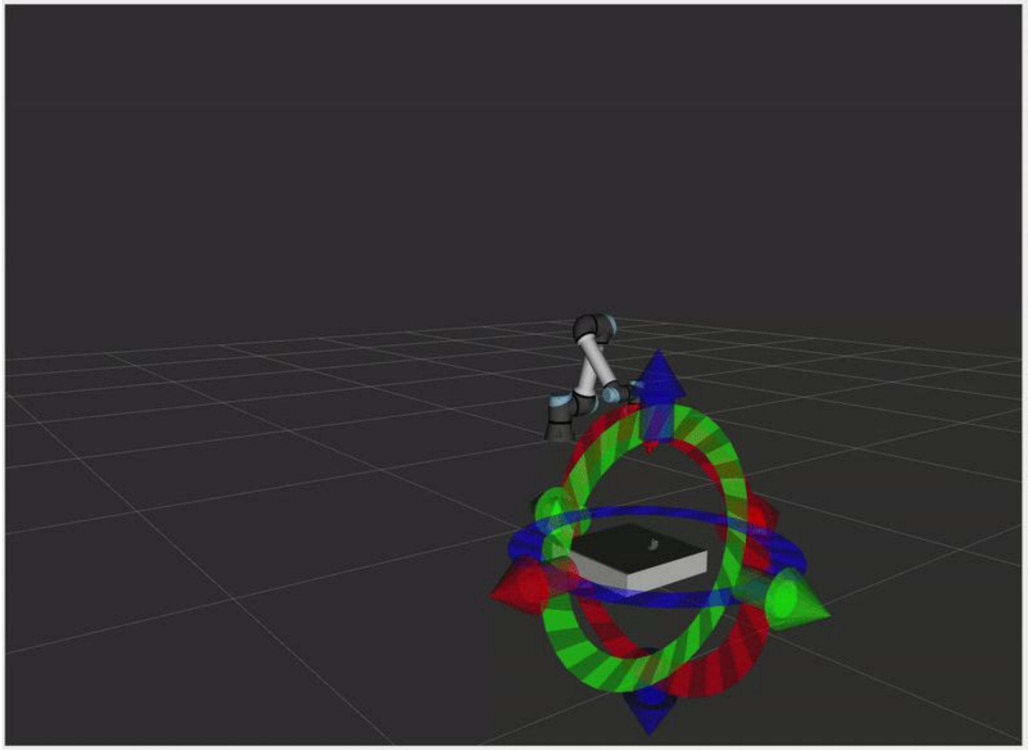

# User Interface



# User Interface




# User Interface



# Motion Planning

Necessary to calculate robot path and joint angles for robot arm to accomplish its trajectory

Features and Capabilities:

- Custom URDF package
- Universal Robotics control package
- Custom MoveIt2 motion planning package
- Action-server for printing

Key Dependencies:

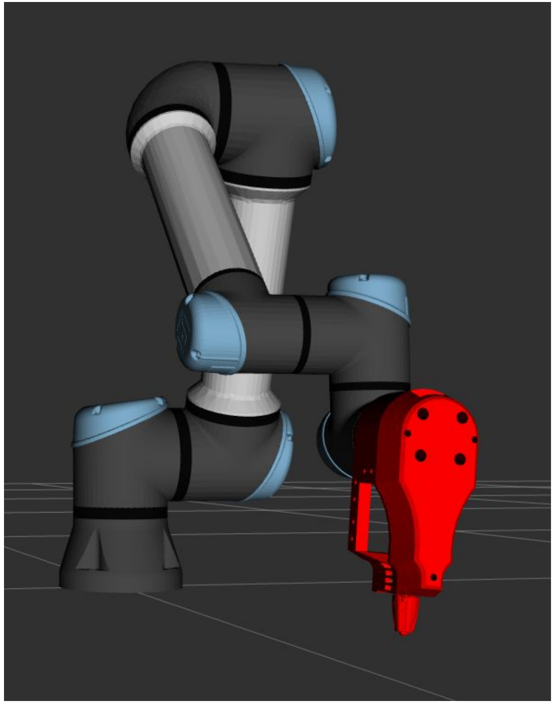
MoveIt2

ROS 2™

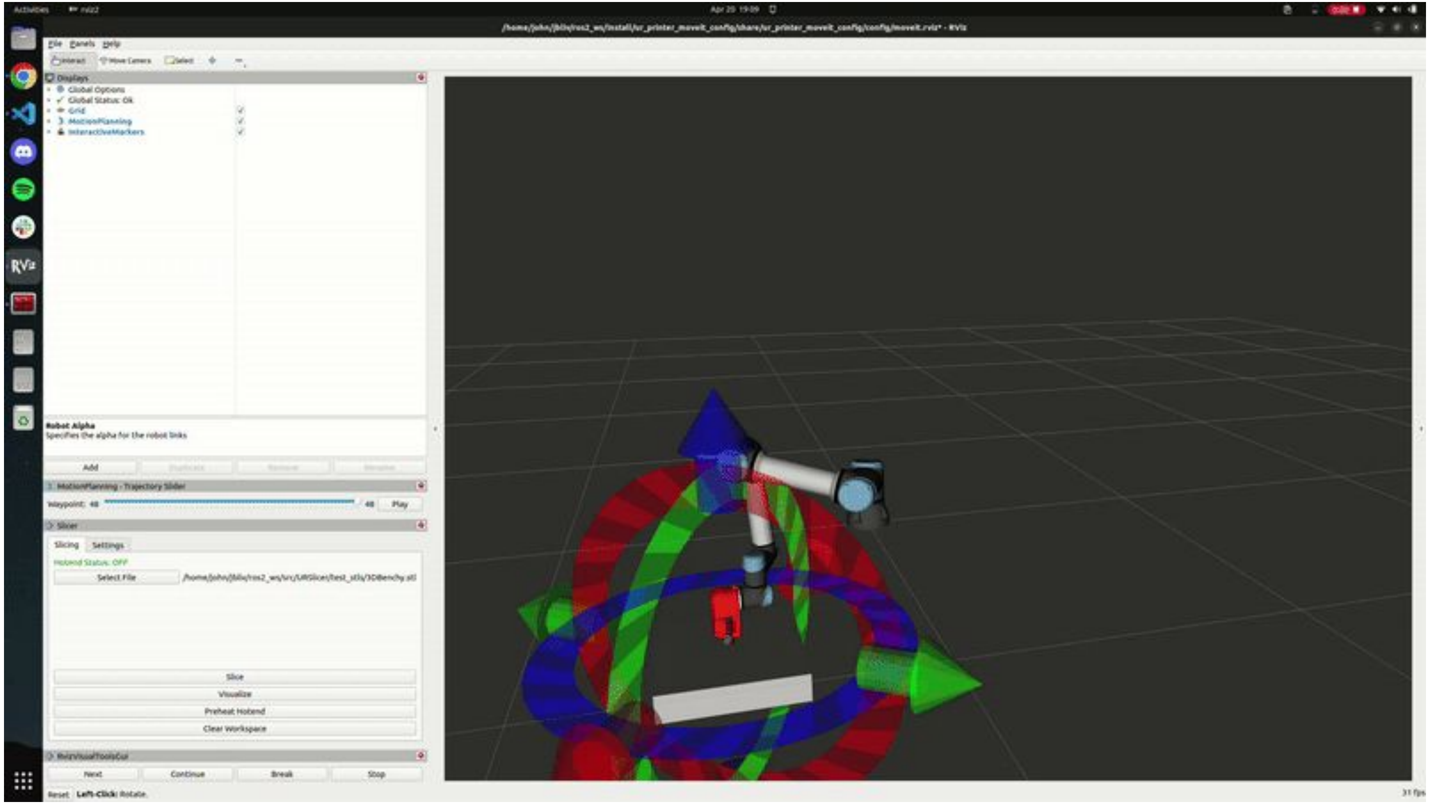
# Motion Planning: Custom URDF

Integrates custom end effector into environment

- Allows for collision aware motion planning

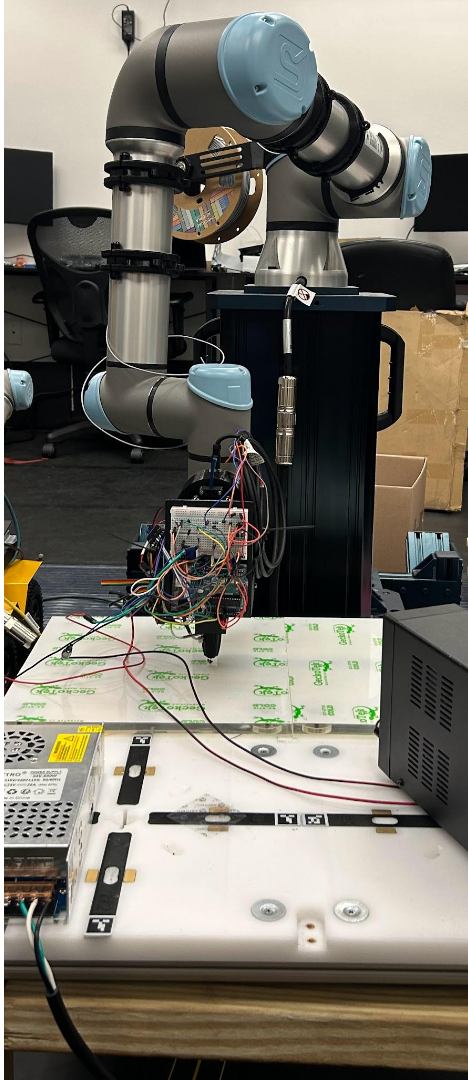


# Motion Planning: Printing Path Planning



# Final Prototype

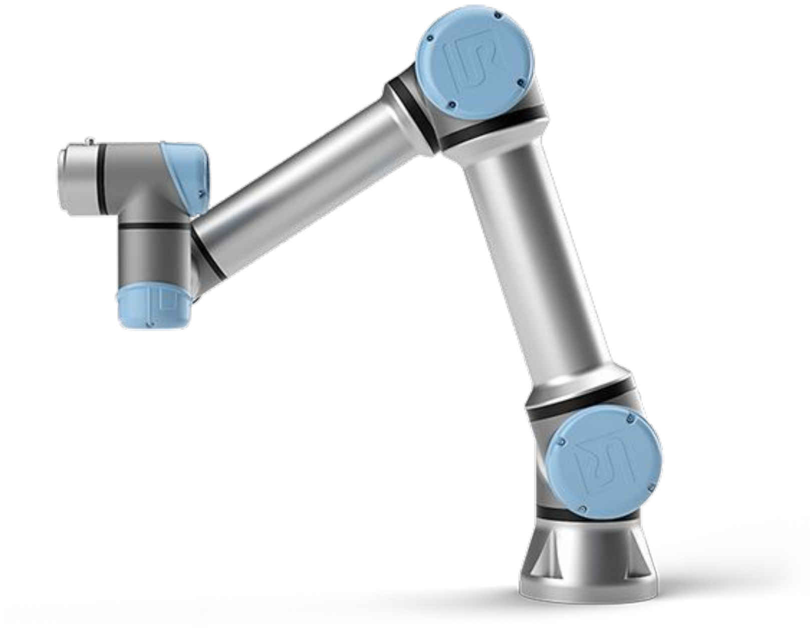
- Arduino based circuitry
