

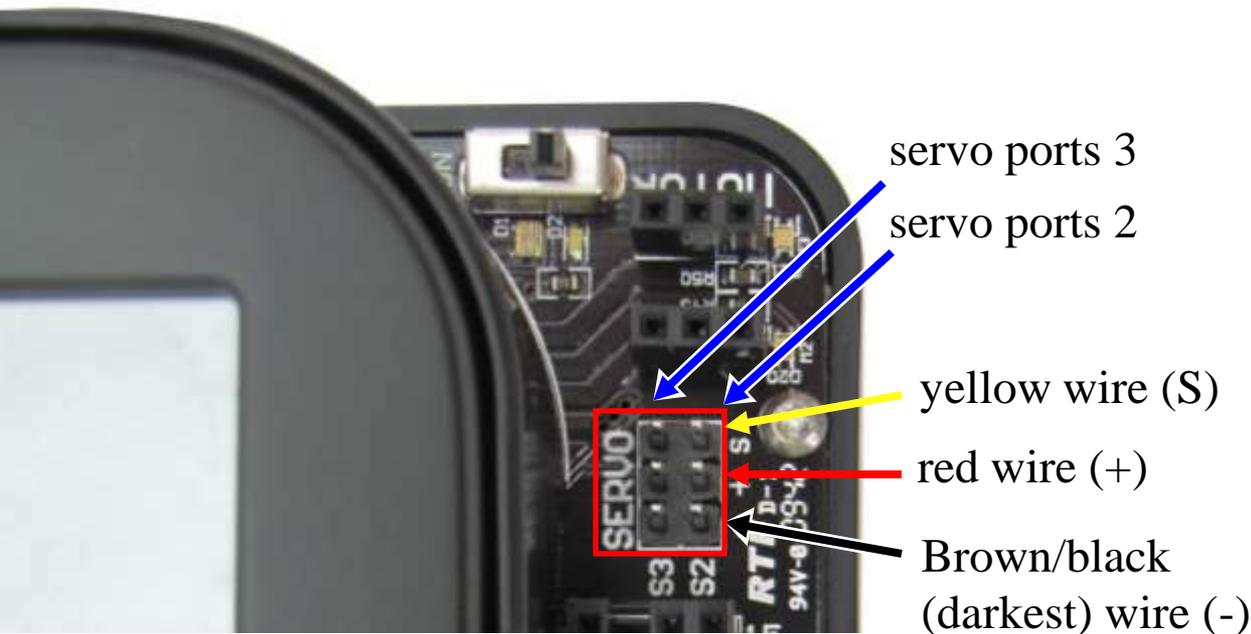
CBC Servo Motor Ports



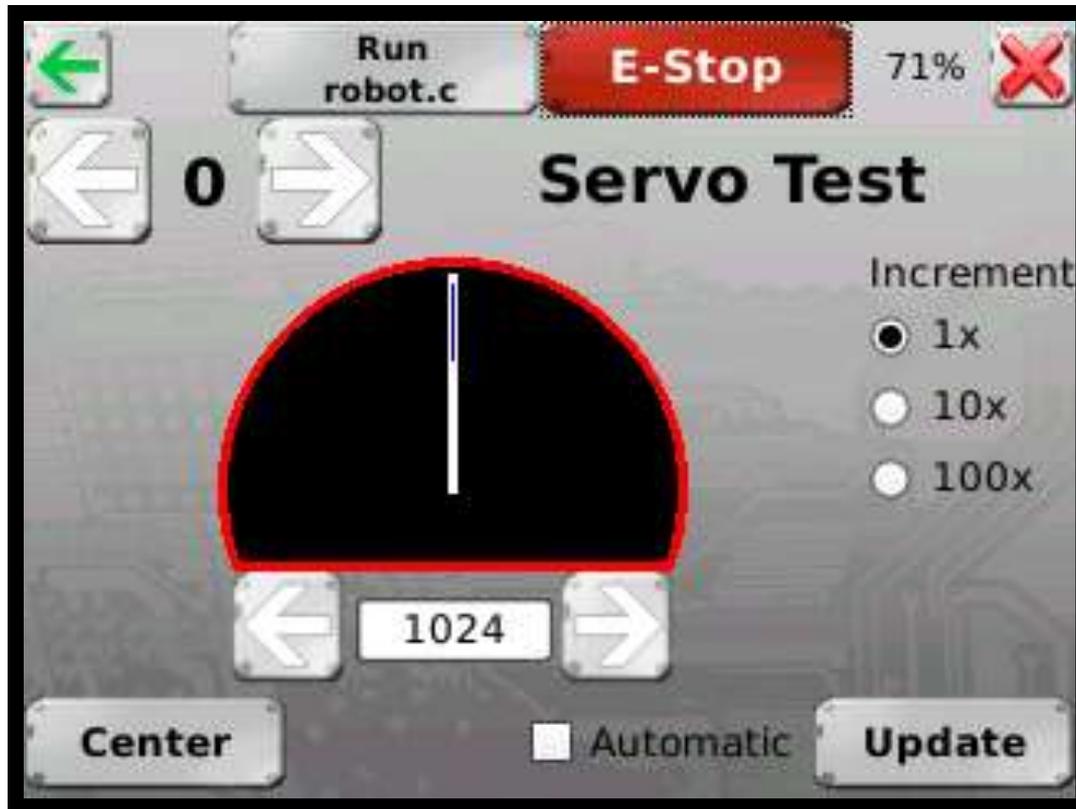
servo ports 0 and 1; servo ports 2 and 3

