**Project Description**

- Multidisciplinary design of an electrical 3D Tube Bender.
- **Our Electrical Engineering Team:**
  - Decided bending methods and designs
  - Precisely calibrated and tested the motors
  - Constructed the motor housing
  - Created a UI and software for motor communication
  - Designed and implemented component interconnections
- **Past Manufacturing Engineering Team:**
  - Designed and produced aluminum die heads
- **Future Electrical and Manufacturing Engineering Teams:**
  - Design a linear feeding system
  - Implement a permanent power system
  - Design a z-axis control for spirals and helices

**Background Information**

**Hardware:** Rotational Servo Motors and Microcontrollers

**Software:** .NET Framework for Object-Oriented C# programming

**Sponsor Requirements:**

- Bend Type L Quarter Inch Copper Tubing
- Electrical Control System
- User Interface
- Bend 3D Shapes, Spirals, and Helices

**Team Information**

- **Caden Sanders**
  - Discipline: Computer Engineering
  - "I am currently pursuing a career in computer or software engineering."

- **DaNae’ Winston**
  - Discipline: Electrical Engineering
  - "I am currently pursuing a career in electrical engineering, with a concentration in micro/nano devices."

- **Jayce Jones**
  - Discipline: Electrical Computer Engineering
  - "I am currently pursuing a career in computer or hardware engineering with an interest in embedded systems."

**Results**

**Functional Diagram**

- **Overview of Entire Bending System**

  The Jrks send PWM to each motor and receive feedback from a potentiometer connected to that motor. The Maestro sends PWM to each Jrk for simultaneous operation. All Jrks and the Maestro connect to the computer through serial USB for error detection.

**Software**

- **Using WinForms and C#**

  The C# code is integrated with a WinForms UI. Each box represents a class. This diagram covers how the user interface works together with the motors. There are .dll libraries included in the code. These libraries include device communication functions which can be called from our program’s executable.