Topic 30

Servo Motors and Servo Winches

Chris Summa

Fall 2004

Goals

Gain familiarization with:

- Servo Motor Anatomy
- IC Commands
- Calibrating a servo
- “Hacking” servo to create a “winch” servo

Servo Motor

Specifications

• Specialized motor for turning to a specific position

• Components:

– DC motor

– Gear reduction unit

– Shaft position sensor

– Electronic circuit that controls the motor's operation

• "Servo" - capability to self-regulate its behavior, i.e., to measure its own position and compensate for external loads when responding to a control signal

• Widely used in hobby radio control applications:

– RC cars: position the front wheel rack-and-pinion steering

– RC airplanes: control the orientation of the wing flaps and rudders



Futaba S3003 Servo Motor

• Positioning applications:

– Shaft travel is restricted to 180 degrees

– Input waveform specifies desired angular position of output shaft

– Electronics measure current position

– If different from desired position, servo is turned on to drive the shaft to the desired position

Servo Motor

Servo Control

• Most hobby servo motors use a standard three wire interface:

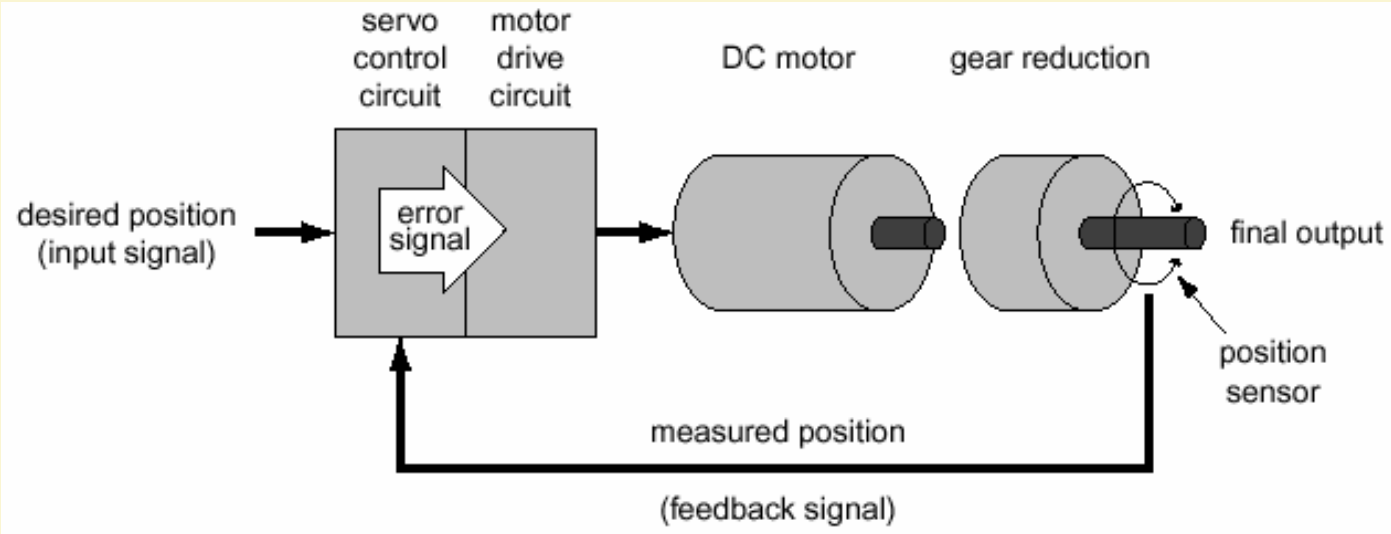
– Power

– Ground

– Control Line

• Power supply is typically 5 to 6 v

The **input** to the **servo motor** is desired position of the output shaft. This signal is compared with a **feedback signal** indicating the actual position of the shaft (as measured by position sensor). An **"error signal"** is generated that directs the motor drive circuit to power the motor. The servo's gear reduction drives the final output.



Servo Motor

Servo Control Signal

• Control line uses a **PWM scheme** for encoding the position signal

• Servo PWM method is different from the speed control PWM

– Speed control PWM: **overall duty cycle** (i.e, percentage of on-time) determines the speed of the motor

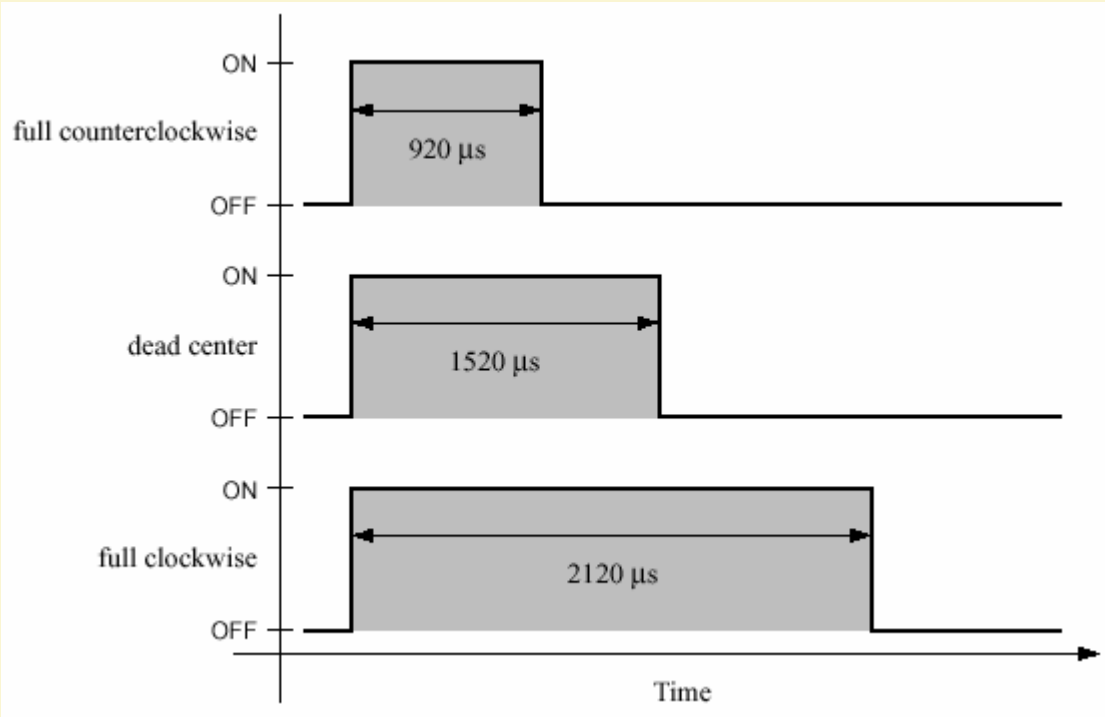
– Servo PWM: **length of the pulse** is interpreted to signify control value

• Waveforms' length:

– **920 μs** - full counterclockwise

– **1520 μs** - center position

– **2120 μs** - full clockwise



Three sample waveforms for controlling a servo motor

Servo Motor

Servo Control Signal

• To complete the servo control, the control puls must be continuously repeated.

• Servo turns off when pulses stop

• For **Futaba servo motors**, the recommended interval between control pulses is 14 to 20 ms

• Servo Timing Signal

– Pulse width must be accurate in μs; otherwise servo exhibits **jitter**

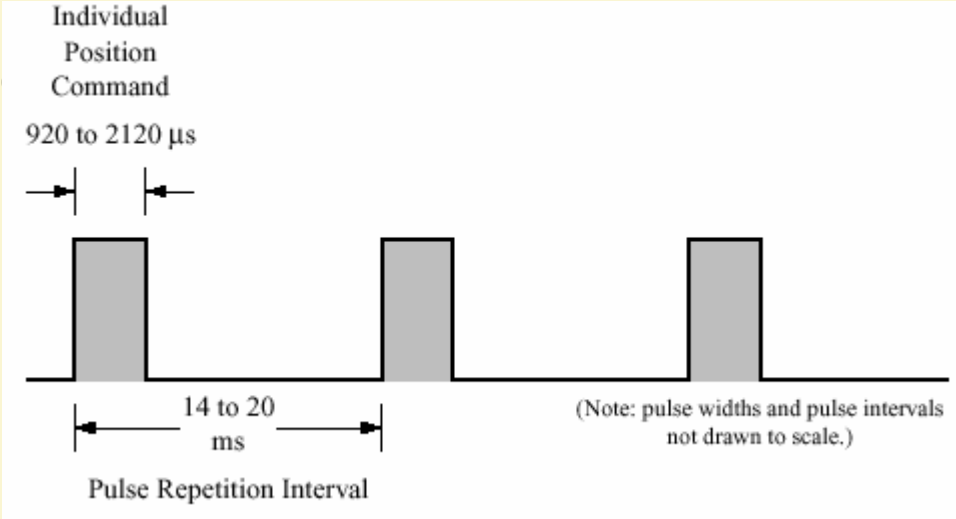
– Interval between pulses may vary 14 - 20 ms; successive pulses need not be exactly same distance apart

• Limits mechanical first, then electrical

– Electronics will try to drive output shaft to a point beyond mechanical limits

• **Do not plug servo motor in backwards!**

-Can cause damage to Servo and Handyboard.



To get the servo motor to continually attempt to reach the desired position, the timing pulse must be repeated at a regular interval

Experiment: find range of motion of different servo motors

Futaba S3003 Servo

Our Servo of choice has the following features:

•**SMT Contruction --**Surface Mount Technology improves reliability in applications where there is vibration.

•**Impact Resistant, Fuel Proof Case --** High impact plastic is designed to withstand minor crashes and the effects of model fuel.

•**Through-the-hole Circuit Board Plating --** Eliminates the need for wires or leads.

•**Heavy Duty Output Shaft --** Designed not to break, and offers versatility when installing the servo horn.

Stock No.	Servo	Torq/Spd 4.8V	Torq/Spd 6.0V	Bearing	Size	Weight (oz.)
FUTM0031	S3003 Standard	44/.23	57/.19	Top	1.6 x 0.8 x 1.4	1.3

Servo Motor

Exercises for your Servo Motor

1. Calculate the **duty cycle** of the servo control pulse, assuming dead center positioning and pulse intervals of 18 ms.

