WTS Technical Manual

Programming guide and advanced documentation

Document No: 23352 Rev a

Sherborne Sensors

.... the first choice in precision

Introduction / Overview

This manual contains advanced information on WTS telemetry range of devices. This includes more detailed information than supplied in the device manuals and also programming information.

To communicate with WTS devices a base station is required. Base stations will offer RS232, RS485 and USB interfaces.
Refer to the Base Station section **WTS-BSi and WTS-BSU** for details on connections and interfacing.

2.4GHz Radio General

Communicating with WTS Devices

To communicate with WTS devices a base station is required. Base stations offer RS232, RS485 and USB interfaces. In this section we will describe the interfaces and how data shall be sent to and retrieved from other WTS devices via the base station.
For details regarding device specific communications you will need to refer to the appropriate device section of the manual.

Packet Types

There are several different packet types, which are used depending on the type of data carried.
Read and Write packets are used to communicate with a device (When it is awake) and can read or write parameter values.
Some devices transmit data at regular intervals and this data does not need requesting.
Woken packets are received when a device is successfully woken.

Packet Structure

All packets conform to the following structure. The Data Packet part changes depending on what packet is being transported.

**Length	**Length	Base Address	Packet Type	Data Packet Structure	*CRC1 LSB	*CRC2 MSB
1 Byte	1 Byte	1 Byte	1 Byte	Variable Bytes	1 Byte	1 Byte
* CRC calculated on this part						
			** Length refers to this section			

This **Transport Packet** is used to carry the **Data Packets** into and out of the target device via the base station.

Where:

- Length bytes are identical and contain the length of just the **Data Packet** section.
- The CRC bytes are CRC 16 values of all bytes from Length up to and including Data section.
- Base Address is the address of the base station used where multiple base stations are deployed. Base station addresses can range from 1 to 16 and is set by DIP switches on the base station. **NOTE:** The WTS-BSU is fixed at address 1.
- The Packet Type byte defines the packet type thus defining the Data Packet Structure. In received packets this byte also indicates Error, Low Battery and Broadcast status.

Handling Base Station Data

The packets arriving at the base station serial or USB port are not hand shaken. Data may arrive as a partial packet or many packets may arrive together. Therefore the recommended best practice to handle data is to place arriving data into a circular buffer and to detect the packets from this buffer by looking for a length byte pair. Then look forward in the buffer at the CRC position (if the buffer contains enough bytes) and check whether the CRC is valid. If so you can extract and use the packet. If not then advance the start of the circular buffer until you find a matching byte pair then check for a valid CRC again.

CRC

The CRC algorithm is identical to that used in Modbus communications and should be calculated for outgoing packets and checked on incoming packets. The following BASIC example is of a function that will calculate the CRC of a string and append the two CRC bytes to the end of the string

```
SUB GenerateCRC16(sTarget AS STRING)
    'reads from buffer
    DIM CRC AS LONG
    DIM LSB AS INTEGER
    DIM C AS LONG
    DIM D AS INTEGER
    DIM Res(1) AS BYTE
    CRC = 65535
    FOR C = 1 TO LEN(sTarget)
        'xor byte
        CRC = CRC XOR ASC(MID$(sTarget, C, 1))
        FOR D = 1 TO 8
            'get lsb
            LSB = (CRC AND 1) = 1
            'move right
            CRC = INT(CRC / 2)
            'if LSB was 1 xor with polynomial
            IF LSB THEN CRC = CRC XOR (&HA001&)
        NEXT D
    NEXT C
    sTarget = sTarget & CHR$((CRC AND 255))
    sTarget = sTarget & CHR$(INT(CRC / 256))
END SUB
```

Packet Type Byte

The Packet Type bytes indicates the type of packet and holds information regarding Error, Low Battery and Broadcast status of received packets.

bit7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Error	LoBatt	Broadcast		Packet Type			
0	0	0	0	0	0	0	0

Bit	Function
Error	Bit indicated an error is present. This is set and reset by the device, which will include this information in the packet sent to the module.
LoBatt	Bit indicated a low battery. This is set and reset by the device, which will include this information in the packet sent to the module.
Broadcast	Used to indicate that a routed packet was broadcast so the receiver knows not to respond.

Value	Type	Description
3	Data Provider	Used to provide unrequested data.
5	Read	Read data from a specific device.
6	Write/command	Write a value or execute a command to a specific device
7	Response ACK	Response - Acknowledged. May also contain data.
8	Response NAK	Response - Not Acknowledged. The command was not recognised.
9	Response Timeout	Response Timed out. A response was not received by the device.
10	Response Data Invalid	Response – Data invalid. The device has reported that the data in a Write was invalid or out of range.

Data Packet Structures

The following structures show how the data is defined within the Data Packet Structure of the overall packet.

Data Provider

These packets are sent at intervals by some devices and contain data. There is no need to request these packets as they arrive automatically. If you have multiple base stations and these are within the range of the transmitting device the packets will arrive from each base station.

Packet Type	Data Tag		Status	Data Type	Data	RSSI	CV
03	00	00	00	00	[...]	00	00

Packet Type

This is 0x3 hex (3 decimal) and may have higher bits set which indicate Error, Low Battery and Broadcast.

Data Tag

Every device that transmits **Data Provider** packets has a configurable 2 byte **Data Tag**. Devices that consume Data Provider Packets can be configured to look for specific Data Tags. The reason we use Data Tags and not just rely on a devices ID for identification is that in a working system multiple devices may be relying on data from a single device. If that device were ever replaced then its unique ID would change and therefore multiple devices would have to be reconfigured. By using a Data Tag we only need to change this tag on the replacement device and the rest of the system will work as required.

Status

The bit values in this byte are used to indicate certain things. Only two bits are allocated a global meaning. The rest are device specific and you will need to refer to the device manual for clarification.

Status Byte							
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit0
X	X	X	X	X	X	Integrity	Shunt Cal

Data Type

This byte defines how the data is formatted in this packet and also indicates the best way to represent the data.

Function	Display As					Data Type		
Bit	7	6	5	4	3	2	1	0
Sample	0	1	1	1	1	0	0	1

Display As	Type	Description
0	Undefined	
1	Numeric	Numeric representation based on Data Type
2	Boolean	The data may be in any format but represents a boolean result where non zero numeric is True and string length > 1 or > 0 is True
3	Text	Can display as ascii text
4	Binary (unprintable)	Unprintable characters
5	Hex	Best represented as hex
6	Bit Map (10110101)	Each bit value should be shown
7	Percent	Numeric or string value has a value 0 - 100

Data Type	Description	Size In Bytes
0	No content/unknown	0
1	UINT8	1
2	UINT16	2
3	INT32	4
4	Float	4
5	String	0-64
6	Binary	0-64

NOTE: See **Data Type Formats** in **Appendix A**

The Display As bits should be used where possible as this can help in presenting the data for display purposes.

Data

This will be of variable length and will depend on the data type.

RSSI

This indicates the signal strength that this packet was received at. See **RSSI & CV** in **Appendix A**.

CV

This indicates correlation value which equates to the quality of the signal when this packet was received. See **RSSI & CV** in **Appendix A**.

See Advanced Data Provider Interface in

Read

The read packets are used to read parameters from a remote device or the base station itself. To talk to the base station just use the base station ID.

Packet Type	To ID			Command
05	00	00	00	00

Packet Type

This is 0x5 hex (5 decimal).

To ID

This is the ID of the device to read from (MSB first).

Command

The command number of the parameter you want to read. You will need to refer to the device manual for this information.

Write

The write packet is used to write parameter values to a device or execute commands.
You can write any supported data format to any other data format parameter but some formats are not very suitable. i.e. You can write an INT32 formatted value to a parameter that is just a UINT8 but if the value exceeds either the target data type limits or any other bounded limits imposed by the device you will receive an INVALID_DATA response.
When executing a command you do not need any data so it is usual to specify the data type as No Content (zero) and not include any data.

Packet Type	To ID			Command	Data Type	Data
06	00	00	00	00	00	[...]

Packet Type

This is 0x6 hex (6 decimal).

To ID

This is the ID of the device to write to from (MSB first). You can use the broadcast ID here of 0xFFFFFF (255 decimal for each of the To ID bytes) but be careful as this will write the value to all devices on the same channel and encryption key. You may have mixed device types so command numbers between devices may be different. Use broadcast with care.

Command

The command number of the parameter you want to write to. You will need to refer to the device manual for this information.

Data Type

Specify the data type of the data you are sending.
Data types are as follows:

Data Type	Description	Size In Bytes
0	No content/unknown	0
1	UINT8	1

2	UINT16	2
3	INT32	4
4	Float	4
5	String	0-64
6	Binary	0-64

NOTE: See **Data Type Formats** in **Appendix A**

Responses to Read and Write…

The response to either a read or write can be as follows:
Responses to Read: ACK, NAK, TIMEOUT
Responses to Write: ACK, NAK, TIMEOUT, DATAINVALID

ACK

If the ACK response is for a write then it will not contain data:

Packet Type	From ID			RSSI	CV
07	00	00	00	00	00

Packet Type
This is 0x7 hex (7 decimal) and may have higher bits set which indicate Error, Low Battery and Broadcast.

From ID
This contains the ID of the device that sent the packet.

RSSI
This indicates the signal strength that this packet was received at. See **RSSI & CV** in **Appendix A**.

CV
This indicates correlation value which equates to the quality of the signal when this packet was received. See **RSSI & CV** in **Appendix A**

If the ACK is in response to a READ then it will contain data:

Packet Type	From ID			Data Type	Data	RSSI	CV
07	00	00	00	00	[…]	00	00

Packet Type

This is 0x7 hex (7 decimal) and may have higher bits set which indicate Error, Low Battery and Broadcast.

From ID

This contains the ID of the device that sent the packet.

Data Type

This byte defines how the data is formatted in this packet and also indicates the best way to represent the data.

Function	Display As					Data Type		
Bit	7	6	5	4	3	2	1	0
Sample	0	1	1	1	1	0	0	1

Display As	Type	Description
0	Undefined	
1	Numeric	Numeric representation based on Data Type
2	Boolean	The data may be in any format but represents a boolean result where non zero numeric is True and string length > 1 or > 0 is True
3	Text	Can display as ASCII text
4	Binary (unprintable)	Unprintable characters
5	Hex	Best represented as hex
6	Bit Map (10110101)	Each bit value should be shown
7	Percent	Numeric or string value has a value 0 - 100

Data Type	Description	Size In Bytes
0	No content/unknown	0
1	UINT8	1
2	UINT16	2
3	INT32	4
4	Float	4
5	String	0-64
6	Binary	0-64

NOTE: See **Data Type Formats** in **Appendix A**

RSSI

This indicates the signal strength that this packet was received at. See **RSSI & CV** in **Appendix A**.

CV

This indicates correlation value, which equates to the quality of the signal when this packet was received. See **RSSI & CV** in **Appendix A**

NAK

This packet is returned if the device receiving the read or write does not recognize the command number.

Packet Type	From ID			RSSI	CV
08	00	00	00	00	00

Packet Type

This is 0x8 hex (8 decimal) and may have higher bits set which indicate Error, Low Battery and Broadcast.

From ID

This contains the ID of the device that sent the packet.

RSSI

This indicates the signal strength that this packet was received at. See **RSSI & CV** in **Appendix A**.

CV

This indicates correlation value, which equates to the quality of the signal when this packet was received. See **RSSI & CV** in **Appendix A**

TIMEOUT

This packet is returned if the device does not respond.

Packet Type	From ID			RSSI	CV
09	00	00	00	00	00

Packet Type

This is 0x9 hex (9 decimal) and may have higher bits set which indicate Error, Low Battery and Broadcast.

From ID

This contains the ID of the device that sent the packet.

RSSI

This indicates the signal strength that this packet was received at. See **RSSI & CV** in **Appendix A**.

CV

This indicates correlation value, which equates to the quality of the signal when this packet was received. See **RSSI & CV** in **Appendix A**

DATA INVALID

This packet is returned if the device has been written to and the data written is invalid.

Packet Type	From ID			RSSI	CV
0A	00	00	00	00	00

Packet Type

This is 0xA hex (10 decimal) and may have higher bits set which indicate Error, Low Battery and Broadcast.

From ID

This contains the ID of the device that sent the packet.

RSSI

This indicates the signal strength that this packet was received at. See **RSSI & CV** in **Appendix A**.

CV

This indicates correlation value, which equates to the quality of the signal when this packet was received. See **RSSI & CV** in **Appendix A**

Pairing…

Pairing is a method of communicating between two devices so that they configure themselves to one or another's radio settings and enables them to identify each other by means of ID and default Data Tag.
Additionally the pairing mechanism can pause a device from performing its default behaviour as some devices operate in a low power mode where they are mostly asleep. This makes communications impossible so the pairing process stops the low power behaviour.
The pairing process is usually initiated by one device (a handheld for example or PC software using a base station) and this enters pairing master mode and is ready to pair for a user defined time period.

Next the other device is put into its pair mode at which time it negotiates with the other device and they will decide on what to do based on their function. Devices are normally put into pair mode by power cycling them. Please refer to the device documentation.

For the scope of this document we are concerned with manually controlling the pairing using a base station.
You would use pairing for the following reasons:

- You have a device whose communications settings are unknown.
- You want to connect to a device that operates in low power mode and is mostly asleep.
- The device you want to talk to may have a normal operational function that you want pausing while you configure it.
- You just want to change the communications settings of a device to match it to a base station settings.

Pair Request

Packet Type	Data Tag		Direction	Config	Duration (optional)
13	00	00	00	00	00

Packet Type

This is 0x13 hex (19 decimal).

Data Tag

You need to supply a Data Tag, which may be useable by the device to which you are pairing. Usually from a base station this is not required and can be set to 0x00, 0x00.
This is present as the same mechanism is used when two devices pair together and in that case they will each want to give the other their default Data Tag.

Direction

The value of this byte determines whether the remote device radio settings are configured to match the base station or if the base station is changed to match the remote device. Both the Channel and the encryption key are matched once pairing has completed.

Value	Meaning
0	The settings in the remote device are changed to match the base station settings.
1	The base station settings are changed to match the remote device.

Config

The value of this byte determines whether the remote device will enter configuration mode, which will inhibit any low power operation, transmission of data provider packets and the ability to enter deep sleep mode. This mode is required otherwise communication whilst configuring could be very poor or impossible.

Value	Meaning
0	Do not change operation.
1	Cause the device to enter config mode to enable it to be configured.

NOTE: after pairing with a device and using the **Config** option it is recommended that the device be power cycled after so that it resumes its normal operation.

Duration

The value of this byte determines whether the base station will be in pair mode for the default time of 5 seconds or whether to use the user defined duration in seconds.
To use the default just omit this byte. If this byte is present its ASCII value will be used to determine how long it will be in pairing mode.

NOTE: While in pairing mode the base station will not operate as normal.

If the remote device enters its own pair mode (non master) then the communications negotiations will take place and the device may come out of any low power modes and a response will be sent to the base station.

Pair Response

This packet will arrive at the base station if another device enters pair mode while the base station is waiting to pair. Once this packet has arrived the base station will be free to talk to the device. It can also determine the device ID and Default Data Tag if it needs this information.

Packet Type	From ID			Data Tag		RSSI	CV
14	00	00	00	00	00	00	00

Packet Type

This is 0x14 hex (20 decimal) and may have higher bits set which indicate Error, Low Battery and Broadcast.

From ID

This contains the ID of the device that paired.

Data Tag

This contains the default Data Tag of the paired device.

RSSI

This indicates the signal strength that this packet was received at. See **RSSI & CV** in **Appendix A**.

CV

This indicates correlation value, which equates to the quality of the signal when this packet was received. See **RSSI & CV** in **Appendix A**

NOTE: After configuration or calibration you should power cycle the paired device to return it to normal operation.

WTS-BSi and WTS-BSu [Base Station]

Overview

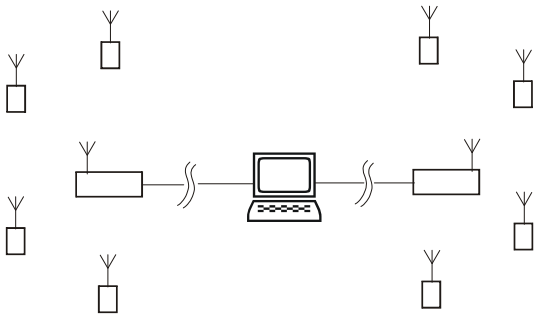
These devices are base stations and interface between the radio and a physical interface for a connection to a PC, PLC or other device.

Addressing

Usually only a single base station is required in a telemetry installation. If a telemetry device is outside the range of the base station a repeater may be deployed.
Some complex topologies may only be realized by using multiple base stations, which may require changes to the **Address** switches. See Multipoint Base Station Section.

Multipoint Base Stations

Sometimes more than one base station is required in a system. This may simply be a central PC with two base stations wired off in opposite directions.



Where one base station handles devices on the left and the other those on the right.
Multiple base stations allow flexibility in routing requests from a PC as each time a packet is sent to a base station it is targeted to a particular base station **Address**. When a packet arrives back at a PC it contains the **Address** of the base station that routed it.
If a base station is the only one connected to a particular serial port then every base station can have **Address** 1 as the PC will send packets to a particular port to select which base station handles a packet, likewise packets arriving back at the PC will be identified by the port that they arrive on.
RS232 devices can only be connected one at time anyway but RS485 allows multiple devices on the same bus. This is where the **Addressing** is vital as it is this that distinguishes between base stations.
When using USB base stations you may only ever have one WTS-BSu connected to a PC at any time. Using WTS-BSi will allow multiple USB base stations to be connected to single PC. Unique **Addresses** are again required in this instance.

Note that broadcast packets can be received by multiple base stations so packets may appear duplicated at the PC end.
It is also possible for the PC to route a packet through all connected base stations by **Addressing** a packet to Address 0.

NOTE: Although it is possible to connect multiple USB base stations to a PC the **T24drv.dll** driver supplied only supports one Base Station **Addressed** as 1.
So when using a WTS-BSi which supports USB but also has a DIP switch to allow **Address** setting the **Address** must be set to 1. The WTS-BSu devices are manufactured with a fixed **Address** of 1.

Connection

The interface can be selected from the DIP switches SW1 as can baud rates for serial interfaces and the Address of the base station.

SW1 Settings (WTS-BSi only)
Switch positions 1 to 4 select the base station **Address**. This should normally be 1.

	1	2	3	4
Address				
1	Off	Off	Off	Off
2	On	Off	Off	Off
3	Off	On	Off	Off
4	On	On	Off	Off
5	Off	Off	On	Off
6	On	Off	On	Off
7	Off	On	On	Off
8	On	On	On	Off
9	Off	Off	Off	On
10	On	Off	Off	On
11	Off	On	Off	On
12	On	On	Off	On
13	Off	Off	On	On
14	On	Off	On	On
15	Off	On	On	On
16	On	On	On	On

Switch positions 5 to 7 set whether serial or USB is used. If USB is not selected then the chosen switch settings control the baud rate for the serial interface. Whether the serial interface is RS485 or RS232 is selected by switch position 8.

	5	6	7
Baud rate / USB			
USB	Off	Off	Off
9600	On	Off	Off
19200	Off	On	Off
38400	On	On	Off
57600	Off	Off	On
115200	On	Off	On
230400	Off	On	On
460800	On	On	On

NOTE:

A baud rate of 9600 (and in some cases 19200) is not suitable for 2 way communication with remote devices as it is too slow and causes timeouts. This baud rate has been included to enable the base station to be connected to a 9600 baud device to allow low rate Data Provider packets to be received.

At any rate below 230400 is may be possible to lose packets at high data rates as the serial cannot keep pace with the radio transmissions.

If USB is not selected as the interface (Switch positions 5 to 7) then this switch position selects whether the serial interface is RS232 or RS485.

	8
232/485	
RS232	Off
RS485	On

Interfaces

RS232

The RS232 interface uses TX, RX and GND to connect to a PC, PLC etc and uses standard RS232 voltage levels.

Handshaking	None
Data Size	8 bits
Stop Bits	1 bit
Parity	None

The baud rate can be selected by setting the DIP switches stated above. NOTE: the base station will require power cycling to utilise a baud rate change.

Example connection to a PC 9 way D serial connector.

PC 9 Way D Plug Pin	Signal Direction	Base Station Connection	
3 (TX)	->	RX	J6 Pin 2 or J7 Pin 3
2 (RX)	<-	TX	J6 Pin 3 or J7 Pin 2
5 (Gnd)		GND	J6 Pin 4 or J7 Pin 5
8 (CTS)	<-	CTS	J6 Pin 1 or J7 Pin 8

RS485

The RS485 interface (This is a 2 wire 485 interface and will not work with 4 wire 485 buses) uses TX, RX and GND to connect to a PC, PLC etc and uses standard RS485 voltage levels.

Handshaking	None
Data Size	8 bits
Stop Bits	1 bit
Parity	None

The baud rate can be selected by setting the DIP switches stated above. NOTE: the base station will require power cycling to utilise a baud rate change.

Example connection

Depending on the RS485 interface or hardware the connections vary and are not standard therefore we can only show the connections to the base station. You must refer to the user manual regarding your RS485 connection to ascertain the correct connections.

PC / PLC Connection	Signal Direction	Base Station Connection	
Refer to RS485 Device User Manual		A	J4 Pin 3 or J5 Pin 3
Refer to RS485 Device User Manual		B	J4 Pin 4 or J5 Pin 4
Refer to RS485 Device User Manual		GND	J4 Pin 5 or J5 Pin 5

NOTE: There are two connectors for RS485, J4 and J5. This is to facilitate easy daisy chaining of devices if required.

Serial Limitations

- When using RS232 or RS485 you should use the fastest baud rate possible. At lower rates data can be lost because it can arrive from the radio faster than the base station can send it serially.
- At 9600 baud you will experience communications problems when configuring devices. This baud rate is too slow for anything other than monitoring data provider packets from devices and even then these should be at a low rate (around 20 per second). The slow baud rates are provided to get low rate data into older systems.
- RS485 is a bus master system and is not ideally suited to full communications with devices when multiple devices are providing data. This is fine for the normal operation of data acquisition but it is recommended that only the device to be configured is active during configuration.

USB

Connection to the base station will be either a captive USB cable (WTS-BSu) or a USB socket B for connection using a standard USB A-B cable (WTS-BSi). There is an optional cable assembly for the WTS-BSi to provide for a USB connection while the device is still fitted to the ABS case.
To communicate with the base station the connected device must use the USB HID Device Class and support USB 2.0 full speed interface (12mbits).
The USB connection will also power the base station.

USB Communications

Using the supplied **T24drv.dll** driver is the easiest way to communicate with the base station. However, if you want to write your own communications software you will need the following information:
Vendor ID: 6017 (0x1781 hex)
Product ID: 2980 (0xBA4 hex)
Incoming packets are read from report 0 and contain 64 bytes of data. There will always be 64 bytes even if there are only a few bytes of valid data. These bytes will need placing into a buffer and your software will need to detect and extract complete packets.
Outgoing data is written to report 0 and must always contain 64 bytes of data. Any unused bytes should be set to zero.
Please note that 65 bytes of data are actually sent and received but the first byte indicates the report number so this is always zero.

Parameter	Value
Vendor ID	0x1781 (6017 decimal)
Product ID	0xBA4 (2980 decimal)
Setup Class	HIDClass
Service Name	HidUsb

Parameter	Value
Connection Information	
ConnectionIndex	0x1
CurrentConfigurationValue	0x1
LowSpeed	FALSE
DeviceIsHub	FALSE
DeviceAddress	0x1
NumberOfOpenPipes	0x2
Pipe #0 Endpoint Descriptor	
bLength	0x7
bEndpointAddress	0x1 [OUT]
bmAttributes	0x3 (USB_ENDPOINT_TYPE_INTERRUPT)
wMaxPacketSize	0x40
bInterval	0x1
Pipe #1 Endpoint Descriptor	
bLength	0x7
bEndpointAddress	0x82 [IN]
bmAttributes	0x3 (USB_ENDPOINT_TYPE_INTERRUPT)
wMaxPacketSize	0x40
bInterval	0x1
Device Descriptor	
bLength	0x12
bcdUSB	0x110
bDeviceClass	0x0
bDeviceSubClass	0x0
bDeviceProtocol	0x0
bMaxPacketSize0	0x8
idVendor	0x1781
idProduct	0xBA4
bcdDevice	0x100
iManufacturer	0x1
iProduct	0x2
iSerialNumber	0x3
bNumConfigurations	0x1

Parameter	Value
Configuration Descriptor	
bLength	0x9
bDescriptorType	USB_CONFIGURATION_DESCRIPTOR_TYPE
wTotalLength	0x29
bNumInterfaces	0x1
iConfiguration	0x0
bmAttributes	0x80 (Bus_Powered)
MaxPower	0x64
Interface Descriptor	
bLength	0x9
bInterfaceNumber	0x0
bAlternateSetting	0x0
bNumEndpoints	0x2
bInterfaceClass	0x3 (Human Interface Device)
bInterfaceSubClass	0x0 (No Subclass)
bInterfaceProtocol	0x0 (None)
iInterface	0x0
Endpoint Descriptor	
bLength	0x7
bEndpointAddress	0x1 [OUT]
bmAttributes	0x3 (USB_ENDPOINT_TYPE_INTERRUPT)
wMaxPacketSize	0x40
bInterval	0x1
Endpoint Descriptor	
bLength	0x7
bEndpointAddress	0x82 [IN]
bmAttributes	0x3 (USB_ENDPOINT_TYPE_INTERRUPT)
wMaxPacketSize	0x40
bInterval	0x1

NOTE: If you do not want to use the supplied communications DLL (T24drv.dll) you may be interested in the following:
We have successfully tested Easy HID, which supplies the mcHID.dll, which is a great generic way of connecting to HID devices. This library is free and was written to ease both the programming of PIC devices and create sample code for VB, Delphi and Visual C++.
The USB interface has also been successfully used with Windows CE but we do not supply any drivers for this operating system.

LED Indication

Two LEDS indicate Power/Mode and Activity.
The red LED indicates mode and should flash at a 2Hz rate. If any errors are detected with the radio then the LED will remain lit.
The green LED flashes once for each packet received or transmitted via radio, USB or serial.

Communications

To configure the base station, you will use the Read and Write mechanisms described in the **Data Packet Structures** section to read and write parameters and execute commands.
The base station may also be receiving packets from other devices. These will be Data Provider Packets and these may arrive at any time.

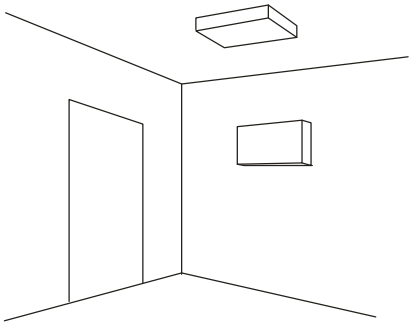
Parameter List

<table>
<tr><th>Parameter</th><th>Command Number</th><th>Description</th><th>Native Data Type</th><th>Read / Write</th></tr>
<tr><td>ID</td><td>3</td><td>Read the unique identifier ID for this device. (3 bytes)</td><td>BINARY 3 byte</td><td>R</td></tr>
<tr><td>Channel</td><td>11</td><td>The radio channel to operate on. (1-16) Requires power cycle or Reset to enable.</td><td>UINT8</td><td>RW</td></tr>
<tr><td>EncKey</td><td>15</td><td>The radio encryption key to operate on. Requires power cycle or Reset to enable. (16 bytes) Not supported in this release.</td><td>BINARY 16 bytes</td><td>RW</td></tr>
<tr><td>UseCSMA</td><td>18</td><td>Select whether to use Carrier Sense Multiple Access techniques on transmission.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Disabled
The Carrier Sense Multiple Access will be disabled. NOT RECOMMENDED</td></tr><tr><td>1</td><td>Enabled
The Carrier Sense Multiple Access will be enabled.</td></tr></table>See Unslotted CSMA/CA in Appendix A</td><td>UINT8</td><td>RW</td></tr>
<tr><td>Power</td><td>12</td><td>Set or read the output power level. (0-100%)</td><td>UINT8</td><td>RW</td></tr>
<tr><td>Name</td><td>10</td><td>Set or read a user defined name. (11 characters)</td><td>STRING 11 bytes</td><td>RW</td></tr>
<tr><td>WakerDuration</td><td>17</td><td>Set or read the duration in milliseconds to wait for a device to wake. Although WAKE commands are sent to the target device the base station actually intercepts this and handles the wake itself. Default = 12000 milliseconds.</td><td>UINT16</td><td>RW</td></tr>
<tr><td>Save</td><td>24</td><td>Save any changes made to parameters. Required before power cycling or issuing a Reset command. Requires 200mS recovery time after executing.</td><td colspan="2">Command</td></tr>
<tr><td>Reset</td><td>25</td><td>Restarts the device and utilises new channel and encryption keys if those have been changed and saved. Note after a Reset the device will be asleep.</td><td colspan="2">Command</td></tr>
</table>

NOTE: All changes require a SAVE command to enable them to survive through power cycle or RESET command.

Installation

The base WTS-BSi should be mounted horizontally on a wall or ceiling so that the side face containing the PCB antenna faces the general direction of the target devices.



The WTS-BSu should also be positioned to present itself to the other radios in a 'landscape' aspect.

Specification

Parameter	Minimum	Typical	Maximum	Units	Notes
WTS-BSi External Supply voltage Range	9	12	32	Volts	
WTS-BSi, WTS-BSu USB Supply Range	4.875	5	5.125	Volts	As defined by USB 2.0 Specification
Average Operational Current (WTS-BSi)	-	TBD	500	mA	
USB Bus Powered Operational Current	100		200		
Operating Temperature Range	-40	-	65	Deg C	
Storage Temperature Range	-40	-	65	Deg C	
Reverse polarity Protection		-	-32	Volts	Maximum Supply level

WTS Wireless Inclinometer

Overview

The WTS-n-nn is a ” sine of angle” acquisition module. This allows wireless remote viewing of ” sine of angle” information using 2.4GHz radio and periodically transmits it.
Between transmissions the device is optionally in a power saving sleep mode to conserve batteries.

Communications

To configure the device you will use the Read and Write mechanisms described in the **Data Packet Structures** section to read and write parameters and execute commands.

Parameter List

Parameter	Command Number	Description	Native Data Type	Read / Write
ID	3	Read the unique identifier ID for this device. (3 bytes)	BINARY 3 bytes	R
Version	53	Read the firmware version.	STRING 11 Bytes	R
Channel	11	Radio Channel	UINT8	RW
EncKey	15	The radio encryption key to operate on. Requires power cycle or Reset to enable. Not supported in this release.	BINARY 16 Bytes	RW
Power	12	Set or read the output power level. (0-100%)	UINT8	RW
Name	10	Set or read a user defined name. (11 characters)	STRING 11 Bytes	RW
Model	51	Read the model number of the device.	STRING 11 Bytes	R
WakeChkInt	16	Set or read the interval in milliseconds that a sleeping device will wake to request a full wake from the base station. (default 3000)	UINT16	RW
UseCSMA	18	Select whether to use Carrier Sense Multiple Access techniques on transmission. Value: 0 — **Disabled** The Carrier Sense Multiple Access will be disabled. NOT RECOMMENDED Value: 1 — **Enabled** The Carrier Sense Multiple Access will be enabled. See **Unslotted CSMA/CA** in **Appendix A**	UINT8	RW
BattLevel	69	The voltage measured on the battery.	FLOAT	R
BattLowLevel	110	Set or read the battery voltage at which the low battery flag will be set in all received packets and in the Status parameter).	FLOAT	RW
LowPowerMode	75	Defines power save mode. 0 = Awake all the time and transmit at TxInterval. 1 = Sleep. Wake at TXInterval, acquire value, transmit value, sleep.	UINT8	RW
TxInterval	76	Time Interval between Transmissions. Set mS	UINT16	RW
SampleTime	78	Set or read the SampleTime in milliseconds for acquiring a reading at each TxInterval interval. The larger the SampleTime the more accurate the reading but at the expense of battery life. (Default 5)	UINT16	RW
SleepDelay	77	Time period before switching to low power sleep mode if no StayAwake command or trigger received. Setting to zero disables.	UINT16	RW
DataTag	119	Set or read the 2 byte Data Tag that is used when transmitting the weight in a Data Provider Packet. Note that default value is set to last 2 bytes of ID.	UINT16	RW

<table>
<tr><td>NumCalPoints</td><td>79</td><td>Number of calibration points required. Range 2 to 9</td><td>UINT8</td><td>RW</td></tr>
<tr><td>CalPoint1</td><td>80</td><td>The % full scale value of this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPoint2</td><td>81</td><td>The % full scale value of this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPoint3</td><td>82</td><td>The % full scale value of this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPoint4</td><td>83</td><td>The % full scale value of this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPoint5</td><td>84</td><td>The % full scale value of this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPoint6</td><td>85</td><td>The % full scale value of this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPoint7</td><td>86</td><td>The % full scale value of this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPoint8</td><td>87</td><td>The % full scale value of this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPoint9</td><td>88</td><td>The % full scale value of this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPointGain1</td><td>89</td><td>The gain to apply to the input at this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPointGain</td><td>90</td><td>The gain to apply to the input at this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPointGain</td><td>91</td><td>The gain to apply to the input at this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPointGain</td><td>92</td><td>The gain to apply to the input at this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPointGain</td><td>93</td><td>The gain to apply to the input at this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPointGain</td><td>94</td><td>The gain to apply to the input at this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPointGain</td><td>95</td><td>The gain to apply to the input at this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPointGain</td><td>96</td><td>The gain to apply to the input at this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPointGain</td><td>97</td><td>The gain to apply to the input at this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPointOffset1</td><td>98</td><td>The offset to apply to the input at this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPointOffset2</td><td>99</td><td>The offset to apply to the input at this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPointOffset3</td><td>100</td><td>The offset to apply to the input at this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPointOffset4</td><td>101</td><td>The offset to apply to the input at this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPointOffset5</td><td>102</td><td>The offset to apply to the input at this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPointOffset6</td><td>103</td><td>The offset to apply to the input at this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPointOffset7</td><td>104</td><td>The offset to apply to the input at this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPointOffset8</td><td>105</td><td>The offset to apply to the input at this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>CalPointOffset9</td><td>106</td><td>The offset to apply to the input at this calibration point.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>Value</td><td>72</td><td>Reads the calibrated output.</td><td>FLOAT</td><td>R</td></tr>
<tr><td>Counts</td><td>70</td><td>Raw A/D counts</td><td>INT32</td><td>R</td></tr>
<tr><td>PFS</td><td>71</td><td>Factory calibrated percent full scale (0=0V, 100=10V)</td><td>FLOAT</td><td>R</td></tr>
<tr><td>DigitalOut</td><td>108</td><td>Read or set the digital output state.<table><tr><th>Value</th><th>Definition</th></tr><tr><td>0</td><td>Turn digital output off</td></tr><tr><td>1</td><td>Turn digital output on</td></tr><tr><td>2</td><td>Make digital output mirror the LED state (V1.05 and later)</td></tr></table></td><td>FLOAT</td><td>RW</td></tr>
<tr><td>Status</td><td>74</td><td>Live status where the bit values indicate status. The following table shows the decimal bit values and their meaning.<table><tr><th>Bit Value</th><th>Definition</th></tr><tr><td>1</td><td>STATUS_SHUNT_CAL</td></tr><tr><td>2</td><td>STATUS_INPUT_INTEGRITY</td></tr><tr><td>4</td><td>Reserved</td></tr><tr><td>8</td><td>Reserved</td></tr><tr><td>16</td><td>STATUS_POWER_UP</td></tr><tr><td>32</td><td>STATUS_BATT_LOW</td></tr><tr><td>64</td><td>STATUS_DIGITAL_INPUT</td></tr><tr><td>128</td><td>STATUS_DIGITAL_OUTPUT</td></tr></table></td><td></td><td></td></tr>
</table>

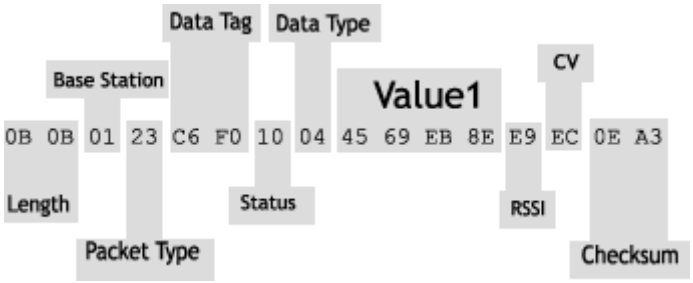
Reset	54	Restarts the device and utilises new channel and encryption keys if those have been changed and saved.	Command	
Save	55	Save any changes made to parameters. Required before power cycling or issuing a Reset command. Requires 500mS recovery time after executing.	Command	
Sleep	56	Sends the TSC into Sleep	Command	
Pause	57	Stops the output streaming to allow configuration	Command	
Resume	59	Streaming continues.	Command	
StayAwake	58	No function other than to reset SleepDelay timer	Command	
SleepTrigger	120	Enter Data Tag to watch for that will trigger sleep mode	UINT16	RW
PauseTrigger	121	Enter Data Tag to watch for that will trigger pause mode	UINT16	RW
ResumeTrigger	123	Enter Data Tag to watch for that will trigger resume mode	UINT16	RW
StayAwakeTrigger	122	Enter Data Tag to watch for that will reset SleepDelay timer.	UINT16	RW
DoSystemZeroTrigger	125	Enter Data Tag to watch for that will reset SleepDelay timer.	UINT16	RW

NOTE: All changes require a SAVE command to enable them to survive through power cycle or RESET command.

Parameters in grey are for factory calibration only – any changes will invalidate calibration and warranty

Data Provider Format

At every TXInterval a Data Provider packet is transmitted that holds 1 value in FLOAT format (See Appendix A).



Measurement Resolution

The noise free resolution is dependent on the Sample Time (SampleTime)

SampleTime (mS)	Noise Free Resolution	Ratio
> 0	15.5 bits	1:50,000
> 9	16 bits	1:65,000
> 49	17.25 bits	1:150,000
> 99	18 bits	1:250,000
> 999	18.75 bits	1:400,000

WTS-HS [Handheld Reader Simple]

Overview

The WTS-HS captures Data Provider data and displays it. The WTS-HS also performs the function of waking the remote device when it is turned on and sending it to deep sleep mode when it is turned off. If no buttons are pressed on the WTS-HS it will turn off after 5 minutes.

Communications

To configure the device you will use the Read and Write mechanisms described in the **Data Packet Structures** section to read and write parameters and execute commands.

Parameter List

<table>
<tr><th>Parameter</th><th>Command Number</th><th>Description</th><th>Native Data Type</th><th>Read / Write</th></tr>
<tr><td>ID</td><td>3</td><td>Read the unique identifier ID for this device. (3 bytes)</td><td>BINARY 3 bytes</td><td>R</td></tr>
<tr><td>Version</td><td>53</td><td>Read the firmware version.</td><td>STRING 11 Bytes</td><td>R</td></tr>
<tr><td>Channel</td><td>11</td><td>Radio Channel</td><td>UINT8</td><td>RW</td></tr>
<tr><td>EncKey</td><td>15</td><td>The radio encryption key to operate on. Requires power cycle or Reset to enable. Not supported in this release.</td><td>BINARY 16 Bytes</td><td>RW</td></tr>
<tr><td>Power</td><td>12</td><td>Set or read the output power level.
(range:0 to 100 default:100)</td><td>UINT8</td><td>RW</td></tr>
<tr><td>Name</td><td>10</td><td>Set or read a user defined name. (11 characters)</td><td>STRING 11 Bytes</td><td>RW</td></tr>
<tr><td>Model</td><td>51</td><td>Read the model number of the device.</td><td>STRING 11 Bytes</td><td>R</td></tr>
<tr><td>InputValue</td><td>60</td><td>The value that is being read by the device.</td><td>FLOAT</td><td>R</td></tr>
<tr><td>DisplayValue</td><td>61</td><td>The value being displayed by the device.</td><td>FLOAT</td><td>R</td></tr>
<tr><td>UseCSMA</td><td>18</td><td>Select whether to use Carrier Sense Multiple Access techniques on transmission.

<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Disabled
The Carrier Sense Multiple Access will be disabled. NOT RECOMMENDED</td></tr><tr><td>1</td><td>Enabled
The Carrier Sense Multiple Access will be enabled.</td></tr></table>
See Unslotted CSMA/CA in Appendix A</td><td>UINT8</td><td>RW</td></tr>
<tr><td>BattLevel</td><td>69</td><td>The voltage measured on the battery.</td><td>FLOAT</td><td>R</td></tr>
<tr><td>OffDelay</td><td>62</td><td>Time period in minutes before switching off if no button is pressed. Setting to zero disables.
(range:0 to 1440 default:5)</td><td>UINT16</td><td>RW</td></tr>
<tr><td>BoundDataTag</td><td>68</td><td>Set or read the 2 byte Data Tag that is used to match a Data Provider packet to use as the display value.</td><td>UINT16</td><td>RW</td></tr>
<tr><td>BoundID</td><td>69</td><td>Set or read the ID of the bound device. Used to wake the remote device.</td><td>BINARY 3 Bytes</td><td>RW</td></tr>
<tr><td>AutoZero</td><td>63</td><td>Set or read the value limit which may be automatically zeroed on start-up.
When the handheld powers up and the input value is within ±AutoZero then the display will be zeroed. The value of the input will be placed in AutoZero. i.e. from this time onwards until powered off the display will show
input value - AutoZero</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>DoSleepWake</td><td>64</td><td>Set or read whether to perform wake and sleep on the paired device when the handheld is powered up and</td><td>UINT8</td><td>RW</td></tr>
</table>

		down. (range:0 to 1 default:1)		
Timeout	65	Set or read the time in seconds that if exceeded between receiving Data Provider packets from the paired device will cause the display to show ------. Default = 3 (range:0 to 65535 default:3)	UINT16	RW
KeepAwakeInt	66	Set or read the interval in seconds between the handheld transmitting StayAwake signals to the paired device. (range:0 to 65535 default:5)	UINT16	RW
DisplayUpdate	74	Set or read the interval between LCD updates. Default = 300	UINT16	RW
PairWait	67	Set or read the period in seconds that the handheld will wait for another device to pair when placed into pair mode. (range:0 to 65535 default:5)	UINT8	RW
Format	70	Set or read the format for the display. Here you can define how the value will be displayed and where the decimal point will appear. By including a non zero value this will define the resolution of the displayed value. i.e. the smallest step size of value changes. Default = 0000.001	STRING 8 bytes	RW
ZeroSupp	71	Set or read whether to show the value on the display with zero suppression. 0 = No zero suppression 1 = Zero suppression	UINT8	RW
ZeroBand	72	Set or read the band within which zero will be displayed. As soon as the value exceeds ±ZeroBand the actual value will be displayed. This will effectively mask small changes after taring the device.	FLOAT	RW
Overload	73	Set or read the value of the display above which instead of the value being displayed 'Overload' will be displayed.	FLOAT	RW
ScaleInLo	75	Set or read a low input value at which you know what display you require. Default = 0	FLOAT	RW
ScaleInHi	77	Set or read a high input value at which you know what display you require. Default = 1	FLOAT	RW
ScaleDisplayLo	76	Set or read a low display value for the input value stated in ScaleInLo. Default = 0	FLOAT	RW
ScaleDisplayHi	78	Set or read a high display value for the input value stated in ScaleInHi. Default = 1	FLOAT	RW
Reset	54	Restarts the device and utilises new channel and encryption keys if those have been changed and saved.	Command	
Save	55	Save any changes made to parameters. Required before power cycling or issuing a Reset command. Requires 500mS recovery time after executing.	Command	

NOTE: All changes require a SAVE command to enable them to survive through power cycle or RESET command.

Power Supply

Recommend using alkaline AA cells as rechargeable are too low voltage and lithium may not be able to supply the current for the radio to start up.

Installation

There are no specific installation instructions.

WTS-HA [Handheld Reader Advanced]

Overview

The WTS-HS captures Data Provider data from multiple devices and displays it. The WTS-HS also performs the function of optionally waking the remote device when it is turned on and sending it to deep sleep mode when it is turned off. If no buttons are pressed on the WTS-HS it will turn off after 5 minutes.
The handheld can operate in two modes. The operation of the buttons and the automatic sleep/wake functions are dependent on these modes.

To attach devices to the handheld we must first ensure that the appropriate devices are transmitting their values at a suitable rate such as the default of 3 per second. Then we can tell the handheld which devices we want to communicate with by either automatic pairing or manual setting of the parameters.

Automatic Pairing

The handheld supports pairing replace an existing device. i.e. initially the handheld would be configured using manual configuration.
If then a WTS-SA device required replacing and this was item 3 in the handheld we would follow the following procedure:

- First select this device even though the display may show ------ because of no communications. (In Result mode you would have to hold the **Next** key to enable selection of individual devices.)
- Next turn off the handheld then press and hold the **Power** key then press and hold the **Tare** key. Now both keys are held down until **PAIRING** is seen on the display. The keys can now be released.
- Now apply power to the WTS-SA within 10 seconds.
- If unsuccessful the display will show **failed**. If successful the **Data Tag** and **ID** of the new device will have been configured within the handheld.
- Depending on how the replacement device was configured it **may** be necessary to reapply the system zero.

Manual Configuration

Use the parameters **ValueDataTag1** through to **ValueDataTag12** and **ValueID1** through to **ValueID12** and enter the Data Tags and IDs of the devices to connect to.
For example, to sum two WTS-H-NR devices whose Ids are FFF123 and FFFABC.
The default data tags for these devices would be F123 and FABC so we would set the following:

ValueDataTag1=F123
ValueID1=FFF123
ValueDataTag2=FABC
ValueID2=FFFABC

We would ensure that the other unused **ValueDataTagx** and **ValueIDx** parameters were set to zero.

Available Modes

Item Mode

Opmode = 0.
Up to 12 individual devices can be connected to and the user can step through each one in sequence.
If DoSleepWake is set then the handheld will wake all configured devices when turned on and send them all to sleep again when turned off. NOTE: When the handheld wakes devices this achieved through the transmission of a broadcast wake. i.e. all devices on the same channel and with the same encryption key will wake.

Key Operation

Key	Operation
Sleep	Send the currently selected device to sleep.
Wake	Will attempt to wake the currently selected device.
Tare	Toggle between
Next	Step to the next device.
F1	If motion detection is activated then the reading must be steady to enable this key. Pressing this key with an unstable reading will do nothing. This transmits a Data Provider packet marked with a Data Tag held in **F1DataTag** and can also contain data as defined by **F1Data**.

Power	Toggles between on and off. Hold for 2 seconds to activate.

Result Mode

OpMode = 1.
Up to 12 individual devices can be summed and the result displayed.
If **DoSleepWake** is set then the handheld will wake all configured devices when turned on and send them to sleep again when turned off. NOTE: When the handheld wakes devices this achieved through the transmission of a broadcast wake. i.e. all devices on the same channel and with the same encryption key will wake.

In this mode there is an option of retrieving a system zero value from an external source. This is activated by supplying the Data Tag to the **ExtZeroDataTag** parameter. When activated the value supplied by the Data Provider packet marked with this tag will be used as the system zero and will be subtracted from the sum of all contributing inputs.
Usually in this mode only the result is displayed (In the initial devices the result will just be the sum) but by holding the **Next** key for 65 seconds will activate the ability to step through each contributing input using the **Next** key. See **AllowNext**

Key Operation when viewing the sum

Key	Operation
Sleep	No effect
Wake	Will attempt to wake any sleeping devices.
Tare	Toggle between displaying gross sum or tared sum.
Next	No effect unless held for 5 seconds to activate individual item view. This can be disabled by setting **NoNext** parameter.
F1	If motion detection is activated then the reading must be steady to enable this key. Pressing this key with an unstable reading will do nothing. This transmits a Data Provider packet marked with a Data Tag held in F1DataTag and can also contain data as defined by F1Data.
Power	Toggles between on and off. Hold for 2 seconds to activate.

Key Operation when viewing an individual item

Key	Operation
Sleep	No effect
Wake	Will attempt to wake any sleeping devices.
Tare	If sum was currently tared then this key will toggle between displaying gross or tared value of current device. If sum view was displaying gross then this key has no effect. If an external system zero is used then only gross values actually supplied to the handheld can be displayed.
Next	Selects next device to view.
F1	If motion detection is activated then the reading must be steady to enable this key. Pressing this key with an unstable reading will do nothing. This transmits a Data Provider packet marked with a Data Tag held in F1DataTag and can also contain data as defined by F1Data.
Power	Toggles between on and off. Hold for 2 seconds to activate.

Communications

To configure the device you will use the Read and Write mechanisms described in the **Data Packet Structures** section to read and write parameters and execute commands.

Parameter List

<table>
<tr><th>Parameter</th><th>Command Number</th><th>Description</th><th>Native Data Type</th><th>Read / Write</th></tr>
<tr><td>ID</td><td>3</td><td>Read the unique identifier ID for this device. (3 bytes)</td><td>BINARY 3 bytes</td><td>R</td></tr>
<tr><td>Version</td><td>53</td><td>Read the firmware version.</td><td>STRING 11 Bytes</td><td>R</td></tr>
<tr><td>Channel</td><td>11</td><td>Radio Channel</td><td>UINT8</td><td>RW</td></tr>
<tr><td>EncKey</td><td>15</td><td>The radio encryption key to operate on. Requires power cycle or Reset to enable. Not supported in this release.</td><td>BINARY 16 Bytes</td><td>RW</td></tr>
<tr><td>Power</td><td>12</td><td>Set or read the output power level. (0-100%)</td><td>UINT8</td><td>RW</td></tr>
<tr><td>Name</td><td>10</td><td>Set or read a user defined name. (11 characters)</td><td>STRING 11 Bytes</td><td>RW</td></tr>
<tr><td>Model</td><td>51</td><td>Read the model number of the device.</td><td>STRING 11 Bytes</td><td>R</td></tr>
<tr><td>UseCSMA</td><td>18</td><td>Select whether to use Carrier Sense Multiple Access techniques on transmission.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Disabled
The Carrier Sense Multiple Access will be disabled. NOT RECOMMENDED</td></tr><tr><td>1</td><td>Enabled
The Carrier Sense Multiple Access will be enabled.</td></tr></table>See Unslotted CSMA/CA in Appendix A</td><td>UINT8</td><td>RW</td></tr>
<tr><td>OffDelay</td><td>62</td><td>Time period in seconds before switching off if no button is pressed. Setting to zero disables. Default = 5</td><td>UINT16</td><td>RW</td></tr>
<tr><td>AutoZero</td><td>63</td><td>Set or read the value limit which may be automatically zeroed on start-up.
When the handheld powers up and the input value is within ±ZeroBand then the display will be zeroed. The value of the input will be placed in AutoZero. i.e. from this time onwards until powered off the display will show
input value - AutoZero</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>DoSleepWake</td><td>64</td><td>Set or read whether to perform wake and sleep on the paired device when the handheld is powered up and down. Default = 1</td><td>UINT8</td><td>RW</td></tr>
<tr><td>Timeout</td><td>65</td><td>Set or read the time in seconds that if exceeded between receiving Data Provider packets from the paired device will cause the display to show ------. Default = 3</td><td>UINT16</td><td>RW</td></tr>
<tr><td>KeepAwakeInt</td><td>66</td><td>Set or read the interval in seconds between the handheld transmitting StayAwake signals to the paired device. Default = 5</td><td>UINT16</td><td>RW</td></tr>
<tr><td>DisplayUpdate</td><td>74</td><td>Set or read the interval between LCD updates. Default = 300</td><td>UINT16</td><td>RW</td></tr>
<tr><td>PairWait</td><td>67</td><td>Set or read the period in seconds that the handheld will wait for another device to pair when placed into pair mode. Default = 5</td><td>UINT8</td><td>RW</td></tr>
<tr><td>Format</td><td>70</td><td>Set or read the format for the display. Here you can define how the value will be displayed and where the decimal point will appear. By including a non zero value this will define the resolution of the displayed value. i.e. the smallest step size of value changes. Default = 0000.001</td><td>STRING 8 bytes</td><td>RW</td></tr>
<tr><td>ZeroSupp</td><td>71</td><td>Set or read whether to show the value on the display with zero suppression.</td><td>UINT8</td><td>RW</td></tr>
</table>

<table>
<tr><td></td><td></td><td>0 = No zero suppression
1 = Zero suppression</td><td></td><td></td></tr>
<tr><td>ZeroBand</td><td>72</td><td>Set or read the band within which zero will be displayed. As soon as the value exceeds ±ZeroBand the actual value will be displayed. This will effectively mask small changes after taring the device.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>Overload</td><td>73</td><td>Set or read the value of the display above which instead of the value being displayed 'Overload' will be displayed.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>ScaleInLo</td><td>75</td><td>Set or read a low input value at which you know what display you require. Default = 0</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>ScaleInHi</td><td>77</td><td>Set or read a high input value at which you know what display you require. Default = 1</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>ScaleDisplayLo</td><td>76</td><td>Set or read a low display value for the input value stated in b. Default = 0</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>ScaleDisplayHi</td><td>78</td><td>Set or read a high display value for the input value stated in ScaleInHi. Default = 1</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>OpMode</td><td>122</td><td>Selects the operational mode.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Items Mode
Each of the configured input values are displayed one at a time and the Next button can be used to step through them.</td></tr><tr><td>1</td><td>Result Mode
The input values are summed and displayed.</td></tr></table></td><td>UINT8</td><td>RW</td></tr>
<tr><td>F1Data</td><td>128</td><td>Set or read what data to transmit when the F1 key is pressed.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Always Gross
The gross value will be transmitted even if the display shows a zeroed net reading.</td></tr><tr><td>1</td><td>As Displayed
The gross or net value will be transmitted depending on what is selected at the time.</td></tr></table></td><td>UINT8</td><td>RW</td></tr>
<tr><td>F1DataTag</td><td>127</td><td>Set or read the 2 byte Data Tag to use in the Data Provider packet that is transmitted when the F1 key is pressed.</td><td>UINT16</td><td>RW</td></tr>
<tr><td>ExtZeroDataTag</td><td>80</td><td>Set or read the 2 byte Data Tag of the Data Provider Packet that will supply a system zero value. This value will be subtracted from the gross or net summed values. This allows the same handheld to be used with different sets of data providers each supplying its own system zero value.</td><td>UINT16</td><td>RW</td></tr>
<tr><td>ExtZeroID</td><td>100</td><td>Set or read the ID of the device acting as external system zero.</td><td>BINARY 3 Bytes</td><td>RW</td></tr>
<tr><td>MotionBand</td><td>123</td><td>Specify an engineering units band that the readings must stay within for the duration of the MotionTime for the reading to be considered steady.</td><td>FLOAT</td><td>RW</td></tr>
<tr><td>MotionTime</td><td>124</td><td>Specify the duration in seconds for the motion detection to operate.</td><td>UINT8</td><td>RW</td></tr>
<tr><td>MsgDuration</td><td>125</td><td>Specifies the duration in milliseconds that messages</td><td>UINT16</td><td>RW</td></tr>
</table>

		are displayed. i.e. the message that shows **Input 1** or **Input 2** as items are selected.		
ItemDuration	126	Specifies the duration in seconds that individual item values are displayed (in Result mode) before automatically switching back to display the sum.	UINT8	RW
ValueDataTag1	81	Set or read the 2 byte Data Tag that is used to match a Data Provider packet to use as the input value 1.	UINT16	RW
ValueID1	101	Set or read the ID of the device acting as input 1. Used to wake the remote device when Wake key pressed in Items Mode.	BINARY 3 Bytes	RW
ValueDataTag2	82	Set or read the 2 byte Data Tag that is used to match a Data Provider packet to use as the input value 1.	UINT16	RW
ValueID2	102	Set or read the ID of the device acting as input 1. Used to wake the remote device when Wake key pressed in Items Mode.	BINARY 3 Bytes	RW
ValueDataTag3	83	Set or read the 2 byte Data Tag that is used to match a Data Provider packet to use as the input value 1.	UINT16	RW
ValueID3	103	Set or read the ID of the device acting as input 1. Used to wake the remote device when Wake key pressed in Items Mode.	BINARY 3 Bytes	RW
ValueDataTag4	84	Set or read the 2 byte Data Tag that is used to match a Data Provider packet to use as the input value 1.	UINT16	RW
ValueID4	104	Set or read the ID of the device acting as input 1. Used to wake the remote device when Wake key pressed in Items Mode.	BINARY 3 Bytes	RW
ValueDataTag5	85	Set or read the 2 byte Data Tag that is used to match a Data Provider packet to use as the input value 1.	UINT16	RW
ValueID5	105	Set or read the ID of the device acting as input 1. Used to wake the remote device when Wake key pressed in Items Mode.	BINARY 3 Bytes	RW
ValueDataTag6	86	Set or read the 2 byte Data Tag that is used to match a Data Provider packet to use as the input value 1.	UINT16	RW
ValueID6	106	Set or read the ID of the device acting as input 1. Used to wake the remote device when Wake key pressed in Items Mode.	BINARY 3 Bytes	RW
ValueDataTag7	87	Set or read the 2 byte Data Tag that is used to match a Data Provider packet to use as the input value 1.	UINT16	RW
ValueID7	107	Set or read the ID of the device acting as input 1. Used to wake the remote device when Wake key pressed in Items Mode.	BINARY 3 Bytes	RW
ValueDataTag8	88	Set or read the 2 byte Data Tag that is used to match a Data Provider packet to use as the input value 1.	UINT16	RW
ValueID8	108	Set or read the ID of the device acting as input 1. Used to wake the remote device when Wake key pressed in Items Mode.	BINARY 3 Bytes	RW
ValueDataTag9	89	Set or read the 2 byte Data Tag that is used to match a Data Provider packet to use as the input value 1.	UINT16	RW
ValueID9	109	Set or read the ID of the device acting as input 1. Used to wake the remote device when Wake key pressed in Items Mode.	BINARY 3 Bytes	RW
ValueDataTag10	90	Set or read the 2 byte Data Tag that is used to match a Data Provider packet to use as the input value 1.	UINT16	RW
ValueID10	110	Set or read the ID of the device acting as input 1. Used to wake the remote device when Wake key pressed in Items Mode.	BINARY 3 Bytes	RW
ValueDataTag11	91	Set or read the 2 byte Data Tag that is used to match a Data Provider packet to use as the input value 1.	UINT16	RW
ValueID11	111	Set or read the ID of the device acting as input 1. Used to wake the remote device when Wake key pressed in Items Mode.	BINARY 3 Bytes	RW
ValueDataTag12	92	Set or read the 2 byte Data Tag that is used to match a Data Provider packet to use as the input value 1.	UINT16	RW
ValueID12	112	Set or read the ID of the device acting as input 1. Used	BINARY 3	RW

		to wake the remote device when Wake key pressed in Items Mode.	Bytes	
AllowNext	129	Determines whether to allow the ability to view individual items when in Result mode. Set to zero to disable or set to a number representing the number of seconds to hold down the **Next** key to activate this feature. Once activated the **Next** key steps through all input values. Once the device is powered off this feature would have to be activated again. (range:0 to 30 default:6)	UINT8	RW
AllowSysZero	130	Determines whether to allow the ability to perform system zero by pressing and holding the **Tare** key. Set to zero to disable or set to a number representing the number of seconds to hold down the Tare key to perform the system zero. (range:0 to 30 default:12)	UINT8	RW
Reset	54	Restarts the device and utilises new channel and encryption keys if those have been changed and saved.	Command	
Save	55	Save any changes made to parameters. Required before power cycling or issuing a Reset command. Requires 500mS recovery time after executing.	Command	
DoSysZero	120	Perform a system zero on all devices. This will remove the current input values so from this point on the current input will give a value of zero. This can be removed by issuing the **RmSysZero** command.	Command	
RmSysZero	121	Reset the system zero settings so values will represent the actual inputs.	Command	

NOTE: All changes require a SAVE command to enable them to survive through power cycle or RESET command.

Power Supply

Recommend using alkaline AA cells as rechargeable are too low voltage and lithium may not be able to supply the current for the radio to start up.

Installation

There are no specific installation instructions.

WTS-HR [Handheld Reader Roaming]

Overview

The WTS-HR is a roaming handheld that can be used to view the reading supplied by an unlimited number of acquisition modules. The acquisition Data Tags or IDs do not need to be known beforehand.

The handheld will automatically wake any device on the same channel and encryption key.
An internal list is maintained of the top ***n*** number of acquisition modules ordered by signal level and a **Next** key on the handheld allows cycling through this list.
The list size (***n***) is variable between 2 and 20 and this enables the viewing experience to be tailored to particular applications.

The acquisition modules are identified by their 4 character hexadecimal Data Tags.

When in communication with a particular acquisition module the LED on that module is activated. This provides visual feedback of the selected and currently viewed module. The LED output can also appear optionally on the digital output.

Communications

To configure the device you will use the Read and Write mechanisms described in the **Data Packet Structures** section to read and write parameters and execute commands.

