



More ▾ Next Blog»

Create Blog Sign In

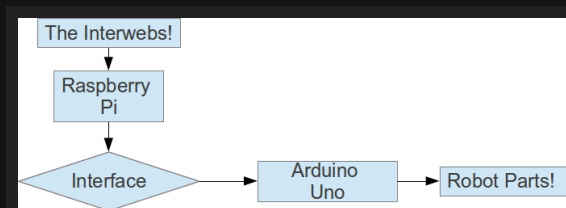
# Dave's Projects

Sunday, February 24, 2013

## WebServo

I recently decided it would be fun to create a robot which I could control over the web (while at my office or wherever), which I could use to play with (torment) my cat Lilu. There are a few steps leading up to the creation of this. While I am waiting for the parts to arrive I started working on the web interface.

I am using an **Arduino Uno** as my micro controller to control the motors and servos which will be on the robot. I found ways to control the uno over the internet, however, they either require the purchase of a breakout board or USB connection to another computer. I own a couple **Raspberry Pis** and figured I could use what I had already to do the job. While (I think) it is totally possible to use just Arduino code to do all of this and use the RPI as the communication device, I found it more convenient given my level of understanding to simply use the RPI GPIO pins as switches for input to the Uno to trigger motor and servo control. Essentially the device will work as follows:

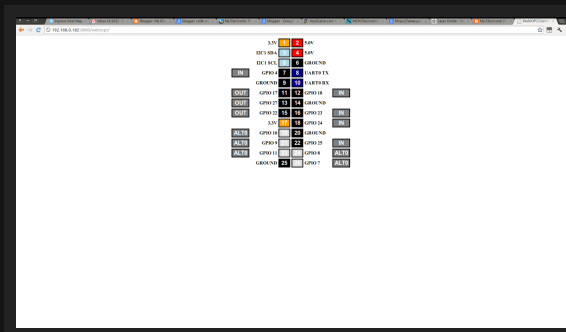


Web controlled robot from RPI and Uno

Since I am still (patiently) waiting for my parts. I will test my web based control using a single servo.

## Raspberry GPIO control over the web

A man by the name of Eric wrote up this wonderful framework called **WebIOPI**, which allows for web-based GPIO control using a python-based server. The installation and usage of this framework is very simple and can be found on his site. The best part of this framework is the ability to customize the UI using the WebIOPI API.



WebIOPI Interface

I thought it would be rather neat to control the device using keyboard input. That is, the left arrow and right arrows will control the motion of the servo. I mapped each keyboard input to it's own RPi GPIO pin, so while the key is pressed the pin is high, and when released the pin is low. This was simple using some jQuery functions along with the WebIOPI API.

### WebIOPI API Usage

```

1 //Use webiopi object to initialize pins
2 webiopi().read(function(){
3 //Init GPIO Pins 17 and 22
4   webiopi().setFunction(17,"OUT");
5   webiopi().setFunction(22,"OUT");
6 });
7
8 //Using the up(37) and down(39) arrows on pin 17,22.
9 //If key is pressed, pin goes high
  
```

### Blog Archive

▼ 2013 (4)

▼ February (4)

USB Webcam on the  
Raspberry Pi

Robot Parts!

WebServo

My Projects

### About Me



**Dave  
Nilosek**



Follow

170

View my  
complete  
profile

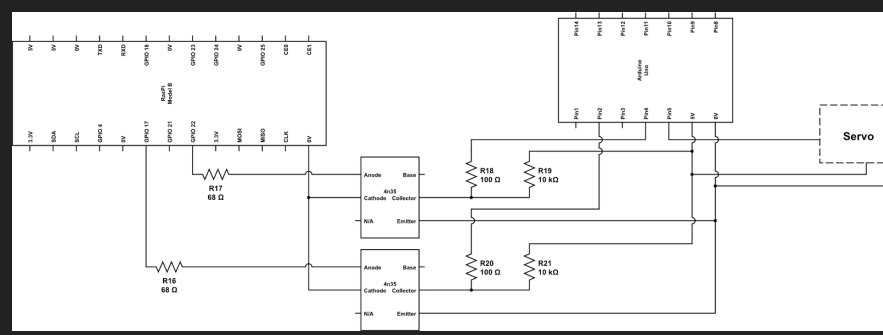
```

10     $(window).keydown(function(event){
11         if(event.keyCode == 37) setValue(17,1);
12         if(event.keyCode == 39) setValue(22,1);
13     });
14     //If key is released, pin goes low
15     $(window).keyup(function(event){
16         if(event.keyCode == 37) setValueK(17,0);
17         if(event.keyCode == 39) setValueK(22,0);
18     });
19     //Use REST API to set pin value
20     function setValueK(gpio, value){
21         $.post('/GPIO/' + gpio + "/value/" + value, function(data){
22             updateValue(gpio,value);
23         });
24     }
25     //Return data
26     function updateValueK(gpio, value){
27         var style = (value == 1) ? "HIGH" : "LOW";
28         $("#gpio"+gpio).attr("class",style);
29     }

```

## Interface circuit for RPi and Arduino

I decided to keep the RPi and Arduino circuits isolated as I feel this would be the safest approach to combining them. This means that they each have their own power sources, which may mean the robot has a little more weight and costs a bit more, however, it means I won't have to figure out how to run it all off of one power source without frying everything. I used an opto-isolator (4n35) as a switch to provide the input for the Arduino. Using the following circuit, when the RPi pin goes high, the Arduino pin will go low. This can be detected in the Arduino code to control the servo.



Circuit diagram (apologies for the lack of standards I am no electrical engineer)

## Arduino Sketch using RPi input

The Arduino sketch here is rather simple. If it detects pin 2 or 4 go low, then it will start to rotate the servo (which in this case is on pin 5). Pin 2 will rotate the servo in one direction, Pin 4 will rotate the servo in the opposite direction.

```

Arduino Sketch
1  #include <servo.h>
2
3  //Servo controller
4  Servo servoMain;
5
6  //Servo Definitions
7  int servoPin = 5;
8  int minP = 750;
9  int maxP = 2250;
10 int stepS = 3;
11 int delayTime = 20;
12
13 //Pin 2 definitions
14 int up = 2;
15 int valUp;
16 int bsUp;
17
18 //Pin 4 definitions
19 int down = 4;
20 int valDown;
21 int bsDown;
22
23 //Keep track of servo deg
24 int deg = 0;
25
26
27 void setup(){
28     //Init Pins
29     pinMode(up, INPUT);
30     pinMode(down, INPUT);
31
32     //Get initial reading
33     bsUp = digitalRead(up);
34     bsDown = digitalRead(down);
35
36     //Init servo
37     servoMain.attach(servoPin,minP,maxP);
38 }
39
40 //Loop
41 void loop(){

```

```

42 //Read pins
43 valUp = digitalRead(up);
44 valDown = digitalRead(down);
45
46 //Only act if changed (reduce calculations)
47 if ( (valUp != bsUp) || (valDown != bsDown) ){
48
49     //If pin 2 is low, add the step to the servo
50     if ( valUp == 0 ){
51         deg += stepS;
52         //Don't go past 180
53         if( deg >= 180){
54             deg = 180;
55         }
56
57         servoMain.write(deg);
58         delay(delayTime);
59     }
60     //If pin 4 is low, subtract the step from the servo
61     if ( valDown == 0 ){
62         deg -= stepS;
63         //Don't go past 0
64         if( deg <= 0){
65             deg = 0;
66         }
67
68         servoMain.write(deg);
69         delay(delayTime);
70     }
71 }
72 }

```

### Example:

In this example I also added two buttons to the page for testing.



Posted by Dave Nilosek at 8:21 PM



+4 Recommend this on Google

Labels: Arduino, Raspberry Pi, Servo

### 4 comments:



**felix dube** June 26, 2013 at 11:07 AM

*This comment has been removed by the author.*

Reply



**Claudiu Mirescu** November 3, 2013 at 12:39 AM

Excellent tutorial. I only have a question: where did you write the jScript code above? in what file? Thank you.

Reply

▼ Replies



**Dave Nilosek** November 3, 2013 at 6:53 AM

In the html file. Check out the WebIO demo to see their format. I basically just adjusted what they had.



**Claudiu Mirescu** November 3, 2013 at 7:27 AM

Did you do any modification to the py file? Whatever I do, it does not seem to work.

Reply

Enter your comment...

Comment as: Google Account ▼

Publish

Preview

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: Post Comments (Atom)

Awesome Inc. template. Powered by Blogger.