

Robot Writing software project

Software Description

This Code is designed to communicate with an Arduino, through the serial port, to control a robot arm that will draw out a string of text using coordinates from a text file translated into g code. An example of what this would look like is on the next page of this document.

At the start of the code including the header files `rs232.h` and `serial.h` makes sure that the code will be able to call functions needed for communicating with the robot (e.g. `CanRS232portbeopened`) which are needed for the code to work and communicate properly.

Once the code begins, the `CanRS232portbeopened` function that is included using the header files makes sure that communication can begin with the robot and will exit the code if not, this ensures any issues with connections are picked up and can be resolved before trying the code again. After communication has begun successfully the code waits for a response of the \$ dollar sign from the Arduino to confirm the Arduino is ready, once received g code is sent to prepare the arm position for drawing. The code will then ask the user for the name of the text file the user has entered and wants to read from, providing an error message and exiting the code if the text file cannot be found, if the text file was opened correctly the code will then ask for the desired font height for the text from the user which will be scaled to provide the correct size when drawn out, the robot will only accept a size between 4 and 10 mm and will ask until a correct size is entered.

If the file has been opened and a correct font size is chosen, the robot will begin to write out the text reading line by line and scaling the values taken from each line before sending off the g code using the `sprint` function to the robot. As the robot is writing there are two functions, checking for breaks with 999 to indicate a new letter/ word is being entered and checking if the text is getting to the 100 mm limit in the x direction, if either scenario occurs the functions will be called to ensure that the text is written out correctly by either leaving a space for a new letter/word or moving to a new line if the x limit is going to be reached.

Finally, once all lines of the text file have been read and written out by the robot, the code will print a message to let the user know. The code will then close the opened file and close the com port printing out another message to let the user know communication has stopped.

Project Files

`MainRobotCode.c` – the code the user will run to draw out their text file

`Serial.h`- defines functions useful for ensuring correct communication over ports and data is being sent and collected properly

`Serial.c`-contains the functions for managing correct communication over the serial ports.

`Rs232.h` – contains functions used for communication over rs232 com ports.

`Rs232.c`- provides useful information and code for serial communication over rs232

Key Data Items

Name	Data type	Rationale
FILE	structure	Stores the data from the text file which can then be easily read line by line using fscanf.
Filename	Char	Used for storing the name of the file the user is opening, will be characters/text
FontHeight	Float	Stores the height from the user, is scaled. Int could also be used but could cause problems
X and Y	Float	Numbers taken from the text file, will be scaled and can end up as decimal so float makes most sense as int wouldn't work
G	Int	Can either be 1 or 0 so no need for float.
Xpos and Ypos	Float	Store the current x and y positions which are used for tracking the current drawing postions
MaxX	Float	Finds leading edge of letter to help correctly position the next letter

Extend table as required

Functions

int CheckNewLetter(float X, float *Xpos, float *MaxX)

parameters:

x – value for x coordinate from text file

x pos – current x position

MaxX – the leading point of current letter

Return value – if 999 is detected will adjust coordinates to make a space for a new letter

int CheckNewLine(float *X, float *Xpos, float *Ypos, float LineSpacing)

X- value for x coordinate from text file

Xpos – current x position

Ypos – current y position

LineSpacing – the distance that the pen will move for a new line

Return value – if x gets to 100 it will start on the next line

Testing Information

Function	Test Case	Test Data	Expected Output
int CheckNewLetter	999 / new letter marker found	999 1 26 Xpos = 10 MaxX=12	Xpos=13 MaxX=0
int CheckNewLine	x=100	G1 X96.00 Y-45.00	Xpos=0 Ypos=-70

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

[Flowchart\(s\) included as separate pdf](#)

Example of g code emulator