- Integrated python slicing algorithm
- ROS2 node based system
    - 9 custom ROS2 packages
- Action-server/publisher-subscriber model



TEXAS ENGINEERING

# Agenda

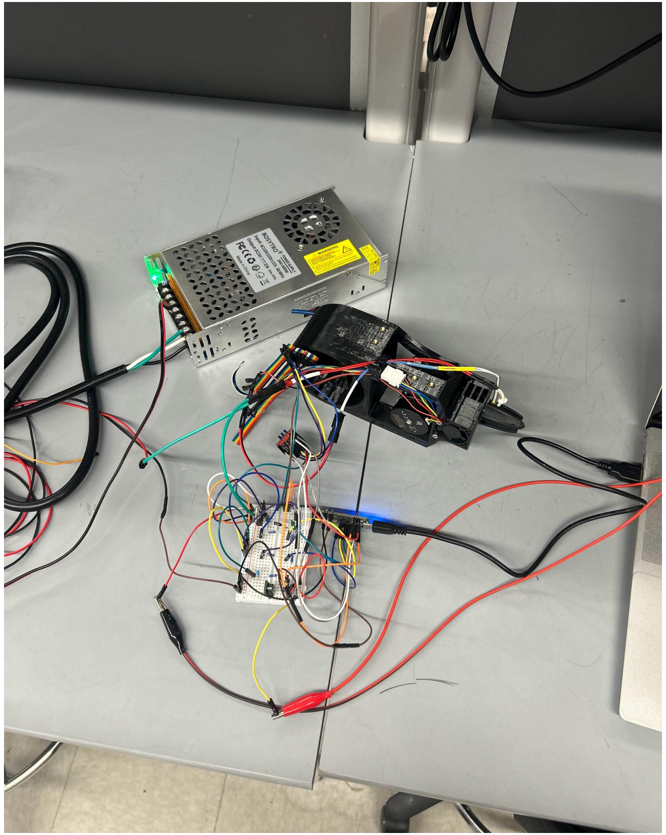
1. Background
2. Project Requirements
3. Design
4. **Testing**
    a. *Hardware/Extruder*
    b. *Slicing Algorithm*
    c. *Robot Simulation*
    d. *Systems Testing*
5. Conclusion



