

# Using the RH96 Remote Head with the BCD996P2 scanner

*Designed and programed by Richard Plummer, KV1W*

*Text and assistance by Rich Carlson, N9JIG*

*Copyright 2019, Richard Plummer & Rich Carlson*

*(Revised 2-23-2019 for USB connection, see page 8)*

The Uniden RH96 remote head was introduced in 2009 for the BCD996T and BCT15 mobile scanners as well as the BR330T and BCD396T handheld scanners. Later, it was updated to work with the newer BC346XT and 396XT handhelds and the BCT15X and BCD996XT mobile scanners when they were introduced. While it was never a big seller (Uniden dropped the RH96 due to poor sales) there were dedicated users of it and many were disappointed when it was found that it would not work on the new BCD996P2 scanner.

While the 996P2 and the older 996XT are very similar in many ways, the USB port used by the 996P2 is not supported by the RH96. The 996P2 is more desirable in many applications; in addition to the support of P25 Phase 2 it also can be upgraded to support DMR/MotoTRBO and ProVoice. Other improvements include the ability to instantly display CloseCall hit information (Frequency, DCS/CTCSS codes) without the need to press a button.

Several discussions were had at the RadioReference forums and elsewhere to see if the RH96 could be made to work with the 996P2. Rich Carlson had found that the RH-96 could be connected to the 996XT and older scanners by using the rear serial port (meant for the GPS receiver) but that requires a custom cable. That did not however work on the 996P2. He also figured out how to bypass the flimsy front panel connector for the RH96 by modifying the radio and adding a port to the rear panel <sup>(1)</sup>. This has the advantage to allow use of the GPS along with the RH96 but again did not make it work the 996P2.

Ric Plummer researched the data strings used by the radios and RH96 and determined that the scanner model number is part of it. The RH96 looks for the model number as part of the data string received from the radio. This tells it how to parse out the rest of the data, properly formats the display to work on that radio and allows the keyboard and other controls to work. The BCD996P2 is not one of the radios included in the RH96 firmware so is not recognized.

Ric Plummer took this approach to determine if the 996P2 data stream could be converted or translated to look like a 996XT. He built an Arduino converter to transfer the data from the radio to the RH96. The Arduino is programmed to convert the data to mimic the 996XT, as far as the RH96 is concerned it is an XT. It strips the BCD996P2 radio identifier and replaces it with

---

<sup>1</sup> <https://forums.radioreference.com/threads/rh-96-rewiring.270715/#post-2007778>

BCD996XT. This was successful, the RH96 worked great with the 996P2 since all the other commands are the same. An Arduino Nano is used in this application.

The next step was to connect to the radio and remote head. Since it has already been determined that the RH96 will work on the XT's rear GPS serial port that was the logical connection. Since the RH96 is a serial device it needs a serial connection to the radio. While Rich Carlson has modified several BCD996XT and BCT15's with 1/8-inch jacks to replace the fragile front serial port on the Uniden scanners, while this required difficult internal modifications to the scanner it allows the GPS receiver to remain connected to the scanner. This approach will not work with the 996P2 as it lacks the front serial port that this particular modification used. The 996P2 has a different daughterboard for the front programming panel and does not retain the 996XT's serial bus. By using the rear GPS port no modification of the scanner or the RH96 is made and no warranty issues are created but precludes the simultaneous use of the GPS feature. We are investigating a way to retain GPS functionality, if successful that will be in a future update.

The device designed by Ric Plummer accepts the 996P2's serial data and processes it so that it reports as a 996XT and sends it to the RH96. This allows two-way communications between the RH96 and the scanner and provided full functionality, identical to that of the 996XT.

The design as created by Ric Plummer is a small project box with the Arduino and an RS232 interface. There is a 12VDC power jack as well as a power pass thru lead to allow one to merely unplug the scanner's power connector, plug it into the device and then plug the pigtail into the power jack of the scanner. This allows a single power lead to power the scanner and converter, the RH96 still requires a separate power source.

There is also a pigtail with the DB9 serial cable to connect to the rear port of the 996P2 and a 1/8-inch stereo jack to accept a standard Male-Male 3-wire audio cable to connect to the RH96. The cable supplied with the RH96 (3-wire 1/8-inch male to a female 4-pin Uniden serial port) is not used and can be set aside.

