PARALLAX INC

Web Site: www.parallax.com
Forums: forums.parallax.com
Sales: sales@parallax.com
Technical: support@parallax.com

Office: (916) 624-8333
Fax: (916) 624-8003
Sales: (888) 512-1024
Tech Support: (888) 997-8267

□Parallax 433.92 MHz RF Transmitter (#27980)
☑Parallax 433.92 MHz RF Receiver (#27981)

General Description

The Parallax 433.92 MHz RF Transmitter allows users to easily send serial data, robot control, or other information wirelessly. When paired with the matched RF Receiver, reliable wireless communication is as effortless as sending serial data. The power-down (PDN) pin may be used to place the module into a low power state (active low), or left floating (it is tied high internally).

Features

- High-speed data transfer rates (1200 ~ 19.2k Baud depending on controller used)
- SIP header allows for ease of use with breadboards
- Compatible with all BASIC Stamp® modules (including BS1 and Javelin Stamp) and SX chips
- As easy to use as simple SEROUT/SERIN PBASIC instructions
- Power-down mode for conservative energy usage (longer battery life)
- Line-of-sight range of 500 feet (or greater depending on conditions)

Application Ideas

- Remote Controlled Boe-Bot® robot
- Wireless data acquisition
- Remote sensors and triggers

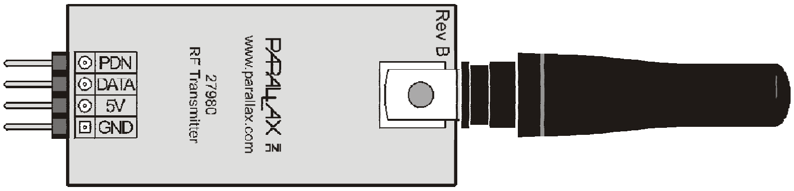
Device Information

Theory of Operation

Short for Radio Frequency, RF refers to the frequencies that fall within the electromagnetic spectrum associated with radio wave propagation. When applied to an antenna, RF current creates electromagnetic fields that propagate the applied signal through space. Any RF field has a wavelength that is inversely proportional to the frequency. This means that the frequency of an RF signal is inversely proportional to the wavelength of the field. The Parallax RF modules utilize a frequency of 433.92 MHz, this works out to be a wavelength of approximately 0.69 meters (2.26 feet, or 7.3e-17 lightyears). 433.92 MHz falls into the Ultra High Frequency (UHF) designation, which is defined as the frequencies from 300 MHz ~ 3 GHz. UHF has free-space wavelengths of 1 m ~ 100 mm (3.28 ~ 0.33 feet or 1.05e-16 ~ 1.05e-17 lightyears).

Pin Definitions and Ratings

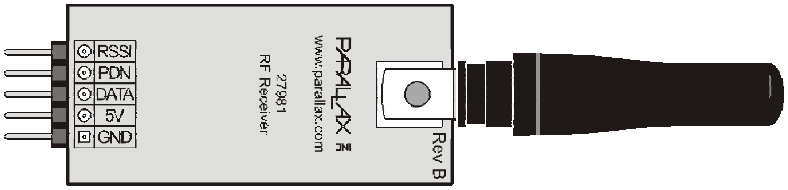
Transmitter (#27980)



Pin	Name	Function
1	PDN	Power Down - active low, pulled high internally so may be left floating
2	DATA	Data Output
3	5v	Power – connect to +5v DC (such as Vdd)
4	GND	Ground → 0 v (such as Vss)

Transmitter Supply Current:	
At Logic High Input:	5.1 mA
At Logic Low Input:	1.8 mA
Low Power mode (PDN):	5 μA

Receiver (#27981)



Pin	Name	Function
1	RSSI	Received Signal Strength Indicator
2	PDN	Power Down - active low, pulled high internally so may be left floating
3	DATA	Data Input
4	5v	Power – connect to +5v DC (such as Vdd)
5	GND	Ground → 0 v (such as Vss)

Receiver Supply Current:	
During Operation (*High or Low*):	5.2 mA
Low Power mode (PDN):	28 μA

PDN

Pulling the power down (PDN) line low will place the transmitter/receiver into a low-current state. The module will not be able to transmit/receive a signal in this state.

RSSI *(receiver only)*

Received Signal Strength Indicator. This line will supply an analog voltage that is proportional to the strength of the received signal.