TEXAS ENGINEERING

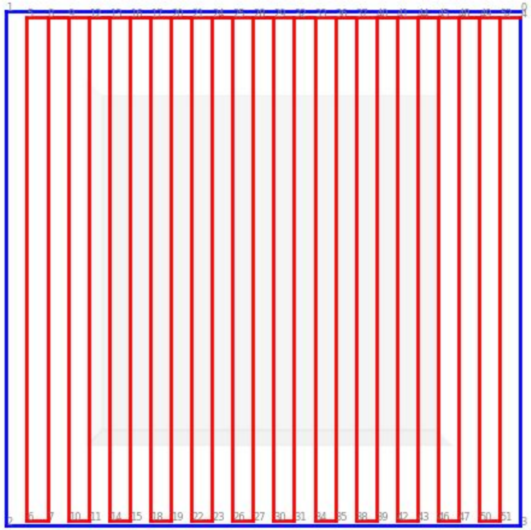
# Isolated Hardware

- Isolated each both systems (Motor & Heating)
  - Checking for proper powering and control
  - Calibrated temperature readings
- Results
  - Upgraded from Arduino Micro to Arduino Uno
  - Motor Driver has its own power supply due to inrush current faulting the UR5e
  - Both systems are fully integrated

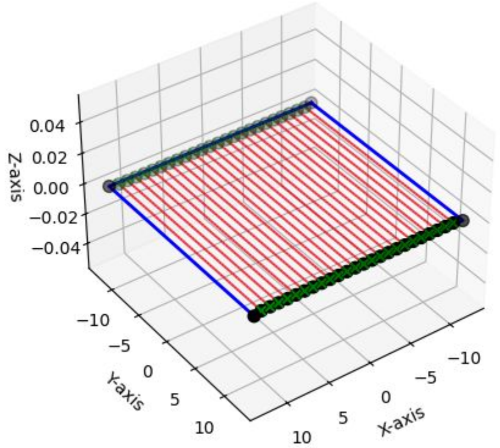


# Slicing Algorithm

- Utilized plotting and visualization for preliminary testing
- Verified data output points for layers with low resolution
- Tested integrated algorithm code with robot arm simulation