Parameter List

<table>
<tr><th>Parameter</th><th>Command Number</th><th>Description</th><th>Native Data Type</th><th>Read / Write</th></tr>
<tr><td>ID</td><td>3</td><td>Read the unique identifier ID for this device. (3 bytes)</td><td>BINARY 3 bytes</td><td>R</td></tr>
<tr><td>Version</td><td>53</td><td>Read the firmware version.</td><td>STRING 11 Bytes</td><td>R</td></tr>
<tr><td>Channel</td><td>11</td><td>Radio Channel</td><td>UINT8</td><td>RW</td></tr>
<tr><td>EncKey</td><td>15</td><td>The radio encryption key to operate on. Requires power cycle or Reset to enable. Not supported in this release.</td><td>BINARY 16 Bytes</td><td>RW</td></tr>
<tr><td>Power</td><td>12</td><td>Set or read the output power level.
(range:0 to 100 default:100)</td><td>UINT8</td><td>RW</td></tr>
<tr><td>Name</td><td>10</td><td>Set or read a user defined name. (11 characters)</td><td>STRING 11 Bytes</td><td>RW</td></tr>
<tr><td>Model</td><td>51</td><td>Read the model number of the device.</td><td>STRING 11 Bytes</td><td>R</td></tr>
<tr><td>InputValue</td><td>60</td><td>The value that is being read by the device.</td><td>FLOAT</td><td>R</td></tr>
<tr><td>DisplayValue</td><td>61</td><td>The value being displayed by the device.</td><td>FLOAT</td><td>R</td></tr>
<tr><td>UseCSMA</td><td>18</td><td>Select whether to use Carrier Sense Multiple Access techniques on transmission.
<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Disabled
The Carrier Sense Multiple Access will be disabled. NOT RECOMMENDED</td></tr><tr><td>1</td><td>Enabled
The Carrier Sense Multiple Access will be enabled.</td></tr></table>See Unslotted CSMA/CA in Appendix A</td><td>UINT8</td><td>RW</td></tr>
<tr><td>BattLevel</td><td>69</td><td>The voltage measured on the battery.</td><td>FLOAT</td><td>R</td></tr>
<tr><td>OffDelay</td><td>62</td><td>Time period in minutes before switching off if no button is pressed. Setting to zero disables.
(range:0 to 1440 default:5)</td><td>UINT16</td><td>RW</td></tr>
<tr><td>KeyTest</td><td>64</td><td>Set or read the state of the keys. Each time the keys are pressed the equivalent bit will be set.
Set to zero to reset. This property is used in ATE to test the keypad.</td><td>UINT8</td><td>RW</td></tr>
</table>

Timeout	65	Set or read the time in seconds that if exceeded between receiving Data Provider packets from the paired device will cause the display to show ------. Default = 3 (range:0 to 65535 default:3)	UINT16	RW
PairWait	67	Set or read the period in seconds that the handheld will wait for another device to pair when placed into pair mode. (range:0 to 65535 default:5)	UINT8	RW
Format	70	Set or read the format for the display. Here you can define how the value will be displayed and where the decimal point will appear. By including a non zero value this will define the resolution of the displayed value. i.e. the smallest step size of value changes. Default = 0000.001	STRING 8 bytes	RW
ZeroSupp	71	Set or read whether to show the value on the display with zero suppression. 0 = No zero suppression 1 = Zero suppression	UINT8	RW
Overload	73	Set or read the value of the display above which instead of the value being displayed 'Overload' will be displayed.	FLOAT	RW
DisplayUpdate	74	Set or read the interval between LCD updates. Default = 300	UINT16	RW
ScaleInLo	75	Set or read a low input value at which you know what display you require. Default = 0	FLOAT	RW
ScaleInHi	77	Set or read a high input value at which you know what display you require. Default = 1	FLOAT	RW
ScaleDisplayLo	76	Set or read a low display value for the input value stated in ScaleInLo. Default = 0	FLOAT	RW
ScaleDisplayHi	78	Set or read a high display value for the input value stated in ScaleInHi. Default = 1	FLOAT	RW
ListSize	79	Set or read the size of the internal list of Data Tags. Can be between 1 and 20.	UINT8	RW
Reset	54	Restarts the device and utilises new channel and encryption keys if those have been changed and saved.	Command	
Save	55	Save any changes made to parameters. Required before power cycling or issuing a Reset command. Requires 500mS recovery time after executing.	Command	

NOTE: All changes require a SAVE command to enable them to survive through power cycle or RESET command.

Power Supply

Recommend using alkaline AA cells as rechargeable are too low voltage and lithium may not be able to supply the current for the radio to start up.

Installation

There are no specific installation instructions.

Windows Driver DLL

Overview

A standard Windows driver DLL is supplied which can be used by many development tools.
This DLL has been created to simplify communications with the WTS range of telemetry devices.
This handles circular buffers for incoming data and provides signalling to the host application of incoming packets via a callback mechanism.
The DLL can be used with a serial port or a USB bus.
Where strings are used these are in a format that can be used by Visual Basic. If used in Delphi or C Builder these functions need to be declared as StdCall.
Below you will find the API declarations for Visual Basic which may be used as a guide for other languages:

```
Public Declare Function VERSION Lib "T24Drv.dll" () As Single

Public Declare Sub INITIALISE Lib "T24Drv.dll" (ByRef iCallbackAddress As Long)

Public Declare Function OPENPORT Lib "T24Drv.dll" (ByVal ComPort As Integer, ByVal Baudrate As Long) As
Integer

Public Declare Function CLOSEPORT Lib "T24Drv.dll" () As Integer

Public Declare Function OPENUSB Lib "T24Drv.dll" () As Integer

Public Declare Sub ENUMUSB Lib "T24Drv.dll" ()

Public Declare Function BIND Lib "T24Drv.dll" (ByVal BaseStation As Byte, ByVal UseRemoteSettings As
Byte, ByVal ConfigMode As Byte, ByVal Duration As Byte, ByRef ID As Long, ByRef DataTag As Long) As
Integer

Public Declare Function BINDASYNC(ByVal BaseStation As Byte, ByVal UseRemoteSettings As Byte, ByVal
ConfigMode As Byte, ByVal Duration As Byte) As Integer

Public Declare Function BINDASYNCPOLL(ByRef ID As Long, ByRef DataTag As Long) AS Integer

Public Declare Function READREMOTE Lib "T24Drv.dll" (ByVal BaseStation As Byte, ByVal ID As Long, ByVal
Command As Byte, ByRef sData As Byte, ByRef Length As Long, ByRef RSSI As Integer, ByRef CV As Integer,
ByRef flags As Integer) As Integer

Public Declare Function WRITEREMOTE Lib "T24Drv.dll" (ByVal BaseStation As Byte, ByVal ID As Long, ByVal
Command As Byte, ByRef sData As Byte, ByVal Length As Long, ByRef RSSI As Integer, ByRef CV As Integer,
ByRef flags As Integer) As Integer
```

Functions and Subs

INITIALISE

Initialise the DLL and setup the callback address so that when packets arrive the hosting application receives a callback.

`SUB INITIALISE(BYREF iCallbackAddress AS LONG)`

Where:

Parameter	Description
iCallbackAddress	long pointer to the address of the callback procedure.

Required before OPENUSB or OPENPORT is called. Pass the address of the callback function. The format of the function (In Visual Basic) is

```
Sub CallBack(ByRef StringPtr As Long, Length As Long)
```

To extract the packet data the following **Visual Basic** example may prove useful:

```
Sub CallBack(ByRef StringPtr As Long, Length As Long)
    Dim PacketString As String
    Dim PacketByteArray(128) As Byte
    'To get the packet into a byte array
    CopyMemory PacketByteArray(0), ByVal StringPtr, Length
    'To get the packet into a string
    PacketString = Left$(StrConv(PacketByteArray(), vbUnicode), Length)
End Sub
```

VERSION

Return the driver version.

`FUNCTION VERSION() AS SINGLE`

Returns:
The version number in floating point format.

OPENPORT

Open a serial port for communications.

`FUNCTION OPENPORT(BYVAL ComPort AS INTEGER, BYVAL Baudrate AS LONG) AS INTEGER`

Where:

Parameter	Description
ComPort	Long variable indicating the COM port to open.
Baudrate	Long value representing the actual required baudrate. The PC port will generate to closest available baudrate to this value.

Returns:

Integer Value	Description
0	Port opened OK.
-1	Invalid settings. The serial port exists but could not be configured to the required settings including the requested baudrate.
-2	Could not open serial port at all. Either the port does not exist or another application has opened this port.

This function must be called before transmitting any packets and once called will activate the callbacks as packets are received by the base station.

OPENUSB

Opens communications with the USB bus. This function does not rely on a base station being present to be a success.

`FUNCTION OPENUSB() AS INTEGER`

Returns:

Integer Value	Description
0	USB bus opened OK.
-2	Could not initialize the USB bus.

This function must be called before transmitting any packets and once called will activate the callbacks as packets are received by the base station.

ENUMUSB

This sub **can** be called every few seconds to allow changes on the USB bus to be enumerated. Most of the time this will change nothing but if a device has been plugged or unplugged from the bus calling this will enumerate the remaining devices and gracefully cope with additions or removals of the base stations. As this can take up to

250mS to complete (When a change has been detected) it is recommended that this be called only every few seconds.

```
SUB ENUMUSB()
```

This should only be called after OPENUSB has been successful. If a base station is present when OPENUSB is called and it is going to stay connected then there is no need to call this method.

CLOSEPORT

Close any open serial ports or USB bus connection.

```
FUNCTION CLOSEPORT() AS INTEGER
```

Returns:

Integer Value	Description
0	Closed OK.
-2	An error occurred while closing the ports.

This function should be called before closing the hosting application.

BIND

Also known as **Pairing**. The BIND function allows connection information to be retrieved from an unknown remote device and to configure the communications settings between that device and the base station. Most devices activate their binding mechanisms by being power cycled but refer to the device manual for details.
This function is blocking and does not return until a bind is successful or the duration has expired.
For a non blocking Bind function see BINDASYNC.

```
FUNCTION BIND(BYVAL BaseStation AS BYTE, BYVAL UseRemoteSettings AS BYTE, BYVAL
ConfigMode AS BYTE, BYVAL Duration AS BYTE, BYREF ID AS LONG, BYREF DataTag AS LONG)
AS INTEGER
```

Where:

Parameter	Description
BaseStation	Represents the base station address. This should be 1.
UseRemoteSettings	Determines whether the remote device will be configured to the base station communications settings or vice versa. Set to 1 to change the base station settings to match those of the remote device or zero to change the remote device settings to match the base station.
ConfigMode	Determines whether the remote device will enter configuration mode. This mode is dependent on the actual device but will generally mean it will stop any automatic transmissions, inhibit low power modes and not act on any automatic sleep mechanisms. This ensures that the binding application can communicate and configure it.
Duration	Sets the duration of the bind attempt in seconds. i.e. how long the base station will wait for the remote device to enter bind mode.
ID	The ID of the bound device.
DataTag	The default Data Tag of the bound device.

Returns:

Integer Value	Description
0	Bind was successful.
1	Bind was not successful. No remote device was detected.
99	Thread conflict detected! Yield (sleep, doevents etc) and retry the function.

Once a successful bind has occurred the hosting application may communicate with the device using READREMOTE and WRITEREMOTE using the ID returned from the bind function.

BINDASYNC

This bind function is non blocking and is useful when you need to control the power supplied to your device to trigger the bind. This function is called first then use BINDASYNCPOLL to test the status and outcome of the bind.

```
FUNCTION BINDASYNC(BYVAL BaseStation As BYTE, BYVAL UseRemoteSettings As BYTE, BYVAL ConfigMode As BYTE, BYVAL Duration As BYTE) As Integer
```

Where:

Parameter	Description
BaseStation	Represents the base station address. This should be 1.
UseRemoteSettings	Determines whether the remote device will be configured to the base station communications settings or vice versa. Set to 1 to change the base station settings to match those of the remote device or zero to change the remote device settings to match the base station.
ConfigMode	Determines whether the remote device will enter configuration mode. This mode is dependent on the actual device but will generally mean it will stop any automatic transmissions, inhibit low power modes and not act on any automatic sleep mechanisms. This ensures that the binding application can communicate and configure it.
Duration	Sets the duration of the bind attempt in seconds. i.e. how long the base station will wait for the remote device to enter bind mode.

Returns:

Integer Value	Description
0	Bind initiation was successful.
99	Thread conflict detected! Yield (sleep, doevents etc) and retry the function.

Now call BINDASYNCPOLL to determine when the bind has completed or failed.

BINDASYNCPOLL

Called after BINDASYNC to determine whether the bind is busy or has completed.

```
FUNCTION BINDASYNCPOLL(BYREF ID As LONG, BYREF DataTag As LONG) As INTEGER
```

Where:

Parameter	Description
ID	The ID of the bound device.
DataTag	The default Data Tag of the bound device.

Returns:

Integer Value	Description
0	Bind was successful.
1	Bind was not successful. No remote device was detected.
99	Busy. Binding is still in progress.

If successful the ID and DataTag parameters will contain the ID and Data Tag of the bound device.

READREMOTE

Reads a parameter from a remote device. The radio modules transparently handle retries. This function is blocking and execution will not continue until a response has been received or the function has timed out.

```
FUNCTION READREMOTE(BYVAL BaseStation AS BYTE, BYVAL ID AS LONG, BYVAL Command AS BYTE, BYREF sData AS BYTE, BYREF Length AS LONG, BYREF RSSI AS INTEGER, BYREF CV AS INTEGER, BYREF Flags AS INTEGER)  AS INTEGER
```

Where:

Parameter	Description
BaseStation	The address of the base station through which to route this packet.
ID	The ID of the remote device.
Command	The command number of the parameter to read.
sData	Pointer to the result data. There must be enough bytes allocated to avoid buffer overflows. Recommend 128 bytes. The first byte indicates the data type. This will be set by the device and you cannot request data of a specific type. See **Data Types and Formats** later.
Length	The number of bytes returned.

RSSI	The radio signal level in dB of the received packet. This parameter will be set on return from this function.
CV	The correlation value of the received packet. A value of 55 is a poorly formed signal whereas 110 is a perfectly formed signal. This parameter will be set on return from this function.
Flags	Contains extra information regarding this packet. The binary value of the flags indicate the following: 1=This packet was broadcast. 2=Remote device reports low battery. 4=Remote device reports an error. This parameter will be set on return from this function.

Returns:

Integer Value	Description
0	Received response OK
1	No response from base station.
2	No response from remote device.
3	NAK response from remote device.
99	Thread conflict detected! Yield (sleep, doevents etc) and retry the function.

WRITEREMOTE

Writes a parameter to a remote device. This function is blocking and execution will not continue until a response has been received or the function has timed out.

```
FUNCTION WRITEREMOTE(BYVAL BaseStation AS BYTE, BYVAL ID AS LONG, BYVAL Command AS
BYTE, BYREF sData AS BYTE, BYVAL Length AS LONG, BYREF RSSI AS INTEGER, BYREF CV AS
INTEGER, BYREF Flags AS INTEGER)  AS INTEGER
```

Where:

Parameter	Description
BaseStation	The address of the base station through which to route this packet.
ID	The ID of the remote device.
Command	The command number of the parameter to read.
sData	Pointer to the data to write. The first byte indicates the data type. See **Data Types and Formats** later. You can write any data type to any parameter and if possible the data will be converted when written. For example you could use strings to write all data if desired.
Length	The number of bytes returned.
RSSI	The radio signal level in dB of the received packet. This parameter will be set on return from this function.
CV	The correlation value of the received packet. A value of 55 is a poorly formed signal whereas 110 is a perfectly formed signal. This parameter will be set on return from this function.
Flags	Contains extra information regarding this packet. The binary value of the flags indicate the following: 1=This packet was broadcast. 2=Remote device reports low battery. 4=Remote device reports an error. This parameter will be set on return from this function.

Returns:

Integer Value	Description
0	Received response OK
1	No response from base station.
2	No response from remote device.
3	NAK response from remote device.
4	Invalid Data response from remote device.
99	Thread conflict detected! Yield (sleep, doevents etc) and retry the function.

Data Types and Formats

The first byte in the data written to and read from devices indicates the data type and thus the format of the data that follows.

Value	Data Type
0	Unknown data type. Can be used when executing commands.
1	UINT8
2	UINT16
3	INT32
4	FLOAT
5	STRING
5	BINARY

See **Appendix A** – **Data Type Formats** for details on the formatting.

Thread Conflicts

The DLL has been designed to allow the host IDE to debug through the callback routines. To achieve this there is a situation where sometimes calls to the functions will not able to be handled correctly (i.e. a callback to the host IDE is in progress and to continue would violate the integrity of the threading). In this case the returned value will be 99 and the host program needs to yield processing if it to succeed in getting a response.
The suggested technique is as follows:

```
Variable = 99
While Variable = 99
       Variable = DLLFUNCTION()
       If Variable = 99 Then Yield
Wend
```

Where **`DLLFUNCTION()`** is where you would place the **`ReadRemote`** or **`WriteRemote`** call. You will need to find the appropriate command in your language to Yield. In **Visual Basic** this is **`DoEvents`** and in **Delphi** this is **`Application.ProcessMessage`**.

Notes

You should only use data from a function if it has returned a zero (0). All other responses indicate an error.

DLL Limitations

- When connecting via USB only one base station is supported and this must be address 1 (set by DIP switches on industrial version).
- This DLL can only open one serial port per host thread.

Appendix A

Data Type Formats

The following data formats are used when communicating with the base station. These formats apply to the raw data in the packets and also to the data parts of the supplied T24drv.dll driver.

Value	Data Type	Number Of Bytes	Example	Notes
1	UINT8	1	01	
2	UINT16	2	00 01	MSB First
3	INT32	4	00 00 00 01	MSB First
4	Float	4	3F 80 00 00	See Floating Point IEEE
5	String	0-64	Hello World	
6	Binary	0-64	"£$%^&*(	

UINT8

Represents an unsigned numeric value from 0 to 255 and consists of a single byte.

Byte							
7	6	5	4	3	2	1	0

UINT16

Represents an unsigned numeric value from 0 to 65535 and consists of 2 bytes. The bytes are in order of significance MSB first.

MSByte								LSByte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

INT32

Represents a signed numeric value from -2,147,483,648 to positive 2,147,483,647 and consists of 4 bytes and is stored in 2's compliment form. The bytes are in order of significance MSB first.

MSByte																												LSByte							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	16	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Sign Bit																																		

Float

Represents a numeric value from n to n and consists of a 4 byte in IEEE 754 format.

MSByte																												LSByte							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	16	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Exponent (7-bit excess-127)							Mantissa (-bit fraction with implicit 1)																													
	Sign Bit																																		

The byte containing the sign and exponent is sent first, with the LS byte of the mantissa being last.
The value of the number is thus

```
(-1)Sign * 2(Exponent-127) * 1.Mantissa
```

Note the ‘assumed 1’ before the mantissa. The exception to this is the special value 0.0, which is represented as 4 zeroes.
The precision of this format is to 7 digits.
eg. a floating-point number of -12345.678 is represented as – [hex] C640E6B6

String

Represents a textual string and is terminated by a NULL (ASCII 0).

Binary

The bytes have no set meaning and are just a string of bytes. These bytes can be any value and may contain non ASCII characters.

RSSI, CV and LQI

Packets received from remote devices have RSSI and CV bytes present at the end of the packet.

RSSI is Received Signal Strength Indication
This indicates the strength of the received signal. This approximates to dB and can be calculated from the RSSI byte which is stored in 2's compliment format. This value also has an offset of 45.
To convert the byte value to RSSI use the following algorithm.

```
RSSI = RSSIBYTE
If RSSI > 127 Then RSSI = ((RSSI - 1) Xor 0xFF) * -1
RSSI = RSSI - 45
```

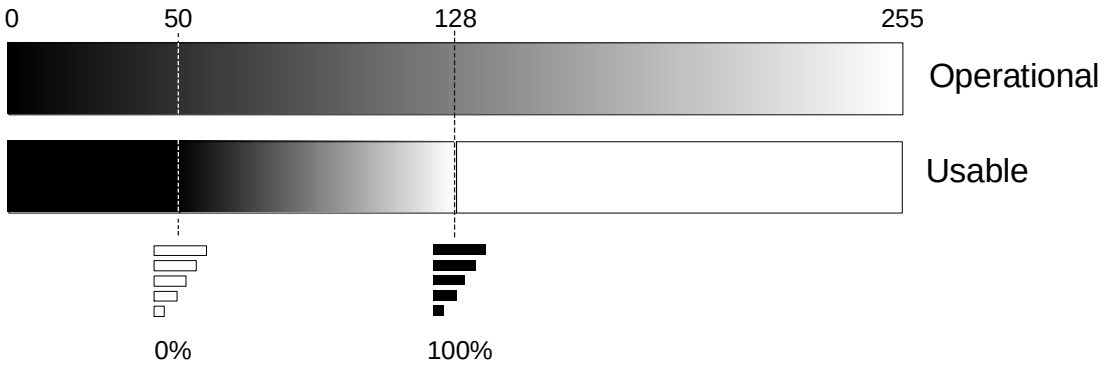
CV is Correlation Value
This indicates the quality of the signal. The value of the CV byte (0-255) needs the most significant bit masking off (AND with &H7F) where a poor CV is around 55 and a good CV is 110.

```
CV = CVBYTE
CV = CV AND &H7F
```

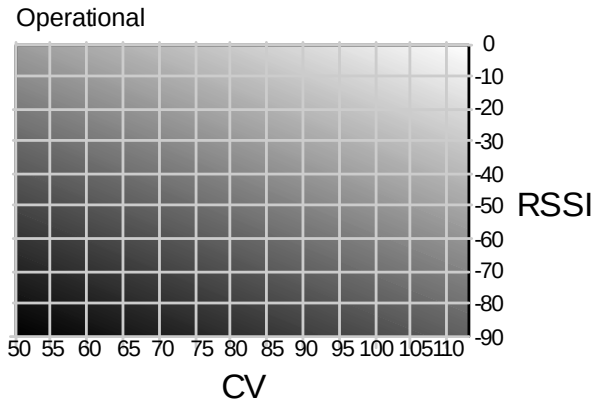
LQI is Link Quality Indication
Sherborne may also refer to **Link Quality** which is derived from the RSSI and CV values:

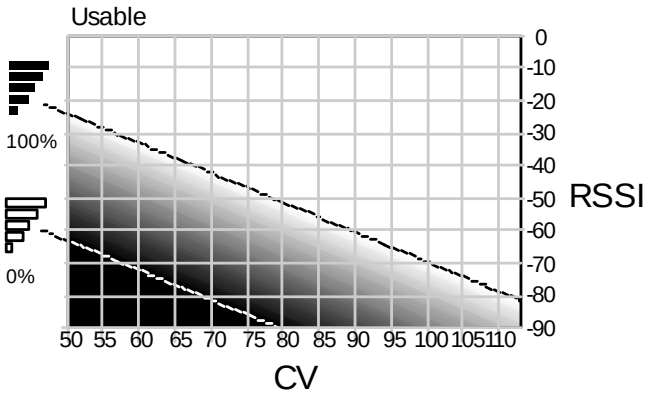
```
LQI = (((94 + RSSI) + (CV - 55)) / 2) * 3.9
```

Which gives an **operational** range of approximately 0 to 255.
This operational range covers the extremes of very poor to very good connection quality so we usually take a portion of this to represent the **usable** range which gives the user a better representation of usable, real-world quality.
The LQI range from 50 to 128 can be thought of as to represent 0-100% **usable** quality.
This reduced portion of the range represents the **usable** range and may be represented, for example, by a signal strength indicator as found on a mobile/cell phone.



The charts below indicate the operational and usable combinations of RSSI and CV where black is poor and white is good.





Unslotted CSMA/CA

Most of the WTS range of devices will allow you to turn off the CSMA (Carrier Sense – Multiple Access).
This is recommended to be turned on but in some circumstances turning it off will increase data rate and reduce latency. For example a single deice transmits at 200Hz. With no other device on this channel CSMA can be disabled giving a more accurate 200Hz transmission.
CSMA is implemented to reduce the collisions between packets from different devices.

With CSMA Disabled

When a device wants to transmit it checks the channel to see if another device is transmitting. If not then the transmission takes place immediately. If the channel is busy then the transmission will occur as soon as the channel has been detected as clear.

With CSMA Enabled

When a device wants to transmit it checks the channel to see if another device is transmitting. If the channel is busy then we wait until it is clear.
Now we back off for a random period.
The first time we back off one of the following periods is randomly selected:

320uS	640uS	960uS

If the channel is now clear then the transmission takes place.
If the channel is busy then a new random backoff period is selected from the following:

320uS	640uS	960uS	1.28mS	1.600mS	1.920mS	2.240mS

If the channel is now clear then the transmission takes place.
If the channel is busy then a new random backoff period is selected from the wider range and the procedure repeated.

Certain packets are transmitted as if CSMA is disabled regardless of the device setting. These include the sleep/wake packets and responses to requests.

Data Tag Control Interface (Advanced)

When acquisition devices are operating in low power mode it is not easy to communicate using the full read/write packets as most of the time the device is asleep. Also in some cases the consumer of the data only knows the Data Tag from the Data Provider packet and does not know the ID of the sender. Therefore we need to utilise a control interface within the Data Provider packet scheme whereby devices such as a handheld can perform rudimentary control on another device while knowing no more than that devices default Data Tag.
Each device supplying data to a consumer only has one defined default data tag. We reuse that tag to enable communicating back to the data provider. This will not affect other consumers of the data as the data provider packet will contain a data type of FF which indicates our internal control interface.
All other consumers will automatically reject the FF data type anyway.
So to control the provider we simply send a data provider packet using the same data tag but containing data of type **FF** the data consists of a single **Function Byte** which has fixed functionality depending on its value. The status byte is not used and may be left at zero.

Value	Fixed Universal Function
0	None
1	SLEEP
2	PAUSE
3	STAYAWAKE
4	CONTINUE
5	DOSYSTEMZERO
6	REMOVESYSTEMZERO
7	SHUNTCALON
8	SHUNTCALOFF
9	DOTARE
A	REMOVETARE
B	LEDONUNTILNEXTTX

Using a Data Tag of FFFF will act as a broadcast data provider control interface and all recipients of an FFFF data tag will check the data type and if this is FF the device may perform the specified function.
To use this interface the sender must reply with the control interface packet within 8 milliseconds of receiving a Data Provider packet.

Packet Type	Data Tag		Status	Data Type	Function Byte
03	00	00	00	**FF**	00

Appendix B

Radio Range

When planning the installation of a radio net it is useful to consider a number of different arrangements and compare their relative merits before deciding on a final layout. One aspect to be considered is the useable signal strength at the receiver input represented by the Received Signal Strength Indicator (RSSI) figure.

RSSI is a negative number related to signal strength in dBm; a smaller number represents a stronger signal so -70 is much better than -80. To obtain a reliable link using Rad24 radios RSSI must be no worse than -85 to -90, beyond this figure packet loss rate increases and link quality drops off sharply. A table illustrating the relationship between receiver input power and RSSI is reproduced at Table 1 below.

Radiated energy diminishes over distance with an inverse square law; signal power is also lost due to absorption and scattering in the air between the transmit and receive antennas, these losses are referred to as Free Space Path Loss and vary according to the wavelength of the signal. At 2.4GHz Free Space Path Loss is given by:

Path Loss = $32.4+20\text{Log}_{10}\, d$

Where:
Path Loss is expressed in dB
d = Path length in metres

Example 1

Path Loss over 80metres = $32.4+20\text{Log}_{10}\, 80$

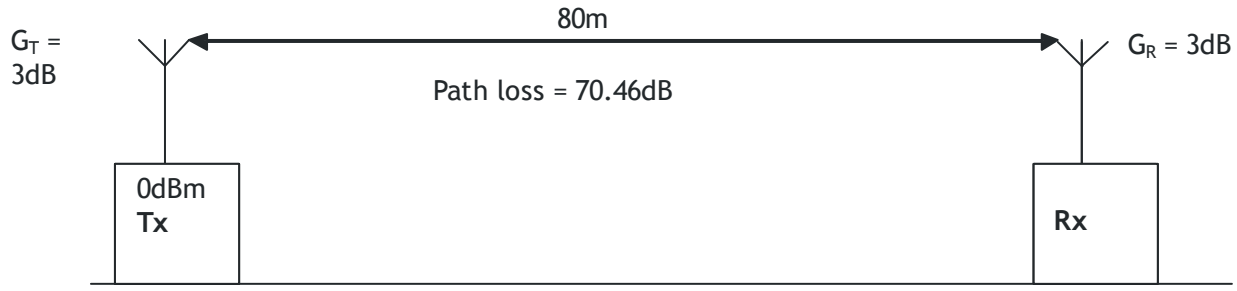
$20\text{Log}_{10}\, 80 = 38.06$

Path Loss = 32.4+38.06 = 70.46 dB

For convenience, Table 2 below gives path losses at range intervals of 5 metres.

Example 2

An estimate of signal power at the receiver input can be made by considering two Rad24i radios in the open separated by a distance of 80m. Rad24i output power is 0dBm and the peak gain of the integral antenna is 3dB as shown at Figure 1 below.



G_T = Gain of transmit antenna
G_R = Gain of receive antenna

Figure 1

Calculation of signal power at the receiver input is achieved by summing all the gains and then subtracting the sum of all the losses, in this case:

Signal power at the receiver input = (G_T+G_R) – Path Loss

or

6 – 70.46 = -64.46dBm

Signal power at the input of the receiver = -64.46dBm

Note that although the manufacturer's data sheet for the chip antenna gives gain as 3dBi Peak, this figure is rarely achieved in practice; between 1 and 1.5dBi is more usual.

Example 3

In practice the radios would be fitted inside enclosures and the antenna may be either inside the enclosure or mounted some distance from the radio and connected to it by an extended feeder. A practical example of this type is illustrated at Figure 2 below.

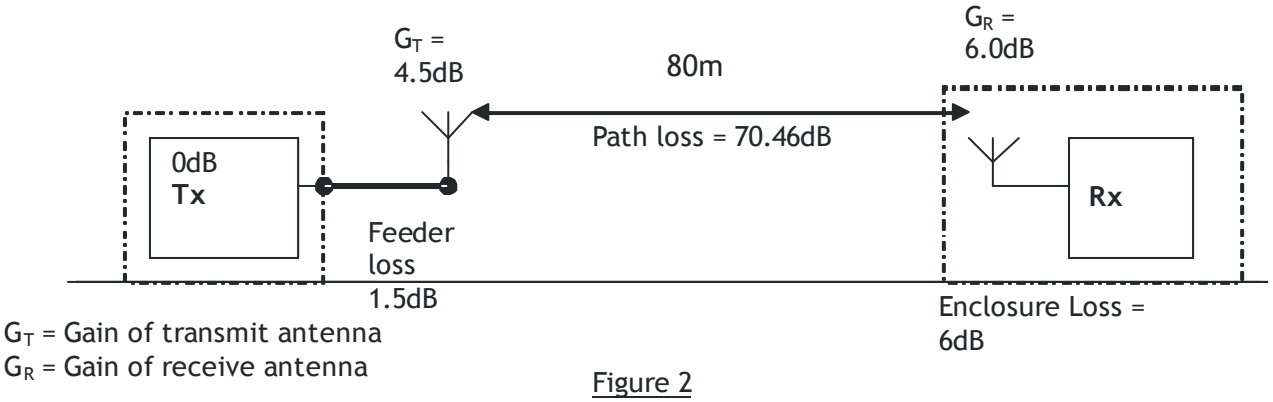


Figure 2

Sum of the gains = G_T+G_R

or: 4.5 + 6.0 = 10.5

Sum of the losses = Feeder Loss + Path Loss + Enclosure Loss

or: 1.5 + 70.46 + 6.0 = 77.96

Signal strength at the receiver input is: 10.5 – 77.96 = -67.46dBm

Refer to Table 1 below that relates receiver input power to RSSI, an input level of -67.46dBm will give an RSSI of around -74.

In this example the gain of the receive antenna and the receiver enclosure loss are identical and therefore cancel out.

Antenna Basics

Gain

For a particular antenna the stated gain figure applies only along the antenna bore-sight and often only in one plane; if the remote point is off-axis relative to the antenna electrical centre-line or bore-sight then the gain will depend on the degree of offset according to the polar diagram for that antenna.

For example, an antenna is advertised as having a gain of 12dBi and a beam-width of 60 degrees; this is normally taken to refer to the half power or -3dB beam-width. In other words the antenna gain at ±30degrees off the bore-sight will be 3db down on the peak gain or 9dBi. Beyond ±30 degrees off-axis the gain will be very much less.

For short vertical antennas of the "rubber duck" type the gain is fairly constant in the horizontal plane but the vertical beam-width may be as little as 12 degrees for a 9dB antenna making it necessary to mount the antenna slightly off vertical for best signal strength in a specific area; this of course means that the pattern on the opposite side might now be pointing into the ground or up in the air.

Polarisation

Antenna polar diagrams show coverage in terms of variations in gain over vertical and horizontal planes relative to the antenna. Depending on its method of construction the antenna will radiate the electric component of the Electro-Magnetic (EM) wave in one plane and the magnetic component in the other (Polar diagrams identify these two components as E for electric and H for magnetic). If the electric component of the EM wave is vertical then the antenna is said to be vertically polarised and vice-versa.
There are some exceptions to this (such as crossed and circular polarisation) but for the sake of simplicity they are not considered here.
For maximum power transfer all antennas on the same radio net must be mounted so that their signals have the same polarisation; it is sometimes possible to take advantage of this to allow operation of adjacent but otherwise unconnected radio nets on the same channel by having the antennas on one net vertically polarised and the other horizontally polarised.

Mounting Requirements

Objects or structures within the operating region of the antenna will distort the horizontal and vertical space pattern so that the antenna polar diagram no longer represents the actual coverage. For this reason, antennas should be mounted to maximise separation from buildings or structures and away from areas where large objects may be temporarily placed. This is usually achieved by fixing the antenna to a mast or tower by means of a bracket that allows adjustment of antenna orientation. The mounting method should also allow for adjustment of height as, very often, a small change in antenna elevation will improve signal strength.

Receiver Input Power and RSSI Relationship

During development of the Rad24 radio it was necessary to understand the relationship between signal power at the receiver input and the indicated RSSI figure. A calibrated signal source was used to set RSSI readings at intervals of 10 and the corresponding input levels recorded at each step. Figures in italics were not measured but extrapolated from the measured values.

Input signal power	Indicated RSSI
(dBm)	Value
-7	-10
-17	-20
-28.5	-30
-36.5	-40
-44.5	-50
-54	-60
-64	-70
-74	-80
-79	*-85*
-84	*-90*

Path Loss at 2.4GHz

Free Space Path Loss in dB = $32.4+20\text{Log}_{10}$ d
Where d = Free space path length in metres

Path length (metres)	Loss (dB)	Path length (metres)	Loss (dB)	Path length (metres)	Loss (dB)	Path length (metres)	Loss (dB)
5	46.37	70	69.30	135	75.00	200	78.42
10	52.40	75	69.90	140	75.32	205	78.63
15	55.92	80	70.46	145	75.62	210	78.84
20	58.42	85	70.98	150	75.92	215	79.04
25	60.35	90	71.48	155	76.20	220	79.24
30	61.94	95	71.95	160	76.48	225	79.44
35	63.28	100	72.40	165	76.74	230	79.63
40	64.44	105	72.82	170	77.00	235	79.82
45	65.46	110	73.22	175	77.26	240	80.00
50	66.37	115	73.61	180	77.50	245	80.18
55	67.20	120	73.98	185	77.74	250	80.35
60	67.96	125	74.33	190	77.97	255	80.53
65	68.65	130	74.67	195	78.20	260	80.69

These figures are for free-space path loss only, when estimating signal power at the receiver input, take the sum of all the gains and then subtract the sum of all the losses.

Power Density

From the radar equation, power density at the target is given by:

$$Pd = \frac{PT\ GT}{4\pi\ r2}$$

Where:

Pd = Power density in W/m2
PT = Transmitter output power in Watts
GT = Antenna gain as a multiple of input power
r = Range to target in metres

From the Rad24 radio specifications:

Power output is 1mW Peak
Chip antenna peak gain is 3dBi or 2 times the input power

Measurement range is 20cm or 0.2m

$$Pd = \frac{1x10-3\ X\ 2}{12.568\ X\ 0.22}$$

$$Pd = 3.978x10-3\ W/m2$$

Pd at 20cm range is 3.978mW/m2

Dividing this by 10,000 to express power density in W/cm2 gives:

397.8nW/cm2

This is the peak power density assuming the RF output is 1mW Continuous Wave; multiply this figure by the duty cycle and the average power density is reduced proportionately. Further, this calculation uses the peak gain of the chip antenna stated as 3dBi in the manufacturers data sheet, in practice this gain is rarely achieved being nearer to 1 to 1.5dBi.

RF Exposure Limits

Power density at the Rad24 Antenna is well below the Whole Body Average SAR (Specific Absorption Rate) of 80mW/Kg exposure limit given at ANSI/IEEE C95.1-2005 and OET Bulletin 65 Edition 97-01 dated August 1997. Under the terms of FCC CFR Title 47 Volume 1 Part 2.1091 and Part 2.1093 this equipment is categorically excluded from routine environmental evaluation for RF exposure.
Article 3(1) of Directive 2004/40/EC of the European Parliament and of the Council dated 29 April 2004 (The Physical Agents Directive) gives an Exposure limit for Whole Body Average SAR as 400mW/Kg. Rad24 Peak ERP is not more than 4mW assuming a 6dB antenna.