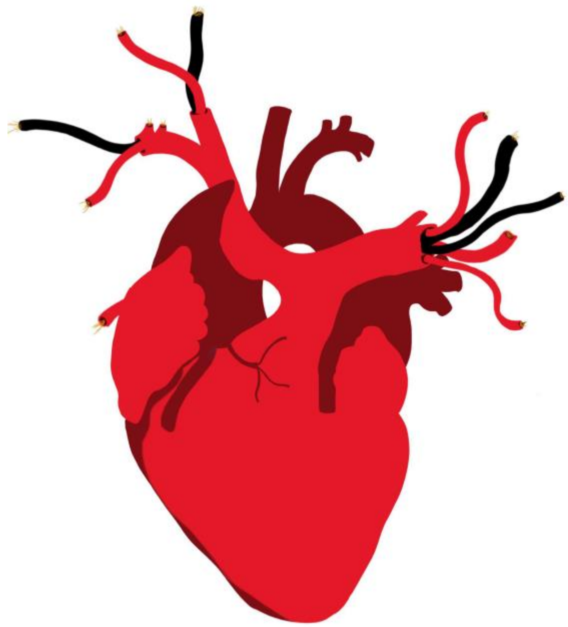
Heart Bytes



Design Document 3

11/11/2022

Eisa Alyaseen - eha33@nau.edu

Alex Anderson - amt646@nau.edu

Abdulrahman Aziz - aaa797@nau.edu

Introduction:

The Heart Bytes team is an electrical engineering capstone team that is working to build a stent crimping machine for W.L. Gore & Associates. The stent crimping machine will consist of several electrical components as well as mechanical systems. Due to this, the Heart Bytes are working alongside a mechanical engineering capstone team to produce the device. The specifications for the device were given to the teams by W.L. Gore & Associates since they were the sponsor for this project. On the electrical side of the stent crimping machine, the team is responsible for getting measurements from sensors that can be related to the diameter and radial force applied to the stent. The team is also responsible for having a motor that will be used to control the size of the iris to actually crimp the stents. The final thing the Heart Bytes team is responsible for is creating a graphical user interface to control the machine.

To give some background information on the use case of our project, a medical stent is a tool that is inserted within patients' blood vessels during surgeries in order to ensure proper blood flow through the vessel. These sorts of medical stents are also used to unclog the arteries of patients with heart problems and high cholesterol patients. Prior to surgery, the stent needs to be crimped to a specific diameter so that it can fit within the patient's blood vessels. Stent crimping machines are devices used to contract medical stents to the desired diameter or until a specified radial force has been exerted on the stent. Stents are metal tubes that are surgically entered into a person's blood vessel to ensure proper blood flow and also to unclog these vessels as spoken before.

In the current market, similar stent crimps do exist in the market. However, the difference between our stent design and the stents that are in the medical field is that those stents are designed with a high budget. In addition, a portion of the stents in the field of medicine is also completely mechanical. The stent that we designed is of a low budget and also is a stent design that operates on both electrical and mechanical parts. As mentioned earlier, the client for this project is W.L. Gore & Associates which is an engineering company that specializes in the use of fluoropolymers in various different applications. One of the larger divisions within the company is the Gore Medical Products Division, which makes use of these fluoropolymers in order to design a variety of medical products.

Engineering Requirements:

A list of the engineering requirements for the project can be seen below. This list was written last semester as a part of a deliverable.

1. General Requirements
   a. The device must have a graphical user interface to display information (GUI)
   b. The device should be able to draw power from the wall for general usage
   c. The device will be able to accept power from a battery
2. Safety Standards Requirements
   a. Warning labels in accordance with OSHA 1915.16
   b. Electric shock warning in accordance with OSHA 1910.137
   c. Emergency stop button in accordance with ISO 13850:2016
   d. Pinch point guard in accordance with OSHA 1910.212 2.
3. Microcontroller Requirements
   a. Connection required for use of a stepper motor
   b. Must be able to run a graphical user interface
   c. A set of seven-segment displays to display the diameter
   d. A set of seven-segment displays to display radial force
   e. Must have memory to store project files
   f. Must be able to supply the power needed to run all of the above components
4. User Input Requirements
   a. There must be two buttons to control the diameter of the stent
   b. One button should be used to increase the diameter
   c. One button should be used to decrease the diameter
   d. There must be two buttons to control the radial force of the stent
   e. One button should be used to increase the radial force
   f. One button should be used to decrease the radial force
   g. There must be an additional button for emergency stops
5. Sensor Requirements
   a. Motor encoder/ Sensor
   b. Must be able to measure distances between 1 to 100 mm
   c. Must be accurate up to 1%
   d. Radial Force Sensor
   e. Must be able to read forces around 132.94 Newtons or 28.9 N/cm
   f. Must be accurate up to 1%
6. Graphical User Interface Requirements
   a. The GUI must be several seven-segment displays, one to display radial force, one for diameter
   b. One set of seven-segment displays should be able to display radial force
   c. One set of seven-segment displays should be able to display diameter
7. Motor Requirements
   a. A functioning stepper motor
   b. Stepper Motor control
   c. It will be used to change the diameter of the aperture of the stent crimp.