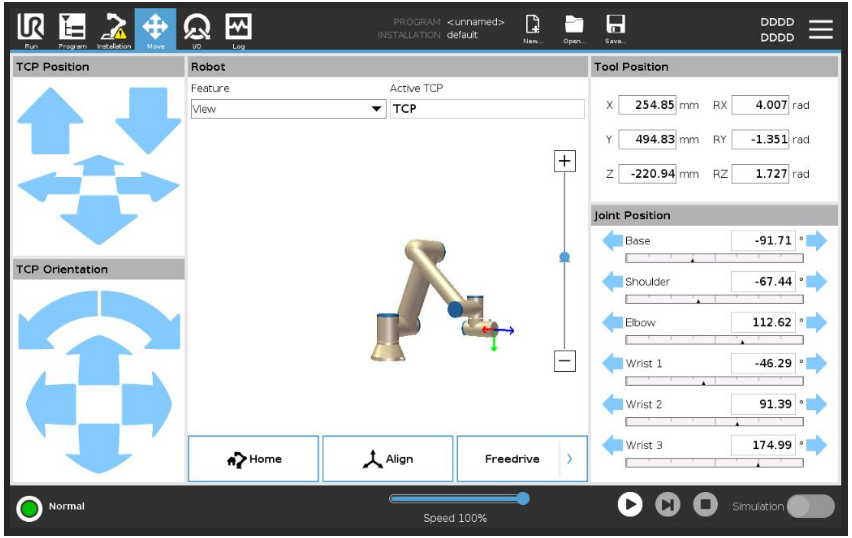


- Results
  - Time of algorithm: *0.44 to 0.73 sec*
  - Works with various inputs: *layer height, infill density, infill angle*
  - Outputs data points for the UR5e robot arm to utilize



# Robot Simulation

- Utilizes Universal Robotics mock arm simulation
- Custom ROS2 Testing Nodes:
  - Bed object creation
  - Mock slicing node
  - Mock hardware integration node



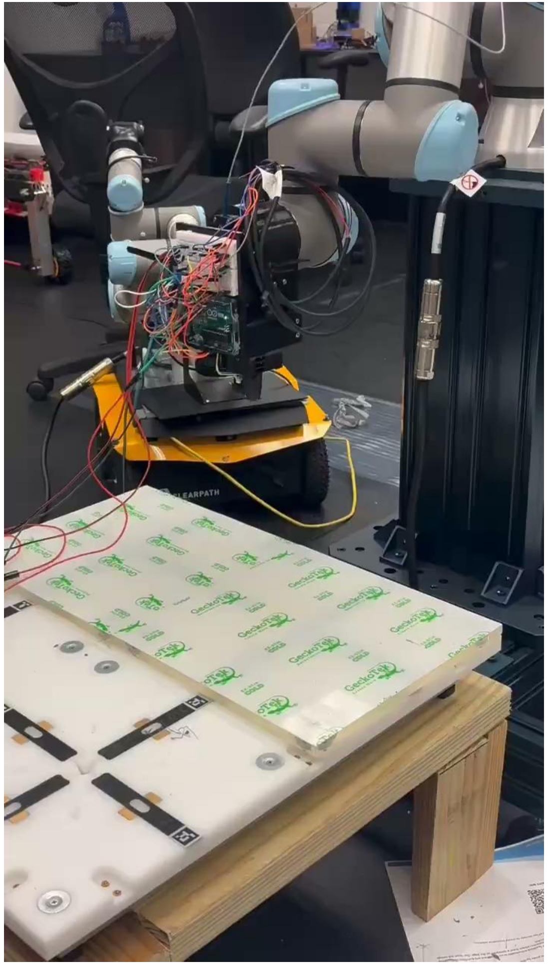
# Systems Testing

Ran a full simulation of the software

- Full integration of the slicing, motion planning, and user interface
- Ran a full software test with the robot to run through layer-by-layer paths

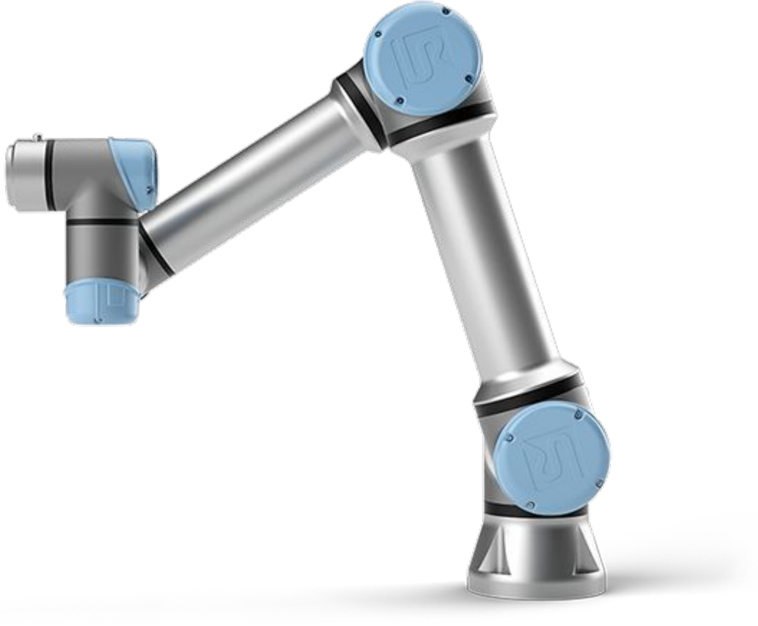
Hardware

- Temperature control is inconsistent
  - Logic communication needs improvement
- Full control of motor



