# Name & Student ID: Angela Rangel, 20361247

# Title: Project Planning

# Outline of the Problem to be Solved

(Maximum 1 page)

The project overall involves creating a software program that enables the writing robot to "draw out" text, based on a given file and text input. This program will read and interpret font data from SingleStrokeFont.txt, scale the text defined by the user input, and generate movement commands in G-code format to then be sent to the Arduino that controls the robotic arm’s movements.

As specified, the software must read the font data from the ‘SingleStrokeFont.txt’ file. After it is read, each character within the font file should be stored in a dynamic memory allocation, to free up the memory space once the robot has finished writing. This also ensures that the software has sufficient memory space and for the program to run at a faster time.

The program should ask the user to input the height of the text ranging from 4mm to 10mm. Create a function that adjusts the X and Y coordinates from the font file by a scale factor of to properly draw out the specified height in real life.

The code should read another text file that is obtained from the user keyboard’s input to draw out the text written within the file. The program should be able to process a file containing text of any length. This can be done by also storing it in a dynamic memory allocation to ensure there’s sufficient memory to run the software and automatically provides the right size array.

For each character within the text file, create a function that translates the font data into G-code commands. This function should include that each new letter that is instructed to be drawn, it must be offset in the X direction so that the letters won’t overlap each other. This is done by the starting position of the next character corresponds to the last position of the previous letter written. Another section to include within this function is to ensure that each text line are drawn with the space of 5mm apart from each other. This is prompted by the LF (ASCII 10) and CR (ASCII 13) codes respectively in the text file.

Finally a function that sends the G-code commands to the Arduino. These commands are then sent to the Arduino to communicate whether to raise or lower the pen and to control the arm’s movements to specific X-Y coordinates. The relevant commands of G-code that is related to the project is shown in Table 1. Once the pen finishes drawing, it should return to the point at origin (0,0) and rest once the pen is in a raised position.

|  |  |
| --- | --- |
| Command | Description |
| F1000 | feed rate, 1000 mm min−1 |
| G0 X Y | Move to the position X,Y |
| G1 X Y | Draw a straight line from the last position to X,Y |
| M3 | Turn on Spindle |
| S0 | Pen up |
| S1000 | Pen down |

**Table 1**: The relevant G-codes needed for the project.

# Key Data Items

|  |  |  |
| --- | --- | --- |
| Name | Data type | Rationale |
| Font Data | Structure | This data item is needed to define how the letter is drawn out by the robot. The font data needs to hold three numerical variables hence why it is a structure. |
| User Input (Text’s Height) | Float | This data item is needed as the user needs to input a height between 4mm to 10mm. This means it is a numerical value that can include decimal values. Hence why it is a float. |
| User Input (Text File) | Strings (char) | Stored as a string as the text file name involves multiple characters that is considered a single item. This also contains ASCII characters hence why it is classified as ‘char.’ This data item is needed to specify which text file the user wants to draw out. |
| Scale factor | Float | This data item is needed to properly draw out the specified height in real life. As it is originally 18 units in the font file. Since the scale factor is a fractional value, the data type needed is a float. |
| G-code commands | Structure | G-code contains ASCII characters hence why it is classified as ‘char.’ This data item is required to provide command to the Arduino and instruct the robotic arm’s movements and positions. |
| **Movement Data:** Coordinates and Pen state (X,Y,P) | Structure | The X-Y coordinates need to be specified |
| **Character Data:** (ASCII Value(C), XYP Structure, Number of required to draw the character (N) | Structure |  |

Extend table as required

# Function Declarations

*Only include functions that you will develop.*

*Example (remove before submission)*

*int TemperatureConversion( int InputTemp, float\* OutputTemp )*

*Parameters:*

*InputTemp – input temperature in degrees C*

*OutputTemp – pointer to return output temperature in degrees F*

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

|  |
| --- |
| **void LoadFontData()**  Parameters: Return Value: 1 if Successful, 0 if failed |
| **void ScaledCoordinates(float ScaledFactor, struct character )** |
| **void GenerateGcode(struct Gcode, Text file)** |
| **void SendtoArduino(char\* command)** |
| **void ProcessText()** |

void ScaleFont ( float Input\_Height, float ScaleFactor, float\* Adjusted\_Height)

Parameters:

Input\_Height –The input height between 4mm to 10mm.

Adjusted\_Height – Pointer to return the height value to scaled so that it draws out the required height in mm.

Return value: Adjusted Height

Void G\_Conversion (Struct FontData[], char\* GCode)

Parameters:

FontData – The font data from the file

GCode – pointer to return output G code command to the Arduino.

Void SendtoArduino (GCode, char Command)

Input: GCode – The G-code command to send to the Arduino

Return value: Command (print out the G code command if successful)

Prints out ‘G Code not found’ if unsuccessful.

# Testing Information

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Test Case | Test Data | Expected Output |
|  |  |  |  |
|  |  |  |  |

*Extend table as required. Note that ‘Function’ includes main()*

# Flowchart(s)

May be included as separate pdf