Name & Student ID

Christopher Collins 20360137

Title

Robot Writing project plan

Outline of the Problem to be Solved

The aim of this project is to develop a code that will be able to read text from file and draw out the text in a size decided by the user. The G code, within the main code, will command the robot how to draw out the text file entered. The size of the text written shall be decided by the user, with the height being set between 4 and 10mm before the drawing starts, the size decided will affect the positioning and so the code will have to make sure the text does not break 100mm wide from the origin of the pen.

When writing the code, a main concern will be checking that the text file is stored and scaled correctly so that when the G code commands are sent to the robot all the letters are correctly sized and stay within the width limit. Another issue that may come about is ensuring that the G code will correctly command the robot to lift the pen up and down to separate out each letter, with this will then come the issue of making sure the robot will begin a new line once the width limit of the area is reached.

Key Data Items

Name	Data type	Rationale
Single stroke Font	Text file	Contains the shape data for the letters that will be drawn out by the robot
Text Data	structure	Will store the data from the text file
Text scale	int	Will scale the data for the letters
Text Area	Integer Array	Decides the area that the text will be written in including the width limit
Pen Position	int	Tracks where the pen is currently within the area, 1 for X direction, one for Y direction
Pen Height	int	Tracks whether the pen is up or down

Extend table as required

Function Declarations

WidthLim (TextAreax, PenPosx, Textscale)

```
If (TextArea-PenPosx < textscale)
……..
```

TextArea – the area where the text is being drawn

PenPosx – the pen position in the x axis

Textscale – the size of the text

Return Value – If the pen position is too close to the width limit then the code will decide to skip to the next line

DrawLetter (PenPosx, PenPosy, TextData, ShapeScale)

```
   G = TextData[i]

   X = TextData[i].x*shapescale/18

   Y = TextData[i].y*shapescale/18

  Sprintf(“G=%d, X=%d, Y=%d\n", G, X, Y)

……
```

PenPosX – records the pens position on x

PenPosY – records the pens position on Y

Text Data - is where the letter data is stored

Return Value – the G code is sent to the robot

Testing Information

Function	Test Case	Test Data	Expected Output
WidthLim	PenPosx = 98	Test Area x = 100	Test Area – penposx < 5 becomes false so moves to next line
DrawLetter	0 18 1	G = 1 X = 0 Y = 18 Scale height = 10	Pen goes down and moves to 0,10. (more lines of text would continue movement until 999)

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

Flowchart(s)

May be included as separate pdf