ELEC4403 Lab 8 – Lab Prep

Main ideas:

- Use Remote Desktop to login
- Use a menu/keys to control
- Calibrate PSD raw values using GUI at start to find appropriate values.
- Use previously defined variable definitions: QQVGA_X = 160, QQVGA_Y = 120, QQVGA_SIZE = QQVGA_X*QQVGA_Y*3(ie. Number of bytes).
- Use LCDSetPrintf(row, column, message) to display on LCDScreen text under the image, also useful for overwriting old values.
- Use a control integer to select which mode to work in
- Use lots of printf statements to how status, error checking
- Draw a line to represent the column with the max red pixels
- Use precompiler #define to specify variables to later easily calibrate
- Use modular blocks of code, eg. One function to drive/control, one function to process the image
- Use <math.h> to do sqrt, typecasting where needed.
- If doing exercise using RGB values, a basic method to verify if a pixel is red (remembering RGB values are 0-255):
    - pixel_red > 50 and pixel_red> pixel_blue*2 and pixel_red>pixel_green*2

Main algorithm details

- Have a start window, to init all
- Move to a new window to show current status (camera/image, binary image and centre of object).
- Have a button to start actual driving/restart process, whilst displaying mode on screen and binary image with current status/mode to see what is going on
- Print more detailed info on command line to see activity, ie all variables, PSD values, actions, without cluttering the screen
- Once motor control hits the can, use VWGetPosition(&x, &y, &phi) to obtain current position, VWTurn(-phi) then calculate distance back using x, y values to VWStraight().
- Use VWDriveWait() in between drive sequences to fully complete driving/action
- Once returned, turn 3.14 radians and start back at idle status, requiring a key press to restart driving. Remove can at this point

Screenshots: