HX11TR Ultrasonic Positioning Device

The HX11TR can be set up to operate as a ultrasonic signal receiver, ultrasonic transmitter, ultrasonic caller and ultrasonic transponder. It is small size and economical. It differs from the previous versions in that it is small size and only has one FCC RJ11 port. This port is RS485, RS422 and RS232 compatible, i.e. it can be connected directly to a microcontroller port, or the serial port of a personal computer PC.

Size: 80 x 40 x 20 mm

Labeling

The first number in the label line is the **primary address** (receiver ID). The second number in the label is the transmitter ID and the third and last number is the transponder ID. The user selects the caller ID.

The label 11510 / 27B / 7B indicates that the receiver ID is 11510, this is a decimal number and its hexadecimal equivalence is 2CF6 or two bytes 2C and F6 the binary code is 0010110011110110, rotate this number to the right and you have 0001011001111011 if the first six digits of the binary code are ignored then you have the transmitter ID which is 10 0111 1011, this translates to 27B hexadecimal. And the transponder ID is the lower significant byte of the receiver ID. The transponder ID is embedded in the time-tag response of the transponder.

Standard factory addressing starts at receiver ID 11362 and goes up from there. This ID equals the hexadecimal value 2C62, this value is represented by the ASCII characters , and b. Device 11362 for example when standard terminal program is used, and the user types in ,b it will reply by sending the contents of its ring buffer. If the ring buffer is empty it will respond with the character #.

Specification	
Supply Voltage	7-16Vdc
Current Consumption	40mA
Range	8m *
Rotational Angle α	+/- 50 degrees *
Shear Angle β	+/- 40 degrees *
Maximum Reception Rate	36 positions/second
Storage Capacity	30 tag encounters **

* See Limitations and Specification Section in Hx11Manual
** Only applies in synchronized modes.

The Serial Communication

Since the I/O port is RS485/RS422 and RS232 compatible, the device can be either connected directly to a PC port or be networked on lines kilometers long. See the RJ12 (RJ11) socket on the bottom.

On the network and in general, pins one and six have no real use. And if the voltage limit on these wires are breached the device may be damaged. Therefore we recommend that you use RJ11 (RJ12) plugs and four-wire telephone cable to connect the devices together. This way you reduce your chances of accidentally damaging a device.

The RS232 serial communication is negated therefore pin numbered four (green wire) should be used when connecting the HX11TR to a PC serial. The HX11 communication is simplex by nature, therefore only one wire and a ground is needed to carry the signal back and forth.

Operation

On startup the HX11TR copies the setup parameters from EEPROM permanent memory into its work registers (temporary registers). Before the parameters are loaded into the work registers, the program does a checksum evaluation of the parameters. If the checksum indicates the parameters may be flawed, it loads its default parameters into the work registers. It flushes the ring buffer (positioning data storage) on start up, and enters monitoring mode. In this mode, it logs the time of arrival of valid ultrasonic signals. If the device has been synchronized, the time of ultrasonic signal arrival is time from last synchronization. Otherwise, if the operation is asynchronous, the time is relative to last overflow of the timer. The time of arrival is appended to the identification code of the ultrasonic signal, received and stored in the HX11X ring buffer. If the identification code of the arriving signal is less than sixteen, then the HX11TR will view this as a call and transpond (respond) with a reply. It does this by mixing the value of the received ID, with it's own (transponder) identification code, and transmitting back that mixture. Note that the mixture is transmitted back after user specified delay (XpDelay). The delays should be selected to minimize the 13mS signal frame conflict. Background Operation (see flowchart below)

When the HX11TR receives $, it clears its timers upon the rising edge of the start-bit belonging to the character following the $. The latency of this synchronization is 200nS, and the timers overflow every 1.048576 seconds.

The device listens for ultrasonic signals and logs the identity and time of signal arrival in a ring (round) buffer. Data stored is referred to as the time-tag, where the time of arrival equals the time elapsed from synchronization, until a fixed point in the ultrasonic signal is detected.

If the device receives its own device ID (16bit) address over the serial lines, it will transmit all the data in the ring buffers. The ring buffer overflows for every 32 ultrasonic time tags received and the oldest time-tag gets overwritten

TxDelay

The constant txDelay is in terms of 4.096 milliseconds, txDelay of 128 equals roughly 524 milliseconds. To put all of this into perspective, suppose txDelay of 11362 was set to 1 and 11364 was set at 10. Then the sonic signal leaves 11362 4mS after sync, given that the signal must travel maximum 4 meters, it takes the signal which is 15mS long, about 12 mS to travel this distance. It takes another 10mS to transmit the results over the serial wire; hence the value 10 x 4 (40mS) is barely adequate. If the value is less than what is required, the signal may not be fully in the round buffer when the device is required to empty its contents. Therefore next time around there may be two values in the round buffers making the sampling erratic.