The Arduino has an on-board USB Mini port to load the programming, otherwise that jack is not used during operation of the radio.

To connect the converter (assuming you use the same design) remove the power connector to the radio and plug it into the converter then plug the power pigtail into the scanner. Plug the DB-9 serial port plug from the converter to the rear serial port of the 996P2. Lastly plug a power lead to the RH96 and connect the converter to the RH96 with an appropriate length 3-wire male to male audio cable. Get a cable with a 90° plug on one or both ends if possible, using a 90-degree plug on the RH96 allows it to not extend below the unit. On the radio end it reduces the possibility of damage by contact.

On the BCD996P2 set the GPS Baud Rate to 9600 in the Menu>Settings>Set GPS Baudrate menu item. You may have to turn this serial port off in the same menu when programming the radio with a serial cable, remember to turn it back on when finished programming.

Once completed the RH96 will replicate the display and controls of the BCD996P2 just as it does with other scanners. One can enjoy the updated feature set of the P2 series scanner and retain the wholesome goodness of the RH96's reduced footprint in the more crowded cabin of today's vehicles. Remember that speaker audio is not handled by the RH96, you will need to mount a remote speaker if the placement of the radio does not lend itself to adequate volume.

The small project box can be attached to the radio by double-stick tape or Velcro. Since it is lightweight and small it does not add to the footprint. Power consumption is miniscule.

### **Legal Stuff:**

Neither the author or designer are interested in producing these devices commercially. This document is produced for individuals to build them for personal use. No warranty is implied or provided, you may use this information at your own risk. You may not use this information for commercial gain nor republish it without full acknowledgement to the author and designer.

### **Build the converter:**

These are the parts used by Ric Plummer to build the converter. All parts were purchased from Amazon. Locate searching for Arduino or the provided Amazon reference numbers.

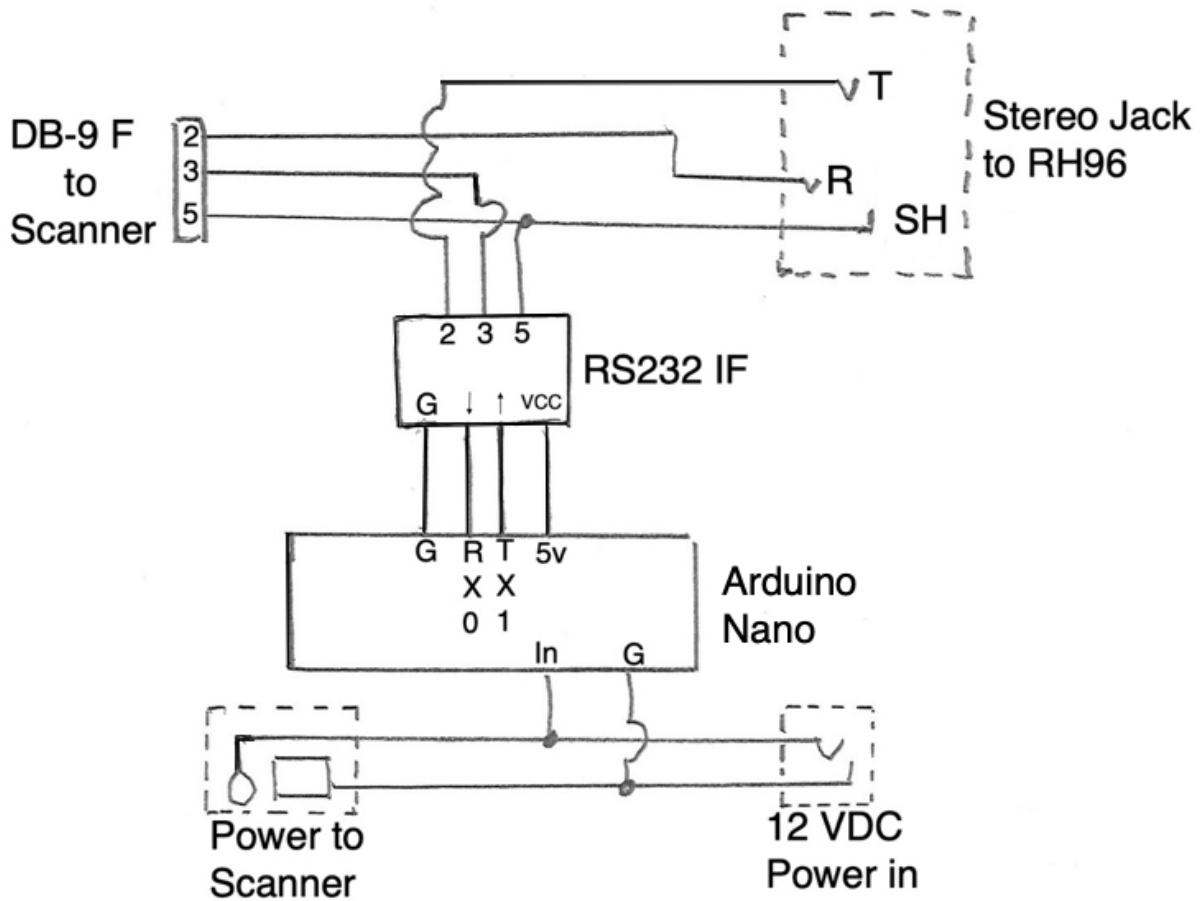
| <b><i>Item</i></b>   | <b><i>Amazon Ref #</i></b> | <b><i>Comments</i></b>      |
|----------------------|----------------------------|-----------------------------|
| Plastic Project Box  | B073Y7FW1Q                 | 2 x 3.5 inch used here      |
| Arduino Nano         | B0713XK923                 | used a knockoff             |
| Mini RS232 TTL conv. | B00OPTOKIO                 | removed connector (*)       |
| Coaxial Power jack   | B07C46XMPT                 | 5.5mm panel mount           |
| 1/8 inch Stereo Jack | LYSB01C3RFHDC-ELECTRNCS    | Panel mount                 |
| Coaxial Power cord   | LYSB01GPL8MVG-CMPTRACCS    | 5.5mm, cut to fit           |
| DB9-Female plug      | B076HPV4WJ                 | with hoods, cut wire to fit |
| Misc. wire           |                            |                             |