Plugging in Servos

- Servo motors plug into the CBC servo ports
- (brown/black-red-yellow cables with 3 prong receptacle)
- The CBC has 4 servo ports numbered 0 & 1 on the left and 2 & 3 on the right
- Plug orientation order is, left to right, brown/black-red-yellow when the CBC is oriented so the screen can be read (or follow the labeling: - + S; the yellow signal wire goes in S)



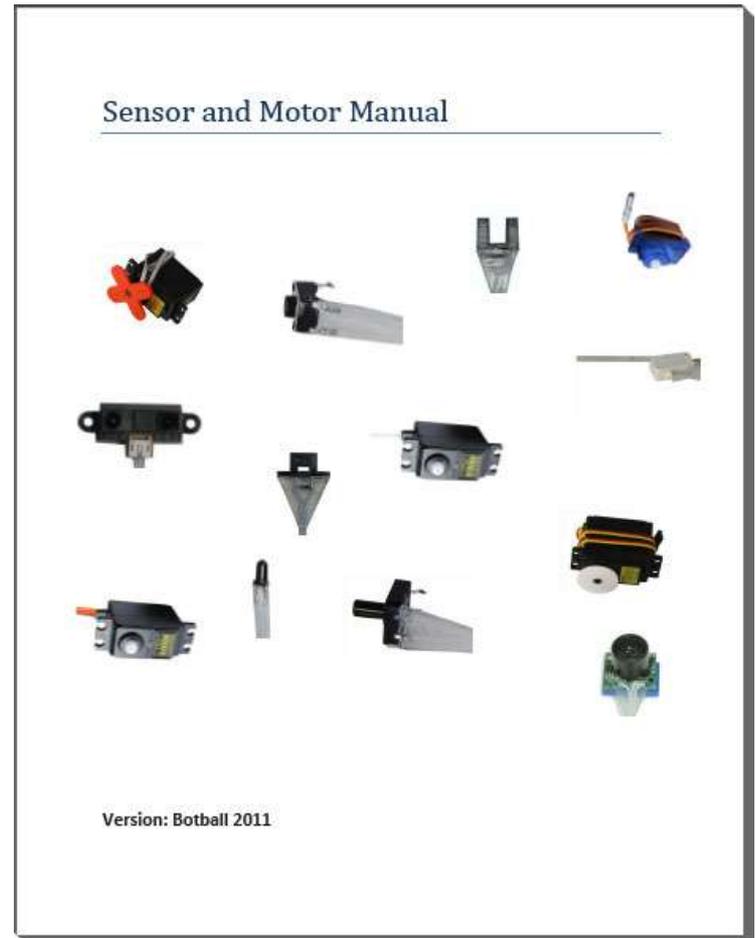
CBC *Servo Test* Screen (under Sensors/Motors)



The CBC *Servo Test* screen can be used to center a servo or quickly determine what position values to use once the servo is installed on a bot - Try it!

Sensor and Motor Manual

For further detail about servos & motors, consult the Sensor and Motor Manual available via KISS IDE help



Using a Loop with a Servo

- A motor can be commanded (via the CBC) to move to fixed rotation at a speed, but a servo command sets only the final position, which it moves to at its *maximum* speed.
- We can move a servo in a more controlled manner, using a loop with the servo function to move it in small steps, with a delay at each step, until the final position is reached.
- Suppose we wanted a servo (#1) to move from position 200 to position 1800 in steps of 100 (~10 degrees), with 0.1 sec between steps
 - we could write 18 `set_servo_position` commands to do this, or
 - this can be done easily with a `for` loop:

```
for (pos=200; pos <1800; pos += 100)
{set_servo_position(1, pos); sleep(0.1)}
```

