# William Chun Qiu 20479482

# System manual

# Software Description

This software is designed to transmit g-code commands via a RS232 serial port to a writing robot to draw out text read from a user input file.

First, the program sends initialization commands to prepare the writing robot. Then it asks the user for a text height between 4mm and 10mm and checks for invalid inputs. After that font data is loaded from the “SingleStrokeFont.txt” text file which contains the data needed to generate g-code commands to raise, lower and move the pen on the writing robot. Next the program prompts the user for a text file name; checks if it can be opened correctly and processes each word in the text file. While processing, the program checks for the width of the words to ensure that the drawn text does not exceed the writing area of width 100mm by moving to the next line if the current line did not have space. G-code commands is then generated for each letter of the words by computing scaled coordinates based on the xOffset, yOffset and scaleFactor which is calculated using the text height divided by eighteen. The accumulated g-code commands are stored in a buffer and sent to the writing robot line by line using the sendCommands function. Finally, after all the text has been drawn, the program moves the pen back to the origin ensuring that the pen is up.

# Project Files

# main.c: Contains the main function, initialization, and logic for reading text and font data, processing it, and generating G-code for robot commands.

# serial.h: Header file containing the interface for serial communication functions (SendCommands, WaitForReply, etc.).

# SingleStrokeFont.txt: File containing the font data for the characters that will be drawn by the robot.

# text.txt: File containing the text that the robot is to draw.

# Key Data Items

|  |  |  |
| --- | --- | --- |
| Name | Data type | Rationale |
| xPosition | float | Stores the current x position of the robot’s pen |
| yPosition | float | Stores the current y position of the robot’s pen |
| gcodeBuffer | Char | Holds the accumulated g-code commands that are sent to the robot |
| scaleFactor | float | Scaling factor to adjust the size of the text based on user input |
| fontData | FontCharacter | Array to store the font data for each character |

# Functions

*Int main():*

*The main function initializes the robot, loads font and text data, processes the text file, generates G-code for drawing, and sends it to the robot.*

*Parameters: none*

*Inputs:*

* *User input for text height*
* *Text file name*

*Outputs:*

* *G-code commands stored in buffer*
* *Sends g-code commands to the robot through serial port*

*Return value:*

* *Returns 0 if program completes successfully*
* *Returns 1 if user provides an invalid text height*

*Void generateGcode():*

*Generates G-code commands to draw a single character at the specified position (xOffset, yOffset).*

*Parameters:*

* *letter – the character to draw*
* *xOffset - The X position for the start of the character*
* *yOffset - The Y position for the start of the character*
* *gcodeBuffer - The buffer where G-code commands are accumulated*
* *bufferIndex - the index indicating where the next G-code command will be stored in the buffer*

*Input:*

* *A character*
* *Current drawing positions*

*Outputs:*

* *Generates G-code commands for the pen movements required to draw the character*
* *The generated G-code is stored in gcodeBuffer starting at position bufferIndex*

*Return value: none*

*Void readTextFile():*

*Reads a text file and generates G-code for each word in the file, adjusting positions and ensuring proper line spacing.*

*Parameters: filename – name of text file to be read*

*Input: a text file containing the words for the robot to draw*

*Output: accumulates the g-code into the gcodebuffer*

*Return value: none*

# Testing Information

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Test Case | Test Data | Expected Output |
| Main() | Text processing and drawing | Text file with characters “Hello World” | G-code commands for drawing ”Hello World” without overlapping |
| generateGcode() | Single character drawing | Character “H” | Correct g-code sequence for drawing “H” with no pen lifting errors |
| readTextFile() | Reading and processing text | Text file “test.txt” | The program should correctly break the text into words and generate g-code without exceeding the width limit |

# Flowchart(s)

Included in a separate pdf