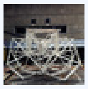
Arduino Bike Speedometer

by amandaghassaei on June 18, 2012

Table of Contents


Author:**amandaghassaei** amandaghassaei.com
Currently working for instructables!

Intro: Arduino Bike Speedometer

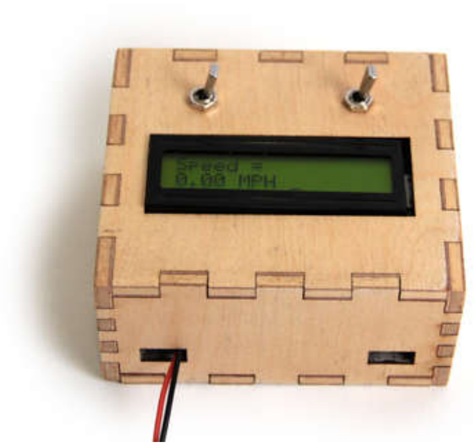
Monitor your road speed using the Arduino. This project uses a magnetic switch (also called a reed switch) to measure the speed of one of the bike's wheels. The Arduino calculates the mph, and send this information out to the LCD screen on the handlebars as you ride. It is compatible with any kind of bike/wheel, simply enter the radius of the wheel in the firmware to calibrate the device for your setup.

Parts List:

(1x) Arduino Uno REV 3 Radioshack 276-128
(1x) Switch-Magnetic Reed Radioshack 55050593
(1x) 10K Ohm 1/4-Watt Carbon Film Resistor Radioshack #271-1335
(1x) 9V Alkaline Battery Radioshack #23-866
(1x) Heavy-Duty 9V Snap Connectors Radioshack #270-324
(1x) PC Board with Copper Radioshack #276-147
(1x) Parallax 27977-RT Serial Backlit LCD Radioshack 276-120
(x2) SPST PC-Mountable Submini Toggle Switch Radioshack #275-645
(2x) Male Header Pins Jameco 103393
(1x) Female Pin Sockets Jameco 308567

Additional Materials:
22 Gauge Wire Radioshack #278-1224
Solder Radioshack #64-013
sand paper
plywood
wood glue
hot glue
screws
zip ties
sugru

Download Arduino IDE



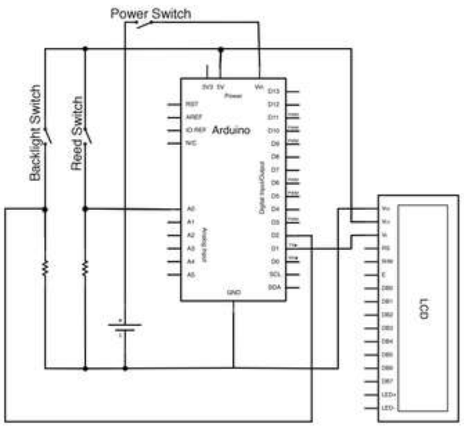
Step 1: Schematic

The schematic for this project is shown above.

It consists of three switches:
-one to connect to a 9V power supply
-one to switch the backlight of the LCD on and off
-a magnetic switch (called a reed switch) which closes each time the wheel completes one full rotation.

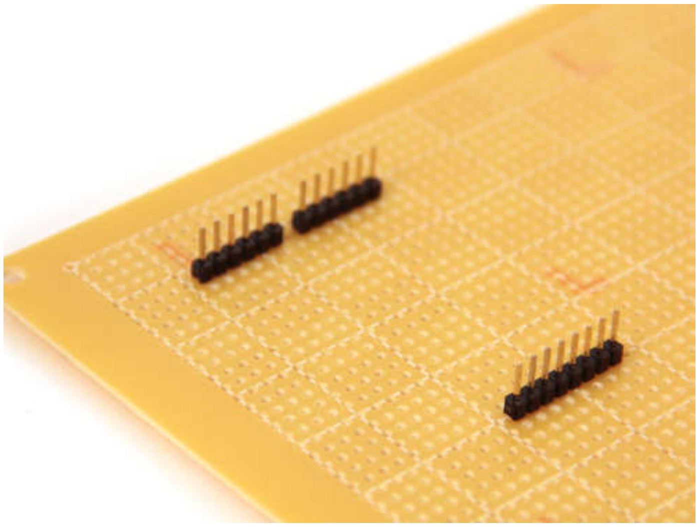
The Parallex LCD is designed to connect to the arduino using only three pins (ignore the labels and the other pins int his schematic). One to 5V, one to ground, and a third to serial out (TX)- on the arduino, serial out is digital pin 1.

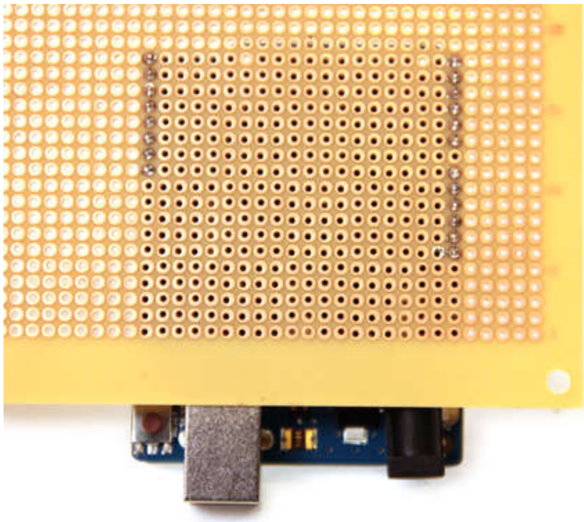
10kOhm resistors are connected to the reed and backlight switches to prevent excess current between 5V and ground (you should never directly connect 5V and ground on the arduino!)



Step 2: Protoboard

Solder three rows of header pins on the protoboard so that the arduino will snap to it as shown in the images above.

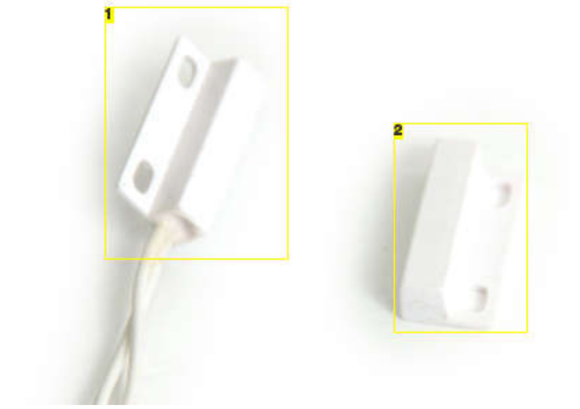




Step 3: Reed Switch

The reed switch is comprised of two pieces, a switch and a magnet. The switch has two wires extending out from it, when a magnet comes near the switch it causes a small mechanical piece to move and close the switch momentarily.

