Jacob Dobson - 20543825

Project planning Document – Robot drawing software

Outline of the Problem to be Solved

The project aims to create a code that can take text from a file and then convert that into G-code that will allow the drawing robot to write that text onto a surface. The steps taken by the code to achieve this result are:

- By reading the file 'SingleStrokeFont.txt' the data for all the 256 ASCII characters are stored in an array.
- The user is then asked to input the desired height of the outputted letters (between 4 and 10 mm). The code will then create the scaling factor to convert the font data into the desired height by using the formula height/18.
- The code will then read the desired text from a txt file making sure to only process and complete one word at a time
- The word will then be converted into G-code, and the x and y position of the pen will be tracked and recorded for each character that is processed and then transmitted to an Arduino via an RS232 connection
- The Arduino will then ensure that the text fits within the 100 mm width and if not will create a new line of text which is spaced 5mm apart from the previous line
- After all the contents of the text file has been drawn onto the surface the pen will be set to the up position and returned to the origin.
- The software should be able to recognise the inputs CR and LF in the input file which announces the end of a line and triggers the corresponding actions of the x and y positions

Key Data Items

Name	Data type	Rationale
Height	float	The height the user wants the text to be in mm. stored as float as used in calculations
PenPosition	int	Sets the position of the pen 0=up, 1=down. Stored as int as only a binary decision is required
XPosition	float	Records the x position of the pen to allow for correct spacing of letters. Stored as float as it will be used for calculations
YPosition	float	Records the y position of the pen to allow for correct spacing between lines and consistent height of letters. Stored as float as it will be used for calculations
MaxWidth	Const float = 100.0	Defines the max width of the line in mm. Stored as float as it will be used for calculations
LineGap	Const float = 5.0	Defines the distance between each line in mm. Stored as float as it will be used for calculations
FontFile	File*	Opens the 'SingleStrokeFont.txt' file using fopen()
TextFile	File*	Opens the text file that contains the desired output using fopen()

CurrentWord	array	Stores the current word being worked on from the text file. An array as it stores all the words one after the other which allows me to access them one by one
ASCII_Characters	array	Stores all the 256 ascii characters in an array

Function Declarations

Only include functions that you will develop.

Float HeightConversion(float Height);

Parameters:

Height – the user inputted height in mm

ScaleFactor – stores the scale factor (height/18)

Return value – the scale factor for the height

Int ReadFont(File* FontFile);

Parameters:

FontFile – the opened file of 'SingleStrokeFont.txt'

Return value – returns 1 if successful, 0 if failed

Int ReadNextWord(File* TextFile, char CurrentWord[]);

Parameters:

TextFile – the opened text file

CurrentWord – array to store the next word

Return value – returns 1 if successful, 0 if failed

Void GcodeConvertor(float *XPosition, float *YPosition, int *PenPosition, float ScaleFactor);

Parameters:

X and YPosition – pointers for the position of the pen at the time

PenPosition – pointer for whether the pen is up or down

ScaleFactor – the scale factor for height depending on the user input

Return value – won't return anything as it is void

Void NewLineCheck(float *XPositoion, float *YPosition, float linegap);

Parameters:

X and YPosition – pointers for the position of the pen at the time

LineGap – gives the spacing for the next line if a new line is needed

Return value – won't return anything as it is void

Void DrawWord(char CurrentWord[], float *XPosition, float *YPosition, int *PenPosition, float ScaleFactor);

Parameters:

X and YPosition – pointers for the position of the pen at the time

PenPosition – pointer for whether the pen is up or down

ScaleFactor – the scale factor for height depending on the user input

Return value – won't return anything as it is void

Testing Information

Function	Test Case	Test Data	Expected Output
Main()	Full working program	Height= 7 and a text file containing a phrase to output	The program reads the font and text file and then writes the text at a size of 7
Main()	Missing text file	make the program check for a text file that doesn't exist	An error should be outputted
HeightConversion()	Maximum height	Height=10	Return scale factor of 10/18
HeightConversion()	Minimum height	Height=4	Return scale factor of 4/18
HeightConversion()	Out of bounds height	Height=11	Output an error and ask to input an allowed height
ReadFont()	Valid font file	SingleStrokeFont.txt	The function returns a 1
ReadFont()	Missing font file	make the program check for a text file that doesn't exist	Function returns a 0
ReadNextWord()	Reading multiple words	Text file containing a string of words	Output the words one after the other and then return a 0
ReadNextWord()	Empty file	A text file with no words in	Outputs 0
GcodeConvertor()	Standard use	XPosition=0, YPosition=0 and a text file containing a letter	Generates the correct Gcode for the letter and updates the coordinates
DrawWord()	Standard use	Text file containing a word and the pen starts at the origin	Draws the word and updates the coordinates
NewLineCheck()	Starting new line	Text file with a word and XPosition =99	Start a new line and print the word
NewLineCheck()	LF and CR test	Text file containing LF and CR	Start a new line

GcodeConvertor	Character not included in font file	A text file containing a character not covered by the font file	Error should be outputted
DrawWord()	Pen up and pen down	Set pen to be up and down	Pens should move up and down according to the position provided in the code
DrawWord()	Offset between characters	A text file containing a word	The word should be printed with adequate spacing between characters

Extend table as required. Note that 'Function' includes main()

AI Statement

For my AI tool I used ChatGPT

Purpose of use:

Identify if I had missed any components within my submission

Prompts given:

"Have I missed anything from my submission" I also uploaded the planning sheet and my draft submission.

Outputs received:

In my general description I have forgotten to add parts regarding the ASCII data being stored in an array and I also forgot to add the statement regarding the LF and CR data in the text file. It recommended I also then add an appropriate data storage for the ascii character in my key data items section. It finally suggested that I need to add more tests in my testing section including: Characters not present in the font file, pen up and pend down test and offset between characters.

What I implemented:

I added lines in the description implementing the forgotten criteria, I also added an ASCII array into my key data items, and then I also added a test for the CR and LF values in the test section as well as the recommended tests provided by the AI engine.