Please ask about problems and questions regarding this tutorial on answers.ros.org (http://answers.ros.org). Don't forget to include in your question the link to this page, the versions of your OS & ROS, and also add appropriate tags.

Servo Controller Example

Description: Tutorial for controlling an R/C servo with rosserial and an Arduino

Tutorial Level:

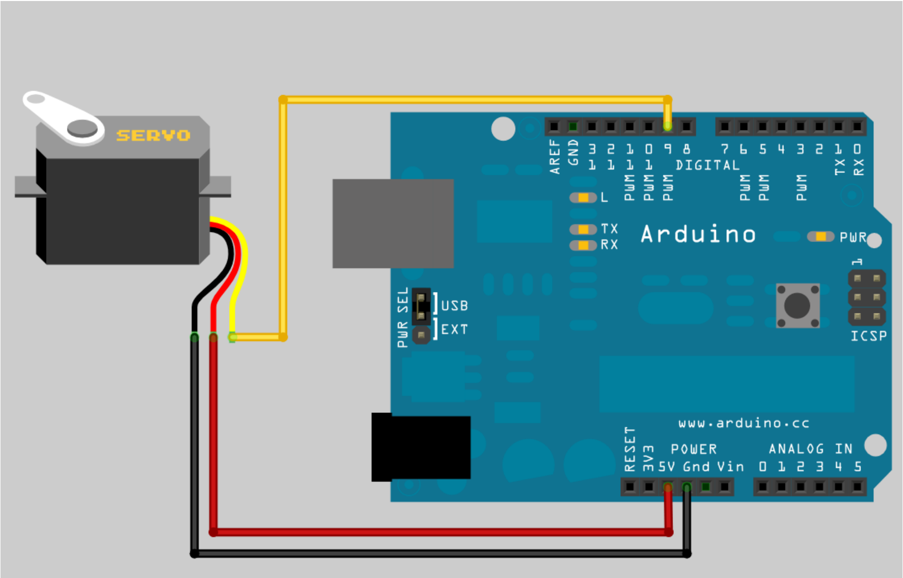
Next Tutorial: IR Ranger (/rosserial_arduino/Tutorials/IR%20Ranger)

electric | fuerte | groovy | hydro | indigo | jade | kinetic

Contents

This tutorial explains how to control an R/C servo through ROS by using an Arduino and rosserial. This can be used to control a release mechanism, a cheap robot arm, ROS powered biped, or anything where you need a cheap actuator. The code provided is a very basic example and shows the control of a single hobby servo.

1. Hardware



This example assumes that you have an Arduino and a hobby r/c servo. The r/c servo can be purchased from your local hobby shop, Towerhobbies (http://www3.towerhobbies.com/cgi-bin/wti0001p?&I=LXUK84&P=ML), Sparkfun (http://www.sparkfun.com/products/9064), etc.

The hobby servo r/c are great little actuators because they are relatively cheap (as low as $10) but contain a gear box and motor control electronics. They are controlled by sending a squarewave pulse of 1-2 milliseconds in width every 20 milliseconds. This typically moves the servo arm from 0-180 degrees. Hobby servo's come in a huge variety of sizes, torques, and angular precision.

2. Code

The code for this tutorial is made extremely simple through the use of the Arduino Servo library. The Servo Library (http://www.arduino.cc/en/Reference/Servo) handles all of the low level control to generate and maintain the servo pulses. All your code needs to do is specify the pin the servo is attached to and then write the angle to the servo object. Underneath, the Servo library uses the Arduino's built in timer interrupts to generate the correct pulses. In this example, we only control one servo, but the same library can be used to control up to 12 servos on most Arduino boards and 48 on the Arduino Mega.

Toggle line numbers

```
/*
 * rosserial Servo Control Example
 *
 * This sketch demonstrates the control of hobby R/C servos
 * using ROS and the arduiono
 *
 * For the full tutorial write up, visit
 * www.ros.org/wiki/rosserial_arduino_demos
 *
 * For more information on the Arduino Servo Library
 * Checkout :
 * http://www.arduino.cc/en/Reference/Servo
 */

#if defined(ARDUINO) && ARDUINO >= 100
  #include "Arduino.h"
#else
  #include <WProgram.h>
#endif

#include <Servo.h>
#include <ros.h>
#include <std_msgs/UInt16.h>

ros::NodeHandle  nh;

Servo servo;

void servo_cb( const std_msgs::UInt16& cmd_msg){
  servo.write(cmd_msg.data); //set servo angle, should be from 0-180
  digitalWrite(13, HIGH-digitalRead(13));  //toggle led
}


ros::Subscriber<std_msgs::UInt16> sub("servo", servo_cb);

void setup(){
  pinMode(13, OUTPUT);

  nh.initNode();
  nh.subscribe(sub);

  servo.attach(9); //attach it to pin 9
}

void loop(){
  nh.spinOnce();
  delay(1);
}
```

The key servo specific areas here are the fact that we made a global Servo object, attached to the correct arduino pin, and then on every servo topic call back, we write the servos new angle to the servo object.

3. Testing

First, startup your roscore and the rosserial python node in their own terminal windows.

```
roscore
```

Now, in a new terminal window, use rostopic pub to control your servo! Simply specify the angle from 0-180 and watch it move.

```
rostopic pub servo std_msgs/UInt16  <angle>
```

Wiki: rosserial_arduino/Tutorials/Servo Controller (last edited 2014-09-22 16:31:23 by AustinHendrix (/AustinHendrix))