# Name & Student ID

Name: Chendan Tang

ID: 20413951

# Title

Development of a Font Drawing System with G-Code Integration for Robotic Arm Control

# Outline of the Problem to be Solved

Summary: The objective of this project is to develop a software system capable of controlling a robotic arm to draw text based on user input, adhering to specific size constraints and utilizing G-Code commands for movement control.

1. **Define function:**

Specifying parameters with meaningful names, indicating their types, and clarifying whether they serve as inputs, outputs, or if their values are modified within the function. Additionally, the expected return values should be documented. This applies exclusively to custom functions created by the user.

1. **Font Data Handling**:

Read and store font data from the provided SingleStrokeFont.txt file, which contains X, Y coordinates and pen up/down data required to draw all ASCII characters. The data should be processed and structured in a way that facilitates efficient retrieval and scaling of characters.

1. **Dynamic Text Scaling**:

Accept user input to set the desired letter height (between 4mm and 10mm) and scale the font's x and y coordinates to ensure the character height aligns with the specified range.

1. **Text File Processing:**

Read text input from a user-specified text file and process each word individually and transmit the data sequentially to the robotic arm.

1. **G-Code Generation and Transmission:**

Generate G-Code commands to control the pen movements of a robotic arm, including raising and lowering actions, and accurately position the arm at specified X, Y coordinates for precise text drawing. Ensure G-Code is transmitted to the Arduino, which in turn relays the commands to the robotic arm for seamless execution and accurate rendering.

# Key Data Items

|  |  |  |
| --- | --- | --- |
| Name | Data type | Rationale |
| Font\_Data | ASCII value | Read from the SingleStrokeFont.txt file |
| User\_Defined\_Height | float | Character height value entered by the user (between 4mm and 10mm) |
| Scale\_Factor | float | To adjust font data to the user-specified height between 4 and 10 mm. |
| XY\_Coordinates | float | The coordinate point used to describe the path that the character draws to |
| GCode\_Commands | string | Commands are generated and passed to the Arduino to control the movement of the robotic arm |
| File\_InputsOutputs | string | The user-provided path to the text file, as well as the text content read from the file |

# Function Declarations

1. *Read and Storing Font Data:*

*int ReadFontData (const char\* filePath, Font\_Data\* fontArray);*

*Parameters:*

*filePath – input string specifying the path to the font data file*

*fontArray – pointer to an array to store parsed font data*

*fontCount – pointer to an integer to return the number of characters read*

*Return value – returns 1 if successful, 0 if failed*

1. *Getting User Input for Character Height:*

*float GetUserDefinedHeight (float minHeight, float maxHeight);*

*Parameters:*

*minHeight – 4mm*

*maxHeight –10mm*

*Return value – returns the user-defined height value*

1. *Scaling X and Y Coordinates:*

*void ScaleXYCoordinates(float inputX, float inputY, float Scale\_Factor, XY\_Coordinates\* outputCoordinates);*

*Parameters:*

*inputX – original X coordinate*

*inputY – original Y coordinate*

*Scale\_Factor – scaling factor based on User\_Defined\_Height*

*outputCoordinates – pointer to XY\_Coordinates to store the scaled values*

1. *Generating G-code Commands:*

*int GenerateGCode(const XY\_Coordinates\* coordinates, int numPoints, char\* gcodeBuffer);*

*Parameters:*

*coordinates – pointer to an array of XY\_Coordinate structures*

*numPoints – number of coordinate points to process*

*gcodeBuffer – pointer to a string buffer to store generated G-Code commands*

*Return value – returns the number of G-Code commands generated, or -1 if an error occurs*

1. *Sending G-Code to Arduino:*

*int SendGCodeToArduino(const char\* GCode\_Commands);*

*Parameters:*

*GCode\_Commands – string containing the G-Code command to be sent*

*Return value – returns 1 if successful, 0 if failed*

1. *Reading Text Input File:*

*int ReadTextFile(const char\* filePath, File\_InputsOutputs\* textBuffer);*

*Parameters:*

*filePath – input string specifying the path to TestData.txt*

*textBuffer – pointer to File\_InputsOutputs to store the text content read*

*Return value – returns 1 if successful, 0 if failed*

# Testing Information

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Test Case | Test Data | Expected Output |
| ReadFontData | Valid file input | "SingleStrokeFont.txt" | Returns 1 and valid fontArray and fontCount |
| GetUserDefinedHeight | User inputs height within range | User input=6mm | Returns 6.0 |
| ScaleXYCoordinates | Valid scaling | Input X=8, Input Y=10, Scale Factor= 0.5 | X=4  Y=5 |
| GenerateGCodeCommands | Generate G-Code | Coordinates = [(4,5), (6,8)] | G-Code string generated correctly |
| *SendGCodeToArduino* | *Send to the Arduino* | *GCode\_Commands* | Returns 1 |
| ReadTextFile | Valid text file input | "inputText.txt" | Returns 1 and fills textBuffer |
| Main () | Complete valid input sequence | Include all the inputs | Successfully read inputs and send G-Code |

# Flowchart(s)

May be included as separate pdf