Total cost for the items will be between \$20 and \$30, depending on quantity and current prices.

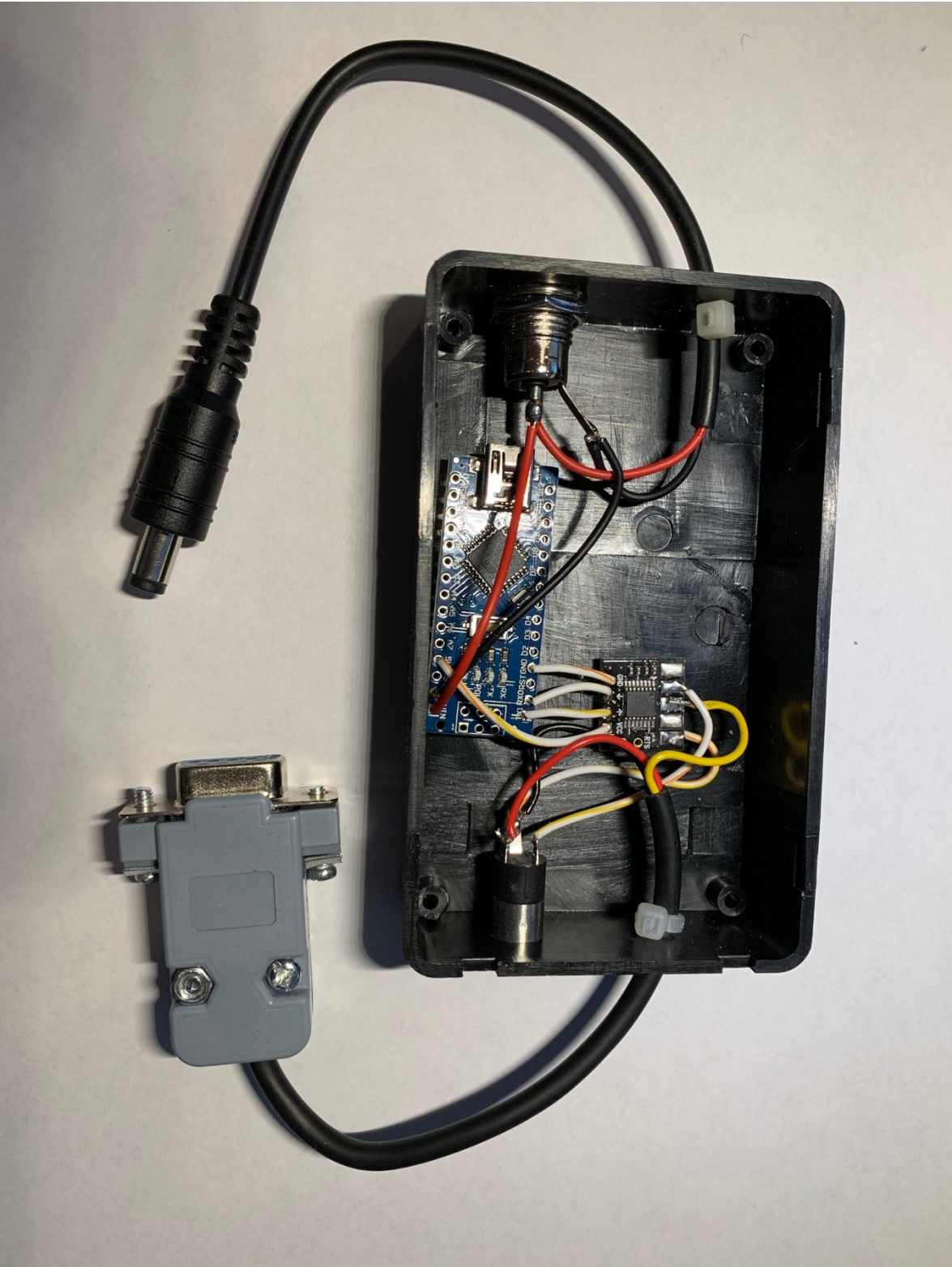
(\*) The RS232 TTL converter board was removed from its DB-9F with a Dremel tool to allow it to fit the package, be sure