

Functionality of the Project:

Before entering the UART command processor, a variety of tests are performed on various peripherals and components. After entering the command processor, the following is the functionality of this project.

Before a command is entered (inside the line accumulator function):

In the function where a line buffer is accumulated (`command_line_interface()`), the `getchar()` is changed to a non-blocking `getchar()`. Thus, while accumulating a line buffer, it is possible to perform another action. I am checking if the DAG is horizontally flat in this function along with accumulating a command. If the DAG is perfectly flat on the table, RED LED is lit and if it is not flat the color transitions to green using PWM. In other words, the closer it is to 0 degrees the more red and less green the LED is and the further it is from 0 degrees the less red and greener it is. This functionality **only works** when the DAG is **not calibrated/ zeroed** as we set the green LED after it is calibrated, and we do not want to mess with the green LED color after it is calibrated. Once the calibration is removed after setting the desired angle, this functionality will again start working.

After a command is entered:

Once the user enters a command, we process the command entered.

The following are the available commands:

Calibrate:

This command is used to calibrate/zero the DAG. Once we enter the handler of calibrate function, the WHITE LED is set. We move the DAG to the desired angle and press the button switch to set the current angle of the DAG as relative zero. Once calibrated the GREEN LED is lit.

Set <angle>:

This command is used to set an angle desired by the user through the command processor. The user enters `set <angle>` (for example, `set 45` or `set 90`) after which we move the DAG from the relative zero angle to the user entered desired angle. The color of the LED is green initially and as we move to the desired angle the color is changed to blue through PWM. Thus, giving the user an idea about the angle. Additionally, the current angle is also displayed on the terminal every single second to provide the user some more insight on the movement of the DAG.

Once the user attains the desired angle, the BLUE LED is lit and the user **MUST** press the touch sensor to come out of this set mode. After coming out of the loop the previously calibrated state is removed and relative zero is set to zero degrees. Since the calibration is removed, the functionality to check if the DAG is flat on the surface will work now in the `command_line_interface()` function.

Info:

This command is used to gain some useful information on the DAG. The relative zero is printed on the screen, the angle with respect to the initial zero is printed on the screen, the calibration status is printed on the screen and the maximum measurable angle is also printed on the screen.

Help:

Prints out information about all the available commands.

Author:

Prints the name of the author.

Corner cases:

1. What happens if we “set” before calibrating?

The Relative zero is assumed to be actual zero. For example, if set 90 before calibrating we need to attain 90 degrees with respect to actual zero degrees.

2. What happens when switch is randomly pressed before entering calibrate command?

When the switch is pressed before entering the calibrate command, the interrupt is triggered, and the global variable is set. So, inside the calibrate handler when we check if a switch was pressed it would return true even though we did not press the switch at that point of time. This is handled by clearing the global variable as soon as we enter the calibrate handler and then check if a switch was pressed.

The same is done with a touch sensor

3. What happens if the user moves the DAG in the opposite direction of the relative zero-degree angle away from the required angle? For example, if the relative zero is 90 degrees and the required angle is 120 degrees what happens if the user moves below 90 degrees?

The color of the LED remains green if the DAG is moved in the wrong direction and “Wrong direction !!!” is printed on the terminal. Once the user corrects their direction and moves towards the desired angle, the LED again starts transitioning from GREEN to BLUE and the current angle and required angle is printed on the terminal.

4. What happens when the user calibrates the DAG’s relative zero to be a high angle? What is the maximum angle that they can measure after that calibration?

Let us assume that the relative zero-degree angle is calibrated to be 160 degrees. Now, the maximum angle that can be measured is 20 degrees (180 – 160 degrees). So, if the user enters an angle greater than 20, I am showing the available range and immediately returning to the command processor without setting the entered angle as it is not possible to set that angle. The calibration is untouched and the relative zero remains 160 degrees.

5. What happens when the user enters an invalid/unknown command?

If the command entered is unknown “unknown command: <entered command>” is printed out to the screen.

If the user enters an invalid command, it is handled appropriately in the command handler function. For example, if the user enters **set 90 90** “Invalid command for set – look at help” is printed. Similarly, **set abcd1** is also handled.

For more information on the command handler’s erroneous input processing, look at the manual testing section of the test results document.