With a month left in the project, the team has fulfilled most of the requirements and has changed some of the requirements with permission from our sponsor at W.L. Gore & Associates. From the general requirements, the team has found a suitable graphical user interface with a touchscreen display and the requirement for the system to be powered from an outlet has been met. The team decided that if the device can run solely off of power from the wall, the machine no longer needed an option for battery power.

The safety requirements have yet to be met, but the team is still working towards meeting these requirements. All of the microcontroller requirements have been met because of the team's choice to use an Arduino Mega microcontroller.

The requirements regarding the inputs and GUI have been mostly reworked. As mentioned above, the team has decided to use a touchscreen display for the GUI. The shift from buttons and seven-segment displays to a touchscreen display has meant that all of those previous requirements do not make sense within the context of the project. The functionality of the new GUI is superior to the GUI that was planned when those requirements were written.

The requirements for the sensors have not all been met. The rotary encoder that is being used for the project, will be able to determine diameters from 1 to 100 mm, but the error will be greater than 1% error. The exact percentage for the error has yet to be calculated. The values listed for radial force were greatly exaggerated, a more realistic range for the force is 0 to 10 N. The sensor selected for radial force can read this entire range and should have an accuracy of 1%.

The team has met all of the requirements regarding the motor.

Design:

The electrical hardware for the stent crimping machine will be constructed of an Arduino Mega microcontroller connected to a set of sensors and a motor. The user inputs and outputs will be handled by a touchscreen display that is also connected to the Arduino. The Arduino board is at the center of the project and the code for the microcontroller is what controls the entire stent crimping machine. A flowchart modeling the Arduino code can be seen in figure 1.

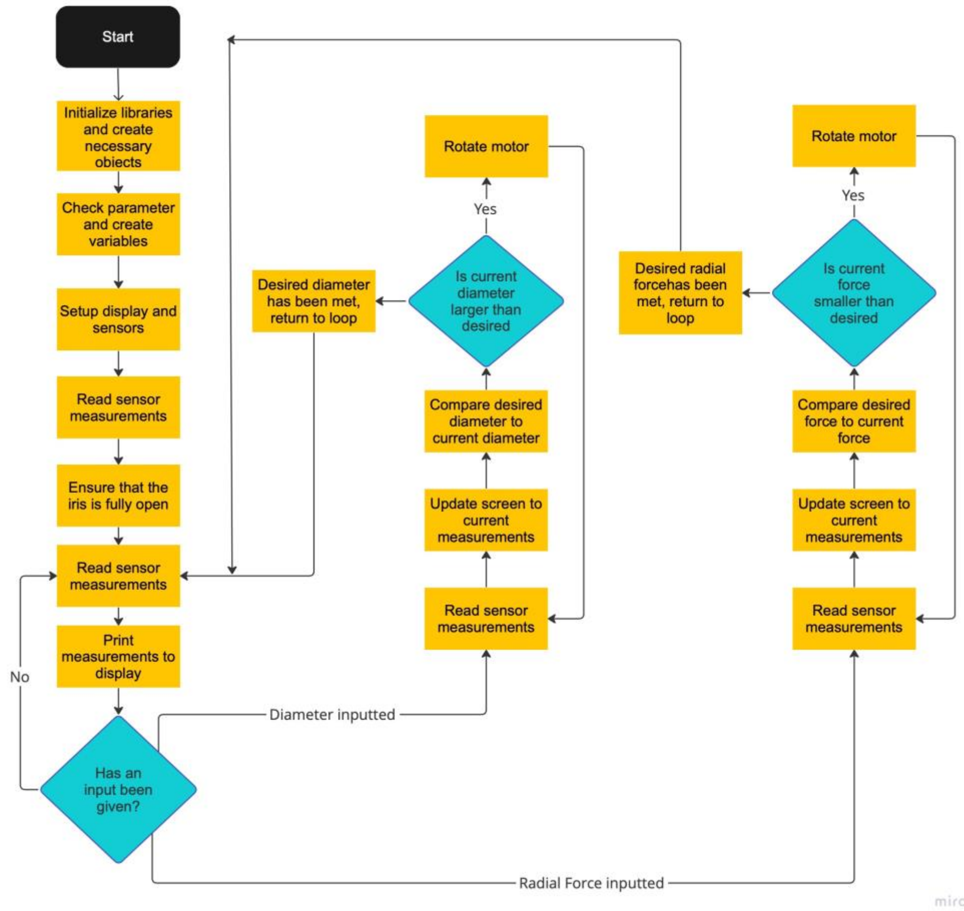


Figure 1: Flowchart Depicting Arduino Processes

The Arduino will start off by going through a setup process shown on the top half of the left-hand side of the flowchart. During this process, the sensors and the motor will be turned on and will be prepared for use. The program's main loop is the bottom three blocks on the left and is the state in which the Arduino will be in while the machine is not actively crimping a stent. When the Arduino receives inputs from the touchscreen display, the Arduino will then be sent to either the set diameter or set radial force feedback loops depending on the inputs chosen. During each iteration of all three of the loops, the Arduino will take measurements from the sensors and print the values to the GUI to ensure the user and the Arduino always has current information on the diameter and force applied to the stent.

The team decided to use a 7” Nextion Intelligent Series display for the graphical user interface. The touchscreen features a built-in microcontroller that processes the touch inputs and the graphics on the display. The touchscreen display utilizes a 4-pin UART connector for power and for receiving and transferring data to the Arduino board. The +5V and ground pins of the UART connection are connected to a board that can input a 5V 1A micro USB cable plugged into an outlet. The RX and TX pins of the UART are connected to the TX1 and RX1 serial ports on the Arduino respectively.

While writing the Arduino code for the touchscreen display, the team struggled with writing methods to receive the UART inputs and process them. While researching the Nextion display, the team found the Easy Nextion Library by Athanasios Seitanis. The library featured all of the functionality needed for this project and ran more efficiently and looked cleaner than the code the team had written. The library is shared using an MIT X11 license and can therefore be used for this project as long as the author is properly credited.

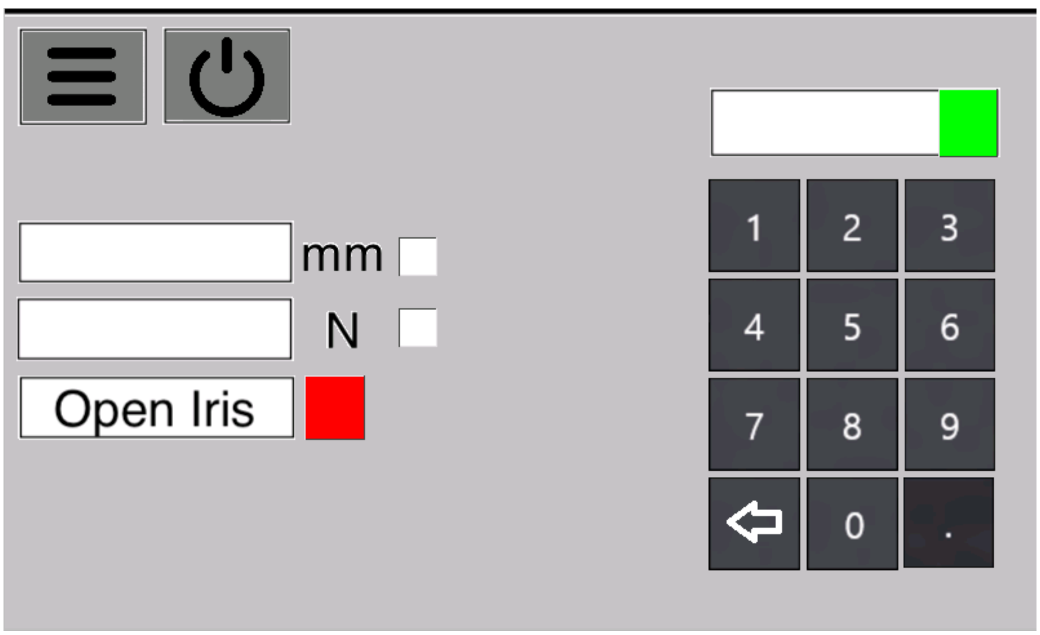


Figure 2: Graphical User Interface

The figure above shows how the screen appears during normal operation. The user is able to select if they want to set the diameter of the stent or the radial force that should be applied to the stent by pressing the checkboxes next to the units. After selecting a mode, the user can use the number pad to input their desired value and press the green button to send the value to the Arduino and begin the crimping process. Input validation will ensure that the user cannot enter a value that cannot be met by the device.

The motor that has been chosen for the project is a NEMA 23 stepper motor that can produce 3NM of torque. The stepper motor will be driven by the DM542T motor driver board connected to an external power supply. The motor will be used to rotate a shaft that is connected to several gears that rotate the iris of the crimping machine. Therefore, the main use of the motor is to control the radius of the aperture of the stent crimper. The NEMA 23 motor is powered by a power source of 24 Volt DC. The motor's movement is controlled by the Arduino and can be set to rotate clockwise or counter-clockwise depending on the desired response. When the motor rotates clockwise, the iris of the crimping machine will increase in diameter. When the motor rotates counter-clockwise, the iris will decrease in diameter. In the past week, the team placed the motor into the stent crimping machine and has begun testing system performance. Initial tests have been successful, and the motor can rotate the crimping mechanism with ease. During use, the shaft of the motor only rotates around 40 degrees.

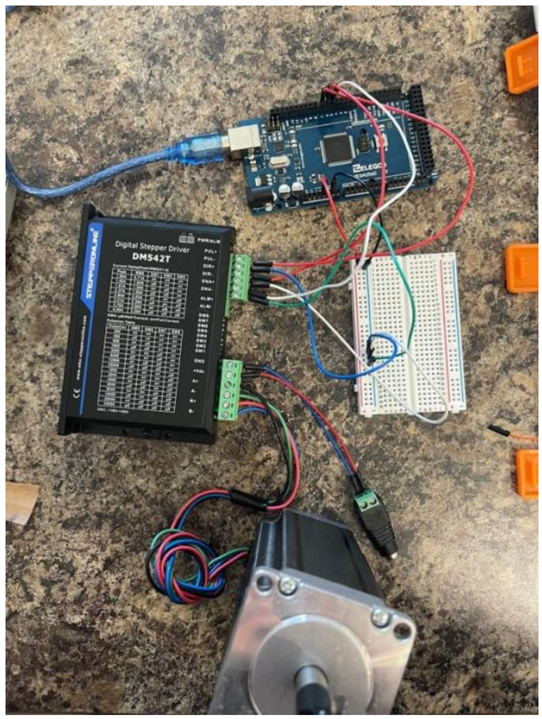


Figure 3: Picture to show the Connections for the Motor

In the above figure, the connections used for the motor can be seen. The A and B connections of the motor are plugged into the correct spots on the driver board. The enable, direction, and pulse connections of the driver board are plugged into 3 digital pins on the Arduino. The power for the driver is supplied by an external power supply using the connector seen in the picture. The NEMA 23 stepper motor is extremely precise in its rotation and it has a 1.8 deg step angle. The rating of the NEMA 23 stepper motor is 4.2 A and has an internal resistance of 0.9 ohms.

To determine the diameter of the stent, the team has chosen to go with a rotary encoder to track the movement of the motor shaft. The measurements of the shaft's movement will be related to the diameter of the stent. The sensor chosen for this project is a Taiss rotary encoder with 600 pulses per revolution. The way it will be used is as the stepper motor rotates, the rotary encoder will translate the rotations of the motor into pulses that can be read by the Arduino. The rotary encoder is a digital sensor that runs off of a 5V pin from the Arduino. The sensor has two digital outputs that are both connected to digital pins on the Arduino.

Figure 4: Rotary Encoder Connections

As mentioned above, the rotary encoder has 600 pulse signals per revolution and the motor shaft will only rotate 40 degrees, which gives the team 66 pulses to determine the diameter. This level of accuracy is less than ideal but is what both the mechanical engineering team and the Heart Bytes have agreed is acceptable.

For determining the radial force that is applied to the stent, the team is using a force-sensitive resistor (FSR) from Pololu. The FSR is 1.5 in^2 and will be placed in between the fins of the crimping mechanism to determine how much force is being applied. The FSR can detect forces from as low as 0.2N up to 20N. This range encompasses the range of forces the stent crimper can apply to the stent, so the team determined it was the best fit for determining the radial force. The FSR is an analog sensor that will be plugged into an analog pin in the Arduino to take measurements. The resistance of the force-sensitive resistor decreases as the pressure on it increases and will have an infinite resistance when no pressure is applied to it. The equation $V_{out} = \frac{R_M * V}{R_M + R_{FSR}}$ will be used to relate the voltage to the force.

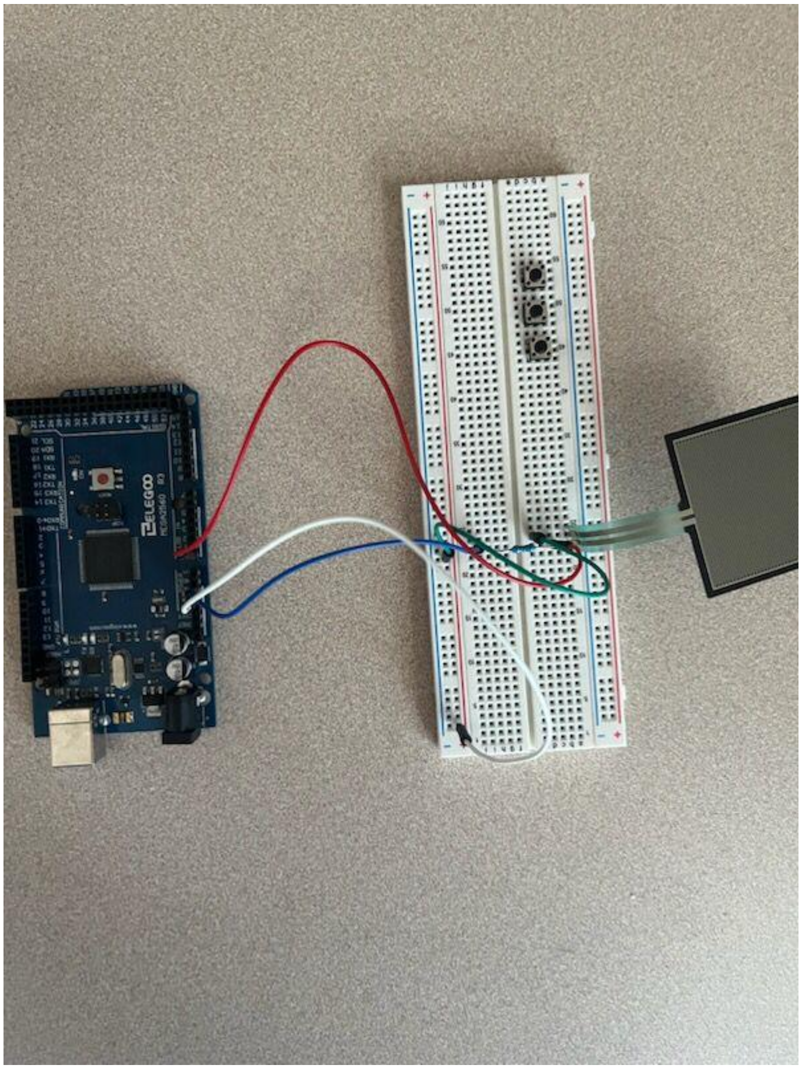


Figure 5: Test Build to Verify the Force-Sensitive Resistor

In figure 5, the connections that are made to get the FSR working can be seen. For testing, the FSR was supplied with a 5V power from the Arduino. The FSR is connected to ground via a 10K ohm pulldown resistor.

The final part of the design for this project is the emergency stop button. The emergency stop button will be placed between the Arduino and the enable pin of the motor driver board. This will allow the user of the machine to stop all mechanical movement within the device.

Plans Moving Forward:

During the final month of work on the project, the Heart Bytes team is working to integrate the electrical components outlined in the design with the crimping mechanism designed by the Mechanical Engineering team. The finalized hardware needs to be completed by November 14th in order to meet the updated 100% completion milestone for the Mechanical Engineering team. Following the integration of components, both teams will work together closely to test and calibrate the stent crimping machine.

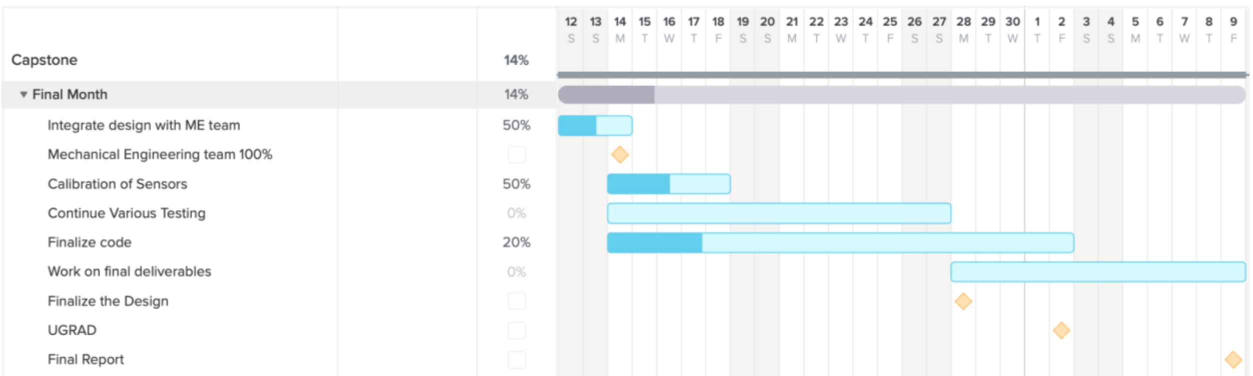


Figure 2: Gantt Chart of the Last Month

The team is currently in the process of creating the final hardware for the stent crimping machine. Currently, the touchscreen display and the motor have been integrated into the device and both work perfectly. Over the next week, the focus will be primarily spent on getting both of the sensors into the final build.

Due to an issue with soldering, the team is currently waiting on a replacement force-sensitive resistor from Amazon. Since the resistor was damaged so late in the testing process, the team has already verified how the sensor is planned to be used. Once the new sensor arrives, the team will be able to solder it and deploy it within the design without any hassle.

The rotary encoder code will need to be rewritten. The method written during the testing of this sensor works properly, but once the method was placed into the full code for the project, errors started occurring. The code for the encoder will be written from the ground up to be as robust as possible to ensure it will work during the next iteration of the full code for the project.

Once these parts are finalized, the Mechanical Engineering team plans to do some rigorous testing methods to ensure the device's functionality meets the requirements. While the ME team is testing the device, the Heart Bytes team will be present in order to alter the code to be in line with the requirements. Once the hardware is finalized, the team will also work to finalize the code. The finalized code will be written to be as efficient as possible and will include proper commenting, proper usage of user-settable parameters, and a menu on the display.

Conclusion:

The Heart Byte team is a team that worked on the project of building a medical stent that operates with both mechanical and electronic features. The stent design of the project had two major aspects which were electrical in nature and mechanical in nature. The mechanical aspects of the stent design were taken care of by the mechanical engineering team working with us. The electrical aspects of the project were handled and completed by the electrical engineering team of Heart Bytes. The electrical system of the stent had three major subsystems. Which were the Motor subsystem, the Diameter sensor subsystem, and the Radial force measurement sensor subsystem. The motor subsystem contained the operation of the stepper motor. The stepper motor is a NEMA 23 stepper motor with a DM542T motor driver. It uses a 24 Volt DC power source as power. The Diameter sensor subsystem contains gear systems and a motor encoder that will act as a diameter sensor to measure the stent aperture diameter. The stent diameter is measured indirectly by the motor encoder by the linearization of the rotation angle of the stepper motor. This is also done with the help of a gear system to which the motor will be attached. The radial force measurement subsystem is the subsystem that enables the measurement of the radial force of the stepper motor as it rotates. At the moment the electrical engineering team and the mechanical engineering team are discussing whether to use a torque transducer or a force-sensitive resistor. Currently, the team is facing issues with the compatibility of the force-sensitive resistor with the mechanical systems that have been already designed by the mechanical engineering team.