XpDelay

The xpDelay is a delay introduced between the instant the transponder receives a call, to the instant the transponder responds to the call. It can be useful to allow some of the residual signal from previous transmission to decay at least partially. The xpDelay is in terms of 4.096 milliseconds. If there is no callers in the system then the xpDelay becomes irrelevant.

IdOverride

Devices will respond to a transmitted Id under 16. So if one device sends a signal with an identification code less than 16, then any device receiving the code will respond by sending back a signal following xpDelay. The calling device will yield the time tag of the signal, (i.e. time of flight).

Modes of Operation

CtlByte = 1 (mode bit 0 set)

Once the time elapsed from synchronization equals time specified by the txDelay, then the device transmits an ultrasonic ID. This ID value is specified on the HX11 label. If a value less than 16 has been written into the override buffer, then this value becomes the transmitted ID, and is referred to as caller ID.

CtlByte = 2 (mode bit 1 set)

Once the time elapsed from synchronization equals time specified by the txDelay, the device transmits the contents of its own ring-buffer over the serial lines.

CtlByte = 3 (mode bit 0 and 1 set)

Once the time elapsed from synchronization equals time specified by the txDelay, then the device transmits an ultrasonic ID. This ID value is specified on the HX11 label. If a value less than 16 has been written into the override buffer, then this value becomes the transmitted ID, and is referred to as caller ID. The device will also clear its own timers, and transmit the contents of its own ring buffer over the serial lines. This operation repeats itself at interval delay equal to txDelay.

CtlByte: Mode bit 2 set

Transponder Disabled. The device will not transpond to any caller.

Control Characters

When certain characters are sent to the HX11TR through the serial connection, the device responds as follows

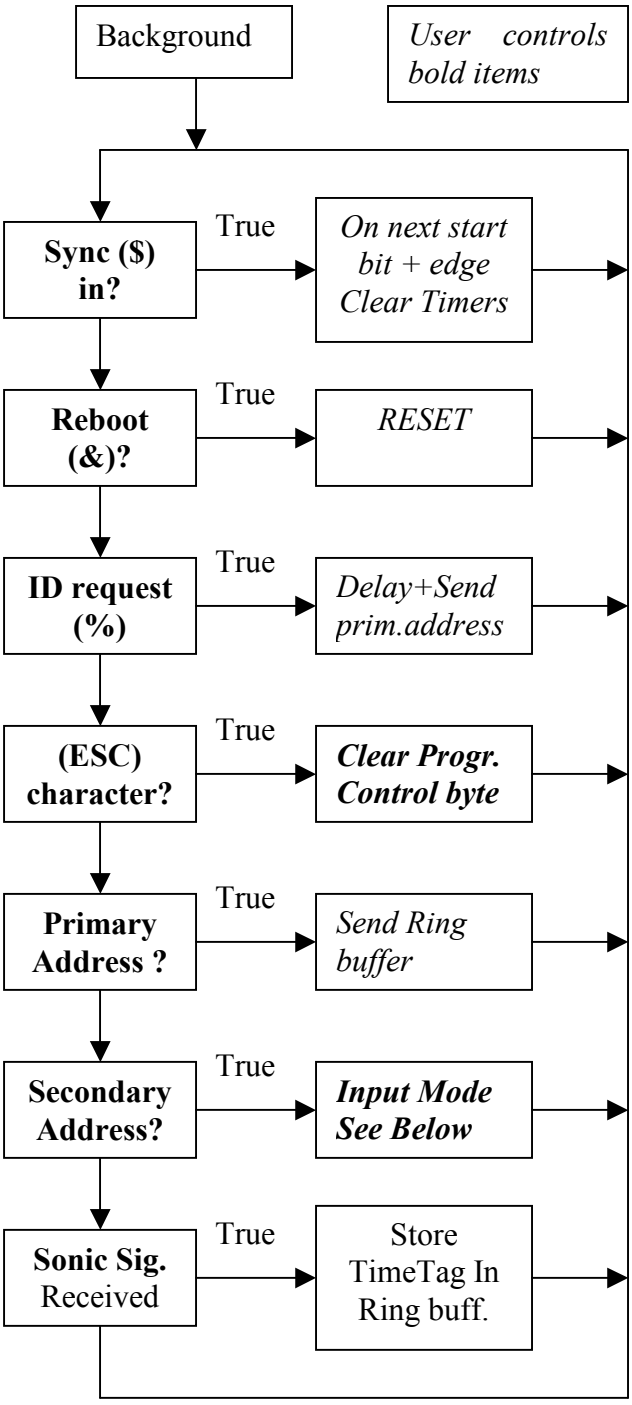
Charcter "$": The timers are cleared upon the receipt of the start bit belonging to the character that follows the character $. Latency 200nS.

Character "%": Upon receipt of this character, the device will respond after a delay, with its primary address in hexadecimal format. The delay is directly proportional to the value of its primary address.

Character "&": When this character is received, the device restarts.

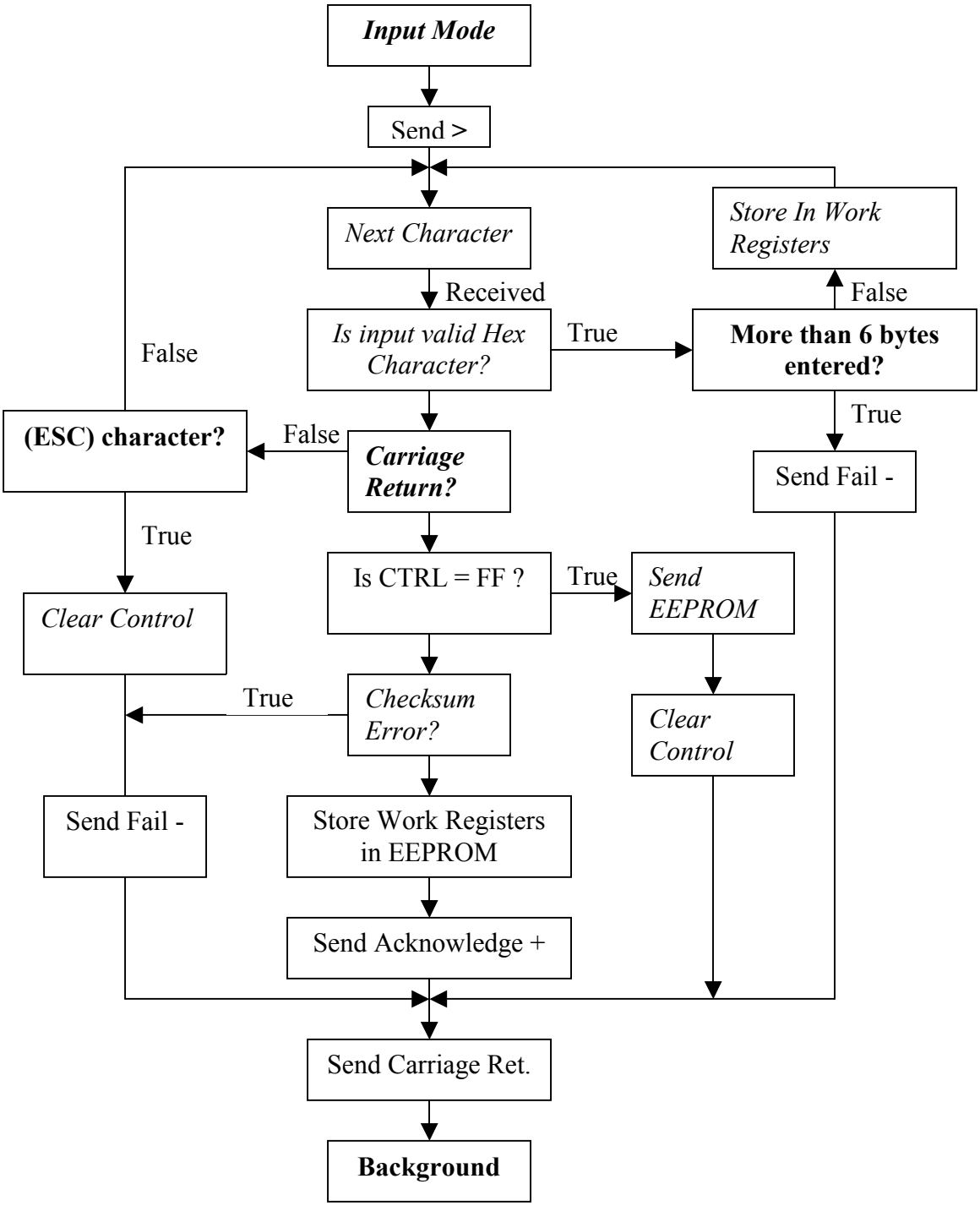
Character "ESC": The device clears the CTLBYTE when it receives this character.

The flow chart below illustrates the HX11TR background operation



Input Mode

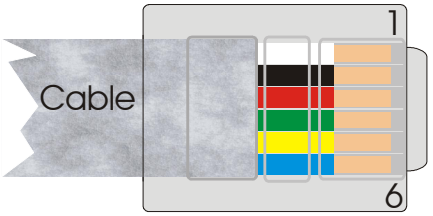
When the HX11TR receives its primary address plus one, the device sends a ">" character prompting for input. All data is entered in hexadecimal form with the alpha characters in capital letters.

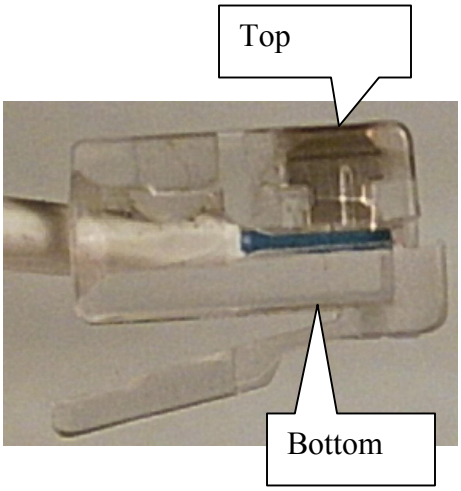


Wire Configuration HX11 devices using RJ12			
Contact	**Wire color**	**Wire function**	Electrical Specs
1	White	Parameter Recovery	0 - 7 Vdc *
2	Black	Ground / Return (Negative)	0V
3	Red	RS485 Positive **	+/- 7V
4	Green	RS485 Negative **	+/- 7V
5	Yellow	Power Input (Positive)	8-16Vdc
6	Blue	Device Hard Reset	0 – 7 Vdc *
** Current forced through the Mode and Reset pins should never exceed 20mA*			
*** To connect the HX11 to a RS232 port on a computer, a RS485 to RS232 or direct cable is required. RS485 is suited for long distance communication.*			

The RJ12 (RJ11)

The images to the right show a RJ12 jack and what we mean by top and bottom. It also shows how the colors are situated. The colors in the illustration to the right are listed 1 thru 6 looking from the top into the plug. This equals looking into the RJ12 socket, where the leftmost pin is pin 1 and the rightmost pin is pin 6.





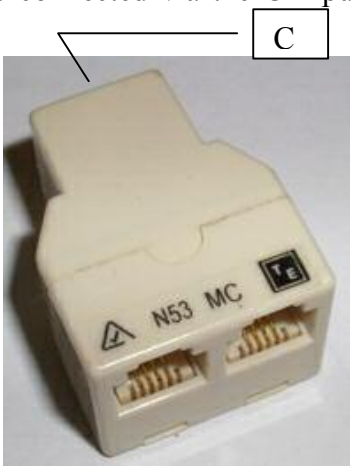
Baud rates and settings

The HX11TR communicates at 19200 baud, it uses one stop bit, no parity, and eight bits transfer.

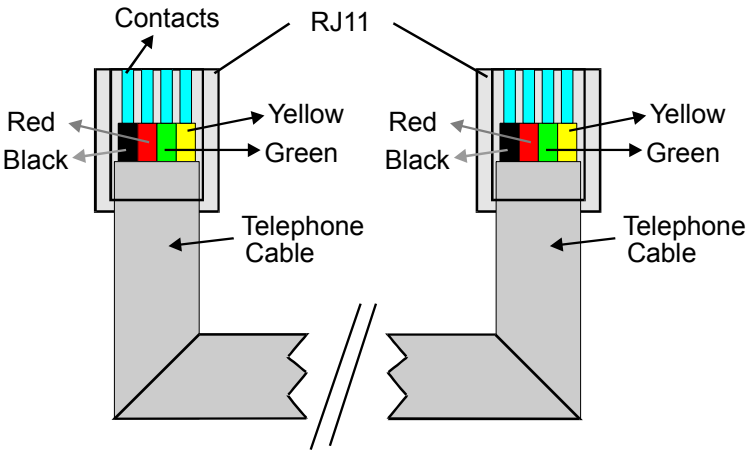
If RS485 transceiver is used to communicate with the hx11tr then the high Z rule for the transmit pin doesn't apply.

Using T connectors

T-Connectors are generally available at hardware stores selling telephone cables and accessories. The T connector joins 3 separate telephone cables, and it has 3 RJ11 or RJ12 sockets A, B and C. **The Hexamite network should have no wires crossed.** Some T connectors have the C input inverted. In this case the cable connected via the C input must be inverted too.



RJ11 and telephone cables

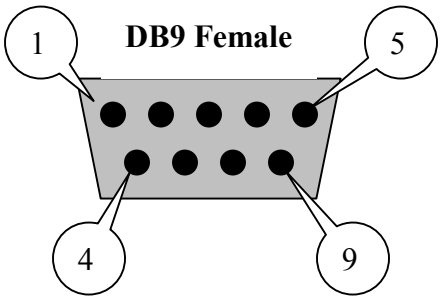


The figure on the right shows how the rj11 needs to be attached to the cable. It is important that the colors match, when looking down on both topsides of the cable end RJ11 plugs. **The cable connections should not be crossed.** Inexpensive tools are available, to help join the RJ11 to the telephone cable. It does not damage the monitors if they are inadvertently inverted. But it could block out the whole multidrop network until removed and corrected.

RS232 wiring

Connecting the HX11TR to a RS232 PC, palmtop, micro controller and etc.

When set up as caller the Hx11TR sends identification and position data streams through its RS485 pins. These pins can be connected directly to the RS232 port of a computer. **Note that the RS232 port receive and transmit pins, should be connected together.** For a private connection to a computer or a Microcontroller consult the following tables.



Pins on D-Subs 9 or 25 pins are always marked with the appropriate numbers for the pins as shown above.

Hx11TR RJ12 Network socket	DB9 (pins)	DB25 (older computers)
2 (Ground / return)	5 (Ground)	7 (Ground)
3 Pos RS485 Not Connected	Not Connected	Not Connected
4 Neg RS485	3 (Tx) and 2 (RX)	2 (Tx) and 3 (Rx)
5 Positive Supply 8-16Vdc	Not Connected	Not Connected

Connecting the HX11TR to a Microcontroller

The HX11TR serial protocol is simplex, both transmitted data and received data are conducted through the same line or pin. If you are only receiving data from the Hx11 devices then connect the serial receive pin on your Microcontroller, to the Pos RS485 i/o or RJ12 network socket pin 3. If you intend to configure the HX11 device, your microcontroller serial transmit and receive pin should be connected together. *And for multidrop and private operation using a standard port pin, all transmit pins should be in high Z or high impedance state when not transmitting.*

HX11TR RJ12 Network socket	Microcontroller
2 (Ground / return)	(Ground)
3 Pos RS485	Rx and (TX)
4 Pos RS485 Not Connected	Not Connected
5 Positive Supply 8-16Vdc	

RS232 to USB

If your interfacing device does not have RS232 port converters from USB to RS232 are readily available. Go to your nearest electronic store. These can be used to bridge the gap between the USB port and the Serial port.

Polling the HX11

When the device receives its binary address over the serial line (RS232/RS485/RS422) it responds with the oldest time-tag stored in its ring buffers, followed by the more recent time-tags until the buffer is empty.

HX11 Output format (time-tag)

The four most significant digits of the time-tag is used for the identification part, the least significant six digits are used for time representation. Following is a typical time-tag response from a HX11 device.

00021507DB #

The above indicates that the ultrasonic identity of transmitter 2 was received at clock count 1378267 from the last reset of the timers (Sync or reception of char. $ on serial lines).

00031399AC 000215080A #

The above content of the ring buffer indicates that the HX device polled, received the ultrasonic identity from transmitter 3 at count 1284524 and transmitter 2 at count 1378314 or (1378314 / 16000000) that is 0.086144625 seconds.

The counter increments sixteen times every microsecond.

Of the sixteen bits used for the ultrasonic identity only ten bits are actually used.

HX11 Output format (time-tag) transponders

The least significant for binary digits (nibble) of the sixteen digit binary identity section represents the caller. The higher significant nibble of the least significant byte in the identification word (16bits) represents the transponder's identity. So:

037F158A44 #

The above indicates that the polled device received the response from transponder 37 to call from caller F (16) at count 1411652.

And similarly

000C1A0290 02C8160B2A #

The polled device received a signal from caller C (12) at count 1704592 and it received the response from transponder 2C (44) to call from caller 8 at count 1444650.

Note that no transponder has the identity 0