# Mathis Bellino – 20342807

# CNC Plotter Coursework

# Outline of the Problem to be Solved

**The program must:**

1. Read font information from 'SingleStrokeFont.txt' storing complete font data in memory.
2. Obtain desired text height (4-10mm) inputted by the user (via keyboard input).
3. Read text to be drawn from a file (filename via keyboard input).
4. Process text file word by word, generating appropriate G-codes for each character, making sure no words are split and the maximum length of a series of words is X.
5. Scale the movements such that letter height (excluding ascenders/descenders) matches user input.
6. Handle the line spacing: 5mm between lines with a width constraint of 100mm max.
7. Transmit G-code commands to Arduino using serial communication functions .
8. Ensure proper pen up/down states and positioning.

# Key Data Items

# **Core Functions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Data type** | | **Rationale** |
| getValidHeight | **Input**: float\* height **Output**: float | | Gets and validates the height of the user’s inputted character height |
| loadFontLibrary | **Input**: const char\*  (font name) **Output**: FontLibrary\* | | Reads entire SingleStrokeFont.txt file into memory. Creates indexed structure containing all character stroke data. Returns NULL if file read fails |
| processText | **Input**: const char\* Font Data\* float height  **Output**: **bool** | | Main text processing function Manages overall text drawing tasks  Returns false if error occurs |
| calculateWordWidth | **Input**: const char\*  Font Data\* float height **Output**: **Float** | | Determines if word fits on current line  Returns calculated width |
| processWord | **Input:** const char\*  Font Data\* float current x  float current y  float height **Output**: **Float** | | Handles individual word processing     Returns ending x position |
| handleLineBreak | **Input**: float\*  float\*  **Output**: **bool** | | Manages new line transitions Updates X and Y coordinates Returns true if line break possible Returns false if at page limit |
| scaleMovement | **Input**: MovementData**\***  float height **Output**: bool | | Scales coordinates to proper height |
| generateGCode | **Input**: const char\* **Output**: FontLibrary\* | Creates and sends GCode commands | |

Extend table as required

# Function Declarations

**bool getValidHeight(float\* height)**

Parameters:

- height - pointer to store validated height value

Return value: returns true if valid height (4-10mm) obtained

**FontLibrary\* loadFontLibrary(const char\* filename)**

Parameters:

- filename - path to SingleStrokeFont.txt

Return value: returns pointer to complete font library structure containing all characters

**bool processText(const char\* filename, FontLibrary\* font, float height)**

Parameters:

- filename - input text file to process

- font - pointer to loaded font data

- height - validated text height

Return value: returns true if text processed successfully

**float calculateWordWidth(const char\* word, FontLibrary\* font, float height)**

Parameters:

- word - string to calculate width for

- font - pointer to font data

- height - text height for scaling

Return value: returns calculated width in mm, -1 if error

**bool processWord(const char\* word, FontLibrary\* font, float\* currentX, float\* currentY, float height)**

Parameters:

- word - string to process

- font - pointer to font data

- currentX - pointer to current X position

- currentY - pointer to current Y position

- height - text height

Return value: returns true if word processed successfully

**bool handleLineBreak(float\* currentX, float\* currentY)**

Parameters:

- currentX - pointer to current X position

- currentY - pointer to current Y position

Return value: returns true if line break possible

**bool scaleMovements(MovementData\* move, float height)**

Parameters:

- move - pointer to movement data to scale

- height - desired text height

Return value: returns true if scaling successful

**bool generateGCode(MovementData\* move, bool penState)**

Parameters:

- move - pointer to movement data

- penState - true for pen down, false for pen up

Return value: returns true if G-code sent successfully

Each function includes:

- Clear parameter list with types and purposes

- Return value explanation

- Uses consistent pointer pattern for outputs

- Follows error handling pattern from skeleton code

# 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