Appendix:

Here is our current code for the project:

```
#include "EasyNextionLibrary.h"
#include <ezButton.h>

String userInput = "";
int diameter = 100;
int radForce = 0;

#define CLK 2 //(A/White)
#define DT 3 //(B/Green)
int lastStateCLK;

//const int maxDiameter = ;
//const int minDiameter = ;

const int ena = 5;
const int dir = 6;
const int pul = 7;
const int interval = 350;
bool pulse = LOW;

EasyNex myNex(Serial1);
ezButton limitSwitch(8);
```

```
void printToScreen()
{
  readSensors();
  String diaString = String(diameter);
  String rfString = String(radForce);

  myNex.writeStr("t1.txt", diaString);
  myNex.writeStr("t2.txt", rfString);
}

void readRotEnc()
{
  int currentStateCLK = digitalRead(CLK);

	if (currentStateCLK != lastStateCLK  && currentStateCLK == 1){

		// If the DT state is different than the CLK state then
		// the encoder is rotating CCW so decrement
		if (digitalRead(DT) != currentStateCLK) {
			diameter --;
		} else {
			// Encoder is rotating CW so increment
			diameter ++;
		}
	}
```

```
            // Remember last CLK state
            lastStateCLK = currentStateCLK;

  //delay(50);
}

void readFSR()
{
  int fsrAnalogPin = 0;
  radForce = analogRead(fsrAnalogPin);
}

/*
  readRadForce: Method to read and convert radial force measurements
  Returns variable with usable radial force reading
*/
void readSensors()
{
 readRotEnc();
 readFSR();
}

void rotateCCW()
{
```

```
  digitalWrite(dir, HIGH);
  pulse = !pulse;
  digitalWrite(pul, pulse);
  delayMicroseconds(interval);
}

void rotateCW()
{
  digitalWrite(dir, LOW);
  pulse = !pulse;
  digitalWrite(pul, pulse);
  delayMicroseconds(interval);
}

/*
  setDiameter: Method used to rotate motor until a set diameter has been reached
  Param: setToValue - diameter the stent should be crimped to
  Returns boolean indicating if process has succeeded
*/
void setDiameter(int setToValue)
{
  limitSwitch.loop();
  bool temp = limitSwitch.getState();

  if(temp == 0)
```

```
  {
    digitalWrite(ena, HIGH);
  }

  myNex.NextionListen();
  readSensors();
  printToScreen();

  Serial.print(setToValue);
  if (diameter > setToValue)
  {
    rotateCCW();
    setDiameter(setToValue);
  }
  setup();
}

/*
  setRadForce: Method used to rotate motor until a radial force has been reached
  Param: setToValue - radial force the stent should be crimped to
  Returns boolean indicating if process has succeeded
*/
void setRadForce(int setToValue)
{
  limitSwitch.loop();
```

```
  bool temp = limitSwitch.getState();


  if(temp == 0)
  {
    digitalWrite(ena, HIGH);
  }


  myNex.NextionListen();
  readSensors();
  printToScreen();


  // Measured radial force is less than specified
  if (radForce < setToValue && temp);
  {
    rotateCW();
    setRadForce(setToValue);
  }
}


void setupMotor()
{
  pinMode(ena, OUTPUT);
  pinMode(dir, OUTPUT);
  pinMode(pul, OUTPUT);
  digitalWrite(ena, LOW);
```

```
  digitalWrite(dir, HIGH);
  digitalWrite(pul, HIGH);
}

void setup(){
  pinMode(CLK,INPUT_PULLUP);
        pinMode(DT,INPUT_PULLUP);
  lastStateCLK = digitalRead(CLK);
  setupMotor();
  limitSwitch.setDebounceTime(50);
  myNex.begin(9600);
  Serial.begin(9600);
  //openUp();
}

void loop(){
  limitSwitch.loop();
  printToScreen();
  myNex.NextionListen();
}

void trigger0(){
  String inputStr = myNex.readStr("t0.txt");
  int inputNum = inputStr.toInt();
```

```
  if(myNex.readNumber("c0.val")==1)
  {
    if(inputNum <= 100&&inputNum > 0)
    {
      setDiameter(inputNum);
    }
    else
    {
      myNex.writeStr("t3.txt", "Enter value within range");
    }
  }
  else if(myNex.readNumber("c1.val")==1)
  {
    if(true)//inputNum <= 10 && inputNum > 0)
    {
      setRadForce(inputNum);
    }
    else
    {
      myNex.writeStr("t3.txt", "Enter value within range");
    }
  }
}

void trigger1()
```

```
{
  return;
}
```