not to damage the PC Board! Both boards are secured to the interior of the project box with sticky foam tape.

**Schematic:**

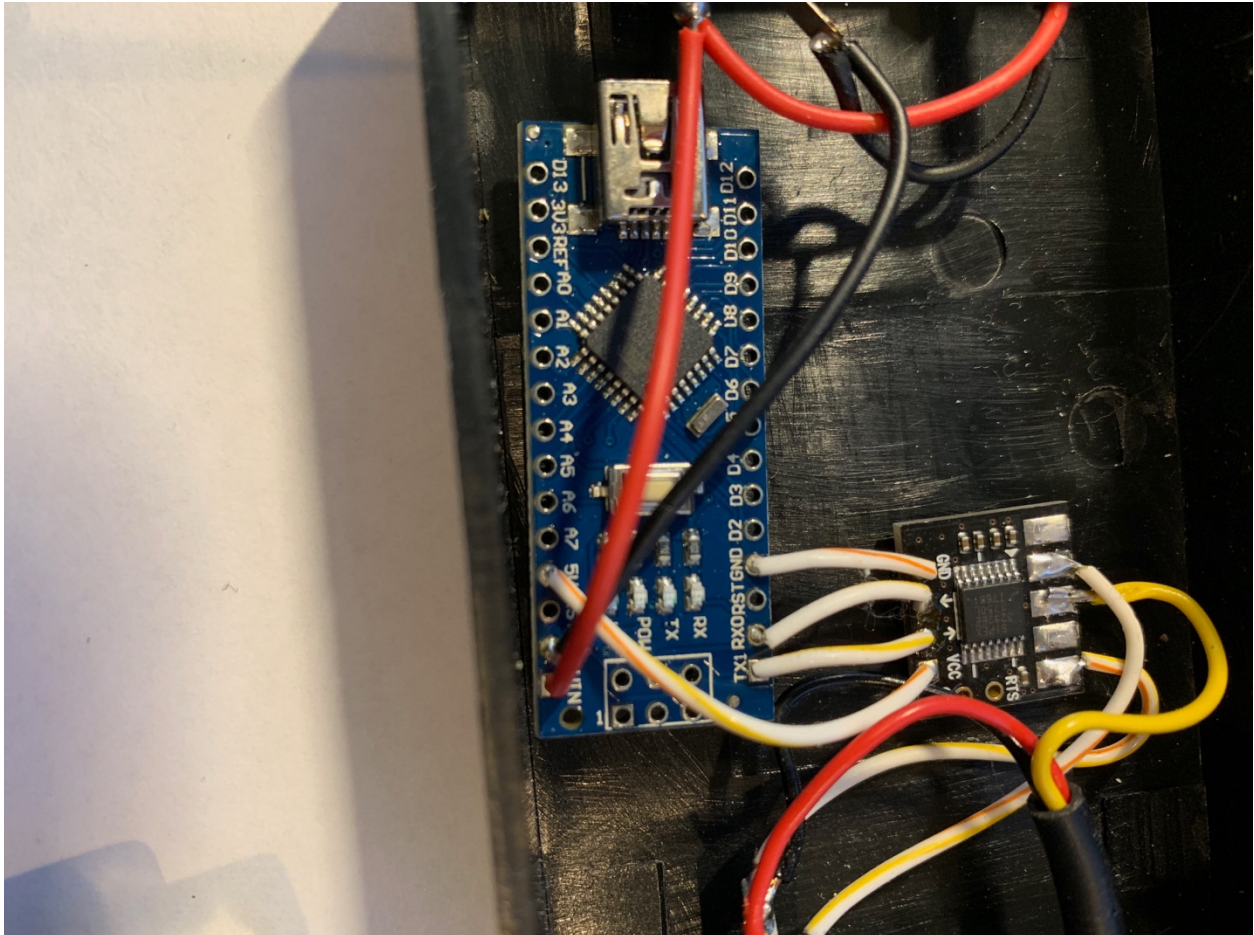
Here is the schematic for the converter using the rear GPS Port:  
(DB-9 connector is female)



