# Cameron McVeigh - 20465184

# Project Planning

# Outline of the Problem to be Solved

(Maximum 1 page)

The programs’ main function is to draw some text provided from a text file using a robot. To allow this to happen the program firstly needs to load the font data from the text file labelled SingleStrokeFont.txt. This file contains the strokes required to draw each ASCII character in the format of X and Y coordinates for horizontal and vertical strokes and Z for pen up or pen down. This data needs to be read and stored in an appropriate datatype for quick access to translate each read characters to movement commands.

The program should then ask for a user input for text font size between 4 and 10 mm. This value then determines the vertical size of each character. The programme needs to apply a scaling factor to the font data array so that each character to be drawn conforms to the user specified height, whilst maintaining width proportions.

The code shall then read text from the text file one character at a time until it reaches a space character, indicating the end of the word. For each character the program translates the ASCII character into the necessary commands from the SingleStrokeFont.txt file, calculating the X and Y offsets for the next character and storing them in another dynamically allocated data type.

The code shall then calculate the width of the current word by summing the scaled widths of each character and the necessary spacing and then calculate the distance left on the current line (i.e. space until end of the line including 5mm spaces between each word), to ensure the word can fit on the current line without any breaks. If the spacing left on the line is less than the word width, the word needs to be wrapped onto the next line by moving the robot arm back to 0 on the x coordinate and down -5mm on the y Coordinate.

For each character the program translates the X ,Y and Z coordinates into G code and storing it in an appropriate data type, detailing where to move the pen and whether it is raised or down. The G-code commands once generated firstly needs to send initial setup commands in G code then the character movements need to be transmitted to Arduino controlling the robot using rs232 serial port connection.

Only once each word is drawn out should the next word be read and processed. After all the text has been read and drawn out by the robot the robot should move back to the origin point (0,0).

# Key Data Items

|  |  |  |
| --- | --- | --- |
| Name | Data type | Rationale |
| FontData | Array of structs | Stores font data for each ASCII character containing X,Y coordinates and pen state. Can be used to quickly translate characters to movement positions. |
| StrokeCoordinates | Struct of ints  ( x, y, z) | To store the X, Y, and Z coordinates for each stroke. |
| StartIndex | Int | Integer used to hold the value of the first line in the array containing the movement strokes for the current character being read. |
| Gcode | Array of strings | Array of strings used to store the representing Gcode lines to be sent to the Arduino. |
| NumMovements | Int | Integer representing the number of strokes for a character, ie the number of lines to be read. |
| AsciiValue | int | Integer to hold the ascii value of the character being read.. |
| Scalingfactor | Float | To store font scaling factor which may have decimal value. |
| textfile | FILE \* | Pointer to the file containing text data |
| FileSize | Int | Integer to hold number of elements in a file.. |
| NumMovements | Int | Integer used to contain the number of movements for the current character. |
| WordWidth | Float | float used to contain the value of the current. |
| RemainingWidth | float | float used to store a value of remaining line space. |
| CurrentXPosition | Float | Current x coordinate on the line, after each word to manage spacing and wrapping. |
| Character | Char | Character currently being processed for Ascii conversion. Uses less memory. |
| SingleStrokeFont | FILE \* | Pointer to the SingleStrokeFont.txt file used to read and store data for each character. |
| G Code | Array of strings | Array of strings representing G code to be sent to the Arduino. |
| MaxLineWidth | Float | To hold the maximum allowable line width (100mm) to easily compare word widths against remaining line space. |
| MovementsArray | Array of structs | Array of structs used to store movement coordinates for the current word. |
| FontSize | float | Integer to hold the user input value between 4 and 10mm. |

Extend table as required

# Function Declarations

*Only include functions that you will develop.*

*Int CheckIfFileIsOpen ( FILE \*file)*

*Parameters:*

*FILE \*file – Pointer to the file to check*

*Return Value – returns 0 if file is open and reading, 1 if the file cannot open or is NULL*

*Int GetAsciiValue ( char character, int \*asciiValue)*

*Parameter:*

*Char character – The character to be converted*

*Int \*asciiValue – Pointer to store the ASCII value of the character*

*Return Value – returns the ascii value of the character being read*

*int PopulateFontData (FontDataArray \*fontDataArray, char \*SingleStrokeFont, int maxSize)*

