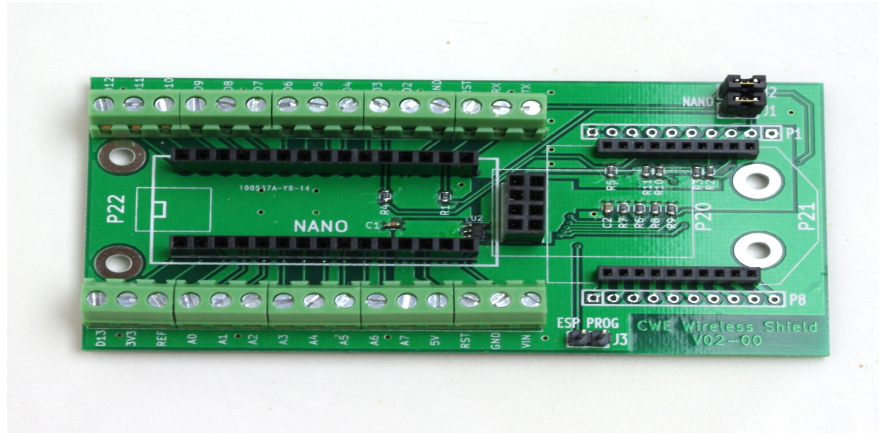




Wireless WiFi Nano Example *using Telnet, ESP8266, and Arduino Nano*



1. Ensure J1 and J2 are jumped.



a) Remove these when programming your Nano.

2. Ensure J3 is not jumped.

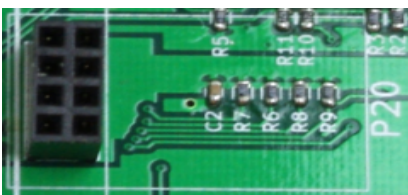


a) Jump this only when programming your ESP.

3. Program your Nano with software from appendix below.
4. Plug your Nano into P22.



5. Plug your ESP8266 into P20.

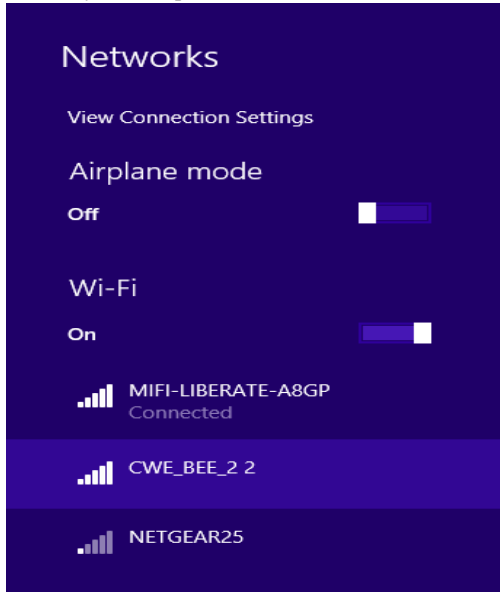




Wireless WiFi Nano Example *using Telnet, ESP8266, and Arduino Nano*



6. Connect your computer/device to “ESP” WiFi connection.



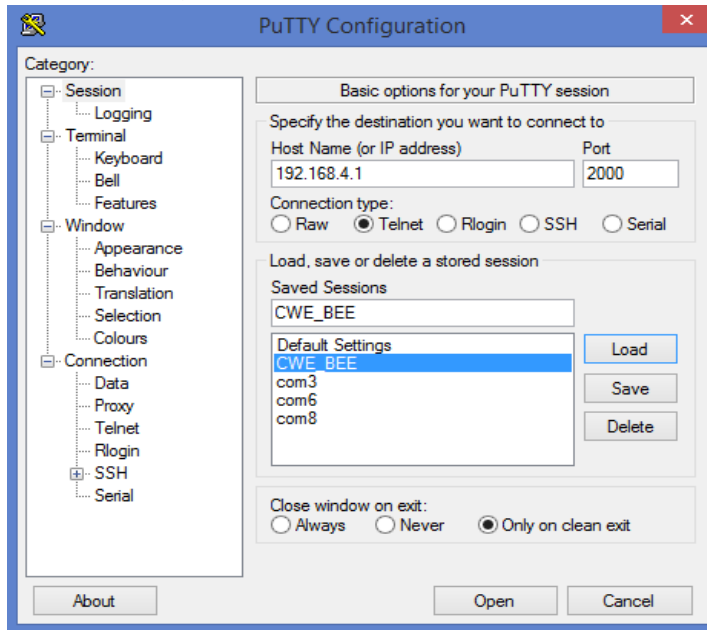
- a) The name in the wireless connections depends how your ESP is configured.
b) Here it is configured as “CWE_BEE_2”.

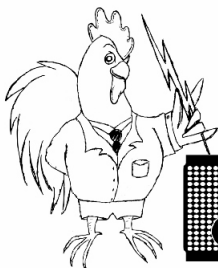


Wireless WiFi Nano Example *using Telnet, ESP8266, and Arduino Nano*



7. Open and connect via PuTTY or equivalent.
8. Connect via Telnet, IP: *192.168.4.1*, and Port: *2000*.

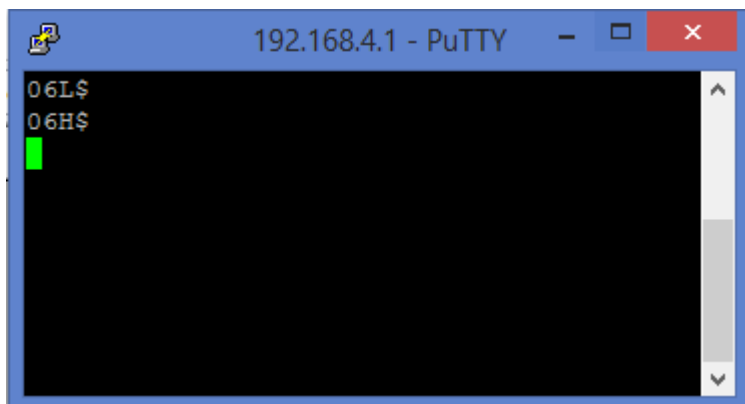




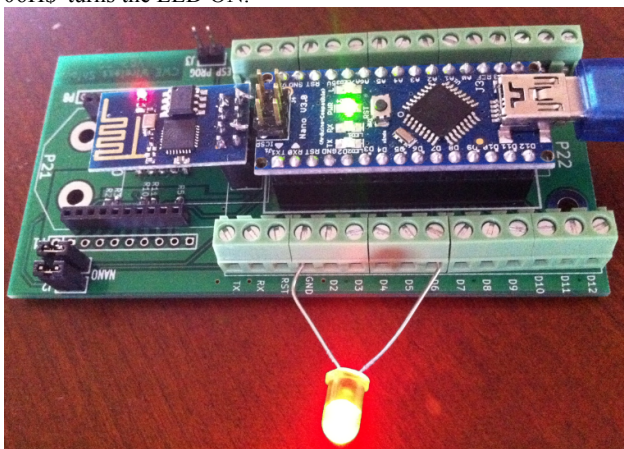
Wireless WiFi Nano Example *using Telnet, ESP8266, and Arduino Nano*



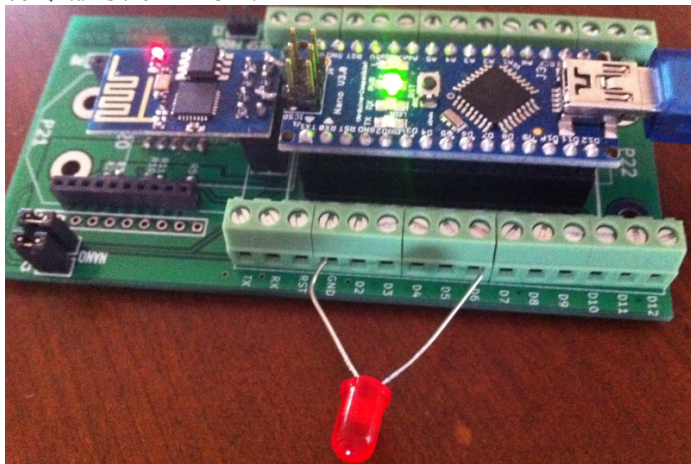
9. Send 'XXH\$' to turn output ON or 'XXL\$' to turn output OFF.
a) Example: '06H\$' turns D6 ON and '06L\$' turns D6 OFF.



- b) '06H\$' turns the LED ON.



- c) '06L\$' turns the LED OFF.





Wireless WiFi Nano Example *using Telnet, ESP8266, and Arduino Nano*



Appendix: Arduino Code

```
/*
This software is provided "as-is," without any express or
implied warranty.
In no event shall CW Electronx be held liable for any
damages arising from the use of the software.
*/

String data, readData;
String pinNum;
String pinStatus;
int pinNumInt;
int sensorValue = 0;

void setup()
{
  //Initialize serial
  Serial.begin(9600);

  //Set the pins to be outputs
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);

  //INITIALIZE AND CONFIGURE ESP8266 (IP: 192.168.4.1
  and PORT: 2000)
  //APPEARS TO TAKE MULTIPLE TIMES TO
  CONFIGURE AS SERVER
  //CLIENT NEEDS TO WAIT UNTIL INIT BEFORE
  SENDING COMMANDS TO SERVER
  Serial.println("AT");
  delay(5000);
  Serial.println("AT+RST");
  delay(5000);
  Serial.println("AT+CIPMUX=1");
  delay(2000);
  Serial.println("");
  Serial.println("AT+CIPMUX=1");
  delay(5000);

  Serial.println("AT+CIPSERVER=1,2000");
  delay(2000);
  Serial.println("");
  Serial.println("AT+CIPSERVER=1,2000");
}

void loop()
{
  //Get Serial
  Command=====

  // Read Serial Data when Available
  while (Serial.available() > 0)
  {
    readData = Serial.readString();

    data = readData.substring(readData.indexOf('.'),
    readData.indexOf('$'));

    //e.i. D02L or D02H ---> Pin#: 2 Status Request: L or H
    pinNum = data.substring(1,3);
    pinStatus = data.substring(3,4);
  }

  //Process and Set Command to
  Outputs=====
  //Convert pin number String to int
  pinNumInt = pinNum.toInt();

  //Check to see if request is in the range of the outputs
  if((pinNumInt < 14) && (pinNumInt > 1))
  {
    setOutput(pinStatus, pinNumInt);
  }

  void setOutput(String statusStringIN, int pinIN)
  {
    if(0 == statusStringIN.compareTo("H"))
    {
      digitalWrite(pinIN, HIGH);
    }
    else if(0 == statusStringIN.compareTo("L"))
    {
      digitalWrite(pinIN, LOW);
    }
    //else Do NOTHING if Command not recognized
  }
}
```