Completed converter in project box (with the cover removed). On the top is the power jack and pass-thru power pigtail, the data port and scanner pigtail are at the bottom. The Arduino is on the left, the RS232 interface on the right.



Detail on the wiring between the RS232 interface (right) and the Arduino (left). The red and black wires to the Arduino are for power, the various wires to the bottom right are for the DB-9 and RH96 serial data.



## Programming:

Here is the programming Ric Plummer wrote to load into the Arduino processor. It is programmed with Arduino IDE Tools.

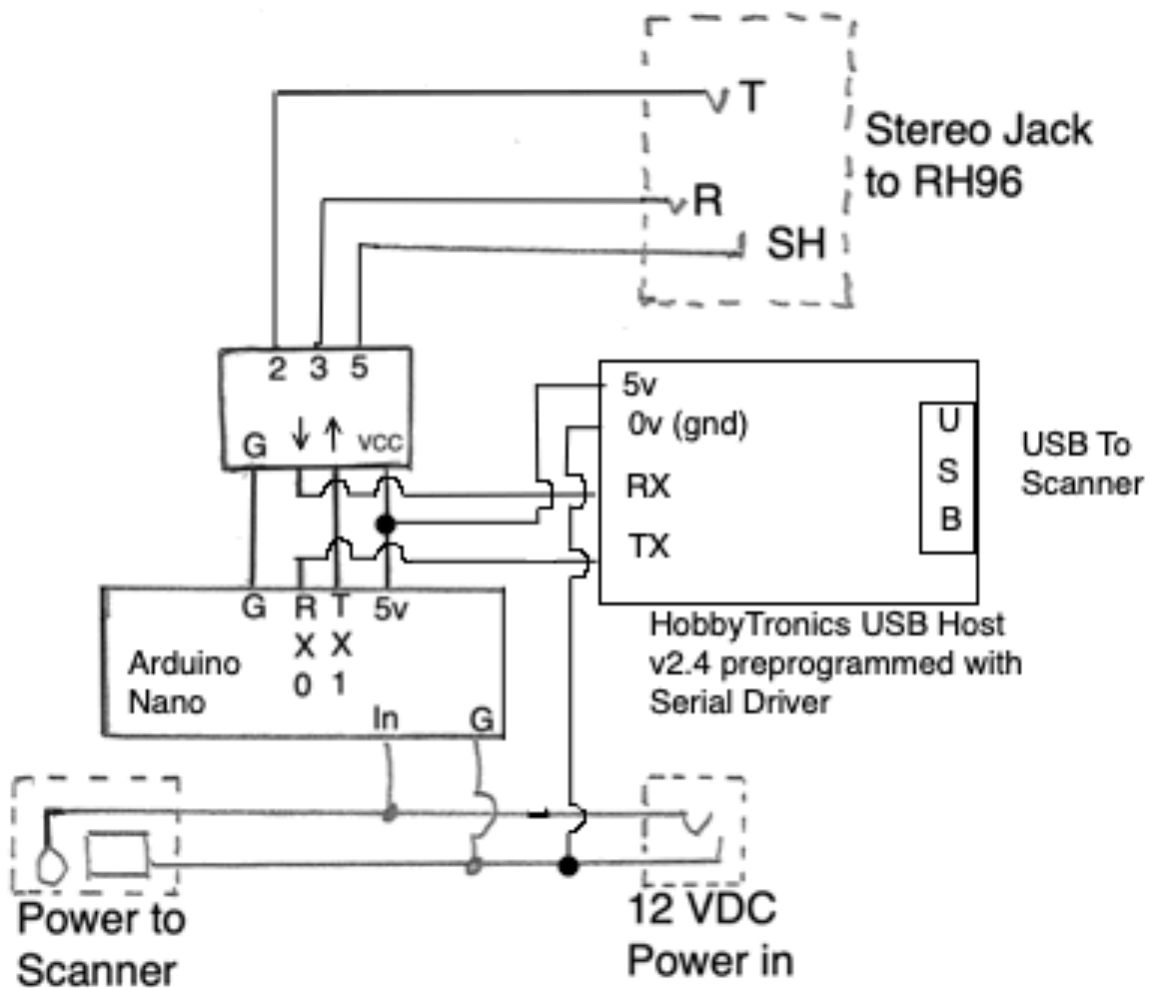
```
//Final version of the RH-96 sketch to use RH-96 with BCD996P2
// Note: the Nano must be loaded before connecting the RS-232 adapter as it will use the USB UART
//and will conflict with loading and testing of the Nano from the PC USB port.
//Once loaded and checked with Serial Monitor, you can connect the RS-232 hardware.
char Str1[200]; //allow storage space for array
char inChar; //for received character
int index = 0; //index into the array of stored characters
int lastindex;
bool bMsgReceived = false;
void setup() {
  Serial.begin(9600);
}
// The loop below will send all <CR> terminated strings to the output as is,
//except changing the P2 to XT in the string MDL,BCD996P2
void loop() {
  // Will Read characters one by one and place into the Array Str1 and terminate the array at \n
  // starting at location 0 and going as high as required to hold the string