*Parameters:*

*FontDataArray \*fontDataArray – Pointer to the Array that will store font data read from*

*SingleStrokeFont.txt*

*Char \*SingleStrokeFont – String representing name of the file to read from*

*Int maxSize – maximum number of FontData strcuts to prevent overflow*

*Return Value – returns 0 if font data was successfully populated, 1 if there was an error*

*Int RetrieveCharacterData (int asciiValue, FontData FontDataArray[], int fontDataSize, FontData outputMovementArray[], int \*numMovements)*

*Parameters:*

*Int asciiValue – ASCII code of the character to retrieve*

*FontData FontDataArray[] – Array containing FontData structs for all characters*

*Int fontDataSize – The total size of fontDataArray*

*FontData outputMovementArray[] – Pointer to array where the retrieved strokes for the character and word will be stored*

*Int \*numMovements – Pointer to an integer where the function will store number of FontData structs to be copied to outputMovementArray*

*ReturnValue – returns 0 if characters data is successfully found and stored, 1 if characters not found*

*Void CreateFactor(int FontSize, float scaleFactor)*

*Parameters:*

*Int FontSize – the user input value between 4-10 for font size in mm*

*Float scaleFactor – Scaling factor needs to be applied to X and Y coordinates of structs.*

*ReturnValue – returns scale factor*

*Float CalculateWordWidth (FontData outputMovementArray[])*

*Parameters:*

*FontData outputMovementArray[] – Pointer to array containing movements for word.*

*ReturnValue – returns the Calculated width of the word in millimetres as a float*

*Float CalculateRemainingWidth ( float Maxlinewidth, FontData outputMovementArray[])*

*Parameters:*

*Float linewidth – total linewidth allowed (100mm)*

*Float currentxPosition – Final x position of previous word*

*ReturnValue – Returns remaining width of line in millimetres as a float*

*void GenerateGCode ( FontData outputMovementArray[], GcodeArray \*gCodeOutput)*

*Parameters:*

*FontData outputMovementArray[] – Array FontData strokes to be converted to G code*

*GcodeArray \*gCodeOutput – An Array where the generated G code commands will be stored*

*Return Value – void*

# Testing Information

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Test Case | Test Data | Expected Output |
| PopulateFontData | Test to ensure that Array is populated with data from SingleStrokeFont text file. | SampleStrokeFont | Array of structs in form x,y,z. Copy of samplestrokefont text file |
| CreateScalingFactor | Ensure scaling factor is only created if user input is between 4-10mm | Using incorrect user input size (eg 11mm) | “The input value must be between 4-10mm” |
| CheckIfFileIsOpen | Invalid file | Empty text file | Return 1 |
| CheckIfFileIsOpen | Valid file | Sample Text file | Returns 0 if successful |
| GetasciiValue | Test to check function can correctly convert Character to ascii value | Character H | Ascii value is 72 |
| RetrieveCharacterData | Test to check that correct lines of FontDataArray are read and stored correlating to character movements. | Character H | 0 0 0  0 18 1  12 0 0  12 18 1  0 9 0  12 9 1  18 0 0 |
| Apply scale factor | Check to ensure that Scale factor is applied to stroke movements correctly. | Character H movement values:  0 0 0  0 18 1  12 0 0  12 18 1  0 9 0  12 9 1  18 0 0  Using User input of 5mm | 0 0 0  0 5 1  3.33 0 0  3.33 5 1  0 2.5 0  3.33 2.5 1  5 0 0 |
| CalculateWordWidth | Test to ensure that function correctly calculates the width of the current word being read. | Word to calculate width: Hello  Data input from array: outputMovemenArray | WordWidth = 90 units |
| GenerateGcode | Test converting output Movement Array coordinates to G code | Using character H movements in outputMovementArray | |  | | --- | | F1000 | | M3 | | S0 | | G1 0 0 | | S1000  G1 X0 Y18 | | S0  G0 X12 Y0 | | S1000  G1 X12 Y18 | | S0  G0 X0 Y9 | | S1000  G1 X12 Y9 | | S0  G0 X18 Y0 | |
|  |  |  |  |

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

# Flowchart(s)

May be included as separate pdf