Digital SWR Meter

Circuit VSWR from Forward and Reverse Power

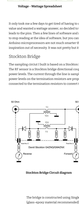
DSI Inline RF Power & VSWR Meter
RF Power 5-1000 watts with Average and Peak

Overall this is an accurate meter, fantastic for a meter under $20, good performance across the HF bands using fairly common parts and simple construction techniques.

The Inline Power and VSWR meter is a valuable tool for the ham radio station. Most commercial units are rather costly, mounted this project as a bench meter to use when I was building a transmitter drive input. The idea to build an inline power meter came out of need rather than inspiration. The design is constructed for several power levels. Read on for detailed information on how forward and reverse power meters work.

Circuit Description

The Wordless bridge circuit is selected over a Reed Bird or other wideband bridge. This is a very simple circuit just a forward and reverse set current transformers and a bridge. The Reed Bird bridge is a very good wideband bridge but requires a balun and is more complicated than a simple bridge. The transformers are toroidal with a single turn primary and a 100 turn secondary. The coupling is for a nominal forward to reverse sensitivity. The coupling factors are usually in the range of 50 to 200 for a 5 watt transceiver. Using 50Ω turns results in 0.1 watt of RF current at the reference receiver. Current rates at 1 watt forward power. The reverse current is 1/10th the forward current. The reverse current is 0.01 watt at 10 voltages to calculate RF power and RF voltage. Voltage from Ham, see "Decide on these options before you build" for circuit information.



The VSWR meter is a bridge circuit. I was spending too much time calculating reed values. The test problem I faced was a power increased the peak reverse voltage 100 V of a 10W diode is 100V. Reed bridge circuit requires 100V diodes. Wordless bridge circuit requires 10V diodes. Wordless bridge circuit requires less component values for bench test. A 10W diode is 10V. Reed bridge would be good enough. I found more readily QRP meters the power range. but I were been looking for a 100 watt meter. Wordless bridge circuit is better for QRP. Reed bridge circuit is better for 100 watt. Reed Bird bridge circuit is the right choice for high power and meter reads. adding amount to the power and budget. I continued on for some time using a spreadsheet with the voltages 10µs, 1V, 10mV and wattage something like the set others.

