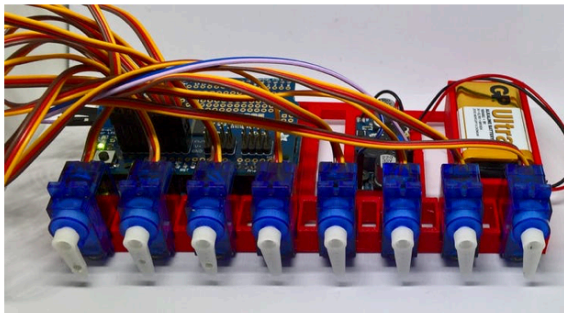


# Calibrating the Servo with Scan Library

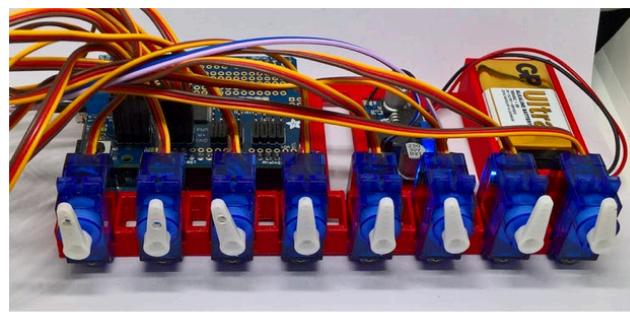
CPRE 288

While using the provided scan library, you will notice that the servo does not always sweep 180 degrees like we would expect.

You might ask, why? Like many physical devices, there are differences in devices, even in new devices, due to variations in components and manufacturing. There are also variations in wear-and-tear over time. **For servo motors, calibration means setting up the servo so that it accurately positions itself when given an angle.** For example, when given 0 degrees, does it align accurately with the 0-degree position? When given 90 degrees? When given 180 degrees? Each servo will have slightly different accuracy. For example, the following images show an experiment in which identical servos were programmed with identical code to rotate from a starting position (pointing down) to 180 degrees (slightly different positions pointing up).



All the servos in their initial minimum starting position



The eight servos in their maximum positions - all slightly different.

(<https://www.hackster.io/jeremy-lindsay/how-accurate-are-my-servos-21f703>)

Physical devices usually don't behave perfectly. Thus an embedded computer system may need to account for the imperfect behavior.

There is a function provided in the scan library that can be used to manually calibrate the servo to more accurately position itself. Note that calibration results are specific to a particular servo. Different servos will have different calibration parameters.

Here is a reference sheet for you to follow to calibrate your servo motor.

## Performing the Calibration

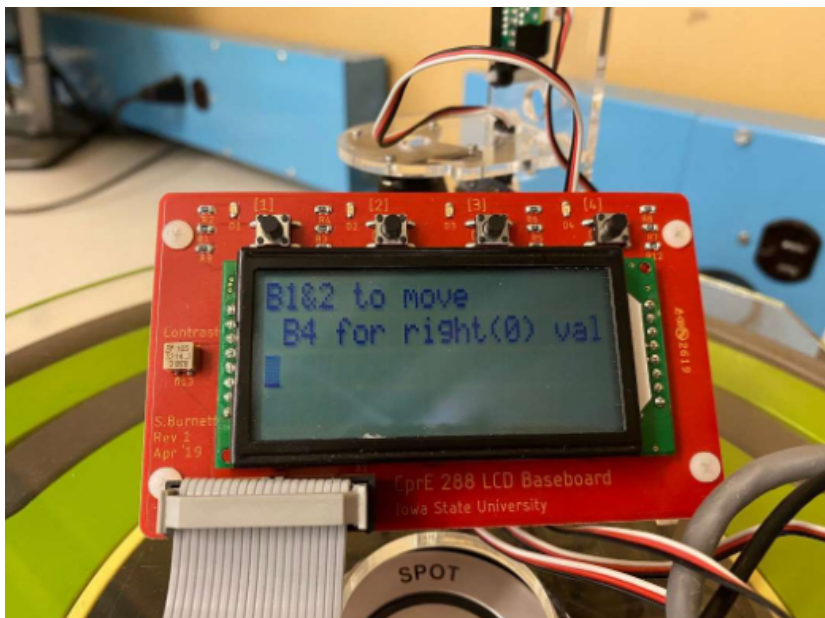
1. Take a look at the contents of `cyBot_Scan.h`. You should see a function called `cyBOT_SERVO_cal()`.
2. Create a short program as follows:

```

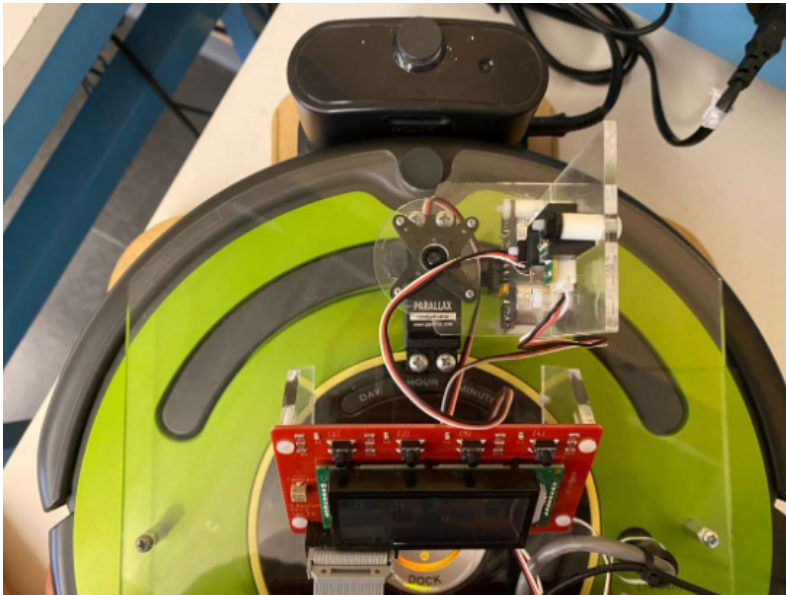
1#include "cyBot_Scan.h"
2#include "cyBot_uart.h"
3#include "Timer.h"
4#include "lcd.h"
5
6int main(void)
7{
8    /* Always good practice to initialize the timer
9     * before you initialize the lcd screen.
10     * This is because the lcd uses timer functions.
11     */
12    timer_init();
13    lcd_init();
14    cyBOT_init_Scan();
15    cyBOT_SERVO_cal();
16
17    return 0;
18}