# Agenda

1. Background
2. Project Requirements
3. Design
4. Testing
5. **Conclusion**
    a. *Results*
    b. *Future Improvements*
    c. *Challenges*
    d. *Acknowledgements*



# Results

| Functional Requirements | Need or Want | Means for Meeting Requirement | Testing/Verification |
|---|---|---|---|
| **Take in 3D file format input** | **N** | **Integrate open source packages into slicing algorithm** | **Ensure slicing data is correct** |
| **Slicing of object** | **N** | **Slicing algorithm** | **Visualization + prototype testing** |
| **Motion planning of robot arm** | **N** | **ROS2 system** | **Simulation + testing** |
| **Integration of extruder and robot arm** | **N** | **Test and run both systems through one interface** | **Testing and calibration** |
| Enable planar FDM printing | N | Integration of all systems | Simulation + prototype testing |
| Enable conformal FDM printing | W | Integration of all systems | Simulation + prototype testing |
| **Minimize volume occupied by robot arm** | **W** | **Optimize motion planning through ROS2** | **Simulation** |

# Future Improvements

- Create plans for a PCB to organize and reduce size of wiring
- Writing documentation so work can continue
- Testing printer settings to increase the quality of the prints
- ROS2 integration of wider project functionalities
- Improve infill generation to accommodate complex geometries

# Challenges

- Having components run on different voltages created more complicated wiring
- Pose repeatability of the UR5e
- Learning new software languages and packages (C++, MoveIt2)
- Planning and executing a large scale software based project

# THANK YOU.

- Ronnie Stone, PhD Student
- Dr. Zhenghui Sha, Faculty Advisor & Sponsor
- Farzana Tasnim, Teaching Assistant
- Dr. Chris Rylander, Professor

The University of Texas at Austin
Cockrell School of Engineering

# Index A: Wiring

## Controlling on/off of stepper and heating element

UR Cable Pin 5 (24V) → Buck Converter → Arduino 5V —Powering Arduino
UR Cable Pin 6 (TO1) → Voltage Divider → Arduino D4 — DO communicating Turn on/off extruder
UR Cable Pin 7 (TO0)→ Voltage Divider → Arduino D5 — DO communicating Turn on/off Heating Element

## Controlling heating element power

Power Supply + → Heating Element +
MOSFET Drain → Heating Element -
MOSFET Source → Common GND
Arduino D6 → MOSFET Gate

## Motor Driver control

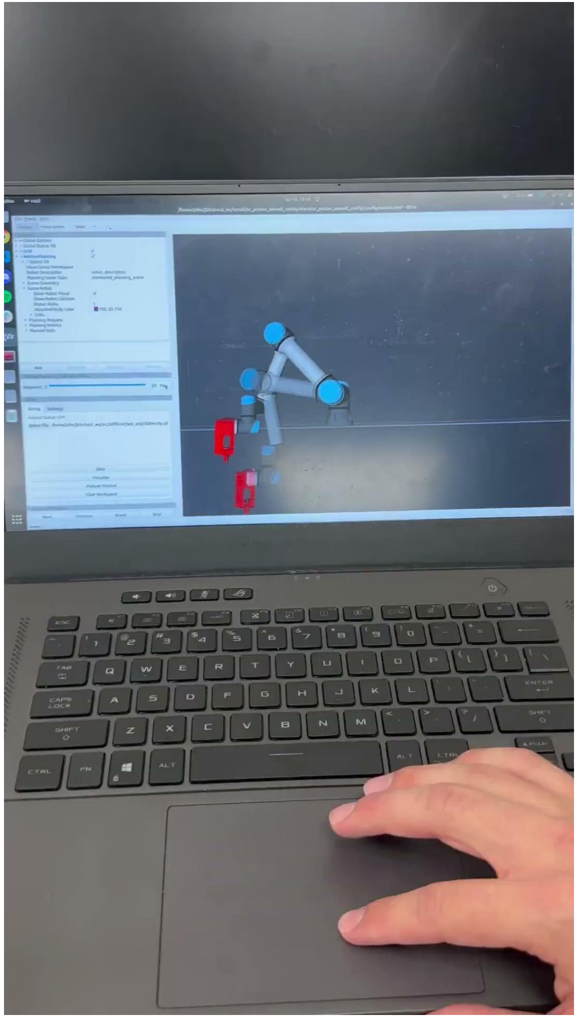
Arduino D8/D9 → Motor Driver DIR/PWM
Arduino D7 → Motor Driver Enable

## Sending temperature reading from Arduino to UR

Arduino A0 ← Thermistor Voltage Divider — Reading temperature
Arduino D10 → 10k Resistor → UR cable pin 2 (AI2) → .1μF capacitor → GND — Sending pwm back to tell what temp is

TEXAS ENGINEERING

# Index B: Media



# Index C: BOM

| Item No. | Component | Specification | # Required Qty | Purpose | Mouser # | Requested Qty | Unit Price | Price w/ Req Qty |
|---|---|---|---|---|---|---|---|---|
| Arduino & Sensors | | | | | | | | |
| 1 | Arduino Micro | 5V logic | 1 | Main controller | 782-A000093 | 1 | $22.10 | $22.10 |
| 2 | 10kΩ Resistor | ¼ W | 1 | Pull-down resistor for thermistor | 603-MFR-25FBF52-10K | 5 | $0.10 | $0.50 |
| UR5e to Arduino Communication (Tool I/O) | | | | | | | | $0.00 |
| 3 | NPN Transistor | 2N2222 or BC547 | 1 | Switching 24V UR5e output to 5V Arduino input | 610-2N2222 | 2 | $3.18 | $6.36 |
| 4 | 1kΩ Resistor | ¼ W | 1 | Current-limiting for transistor base | 279-H81K0FCA | 4 | $0.92 | $3.68 |
| 2 | 10kΩ Resistor | ¼ W | 1 | Pull-down for transistor switching | | | | $0.00 |
| 5 | 24V to 5V Buck Converter | 24V to 5V step-down | 1 | Powering Arduino from UR5e | 580-OKI78SR5/1.5W36C | 2 | $4.97 | $9.94 |
| Setpoint Transmission (UR5e → Arduino) | | | | | | | | $0.00 |
| 2 | 10kΩ Resistor | ¼ W | 2 | Voltage divider for 24V digital signals | | | | $0.00 |
| 6 | 2.2kΩ Resistor | ¼ W | 2 | Voltage divider for 24V digital signals | 603-MFR-25FBF52-2K2 | 3 | $0.10 | $0.30 |
| 7 | Relay Module | 5V relay (single-channel) | 1 | Controls the heating element | 653-G5LE-1-DC5 | 2 | $1.39 | $2.78 |
| Heating Element Control | | | | | | | | $0.00 |
| 8 | MOSFET | IRLZ34N (Logic-level) | 1 | If using DC heater instead of relay | 942-IRLZ34NPBF | 2 | $1.52 | $3.04 |
| 4 | 1kΩ Resistor | ¼ W | 1 | Gate resistor for MOSFET | | | | $0.00 |
| 9 | Flyback Diode | 1N4007 | 1 | Protects MOSFET/relay from voltage spikes | 637-1N4007 | 2 | $0.10 | $0.20 |
| Wiring & Connectors | | | | | | | | $0.00 |
| 10 | Breadboard | 400 tie | 1 | Testing before making it permanent | 426-FIT0096 | 2 | $2.90 | $5.80 |
| 11 | M - M wires | Male to Male wires | 2 | Test Wires for breadboard | 932-MIKROE-513 | 2 | $3.60 | $7.20 |
| 12 | M - F Wires | Male to Femal | 1 | Test Wires for breadboard | 932-MIKROE-512 | 1 | $3.60 | $3.60 |
| Motor Driver | | | | | | | Total w/o Tax & Shipping | $65.50 |
| 13 | Male m8 connector for tooll | Male m8 connector for toolhead | 1 | Male m8 connector for toolhead | 10-04505 | | Mouser Cart | Click |
| 14 | Motor Driver | 2.8A | 1 | Motor Driver | TMC2209 | | | |

# Index D: Gantt Chart

Link to Gantt Chart

# Index E: Github

[Link to Github Repo]