2. For each variety of servo motor that you have on hand, experimentally determine the **control pulse values** corresponding to the limits of rotary travel.

3. Write **wrapper functions** for setting the servo control globals and protecting the servo motors from out of bounds values.

4. Write a function to cause the servo motor to **“sweep”** its position back and forth.

5. With a servo that has been modified for **continuous rotation**, determine how far away from the center position the control signal needs to be to get the motor to run at its maximum speed. What does this tell you about the control function mapping the error signal into motor power?

IC with DC Motors and Servo Motors

Library Functions:

- Output Control
  - DC Motors
    - motors 0, 1, 2, 3
    - p=100 full on in fd direction
    - p=-100 full on in bk direction
  - Servo Motors

```
void fd (int m)
void bk (int m)
void off (int m)
void alloff()
void ao()
void motor(int m, int p)
```

```
fd((3);
bk(0);
off(1);
```

```
init_expbd_servos(1);  // Initializes servo, done FIRST
                       // Set to 0 to allow no pulses to be sent to servo

servo#(int pulse); // Sends pulse to servo, global is actually TWICE pulse in microsecs
                   // Replace # with specific expansion board port number (0-5)
```

Copyright Prentice Hall, 2001

From Servo Motor to Servo Winch

Continuous Rotation - "Winch" Servo





• Servo motor's output shaft rotates back and forth with a sweep of travel of about 180 degrees

• ***Winch servo*** rotates continuously

– Control signal specifies the speed and direction of rotation, rather than the desired angular position

– Useful for wide variety of applications, including robot's main drive motors

• Conversion:

– Feedback potentiometer is replaced by a pair of fixed resistors, which mimic the center position of the potentiometer; when the control signal deviates from center, the servo's control electronics drive the motor one way or the other in a vain attempt to get the servo to move away from center

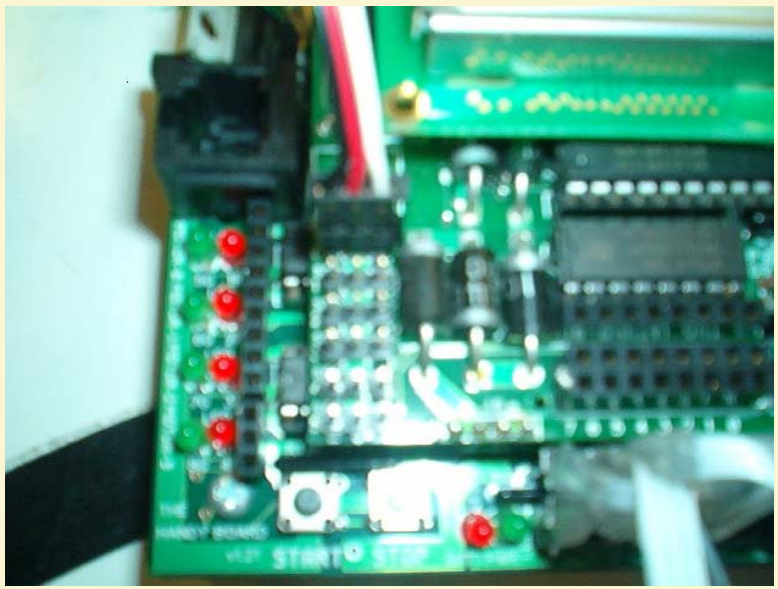
– Result is that the servo spins continuously with user-controllable speed and direction

• **Easier Method:**

– Remove the potentiometer and control components, leaving only the DC motor and gearing. This will allow the servo to be used as a DC motor with standard IC motor commands.