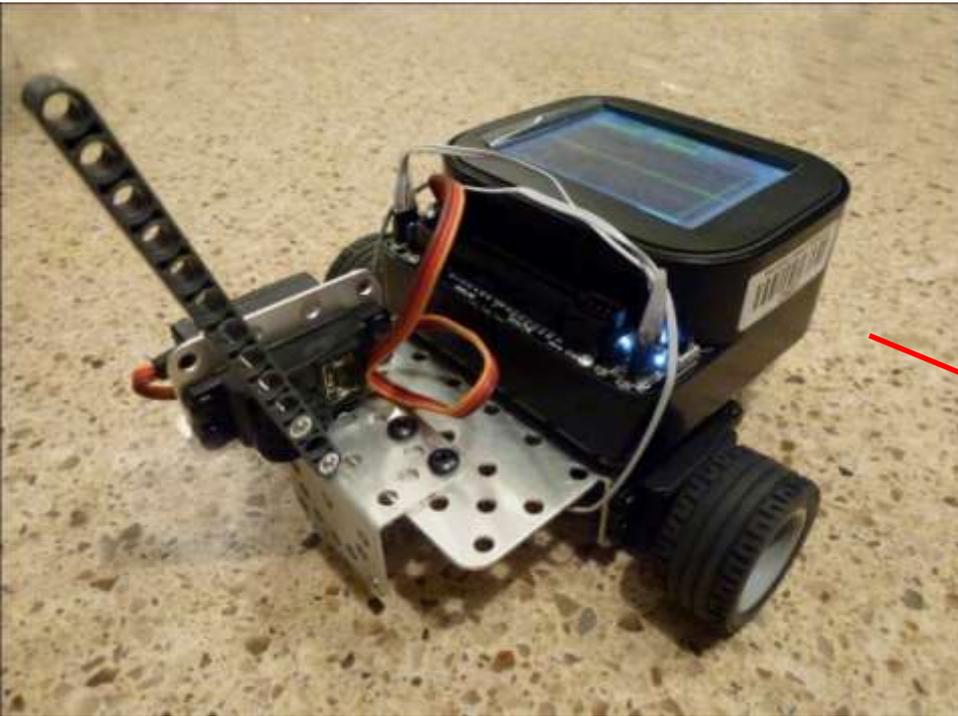
Moving a Servo in a Program

[Assume servo #1, with an arm, is attached to the Demobot]

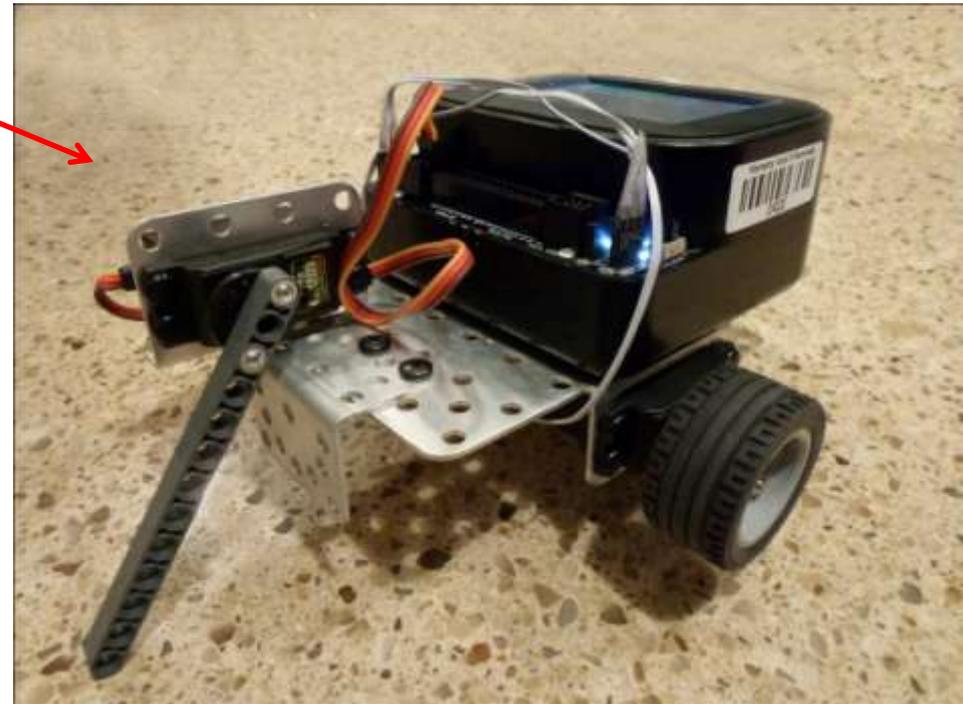
Suppose we want to move an arm down in response to a 'down arrow' on the display screen until it reaches to maximum position; use 'B' button to quit:

```
int main() {
    // preset servo 1 position
    int pos =200, max=1800;
    set_servo_position(1,pos);
    enable_servos(); // turn on servos
    sleep(.1); // pause while it moves
    printf("advance using down arrow\n\n use 'B' to quit\n");
    for(pos=200;pos <max;pos += 100)
        {set_servo_position(1,pos);
         printf("servo at %d\n",pos);
         sleep(0.1) // pause before next move
         while(!down_button() && !b_button()); //wait for button
         if(b_button()) break;
        }
    disable_servos();
    return 0;
}
```

Expected Behavior



Before



After

Moving a Servo in a Program

- Reflection -

- Notice that the 'for' loop is a special case of a while loop where desired activity is fixed.
- Suppose you wanted the arm to move up and down in response to the respective arrow? Try this loop:

```
while(pos>=200 && pos <1800 && !b_button()) {  
    if(down_button()) {pos+=100;  
        set_servo_position(1,pos);}  
    if(up_button()) {pos-=100;set_servo_position(1,pos);}  
    printf("servo at %d\n",pos);  
    sleep(.1);}  
}
```