Solder a 10kOhm (current limiting) resistor between A0 and ground on the protoboard. Connect long pieces of stranded wire to A0 and 5V- these wires will wrap around the bike and attach to the reed switch.



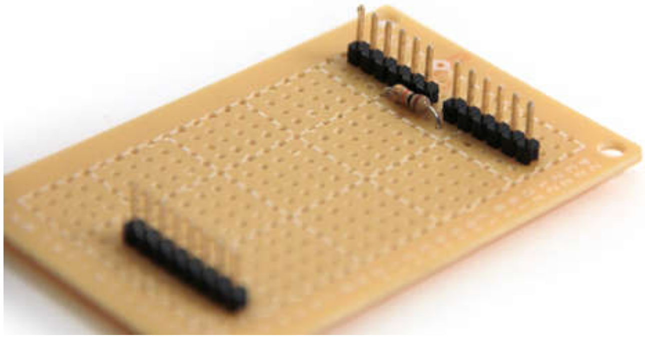
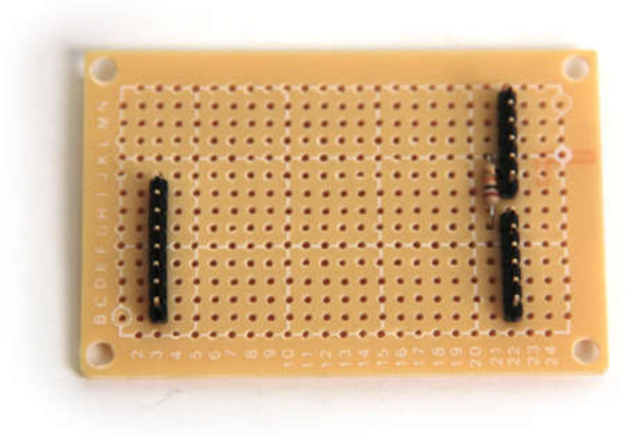
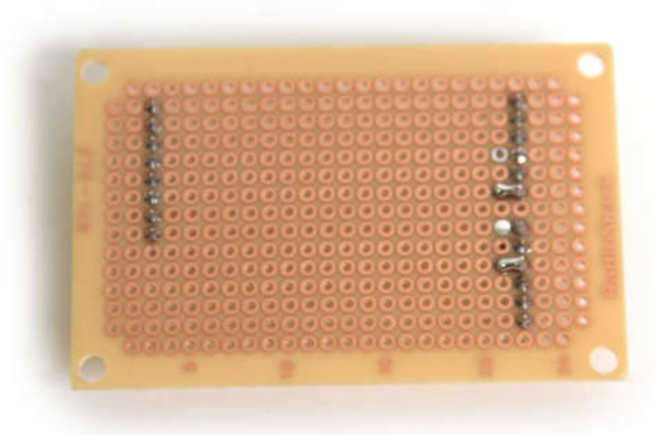


Image Notes
1. switch
2. magnet





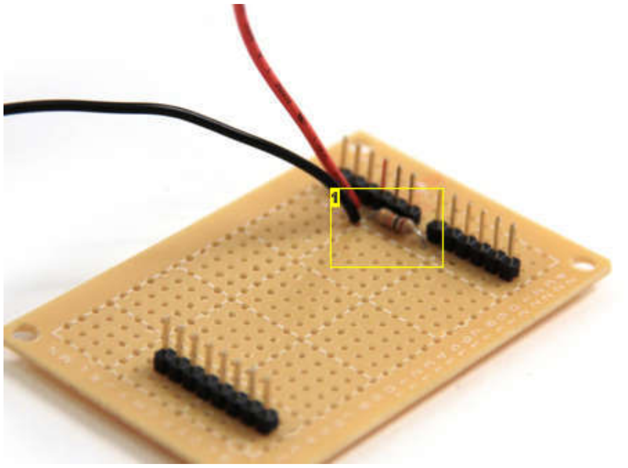


Image Notes
1. i made a mistake here- the black wire should be connected to A0

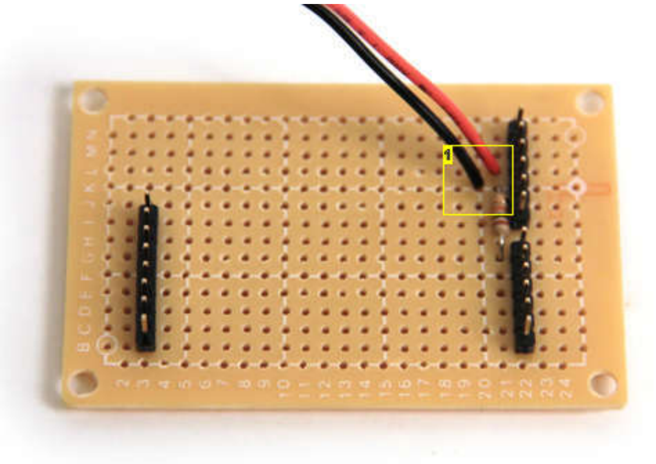


Image Notes
1. i made a mistake here- the black wire should be connected to A0

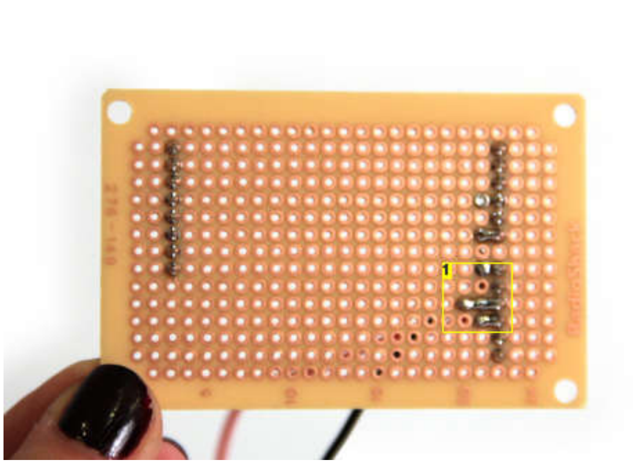
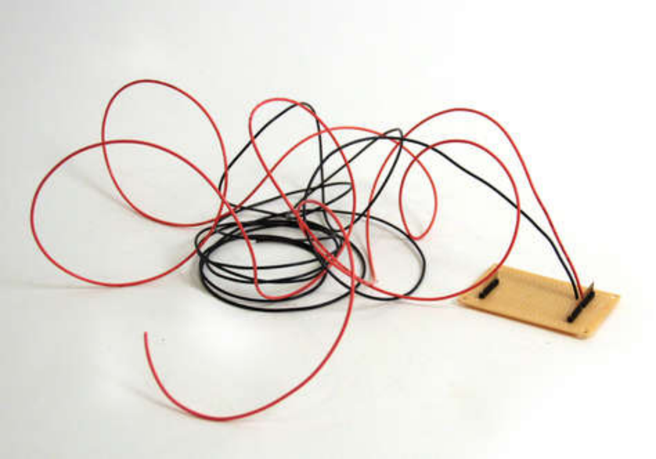