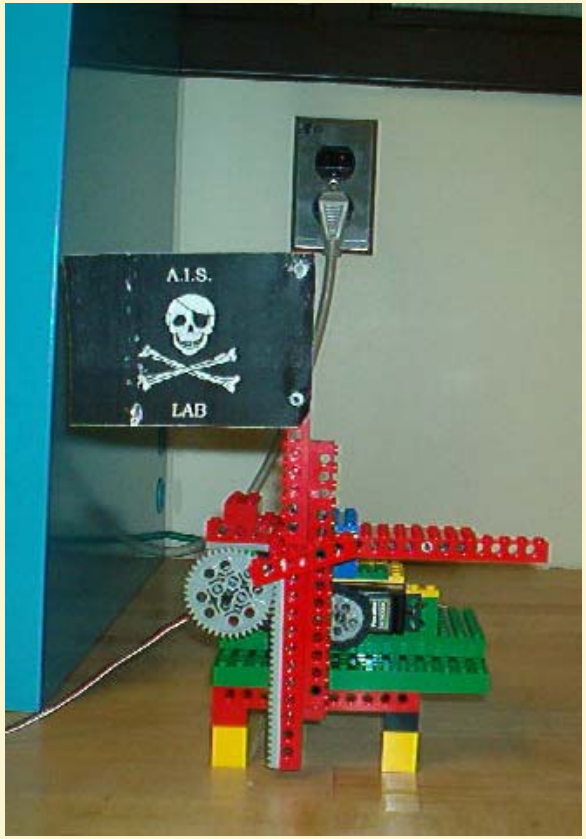
Servos and the Handyboard

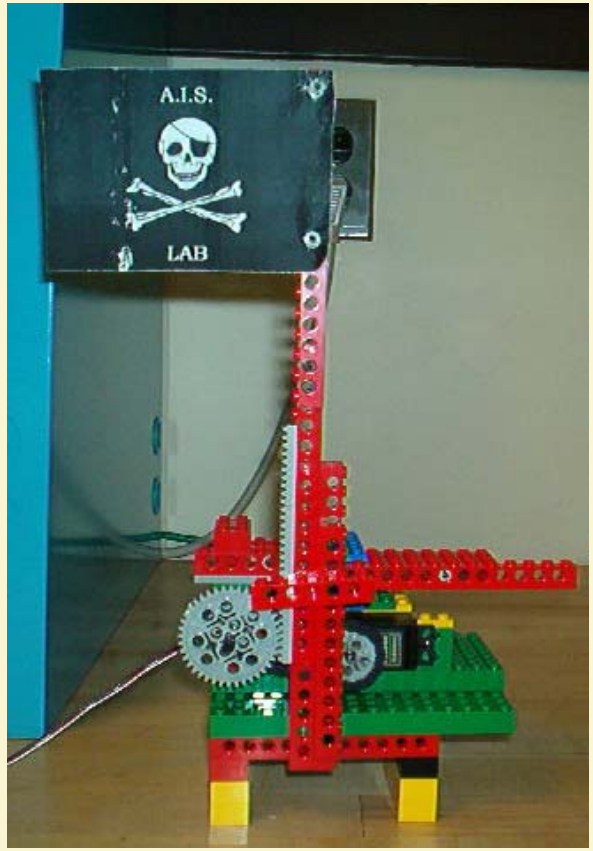
- There are 6 servo ports on the Handyboard expansion board. (Labeled from 0-5)
- Wire orientation as shown. From left to right: black, red and white



A Simple Example

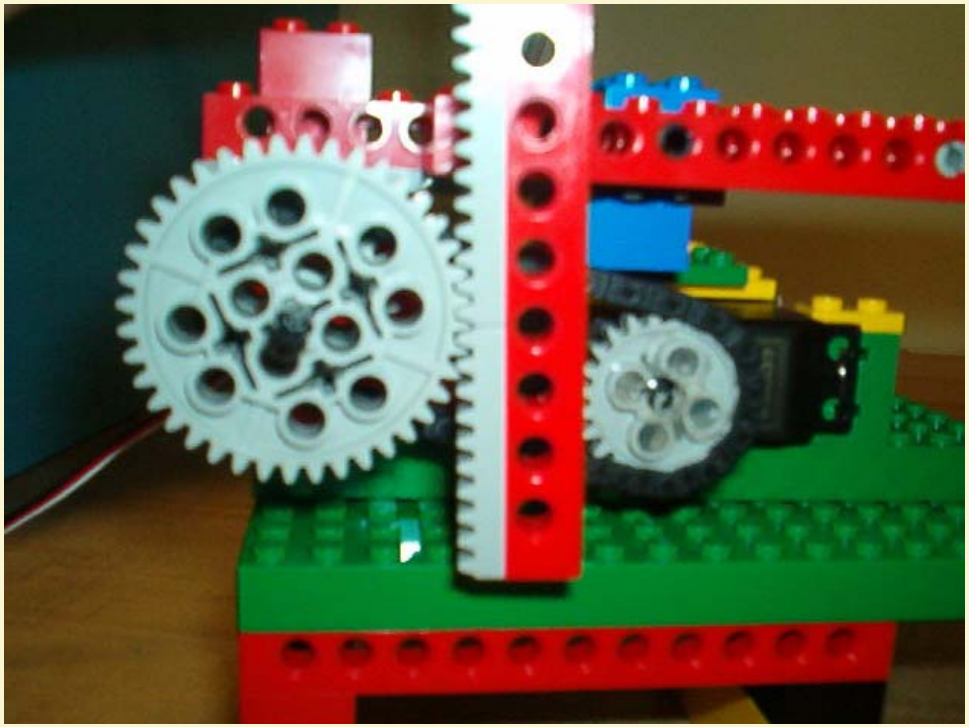
- Here is a simple example of servo control. This flag raising device is fully lowered when the servo shaft angle is 0 degrees, and fully raised when the servo shaft angle is 180 degrees.





A Simple Example

- In this example, we are converting a specified angular rotation into linear displacement. It is important to calculate the range of your output displacement based on the input servo shaft angles. This concept is applied in rack and pinion steering in RC vehicles.