Calibration

When initiating communication between the RF modules, a sync pulse should be sent to re-establish the radio connection between the modules. Sending several characters can accomplish this, however sending a pulse (which maintains a high state during the synchronization) is more efficient:

√ For BS1s the following code line sends an appropriate sync pulse:

```
PULSOUT 1, 300
```

√ For BS2s the following code line sends an appropriate sync pulse:

```
PULSOUT 8, 1200
```

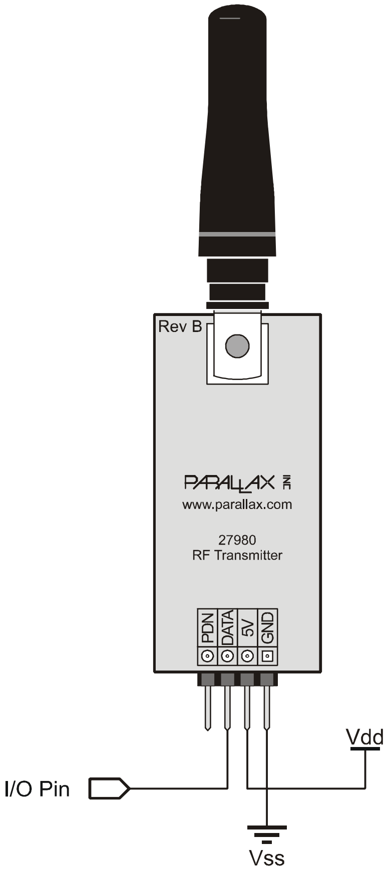
√ For Javelin Stamp modules the following code line sends an appropriate sync pulse:

```
CPU.pulseOut(300, CPU.pin8);
```

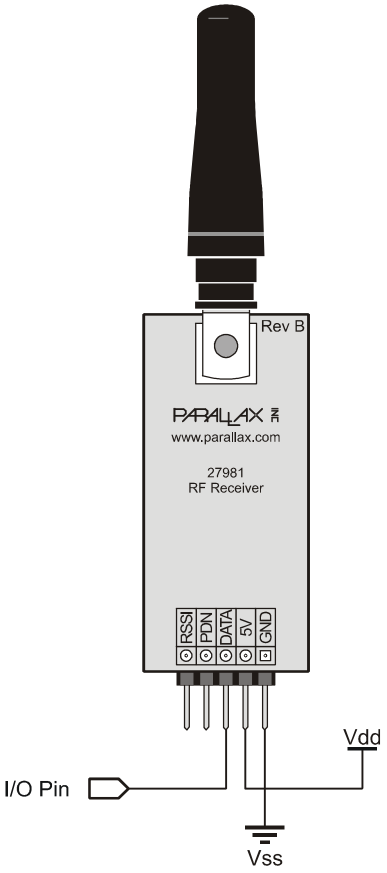
(Note: this line assumes that I/O pin 8 is connected to the DATA line on the transmitter)

Connection Diagrams

Transmitter (#27980)

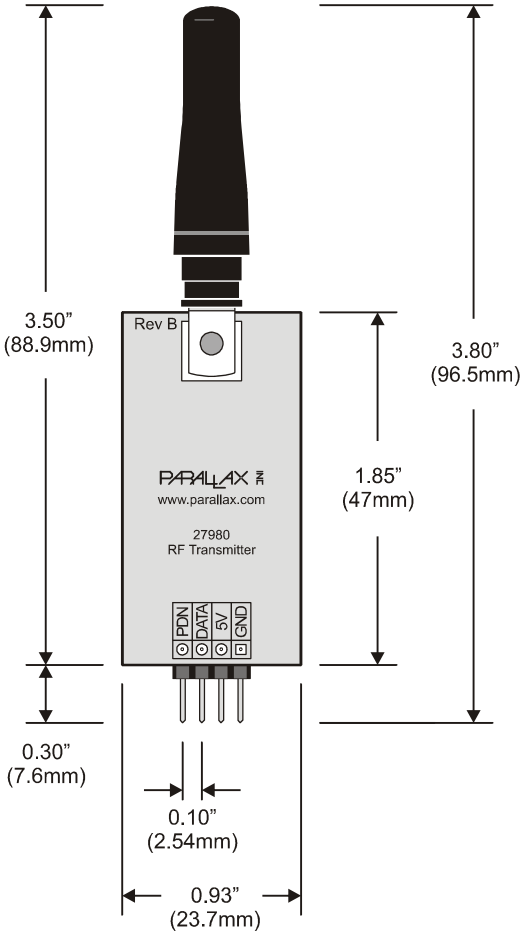


Receiver (#27981)

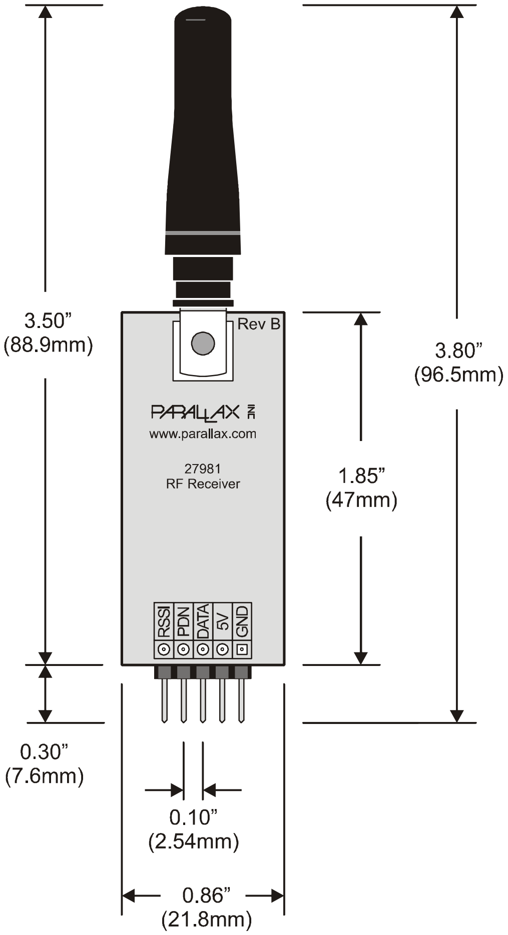


Module Dimensions

Transmitter (#27980)



Receiver (#27981)



Source Code

BASIC Stamp® 1 Example Programs

The following BS1 programs will transmit (Tx) and receive (Rx) two word-sized counters (four bytes of data). DATA lines on each module should be connected to I/O pin 1 of each of the BS1 modules. The 5v and GND lines should be connected to a +5 v source (Vdd) and ground (Vss) respectively (for both Tx and Rx). The BS1 on the receiving end should remain connected to the computer to view the DEBUG statements.

TxCode_v_1.0.bs1

```
' TxCode_v_1.0.bs1
'{$STAMP BS1}
'{$PBASIC 1.0}
'
'Parallax 433.92 MHz RF Transmitter (#27980)  Sample Tx Code (BS1)
'Connect I/O Pin 1 of the BS1 to the DATA line on the RF module
'Connect +5v (Vdd) to the 5v line, connect Ground (Vss) to the GND line
'This code will transmit two word sized counters, byte at a time.
'
'Note the PULSOUT instruction: this helps the receiver sync with
'the transmitter, especially after lapses in communication.
'This code transmits at 2400 baud, inverted.


SYMBOL x  =  W1
SYMBOL y  =  W2

Start:

  PULSOUT 1, 300                               'Sync pulse for the receiver
  SEROUT 1, N2400, ("!", B3, B2, B5, B4)
  PAUSE 100
  x = x + 1
  y = y + 1
  PAUSE 150

GOTO Start
```

RxCode_v_1.0.bs1

```
'RxCode_v_1.0.bs1
'{$STAMP BS1}
'{$PBASIC 1.0}
'
'Parallax 433.92 MHz RF Receiver (#27981) Sample Rx Code (BS1)
'Connect I/O Pin 1 of the BS1 to the DATA line on the RF module
'Connect +5v (Vdd) to the 5v line, connect Ground (Vss) to the GND line
'This code will look for an "!", then read in four bytes.
'The first two bytes will be the high and low bytes of a word variable (x),
'the second two bytes will be the high and low bytes of a second word variable (y).
'The values of x and y will be sent out to the DEBUG terminal.
'This code receives at 2400 baud, inverted.

SYMBOL x  =  W1
SYMBOL y  =  W2

Start:
  SERIN 1, N2400, ("!"), B3, B2, B5, B4
  DEBUG x, y

GOTO Start
```

BASIC Stamp® 2 Example Programs

The following BS2 programs will transmit (Tx) and receive (Rx) two word-sized counters (four bytes of data). DATA line on the transmitter should be connected to I/O pin P8 of the BS2. The DATA line on the receiver should be connected to I/O pin P7 of another BS2. The 5v and GND lines should be connected to a +5 v source (Vdd) and ground (Vss) respectively (for both Tx and Rx). The BS2 on the receiving end should remain connected to the computer to view the DEBUG statements.

TxCode_v_1.0.bs2

```
' TxCode_v_1.0.bs2
'{$STAMP BS2}
'{$PBASIC 2.5}
'
'Parallax 433.92 MHz RF Receiver (#27981) Sample Tx Code (BS2)
'Connect Pin 8 of the BS2 to the DATA line on the RF module
'Connect +5v to the 5v line, connect Ground to the GND line
'This code will transmit two word sized counters, byte at a time.
'
'Note the PULSOUT instruction: this helps the receiver sync with
'the transmitter, especially after lapses in communication
'This code transmits at 9600 baud, inverted.

x   VAR   Word
y   VAR   Word

DO
  PULSOUT 8, 1200                             'Sync pulse for the receiver
  SEROUT  8, 16468, [ "!", x.HIGHBYTE, x.LOWBYTE, y.HIGHBYTE, y.LOWBYTE ]
  x = x + 1
  y = y + 1
  PAUSE 10
LOOP
```

RxCode_v_1.0.bs2

```
'RxCode_v_1.0.bs2
'{$STAMP BS2}
'{$PBASIC 2.5}
'Parallax 433.92 MHz RF Receiver (#27981) Sample Rx Code (BS2)
'Connect I/O Pin 7 of the BS2 to the DATA line on the RF module
'Connect +5v (Vdd) to the 5v line, connect Ground (Vss) to the GND line
'This code will look for an "!", then read in four bytes.
'The first two bytes will be the high and low bytes of a word variable (x),
'the second two bytes will be the high and low bytes of a second word variable (y).
'The values of x and y will be sent out to the DEBUG terminal.
'An optional LED (with proper current limiting resistor) may be connected to the
'BS2 I/O pin 0, this will blink when data is received.
'This code receives at 9600 baud, inverted.

x   VAR   Word
y   VAR   Word

DO
```

```
  LOW 0
  SERIN 7, 16468, [WAIT("!"), x.HIGHBYTE, x.LOWBYTE, y.HIGHBYTE, y.LOWBYTE]
  HIGH 0
  DEBUG ? x
  DEBUG ? y
LOOP
```

Javelin Stamp Application Example

The following two programs demonstrate a simple transmit (Tx) and receive (Rx) communication utilizing the RF modules and Javelin Stamps.

Note: For the transmit (Tx) code connect I/O pin 8 of a Javelin Stamp to the DATA line on the RF transmitter module. For the receive (Rx) code, connect I/O pin 7 of another Javelin Stamp to the DATA line on the RF receiver module.

RFmoduleTX.java

```
/***
RFmoduleTX.java
This code is for simple demonstration of the Parallax 433.92 Mhz RF Transmitter
(#27980)
Connect Javelin I/O pin 8 to the RF Tx module DATA line.
Connect +5v (Vdd) to the RF Tx module 5v line.
Connect ground (Vss) to the RF Tx module GND line.
This code will transmit at 9600 baud, inverted.
This code is meant to run in conjunction with RF Rx module (#27981) running code set
to receive at 9600 baud.
***/

import stamp.core.*;

public class RFmoduleTx {

  static Uart txUart = new Uart (Uart.dirTransmit, CPU.pin8, Uart.invert,
                           Uart.speed9600, Uart.stop1);

  static byte x = 0;
  static byte y = 0;

  public static void main()  {

    while(true) {
      if (x >= 10) x = 0;
       else x++;
      if (y >= 10) y = 0;
       else y++;
      CPU.pulseOut(300, CPU.pin8);  //Sync pulse to help Rx after periods of no
communication.
      txUart.sendString("!");       //Start char that Rx will look for
      txUart.sendByte(x);           //Sending 2 words: high byte, low byte
      txUart.sendByte(y);
      CPU.delay(25);                //help set spacing
    }                               //end while
  }                                 //end main
}                                   //end class
```

RFmoduleRX.java

```
/***
RFmoduleRx.java
This code is for simple demonstration of the Parallax 433.92 Mhz RF Reciever (#27980)
Connect Javelin I/O pin 7 to the RF Rx module DATA line.
Connect +5v (Vdd) to the RF Rx module 5v line.
Connect ground (Vss) to the RF Rx module GND line.
This code will receive at 9600 baud, inverted.
This code is meant to run in conjunction with RF Tx module (#27981) running code set
to transmit at 9600 baud.
***/

import stamp.core.*;

public class RFmoduleRx {

  static Uart rxUart = new Uart (Uart.dirReceive, CPU.pin7, Uart.invert,
                                 Uart.speed9600, Uart.stop1);

  static StringBuffer buffer = new StringBuffer(128);
  static byte c;
  static byte x = 0; static byte y = 0;

  static void bufferMessage(){
    c=0;
    do{
      c = (byte)rxUart.receiveByte();
    } while (c != 33);
    x  = (byte)rxUart.receiveByte();
    y  = (byte)rxUart.receiveByte();
   }                                //end bufferMessage

  public static void main()  {

    while(true) {
      buffer.clear();
      bufferMessage();
      System.out.println(x);
      System.out.println(y);
    }                               //end while
  }                                 //end main
}                                   //end class
```

FCC Notice and Liability Disclaimer

These modules (boards) are not FCC approved. They are designed to comply with FCC Part 15 Rules and Regulations. They are not in a finished product form. They are strictly intended for experimental purposes only. If you wish to use these modules in an actual product (a non-experimental capacity), the module must first be designed into the product then the whole product must be approved by the FCC.

Parallax, Inc. is not responsible for special, incidental, or consequential damages resulting from any breach of warranty, or under any legal theory, including lost profits, downtime, goodwill, damage to or replacement of equipment or property, and any costs of recovering, reprogramming, or reproducing any data stored in or used with Parallax products.