Image Notes
1. I made a mistake here- the black wire should be connected to A0



Step 4: Mount Reed Switch on Wheel

Secure both the magnet and reed switch to your bike wheel with electrical tape (either wheel is fine). As shown in the images above, the magnet connects to one of the tire spokes and the reed switch connects to the frame of the bike. This way, each time the bike wheel turns the magnet moves past the switch. Connect the leads form the reed switch to the long wires from your protoboard (orientation does not matter here- it's just a switch)

Use the code below to test your reed switch. When the magnet on the wheel moves past the switch, the arduino should print ~1023, otherwise it will print ~0. Open the serial monitor (Tools>>Serial Monitor) in Arduino IDE to test for your own setup. If the magnet does not seem to be affecting the reed switch, try repositioning it or even adding a stronger magnet if you have one.

```
//arduino bike speedometer w serial.print()
//by Amanda Ghassaei 2012
//http://www.instructables.com/id/Arduino-Bike-Speedometer/

/*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation; either version 3 of the License, or
* (at your option) any later version.
*
*/

#define reed A0//pin connected to read switch

//storage variable
int reedVal;

void setup(){
 Serial.begin(9600);
}


void loop(){
 reedVal = analogRead(reed);//get val of A0
 Serial.println(reedVal);
 delay(10);
}
```

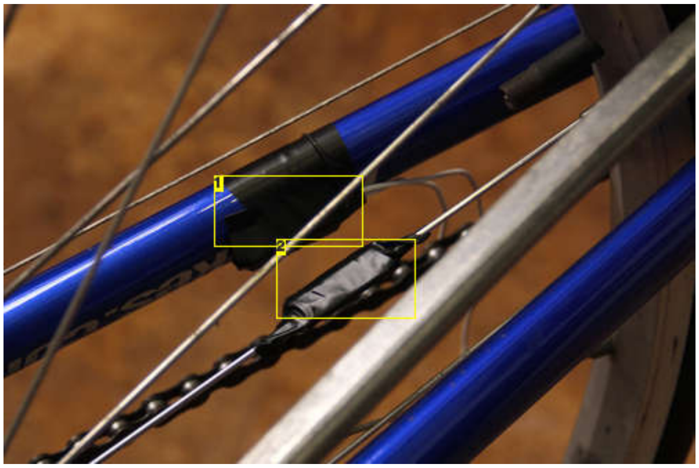


Image Notes
1. switch attached to bike frame
2. magnet attached to spoke

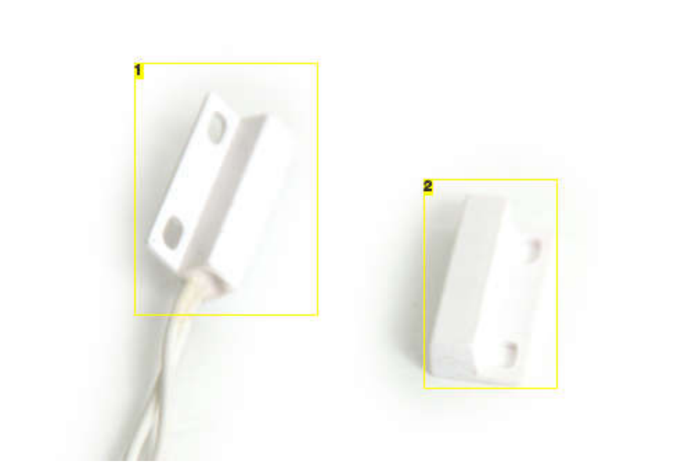
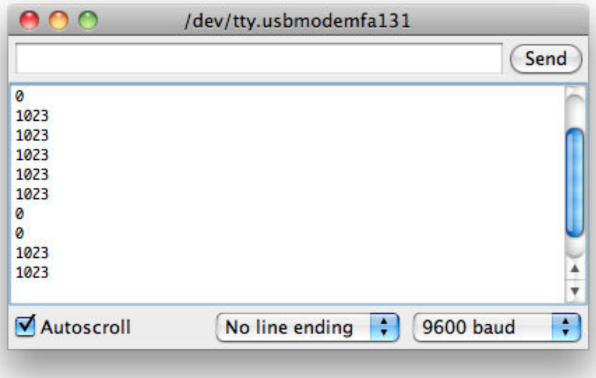


Image Notes

1. switch
2. magnet



```
0
1023
1023
1023
1023
1023
0
0
1023
1023
```

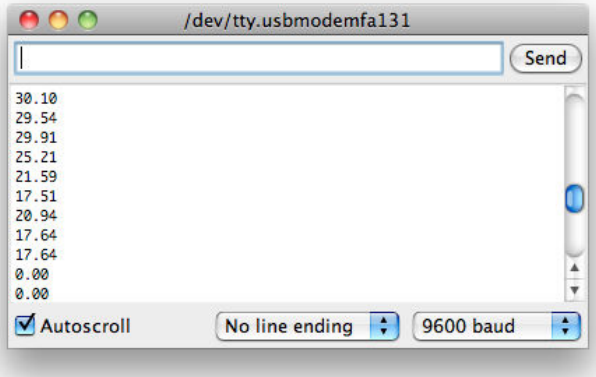
☑ Autoscroll   No line ending   9600 baud