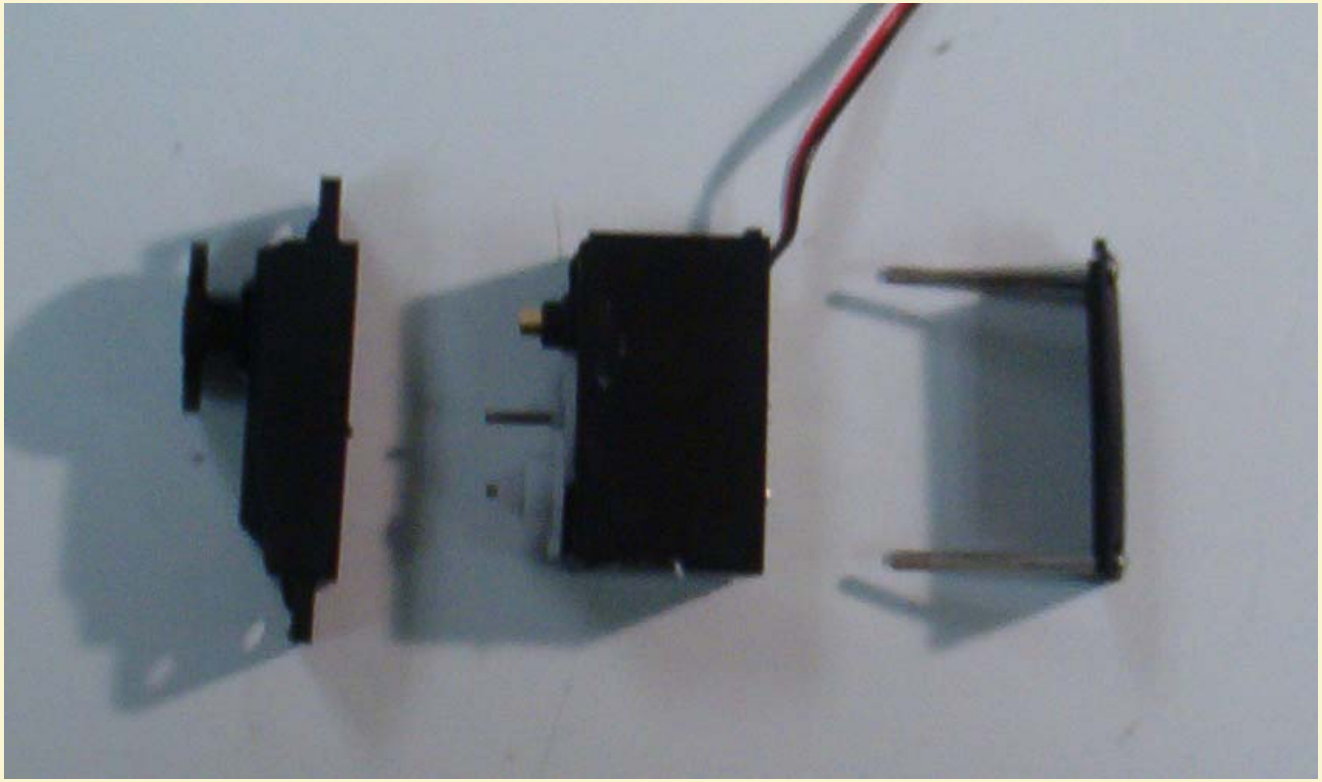
“Hacking” Your Servo

- How to create a “winch” servo from your stock Futaba servo.

- Benefit – Servos have very high gear reduction. The high output torque is very convenient for main drive motors.

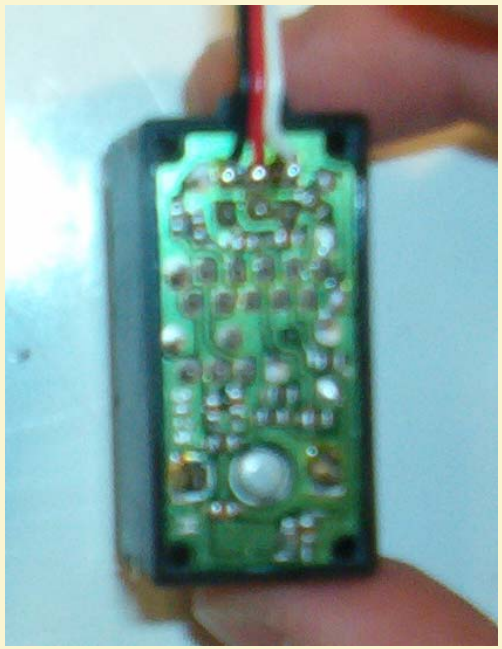
“Hacking” Your Servo

- **Step 1:** Disassemble servo: Take out four screws in the back of the servo and pull housing apart.



“Hacking” Your Servo

- **Step 2:** Remove servo potentiometer from housing. Portion of printed circuit board must be cut and removed. Remove enough of the PC board to expose the motor’s negative and positive leads. The rest of the PC board is not integral to the function of the motor. The remaining portion only serves to hold the DC motor in place.