Understanding how the Wordless bridge circuit works and why automatic Reed Bird bridge circuit is better requires a little math. Reed Bird bridge circuit is better because it provides a fixed source of baseline VSWR. Reed Bird bridge circuit is better because it provides a fixed source of baseline VSWR. Wordless bridge circuit is better because it provides a fixed source of baseline VSWR. Wordless bridge circuit is better because it provides a fixed source of baseline VSWR. Wordless bridge circuit is better because it provides a fixed source of baseline VSWR. **YouTube KC5VHF** KC5VHF Wordless Bridge Circuit Analysis. Wordless bridge circuit is better because it provides a fixed source of baseline VSWR

```
      {
      ///////////// Peak Hold ///////////
       if ( Pfwd > MaxFwd )
            {
              MaxFwd = Pfwd;
            }
         if ( Prev > MaxRev )
            {
              MaxRev = Prev;
            }
         if ( SWR > MaxSwr )
            {
              MaxSwr = SWR;
            }
      ////////////////////////////////// average value integrator   Select % values to add up to 100% //   10
/ 90 is fast,  5 / 95 is medium,   2 / 98 is slow
      AvrFwd = ( AvrFwd * 0.95 ) + ( Pfwd * 0.05 );   // average value integrator
      AvrRev = ( AvrRev * 0.95 ) + ( Prev * 0.05 );   // average value integrator
      AvrSwr = ( AvrSwr * 0.95 ) + ( SWR * 0.05 );   // average value integrator
      ////////////////////////////////
      lcd.setCursor (0, 0);  // go to Column, Row
      lcd.print("=== Transmitting ===");
      lcd.setCursor (0,1);
      lcd.print("FWD  ");
      lcd.setCursor (5,1);           // go to Column, Row
      lcd.print(Pfwd, 1);      // displays forward power as xx.x value    //  ",1" means one decimal place
      lcd.print("  ");     // to clear trailing digits

            //  FWD bargraph
            lcd.setCursor (10,1);           // go to Column, Row
            lcd.print("          ");     // to clear last bar
            lcd.setCursor (10,1);           // go to Column, Row
            float  bar = 12 - ( Pfwd / 2 );
            bar = constrain(bar, 0, 10); // limit max bar
            for (bar; bar < 11; bar = bar + 1)
            {  lcd.print(">");   }    // bargraph
            //  end bargraph

      lcd.setCursor (0,2);           // go to Column, Row
      lcd.print("REV  ");   // displays menu selected // Clearing not required always 16 Char
      lcd.setCursor (5,2);           // go to Column, Row
      lcd.print(Prev, 1);      // displays forward power as xx.x value    //  ",1" means one decimal place
      lcd.print("  ");     // to clear trailing digits

            //  REV bargraph
            lcd.setCursor (10,2);           // go to Column, Row
            lcd.print("          ");     // to clear last bar
            lcd.setCursor (10,2);           // go to Column, Row
            bar = 12 - Prev;
            bar = constrain(bar, 0, 10); // limit max bar
            for (bar; bar < 11; bar = bar + 1)
            {  lcd.print("<");   }    // bargraph
            //  end bargraph

      lcd.setCursor (0,3);           // go to Column, Row
      lcd.print("SWR  ");   // displays menu selected // Clearing not required always 16 Char
      lcd.setCursor (5,3);           // go to Column, Row
      lcd.print(SWR, 1);      // displays forward power as xx.x value    //  ",1" means one decimal place
      lcd.print("  ");     // to clear trailing digits

            //  SWR bargraph
            lcd.setCursor (10,3);           // go to Column, Row
            lcd.print("          ");     // to clear last bar
            lcd.setCursor (10,3);           // go to Column, Row
            bar = 12 - SWR;
            bar = constrain(bar, 0, 10); // limit max bar
            for (bar; bar < 11; bar = bar + 1)
            {  lcd.print("#");   }    // bargraph
            //  end bargraph


      last_Vfwd = Vfwd;
      last_Vrev = Vrev;

      }
////////////////////////////////////////////////////////////////////////////////////////////////////

      if ( RFoff > 10000 )  //  no RF time out
        {
      delay (200);     // delay
      lcd.setCursor (0, 0);  // go to Column, Row
      lcd.print("No RF, Average/Peak ");

      lcd.setCursor (0, 1);  // go to Column, Row
      lcd.print("FWD  ");
      lcd.setCursor (5, 1);  // go to Column, Row
      lcd.print(AvrFwd, 1);      // displays forward power as xx.x value    //  ",1" means one decimal place
      lcd.print(" ");       // one space to clear last if > 10 (x.x vs xx.x)
      lcd.setCursor (10, 1);  // go to Column, Row
      lcd.print(" / ");
      lcd.setCursor (14, 1);  // go to Column, Row
      lcd.print(MaxFwd, 1);      // displays forward power as xx.x value    //  ",1" means one decimal place
      lcd.print("  ");        // one space to clear last if > 10 (x.x vs xx.x)

      lcd.setCursor (0, 2);  // go to Column, Row
      lcd.print("REV  ");
      lcd.setCursor (5, 2);  // go to Column, Row
      lcd.print(AvrRev, 1);      // displays forward power as xx.x value    //  ",1" means one decimal place
      lcd.print(" ");       // one space to clear last if > 10 (x.x vs xx.x)
      lcd.setCursor (10, 2);  // go to Column, Row
      lcd.print(" / ");
      lcd.setCursor (14, 2);  // go to Column, Row
      lcd.print(MaxRev, 1);      // displays forward power as xx.x value    //  ",1" means one decimal place
      lcd.print("  ");        // one space to clear last if > 10 (x.x vs xx.x)

      lcd.setCursor (0, 3);  // go to Column, Row
      lcd.print("SWR  ");
      lcd.setCursor (5, 3);  // go to Column, Row
      lcd.print(AvrSwr, 1);      // displays forward power as xx.x value    //  ",1" means one decimal place
      lcd.print(" ");       // one space to clear last if > 10 (x.x vs xx.x)
      lcd.setCursor (10, 3);  // go to Column, Row
      lcd.print(" / ");
      lcd.setCursor (14, 3);  // go to Column, Row
      lcd.print(MaxSwr, 1);      // displays forward power as xx.x value    //  ",1" means one decimal place
      lcd.print("  ");        // one space to clear last if > 10 (x.x vs xx.x)

      RFoff = 12000;  //     prevents overflow at 32000 and flags average, peak and display clean up on next
transmit
       }


  }   // end of main loop                 // end of main loop                 // end of main loop
// end of main loop                // end of main loop
```

Trust but Verify!

Looking at the example case of 9 Watts RF forward measure the voltage in the cable from the Stockton Bridge using a DVM verify the tip to shield is 2 VDC. Using a 1.5 VDC source (single cell flashlight battery) apply positive to the Display/Arduino Nano input tip and negative to the shield. You should read approximately 5 Watts forward power on the display. The calibration table shows a range of common values for testing. **DO NOT EXCEED 5 Volts! DO NOT REVERSE POLARITY! You will distroy the Arduino Nano.** When using the Arduino Nano the reference voltage will default to the onboard 5 volt reference that varies +/- 1/10 volt resulting a small wattage error. A extra 1/10 V will make the 2.5 W look like 2.8 W or 31 W be 32 W. This is not a fault in Arduino, the same error is found in the standard LM7805 series regulators. The typical 3 digit DVM has similar errors so unless you have a calibrated references these small errors are within expected tolorence.

RF Watts passing through bridge	**Bridge Output VDC*** at 3.5 mm cable
2.5	1
5	1.5
9	2
18	3
31	4

Calibration Table

**Note: Wattage approximate, good for typical DVM,*

see voltage reference discussion.

Closing Thoughts

I did not anticipate when I built this meter how well it would worked on the air and has proven to be a useful test instrument. I use one on my bench and later built one into my tuner. When I transmit SSB the averaging keeps the reading meaningful although averaging does nothing on CW. Even with the variations in reference voltages and diodes I have found the accuracy is better than +/- 5% from 3 to 30 Watts on a dummy load.

Overall this is a accurate QRP meter, fantastic for a meter under $20, but under 1/2 Watt here is something I noticed and most will never see is the power is over stated my power below 0.5 Watt. Not bad considering 5W is the minimum for most designs. This means the meter will show 0.4 W when it is actually 0.3 W but **this is a limitation of the diode voltage drop** correction in the software. For more details on accuracy see the Diode Selection discussion.

Thank you Free DNS for your help