Step 5: Test Switch

Load the code below onto the Arduino. Turn on the serial monitor. It should output 0.00. Start turning the bike wheel, you should see a print of the current mph each second.

```
//arduino bike speedometer w serial.print()
//by Amanda Ghassaei 2012
//http://www.instructables.com/id/Arduino-Bike-Speedometer/

/*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation; either version 3 of the License, or
* (at your option) any later version.
*
*/

//calculations
//tire radius ~ 13.5 inches
//circumference = pi*2*r =~85 inches
//max speed of 35mph =~ 616inches/second
//max rps =~7.25

#define reed A0//pin connected to read switch

//storage variables
int reedVal;
long timer;// time between one full rotation (in ms)
float mph;
float radius = 13.5;// tire radius (in inches)
float circumference;

int maxReedCounter = 100;//min time (in ms) of one rotation (for debouncing)
int reedCounter;


void setup(){

 reedCounter = maxReedCounter;
 circumference = 2*3.14*radius;
 pinMode(reed, INPUT);

 // TIMER SETUP- the timer interrupt allows precise timed measurements of the reed switch
 //for more info about configuration of arduino timers see http://arduino.cc/playground/Code/Timer1
 cli();//stop interrupts

 //set timer1 interrupt at 1kHz
 TCCR1A = 0;// set entire TCCR1A register to 0
 TCCR1B = 0;// same for TCCR1B
 TCNT1  = 0;
 // set timer count for 1khz increments
 OCR1A = 1999;// = (1/1000) / ((1/(16*10^6))*8) - 1
 // turn on CTC mode
 TCCR1B |= (1 << WGM12);
 // Set CS11 bit for 8 prescaler
 TCCR1B |= (1 << CS11);
 // enable timer compare interrupt
 TIMSK1 |= (1 << OCIE1A);

 sei();//allow interrupts
 //END TIMER SETUP

 Serial.begin(9600);
}


ISR(TIMER1_COMPA_vect) {//Interrupt at freq of 1kHz to measure reed switch
 reedVal = digitalRead(reed);//get val of A0
```

```
  if (reedVal){//if reed switch is closed
    if (reedCounter == 0){//min time between pulses has passed
      mph = (56.8*float(circumference))/float(timer);//calculate miles per hour
      timer = 0;//reset timer
      reedCounter = maxReedCounter;//reset reedCounter
    }
    else{
      if (reedCounter > 0){//don't let reedCounter go negative
        reedCounter -= 1;//decrement reedCounter
      }
    }
  }
  else{//if reed switch is open
    if (reedCounter > 0){//don't let reedCounter go negative
      reedCounter -= 1;//decrement reedCounter
    }
  }
  if (timer > 2000){
    mph = 0;//if no new pulses from reed switch- tire is still, set mph to 0
  }
  else{
    timer += 1;//increment timer
  }
}

void displayMPH(){
  Serial.println(mph);
}

void loop(){
  //print mph once a second
  displayMPH();
  delay(1000);
}
```

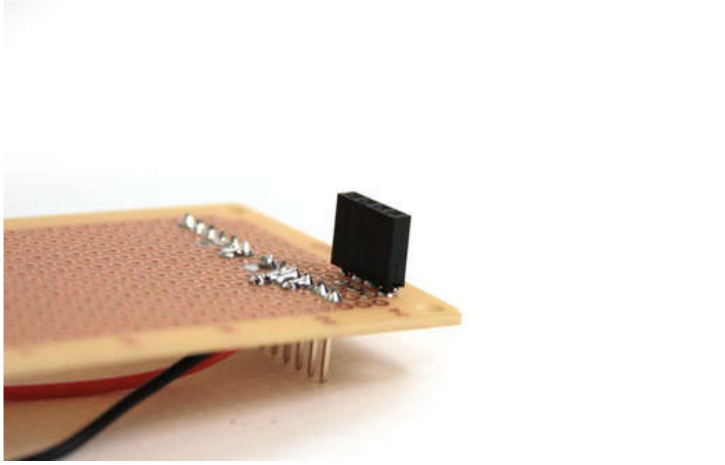


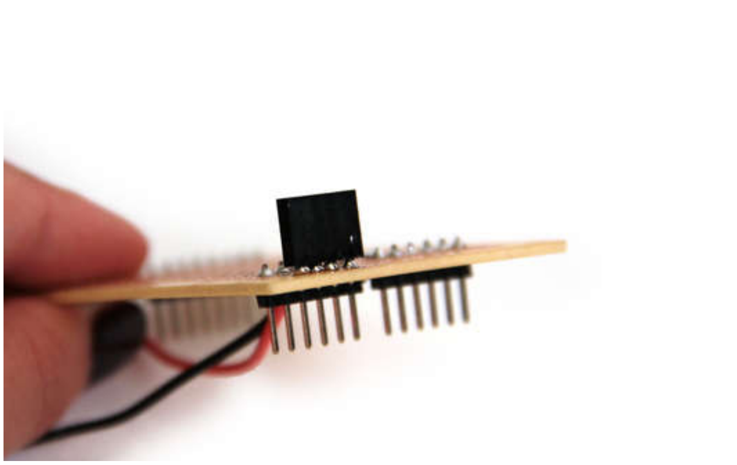
```
30.10
29.54
29.91
25.21
21.59
17.51
20.94
17.64
17.64
0.00
0.00
```

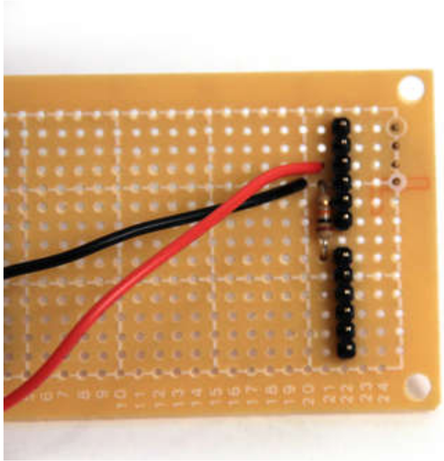
☑ Autoscroll   No line ending   9600 baud

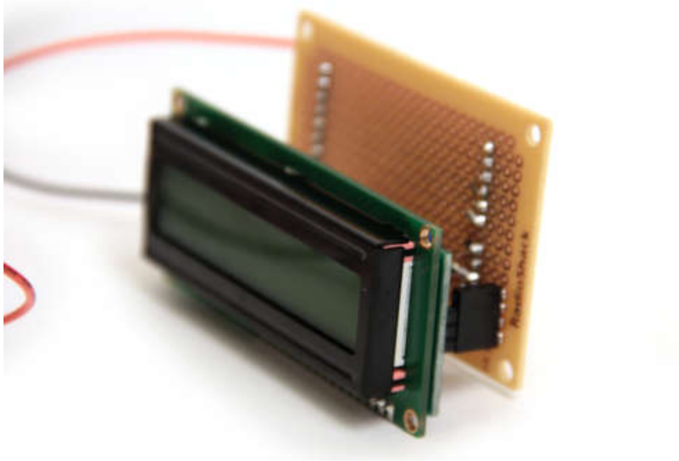
Step 6: LCD

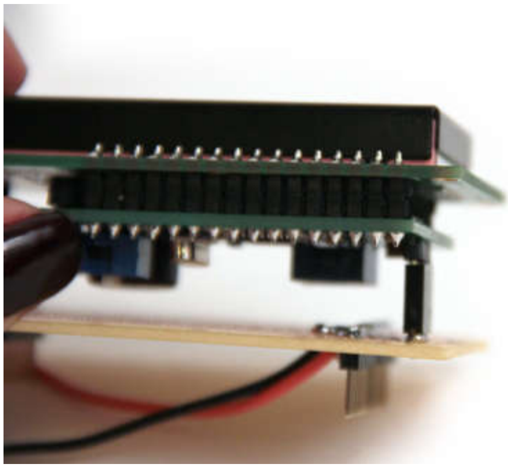
Solder a row of female header sockets on the copper side of the protoboard- three of these will be used to connect to the LCD screen. The LCD should fit nicely on top of the protoboard.





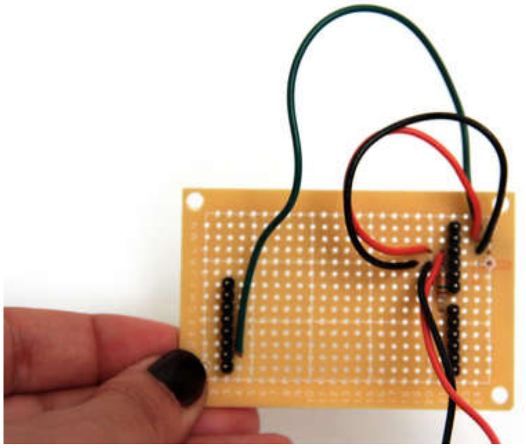


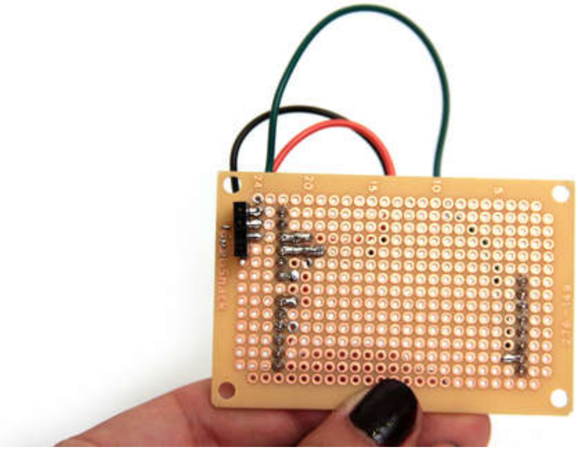




Step 7: Install Parallax LCD Library

Connect Arduino 5V, Ground, and TX (Arduino digital Pin 1) to the LCD socket. Read the labels on the LCD pins to make sure you have everything oriented correctly.





Step 8: Parallax LCD

The underside of the Parallax LCD has two switches and a potentiometer. The pot controls the contrast of the display- you can adjust this to what you like. The switches must be set as they are shown in the image above for proper functioning.

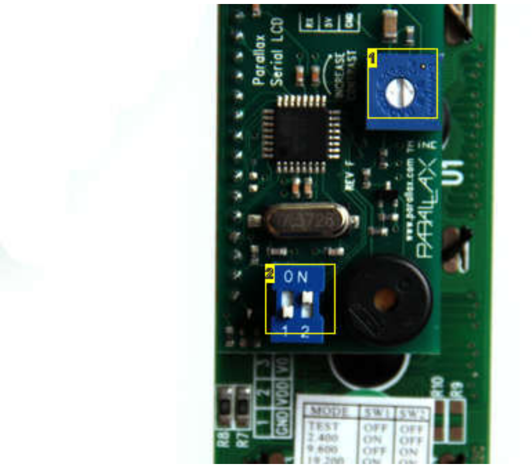


Image Notes
1. Adjust contrast
2. switches must be in this configuration to send messages to LCD

Step 9: Test LCD

Test the following code. For some reason my LCD starts making noise and displaying random characters when I first upload, but works fine once I unplug and reconnect the usb connection. I think this may have something to do will interference from the arduino communicating with the computer via digital pin 1 (TX) during the upload.

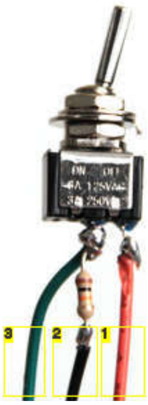
The LCD should display "Hello World" when it is turned on.

```
//test of parallax 2x16 lcd
//by Amanda Ghassaei 2012
//http://www.instructables.com/id/Arduino-Bike-Speedometer/

/*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation; either version 3 of the License, or
* (at your option) any later version.
*
*/

//this code should print "Hello World" on the LCD and the backlight switch connected to digital pin 2 should work


//Serial.write(13);//start a new line

void setup() {
 Serial.begin(9600);
 pinMode(1,OUTPUT);//tx

 Serial.write(12);//clear
 Serial.write("Hello World");
}

void loop() {
}
```

Step 10: Backlight Switch

Wire a switch as shown in the image above. Connect a 10kOhm resistor and a green wire to one lead, and a red wire to the other.

Connect the red wire to Arduino 5V, the other side of the resistor to ground, and the green wire to D2.



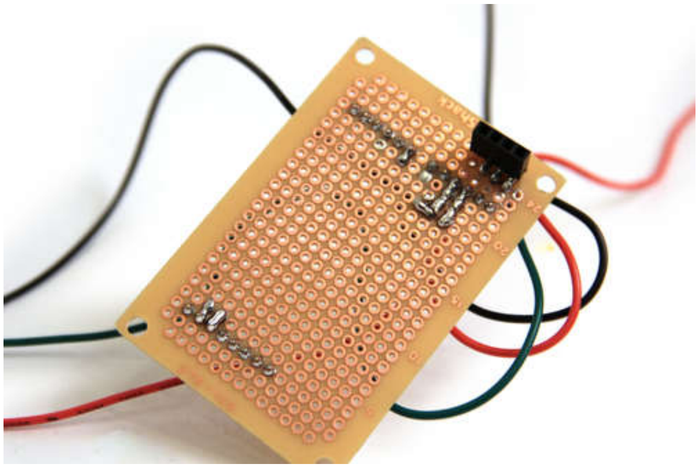


Image Notes
1. to 5V
2. to ground
3. to digital pin 2

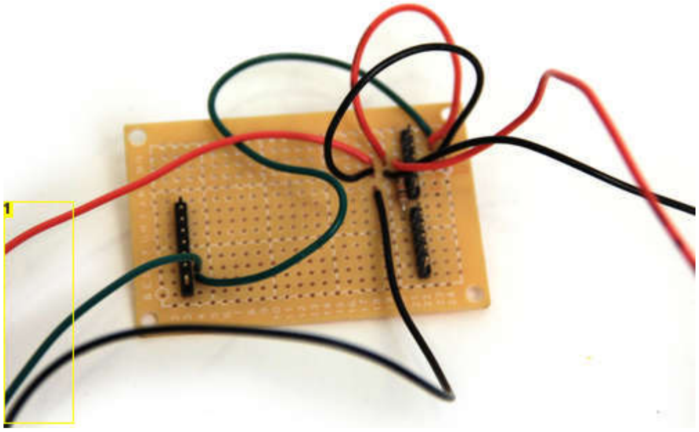


Image Notes
1. to switch

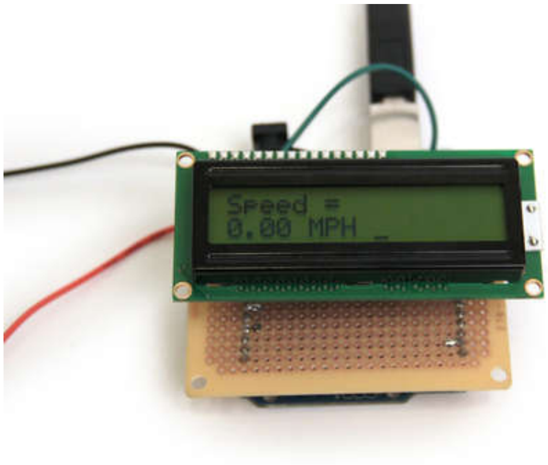
Step 11: Final Speedometer Code

Upload the following code onto the Arduino. Test to make sure the backlight switch works and the speed displays properly. (Again, you may have to unplug the board after loading the firmware and plug it back in again to get it to work properly.)

Measure the radius of your tire wheel (in inches) and insert it in the line: float radius = "";

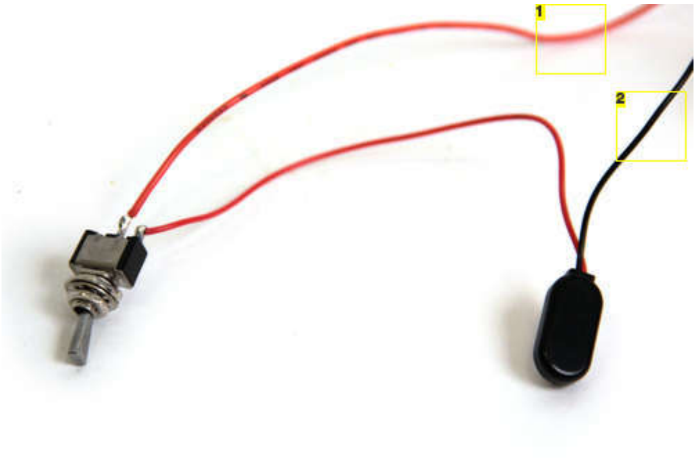
```
//bike speedometer
//by Amanda Ghassaei 2012
//http://www.instructables.com/id/Arduino-Bike-Speedometer/

/*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation; either version 3 of the License, or
* (at your option) any later version.
*
*/

//outputs speed of bicycle to LCD

//calculations
//tire radius ~ 13.5 inches
//circumference = pi*2*r =~85 inches
//max speed of 35mph =~ 616inches/second
//max rps =~7.25

#define reed A0//pin connected to read switch

//storage variables
```

```
float radius = 13.5;// tire radius (in inches)- CHANGE THIS FOR YOUR OWN BIKE

int reedVal;
long timer = 0;// time between one full rotation (in ms)
float mph = 0.00;
float circumference;
boolean backlight;

int maxReedCounter = 100;//min time (in ms) of one rotation (for debouncing)
int reedCounter;


void setup(){

 reedCounter = maxReedCounter;
 circumference = 2*3.14*radius;
 pinMode(1,OUTPUT);//tx
 pinMode(2,OUTPUT);//backlight switch
 pinMode(reed, INPUT);

 checkBacklight();

 Serial.write(12);//clear

 // TIMER SETUP- the timer interrupt allows preceise timed measurements of the reed switch
 //for mor info about configuration of arduino timers see http://arduino.cc/playground/Code/Timer1
 cli();//stop interrupts

 //set timer1 interrupt at 1kHz
 TCCR1A = 0;// set entire TCCR1A register to 0
 TCCR1B = 0;// same for TCCR1B
 TCNT1  = 0;
 // set timer count for 1khz increments
 OCR1A = 1999;// = (1/1000) / ((1/(16*10^6))*8) - 1
 // turn on CTC mode
 TCCR1B |= (1 << WGM12);
 // Set CS11 bit for 8 prescaler
 TCCR1B |= (1 << CS11);
 // enable timer compare interrupt
 TIMSK1 |= (1 << OCIE1A);

 sei();//allow interrupts
 //END TIMER SETUP

 Serial.begin(9600);
}

void checkBacklight(){
 backlight = digitalRead(2);
 if (backlight){
   Serial.write(17);//turn backlight on
 }
 else{
   Serial.write(18);//turn backlight off
 }
}

ISR(TIMER1_COMPA_vect) {//Interrupt at freq of 1kHz to measure reed switch
 reedVal = digitalRead(reed);//get val of A0
 if (reedVal){//if reed switch is closed
   if (reedCounter == 0){//min time between pulses has passed
     mph = (56.8*float(circumference))/float(timer);//calculate miles per hour
     timer = 0;//reset timer
     reedCounter = maxReedCounter;//reset reedCounter
   }
   else{
     if (reedCounter > 0){//don't let reedCounter go negative
       reedCounter -= 1;//decrement reedCounter
     }
   }
 }
 else{//if reed switch is open
   if (reedCounter > 0){//don't let reedCounter go negative
     reedCounter -= 1;//decrement reedCounter
   }
 }
 if (timer > 2000){
   mph = 0;//if no new pulses from reed switch- tire is still, set mph to 0
 }
 else{
   timer += 1;//increment timer
 }
}

void displayMPH(){
 Serial.write(12);//clear
 Serial.write("Speed =");
 Serial.write(13);//start a new line
 Serial.print(mph);
 Serial.write(" MPH ");
 //Serial.write("0.00 MPH ");
}

void loop(){
 //print mph once a second
 displayMPH();
 delay(1000);
 checkBacklight();
}
```



Step 12: Battery

Wire the battery connector and switch in series as shown in the first image above. Connect the read lead from the switch to Arduino Vin and the black wire from the battery connector to Arduino ground.



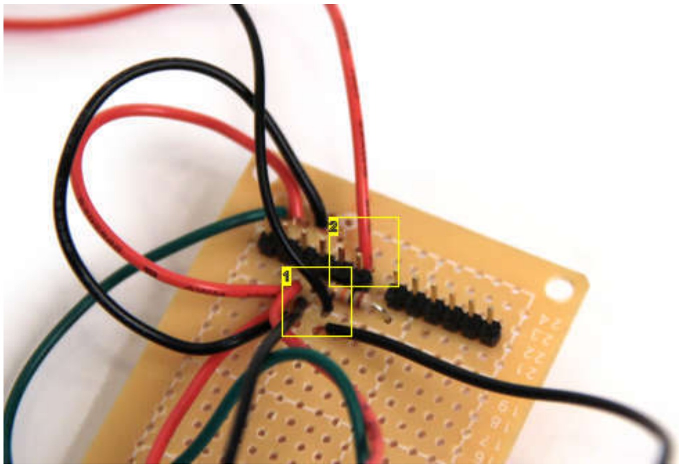


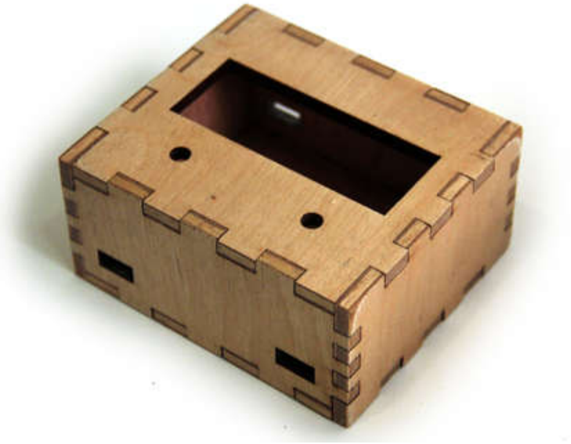
Image Notes
1. to arduino Vin
2. to arduino ground

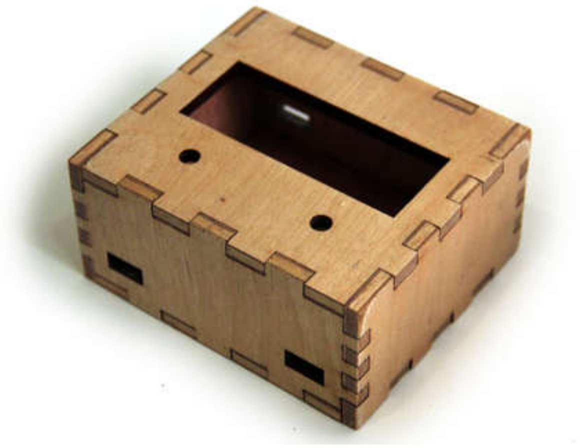
Image Notes
1. black lead of battery connect to arduino ground
2. read lead from switch connects to arduino Vin

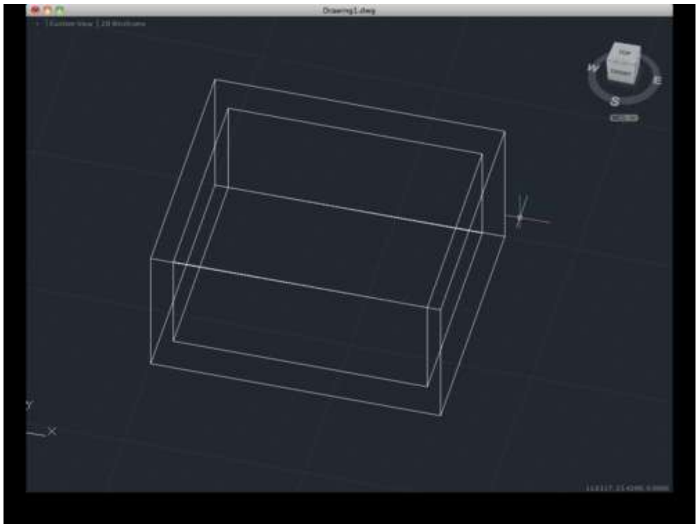
Step 13: Enclosure

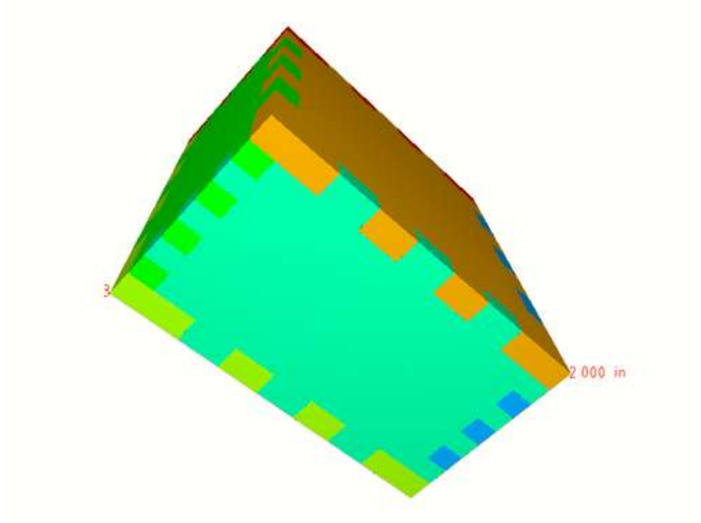
I cut my project enclosure from 1/4" ply on an epilog 120Watt laser cutter. The dimensions of the enclosure are 3.5"x4"x2". I modeled the box in AutoCAD and generated the laser cut files (with finger joints) in Autodesk 123D Make. Then I added two holes for the switches and a rectangular opening for the LCD screen. I also added some holes on the bottom of the enclosure to make attaching it to the bike easier.

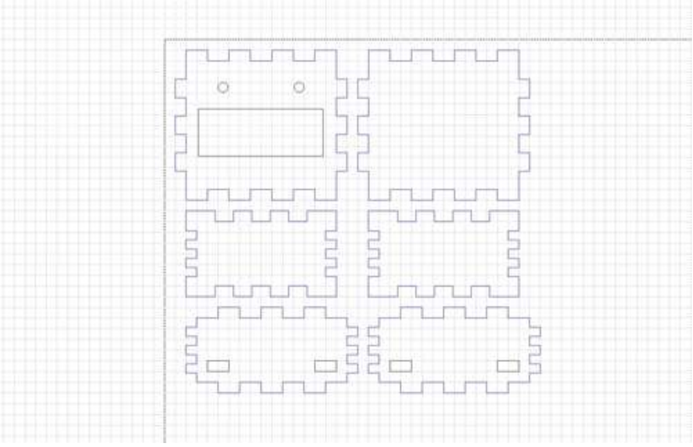
I glued the project enclosure together with wood glue and sanded the edges down. I finished the enclosure with some clear polycrylic.











File Downloads

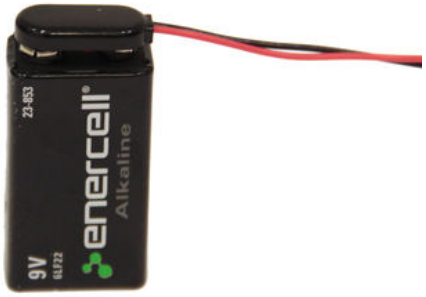
enclosure.dwg (229 KB)
[NOTE: When saving, if you see .tmp as the file ext, rename it to 'enclosure.dwg']

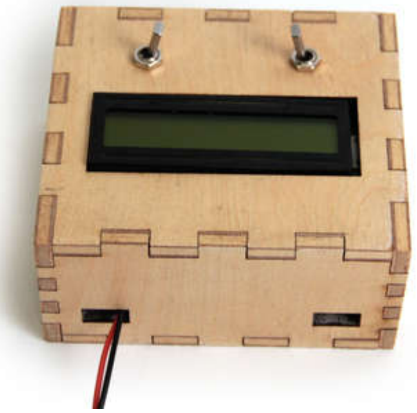
enclosure.stl (1 KB)
[NOTE: When saving, if you see .tmp as the file ext, rename it to 'enclosure.stl']

enclosure.cdr (14 KB)
[NOTE: When saving, if you see .tmp as the file ext, rename it to 'enclosure.cdr']

Step 14: Install Components in Enclosure

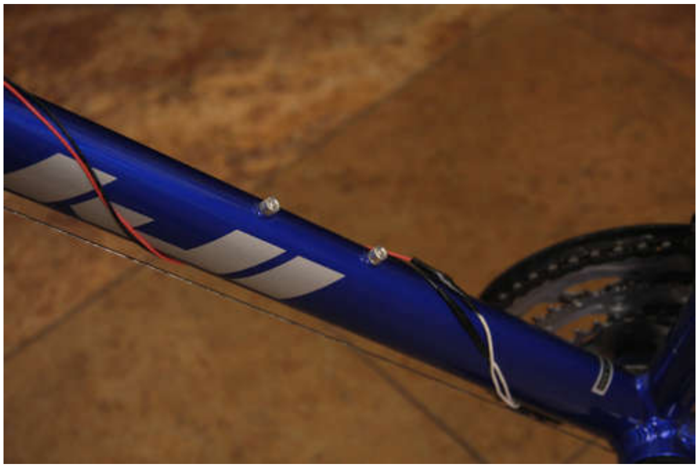
Secure the switches onto the enclosure with nuts. Glue or screw the lcd to the underside of the front panel. Fit the Arduino and Protoboard as well as the battery into the enclosure and secure with velcro or glue. Screw or fasten the enclosure shut.

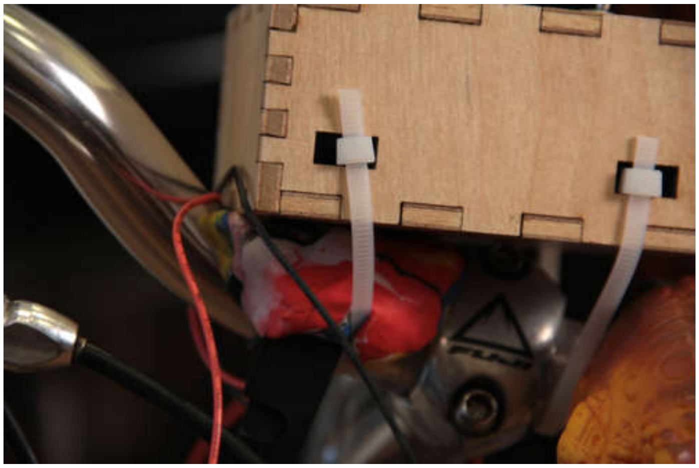


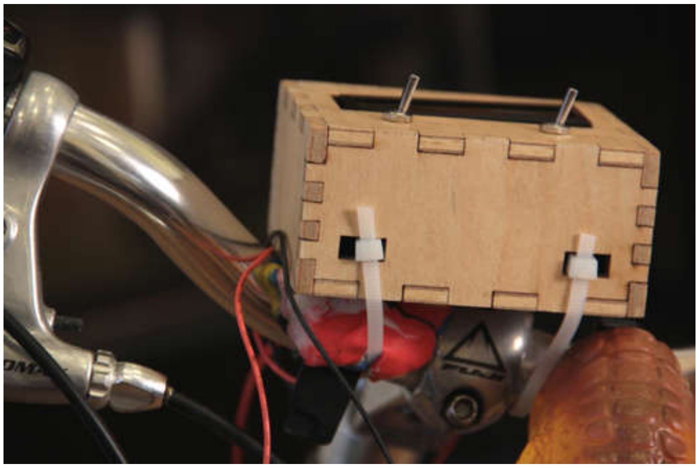


Step 15: Attach to Bike

Wrap the reed switch wires around the bike frame, away from any moving bike parts. I used sugru and some zip ties to attach the speedometer to the handle bars.

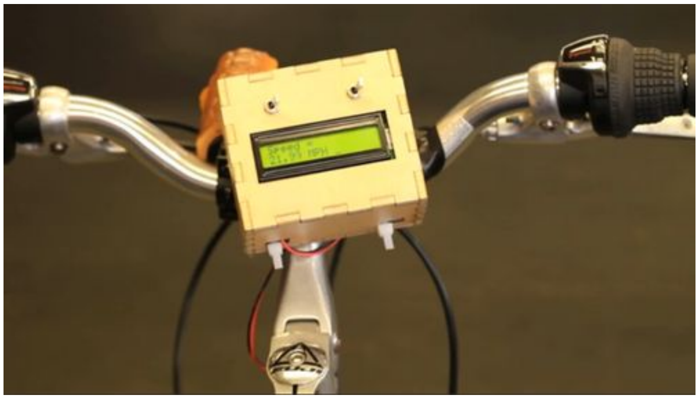






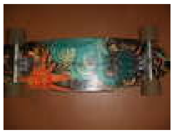
Step 16: Take it Out on the Road

You should be ready to hit the road. Don't let the awesomeness of your new bike speedometer distract you from road hazards!

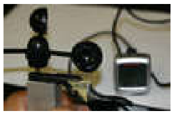


Related Instructables

Wireless Altoids Cycle Computer by Alexdlp



Arduino Skateboard Speedometer by leonardor



Digital Anemometer (wind meter) by dan

Arduino Timer Interrupts by amandaghassaei

Amp up your VW Bug Speedometer by kcrichar

Arduino Bicycle Alarm and Lights by cheeyah