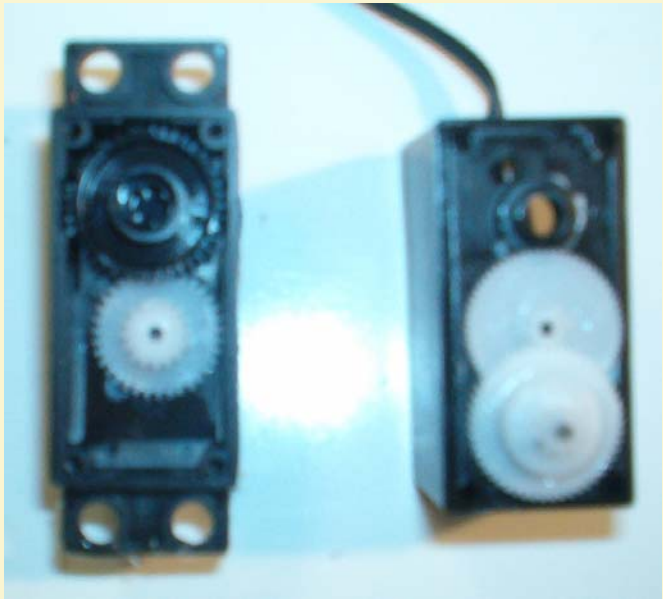
“Hacking” Your Servo

- **Step 3**: What is left is the DC Motor and gearing. DC Motor must be rewired. A motor connection must first be created (read Soldering Lecture to create connection). The connection will be soldered to the exposed leads on the DC motor remaining in the servo housing. It is not important which wire is soldered to which lead, since interchanging the positive and negative signal will only reverse the motor direction. Servo motor will now be used on the Handyboard DC motor ports, and will be used with standard IC motor functions:

```
void motor(int m, int p);
ao();
fd(int m);
bk(int m);
```

Gear Reduction

- **Exercise:** While you have your servo apart, take a look at the gear train. Gears can easily be pulled from the housing, but remember where they go when putting them back. Can you determine the total gear reduction of the system? (Hint: Get a magnifying glass and count the teeth!)



Servo Calibration

- No two servos are exactly the same.
- Each servo must be specifically calibrated to achieve desired angular positioning.

- **Step 1:** Choose a plate to attach to your servo.

  -The round plates are useful for attaching gears.

  -The spoked plates are useful for calibration (one spoke as reference)

  -All of these plates are useful for attaching sensors.



Servo Calibration

- **Step 2:** Use interaction window to input different pulse lengths to servo. Find pulse values that match (as closely as possible to the naked eye) to 0 and 180 degrees of motion.

- **Step 3:** Create an IC function to match a desired angular input (in degrees or radians)

Servo Calibration

- Example Function:

```
//servo0 function allows user to input rotation in degrees in IC
//200 Value corresponds to initial condition of 0 degree position
//3700 value corresponds to final 180 degree position

void servo0_deg(float degrees)
{
   servo0=(int)(degrees/180.*3700.+200.);
}
```

- Remember to use init_expbd_servos(1);

  to initialize servo!

Fixing The “Servo Jitter”

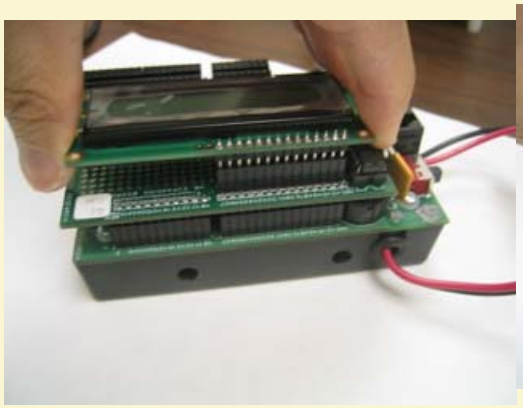
When servos are used with the Handyboard, they will tend to jitter (as in, repeatedly turn back and forth). This is due to a ground loop issue that must be resolved. Currently, the ground planes are at two different potentials, causing jitter as the device tries to get back to either ground potential.

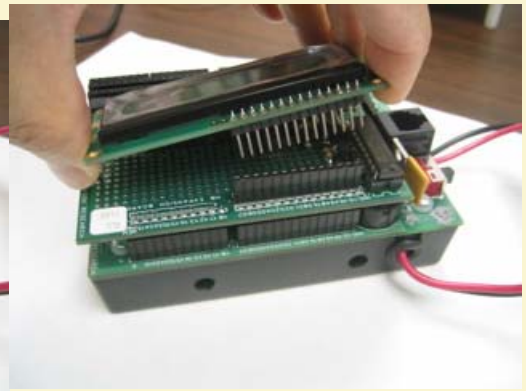
The following slides show a step by step procedure to rid yourself of the jitter problem.

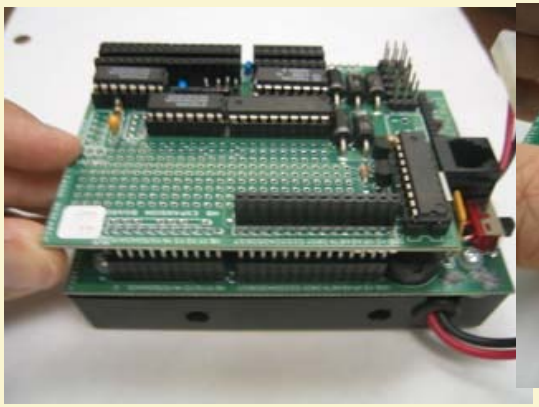
**An alternative method is currently under development to fix the jitter. This requires placing a capacitor of an appropriate size in an appropriate location. As the course progresses, the information pertaining to the jitter fix will be revised.*

