# Akshay & 20485507

# SystemManualAkshayNair

# Software Description

(Maximum 1 page)

# Project Files

(Maximum 1 page)

# Key Data Items

|  |  |  |
| --- | --- | --- |
| Name | Data type | Rationale |
| FontCharacter | Struct | To store the movement data for each ASCII character, including its dx, dy, and pen states. |
| fontData | FontCharacter[128] | Holds the font data for all 128 ASCII characters for efficient lookup during G-code generation. |
| scaleFactor | float | Scaling factor for font size, calculated based on user-defined text height. |
| xPosition, yPosition | float | Tracks the current X and Y positions of the pen, ensuring accurate movement. |
| dx, dy | float\* | Arrays within FontCharacter to store relative movements in X and Y for each character. |
| penState | int\* | Array within FontCharacter to indicate whether the pen is up or down during movements. |
| BUFFER\_SIZE | #define constant | Limits buffer size for file reading, ensuring memory safety when processing lines or commands. |
| scaleFactor | float | Converts user-defined text height into proportional scaling for movements and spacing. |
| LINE\_SPACING | #define constant | Defines the fixed spacing between lines to ensure consistent multi-line text rendering. |
| CHAR\_WIDTH | #define constant | Default width for characters, used to calculate horizontal spacing during G-code generation. |
| fontFile | char[] | Stores the name of the file containing the font data (SingleStrokeFont.txt). |
| textFile | char[] | Stores the name of the input text file provided by the user for generating drawing commands |
| gcodeBuffer | char[] | Temporary buffer for storing G-code commands to be written to the output file. |

Extend table as required

# Functions

**1. float getTextHeight(float userHeight)**

**Short description**: Converts the user-defined text height into a scaling factor for font size.

**Parameters**:

* userHeight – Height of the text in mm (between 4mm and 10mm).

**Return value**:

* Returns the scaling factor to be applied to character rendering.

**2. \**void readFontData(const char fileName)***

**Short description**: Reads font data from the specified font file and populates the global fontData array with movement and pen state information for ASCII characters.

**Parameters**:

* fileName – Name of the font data file (e.g., "SingleStrokeFont.txt").

**Return value**:

* None. If the file fails to open, the program exits with an error.

**3. \**void readTextFile(const char fileName)***

**Short description**: Reads the user-specified text file and generates G-code for the text line by line, writing the commands to an output file.

**Parameters**:

* fileName – Name of the input text file containing the text to be drawn.

**Return value**:

* None. If the file fails to open, the program exits with an error.

**4. \**void generateGcode(char letter, float xOffset, float yOffset, FILE outputFile)***

**Short description**: Generates G-code for a given character based on its font data, applying the current scaling factor and offsets, and writes it to the output file.

**Parameters**:

* letter – ASCII character to generate G-code for.
* xOffset – X-coordinate offset for character placement.
* yOffset – Y-coordinate offset for character placement.
* outputFile – File pointer to the G-code output file.

**Return value**:

* None.

**5. \**void resetPenToOrigin(FILE outputFile)***

**Short description**: Writes G-code to return the pen to the origin (0, 0) at the end of the drawing process.

**Parameters**:

* outputFile – File pointer to the G-code output file.

**Return value**:

* None.

**6. \**void SendCommands(char buffer)***

**Short description**: Sends a command buffer to the robot via serial communication and waits for acknowledgment.

**Parameters**:

* buffer – Command string to be sent to the robot.

**Return value**:

* None.

# Testing Information

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Test Case | Test Data | Expected Output |
| getTextHeight | Valid text height input | userHeight = 8.0 | Returns 0.444 (8.0 / 18.0). |
|  | Invalid text height input (less than 4mm) | userHeight = 3.5 | Error message: "Error: Height must be between 4mm and 10mm." |
|  | Invalid text height input (greater than 10mm) | userHeight = 11.0 | Error message: "Error: Height must be between 4mm and 10mm." |
| readFontData | Valid font file input | File: SingleStrokeFont.txt (contains valid data) | Populates the fontData array with character movement and pen state information. |
|  | Missing or invalid font file | File: missing.txt | Error message: "Error opening font file". Program exits. |
| readTextFile | Valid text file input | File: text.txt (contains "HELLO") | Generates G-code in gcode\_output.txt for "HELLO". |
|  | Missing or invalid text file | File: missing.txt | Error message: "Error opening text file". Program exits. |
| generateGcode | Valid character input | letter = 'A', xOffset = 10.0, yOffset = 20.0 | Generates G-code movements for the character 'A', scaled and offset. Writes to output file. |
|  | Invalid character input (ASCII not in font) | letter = '@' | No G-code is generated for the character. |
| |  | | --- | | resetPenToOrigin |  |  | | --- | |  | | Reset the pen to origin after drawing | N/A | Writes "G0 X0 Y0" to gcode\_output.txt. |
| SendCommands | Valid buffer input | buffer = "G1 X10 Y10\n" | Sends command "G1 X10 Y10" to the robot. Waits for acknowledgment. |
|  | Empty buffer | buffer = "" | Sends an empty command to the robot. No errors are raised. |

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

# Flowchart(s)

May be included as separate pdf