Arduino: Playground

The Software Servo Library can drive servos on all of your pins simultaneously. The API is patterned after the wiring.org servo library but the code is different. You are not limited to 8 servos, but you **must** call the Servo::refresh() method at least once every 50ms or so to keep your servos updating.

Standard Methods

attach(int)
  Turn a pin into a servo driver. Calls pinMode. Returns 0 on failure.
detach()
  Release a pin from servo driving.
write(int)
  Set the angle of the servo in degrees, 0 to 180.
read()
  return that value set with the last write().
attached()
  return 1 if the servo is currently attached.

Extra Methods

refresh()
  You must call this at least once every 50ms to keep the servos updated. You can call it as often as you like, it won't fire more than once every 20ms. When it does fire, it will take from .5 to 2.5 milliseconds to complete, but won't disable interrupts.
setMinimumPulse(uint16_t)
  set the duration of the 0 degree pulse in microseconds. (default minimum value is 544 microseconds)
setMaximumPulse(uint16_t)
  set the duration of the 180 degree pulse in microseconds. (default maximum pluse value is 2400 microsconds)

Safety Quirk

Even though you attach a servo, it won't receive any control signals until you send its first position with the write() method to keep it from jumping to some odd arbitrary value.

Size

The library takes about 850 bytes of flash and 6+(8*servos) bytes of SRAM.

The Code

You can find the code in http://www.arduino.cc/playground/uploads/ComponentLib/servo.zip. Unpack that into your lib/targets/libraries folder to add the library. *I forgot to paste in a license to those files, they are free for all uses. When I figure out how to replace an attachment in the wiki I'll put up a version with proper licenses.*

Limitations

This library does not stop your interrupts, so millis() will still work and you won't lose incoming serial data, but a pulse end can be extended by the maximum length of your interrupt handles which can cause a small glitch in the servo position. If you have a large number of servos there will be a slight (1-3 degrees) position distortion in the ones with the lowest angular values.

An Example

The following code lets you send strings like "90s" and "80w" to position servos on pin 14 and 15 (analog in 0 and 1) to 90 degrees and 80 degrees. You can also use "d" to detach the servo on pin 15 and "a" to reattach it. Ok, it is a silly program but it works for testing.
#include <Servo.h>

Servo servo1; Servo servo2;

void setup() {

```
   pinMode(1,OUTPUT);
   servo1.attach(14);
   servo1.setMaximumPulse(2200);
   servo2.attach(15);
   Serial.begin(19200);
   Serial.print("Ready");
```

}

void loop() {

```
   static int v = 0;
```

```
   if ( Serial.available()) {
     char ch = Serial.read();
```

```
     switch(ch) {
       case '0'...'9':
         v = v * 10 + ch - '0';
         break;
       case 's':
         servo1.write(v);
         v = 0;
         break;
       case 'w':
         servo2.write(v);
         v = 0;
         break;
       case 'd':
         servo2.detach();
         break;
       case 'a':
         servo2.attach(15);
         break;
     }
   }
```

```
   Servo::refresh();
```

}