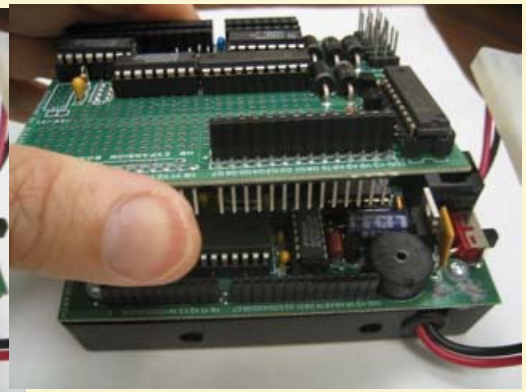
Fixing The “Servo Jitter”

Step 1: Remove display and expansion board



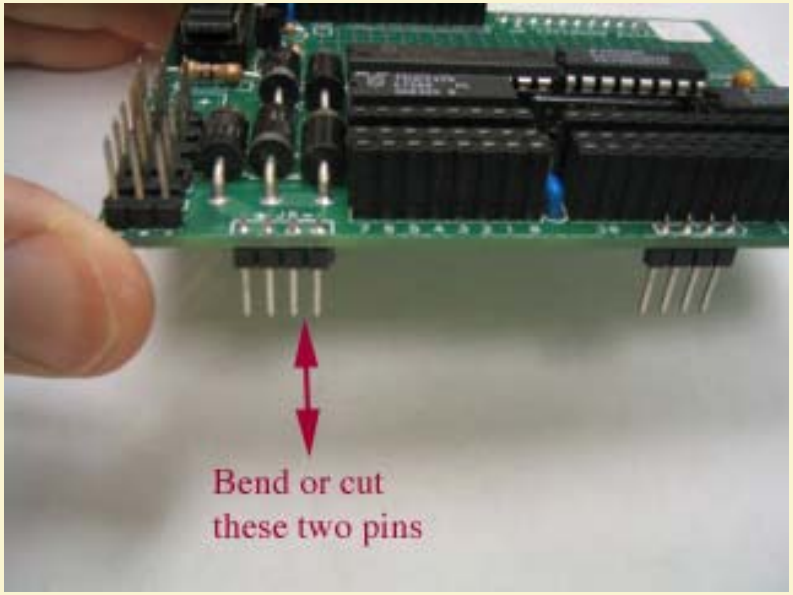


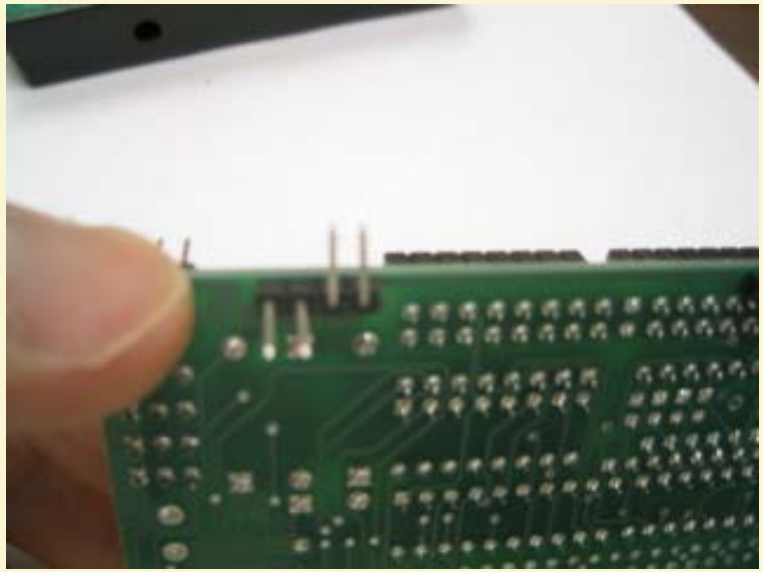




Fixing The “Servo Jitter”

Step 2: The ground potentials in question are the battery ground and the logic ground. An inductive element that isolates these two grounds must be bypassed. We must first remove the battery ground supply as shown below. Bend the two pins as shown on the expansion board. Be sure to ***bend*** the pins and not break them!

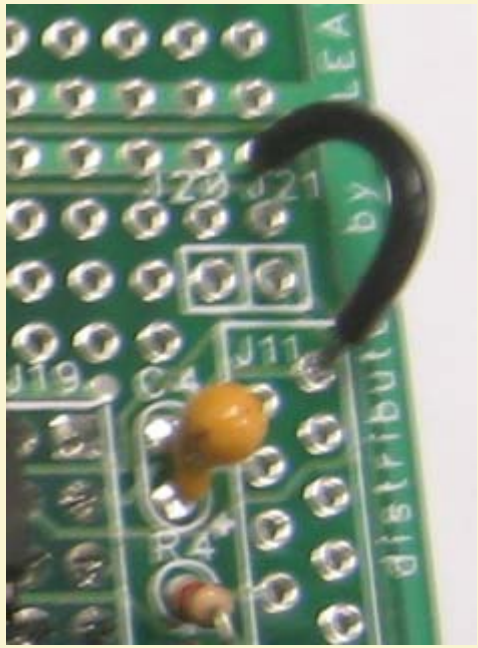


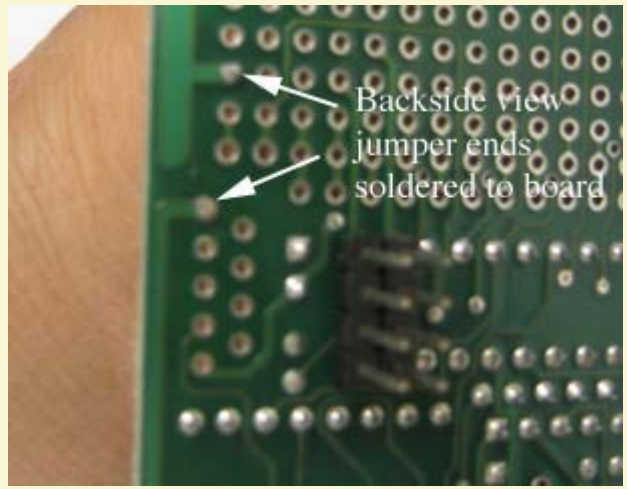


Fixing The “Servo Jitter”

Step 3: The last step is to jumper the servo ground directly to the logic ground. This is accomplished by soldering a small piece of wire to the expansion board as shown. The image on the right is the backside of the board.

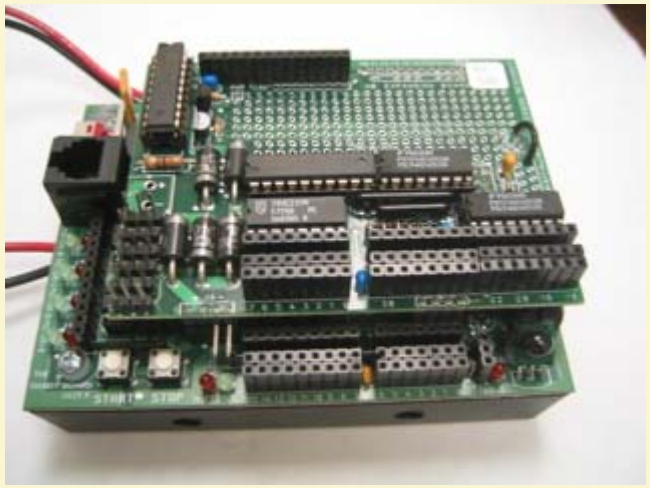






Fixing The “Servo Jitter”

Step 4: Simply reattach the expansion board to the Handyboard, then reattach the display.

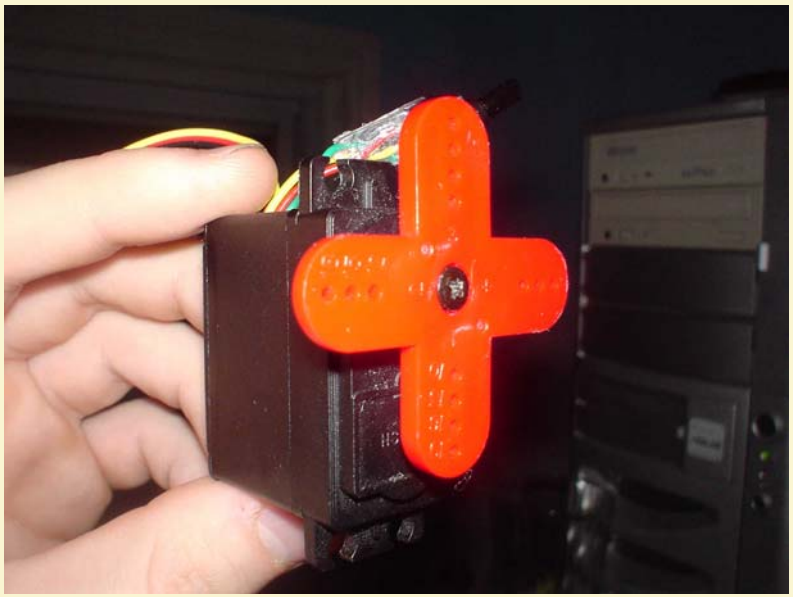




Rate Servos

Position Servos: The servos we have been dealing with up until now are designed to retrieve a pulse signal and move to a certain position.

Rate Servos: As position servos retrieve a signal and move to a respective position, rate servos spin a certain rate based on the signal. Rate servos are used where high torque is a must (they are generally low speed). Therefore, under large load conditions the rate will remain relatively the same as if the servo had no load attached. The rate servo we will be using is shown below.



Rate Servos

Programming the Rate Servo:

The rate servo can be calibrated similarly to the position servo. The only difference is that it will spin at a certain rate, in a certain direction, as opposed to go to a certain position.

First, take a look at the pod shown below. Turn the knob so that it is halfway in between it's available turning range.



Rate Servos

Programming the Rate Servo:

Next, experiment by sending pulses to the servo to find out what pulse values will cause the servo to spin full speed counterclockwise and full speed clockwise. Send a signal halfway between this value to the servo. With the pot in it's middle position as previously described, narrow down the signal so that there is NO SERVO MOVEMENT.

Calibrate using the position servo methods mentioned earlier in the lecture. Instead of referencing a certain angle in degrees, reference a range of speeds for the rate servo. Try –100 to +100, as this will correspond to the programming of the DC motors used in this course.

The pot in question is a zero-offset for the servo. The pot should be secured in this position (it can freely move about at first). This way, sending a range of pulses will consistently achieve an expected output rotation speed.