```

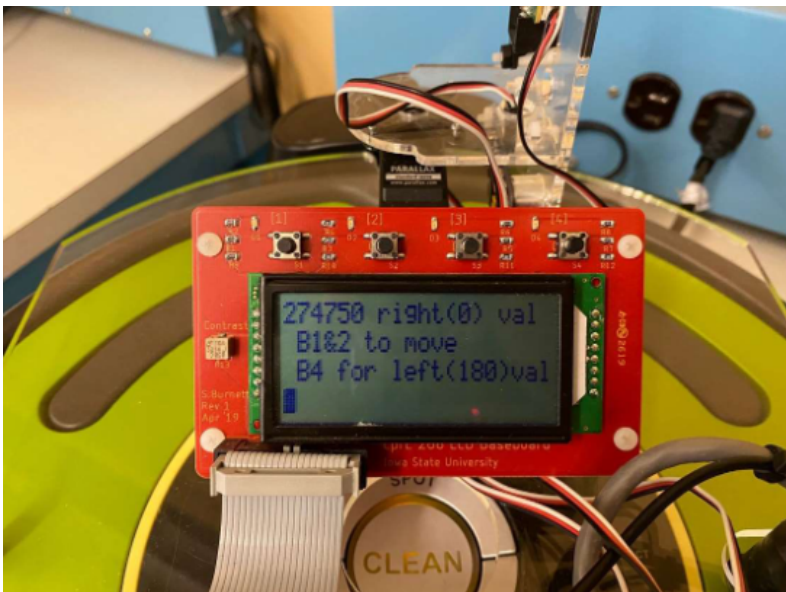
- Flash it on the board and you should be prompted with a short menu on the LCD display as shown below:



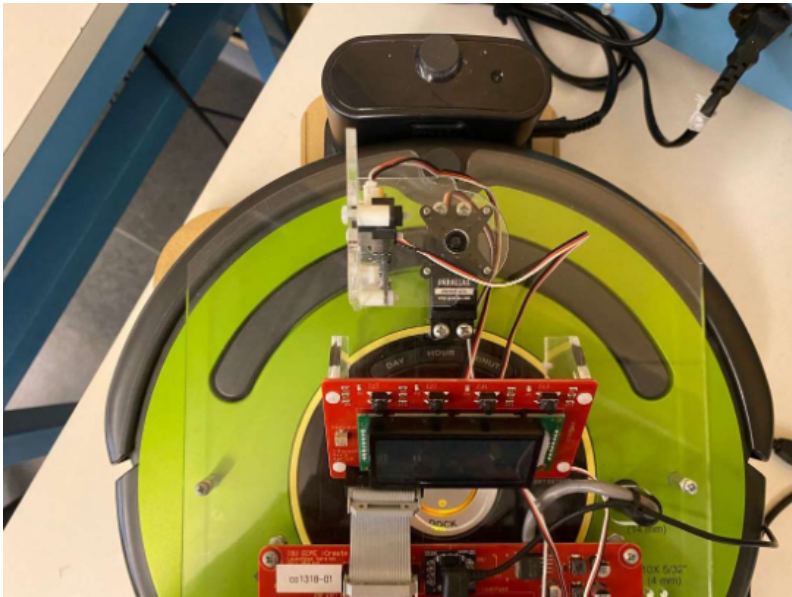
- Use buttons 1 and 2 to move the servo either left or right. In this case, press button 2 to move the servo all the way to the right until it looks like it's at 0 degrees. This is shown below:



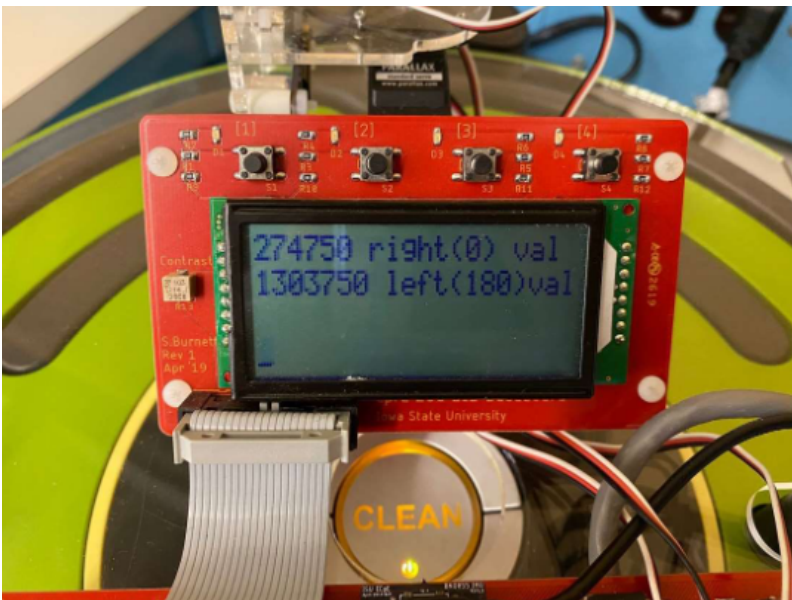
5. Press button 4 to confirm. The display is shown below after pressing button 4. The value shown is the calibration parameter for 0 degrees.



6. Repeat the same process for 180 degrees, pressing button 1 to move the servo all the way to the left, and then press button 4.



7. The display is shown below after pressing button 4. The calibration parameters for 0 degrees and 180 degrees are shown. Write down these two values. You will use them in your code.



## Using the Calibration Values

1. If you skimmed through the scan header file, you should notice that there are two **int** variable declarations (declared in the **HEADER** file):
  - `int right_calibration_value;`
  - `int left_calibration_value;`



These are effectively global variables that you initialize in the **MAIN** file. Your assignments to these variables **must be placed AFTER the call to cyBOT\_init\_Scan()**, because the scan initialization function initializes these variables to default values that are not specific to your bot.

2. Going back to the main method from step 2 above, comment out or delete cyBOT\_SERVO\_cal(), and insert assignments to these variables, assigning the values you found from step 7 above. The program below shows the new assignments (ignore the values). You will use the values obtained from your calibration procedure.

```
#include "cyBot_Scan.h"
#include "cyBot_uart.h"
#include "Timer.h"
#include "lcd.h"

int main(void)
{
    /* Always good practice to initialize the timer
     * before you initialize the lcd screen.
     * This is because the lcd uses timer functions.
     */
    timer_init();
    lcd_init();
    cyBOT_init_Scan();
    //cyBOT_SERVO_cal();
    right_calibration_value = 332500;
    left_calibration_value = 1303750;

    int i;
    cyBOT_Scan_t scan;

    for (i=0; i <= 180; i+=2)
    {
        cyBOT_Scan(i, &scan);
    }

    return 0;
}
```

3. With these calibration parameters, your servo should now be calibrated. The program above also illustrates a simple way to test the calibration, i.e., the for loop.
4. Once tested, you don't need to keep the testing loop, however, you will keep the assignments to the variables in your main program. You should put the CyBot number in comments, so you know which bot these values are for. With these values assigned in your main program, they will be used by cyBOT\_Scan() to more accurately position the servo. **NOTE: DIFFERENT BOTS HAVE DIFFERENT CALIBRATION VALUES. VALUES FOR ONE BOT MIGHT NOT WORK FOR ANOTHER.**