  while (Serial.available() > 0) {
    Str1[index] = Serial.read(); //Read a Character and store in array
    lastindex = index;
    if (Str1[index] == 13) { //Terminate on "New Line" aka <CR>, must be in stored string
      bMsgReceived = true;
      break;
    } else {
      index++; //Index to next array location
    }
  }
  if (bMsgReceived) {
    // Here we check if the response is an MDL and if P2 in Str1[10&11] then change to XT
    if (Str1[0]=='M' && Str1[1]=='D' && Str1[2] == 'L' && Str1[10]!='P' &&
        Str1[11]!='2') {
      Str1[10]='X';
      Str1[11]='T';
    }
    //Now output the string in the array Str1, terminated by a 13, then back to loop to do it again
    index = 0;
    while (index <= lastindex) {
      Serial.print(Str1[index]);
      index++;
    }
    index = 0;
    bMsgReceived = false;
  }
}
```

## Addendum: USB Edition

Ric Plummer designed a second version to use the front USB port of the BCD996P2 to retain GPS capabilities. Basically, this meant using a USB host to convert the RS232 interface from the converter to the USB format understood by the radio's USB port. The previous hardware and programming still applies with the addition of the serial host.

Here is the schematic for the USB edition. A USB-A to USB-Mini cable is used between the converter and the front USB port on the BCD996P2.

*(Thanks to Mark Nelson for catching an error, corrected 2-24-19)*





Here is a photo of the completed converter box for USB use:

