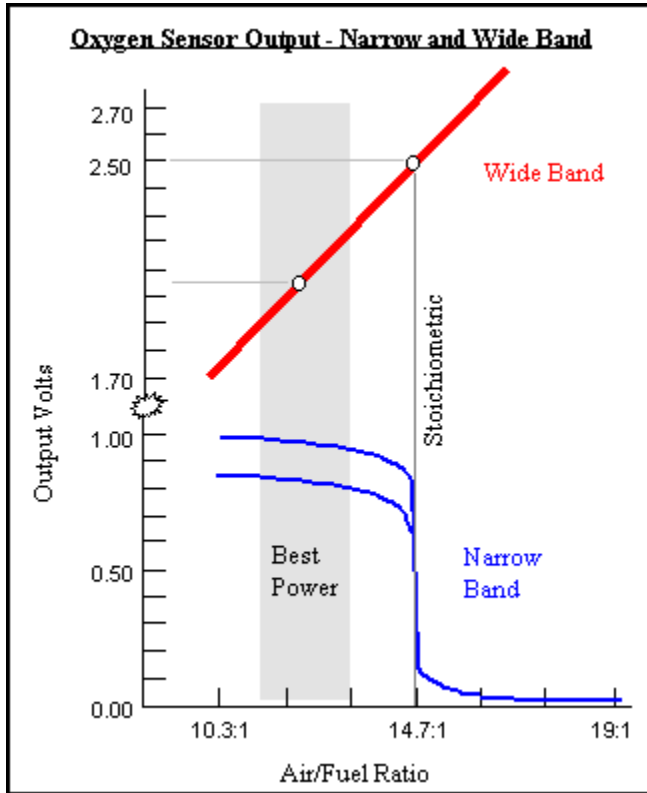


**Simulated Wideband Oxygen Sensor
Using the Arduino Board.**

**Dr. Pushkin
EE290
Timothy Pratt
28 – April - 09**

Simulated Wideband Oxygen Sensor



A typical narrow band oxygen sensor is only good for telling whether or not your Air Fuel Ratio is at 14.7:1 which is the stoichiometric ratio at which gasoline burns most efficiently in air. Unfortunately for car guys this is not the ratio at which engines make the most power that is around 12.5:1 which is a richer (more fuel being burnt) mixture. As can be seen from the graph there is a sharp drop with a large varying of voltages at stoich, and at 12.5 there is a very short range. And that depends on the temperature of the exhaust gas even with a heated oxygen sensor. To give a voltage the oxygen sensor switched back and forth between a higher and a lower voltage then the value and that average is the A/F level that you have.

What I am doing is running the voltage of the oxygen sensor through a series of Op-Amps that uses no current, takes the voltage of the oxygen sensor and multiplies it by 5 (to simulate the wideband), and finally reverses the voltage back to a positive voltage. Because I have stretched the voltage of the oxygen sensor it will give me a flatter curve to read.

My Program has three whiles and two ifs for just reading the voltage switch. The first while makes sure that the high and low both run at least once. When it's in the high while, it will run through an if to see if the voltage input is higher than the last set high input, if it is then it sets the high to the last input value. Because it went into my high loop once it sets my high checker Boolean to true. When it goes into a low (when the input is less than 512) it goes into my low loop and does the same thing, and sets low Boolean to true.

Once both Booleans are set to true it exits the loop, averages the high and low values, prints the voltage output to serial, and finally runs through a few loops to see if it's a rich, lean, or stoichiometric mixture. Within the loops when it decides what the mixture is, the program writes the RICH, LEAN, or STOICH, and then turns on a set of LEDs: the red one if RICH, the green one if LEAN, and both if in STOICH. Then it resets the Booleans, high, and low to nominal values to start the while loop again.

CODE

```
int analogPin = 0;
int input = 512;
int output = 0;
int ghigh = 512;
int glow = 511;
float mili = 0;
boolean H = false;
boolean L = false;
int ledPinR = 7;
int ledPinL = 6;
void setup()
{
  Serial.begin(9600);      // setup serial
  pinMode(ledPinR, OUTPUT);
  pinMode(ledPinL, OUTPUT);
}

void loop()
{
  //input = analogRead(analogPin); // read the input pin
  while (H != false && L != false);
  {

    while (512 <= input)
    {
      if (ghigh < input)
      {
        ghigh = input;
      }
      input = analogRead(analogPin);
      H = true;
    }

    while (511 >= input)
    {
      if (glow > input)
      {
        glow = input;
      }
      input = analogRead(analogPin);
      L = true;
    }
  }
}
```

```
}  
}
```

```
output = ((ghigh + glow)/2);  
mili = output * 0.0049;  
Serial.println();  
Serial.println(mili);  
if (mili < 2.3)  
{  
  Serial.println("RICH");  
  digitalWrite(ledPinR, HIGH);  
  digitalWrite(ledPinL, LOW);  
  delay(1000);  
}  
else if (mili > 2.7)  
{  
  Serial.println("LEAN");  
  digitalWrite(ledPinL, HIGH);  
  digitalWrite(ledPinR, LOW);  
  delay(1000);  
}  
else  
{  
  Serial.println("Stoich");  
  digitalWrite(ledPinL, HIGH);  
  digitalWrite(ledPinR, HIGH);  
  delay(1000);  
}  
glow = 511;  
ghigh = 512;  
H = false;  
L = false;  
}
```

