Universal Receiver User Manual

Version 1.0

Table of Contents

1 WEEE & Battery Directive

Waste Electrical and Electronic Equipment.
HWM-Water Ltd is a registered producer of Electrical and Electronic Equipment in the United Kingdom (registration number WEE/AE0049TZ). Our products fall under category 9 (Monitoring and Control Instruments) of *'The Waste Electrical and Electronic Equipment (WEEE)'* Regulations. We take environmental issues seriously and fully comply with the requirements for collection, recycling and reporting of waste products.

HWM-Water Ltd is responsible for WEEE from customers in the United Kingdom provided that:
The equipment was produced by HWM-Water Ltd (Palmer Environmental/Radcom Technologies/RadioTech/ASL Holdings Ltd) and supplied on or after 13th August 2005.
Equipment supplied before 13th August 2005 and directly replaced HWM-Water Ltd products manufactured since 13th August 2005.

HWM-Water products supplied after 13th August 2005 can be identified by the following symbol:

Under HWM-Water Ltd Terms and Conditions of Sale, customers are responsible for the cost of returning WEEE to HWM-Water Ltd and we are responsible for the costs of recycling and reporting on that waste.

Instructions for returning WEEE:
Ensure that the WEEE meets one of the two conditions above.
The waste will need to be returned in accordance with the regulations for transporting data loggers with lithium batteries.

a. Pack loggers in strong, rigid outer packaging to protect them from damage.
b. Attach a Lithium Warning Label to the package.
c. The package must be accompanied by a document (e.g. consignment note) that indicates:
    i. The package contains lithium metal cells.
    ii. The package must be handled with care and a flammability hazard exists if the package is damaged.
    iii. Special procedures should be followed in the event the package is damaged to include inspection and re-packing if necessary.
    iiii. A telephone number for additional information.
d. Refer to the ADR regulations on shipping dangerous goods by road.

Return the WEEE to HWM-Water Ltd using a licensed waste carrier.
In accordance with the regulations, customers outside the United Kingdom are responsible for WEEE.

The Battery Directive
As a distributor of batteries HWM-Water Ltd will accept old batteries back from customers for disposal, free of charge, in accordance with the Battery Directive.

PLEASE NOTE: All lithium batteries **MUST** be packaged and returned in accordance with the relevant regulations for transporting lithium batteries.

A licensed waste carrier must be used for transporting all waste.
For more information on WEEE compliance or the Battery Directive please e-mail CService@hwm-water.com or phone +44 (0)1633 489 479.

If further support or assistance is required, please contact
HWM Technical Support on 01633 489479 (option 5)
or e-mail support@HWM-Water.com

2 Introduction

The Universal Modbus Receiver has been designed to provide a complete solution for fixed data collection. It is designed to work in conjunction with the AMR range of transmitters. The receiver decodes the data, verifying the incoming data using a 16-bit CRC check (***Appendix 2***) and passes the data packet through to the serial port.

Receiver Frequency:	153.10MHz or 169.40625MHz or 433.92MHz or 869.85MHz.
Modbus Mode:	Modbus Slave Remote Terminal Unit (RTU) supported. Slave address configurable.
Number of Devices Supported:	255 Channels.
Communications Protocol:	RS232 & RS485.
Communications Setup:	Fixed Setup 2400/4800/9600Baud, No Parity, 8-bits, 1-Stop. Programmable Option (default 19200, Even Parity, 8-bits, 1-Stop).
Function Codes Supported:	0x03 (03_{dec}) Read Holding Registers 0x10 (16_{dec}) Write Multiple Registers 0x42 (66_{dec}) Special Function – Program Unit ID 0x44 (68_{dec}) Special Function – Set Serial Protocol

Operating Environment:
Operating Temperature Range: -10°C to +40°C (Recommended for indoor use).
Dimensions: 98 x 198 x 82mm (approx.) excluding antenna, connectors and glands.
Protection Rating: IP65 (when sealed).

Power Supply:
Operating Voltage: $12V_{DC}$ from plug top power supply.

Enclosure:
Die-cast Aluminium – painted grey.

3 Installation

As with any radio communication system, the Universal Receiver should be connected to a clean and stable power supply. If using a switch mode power supply, be aware these types can be rich in harmonics that can cause the high gain receiver module to block the reception of data. The frequency of offending harmonics can shift with temperature, time and load.

For maximum transmission range the antenna of both the transmitter and receiver should point upward (vertical polarization) and should be kept clear of obstructions, particularly metallic surfaces and bodies.

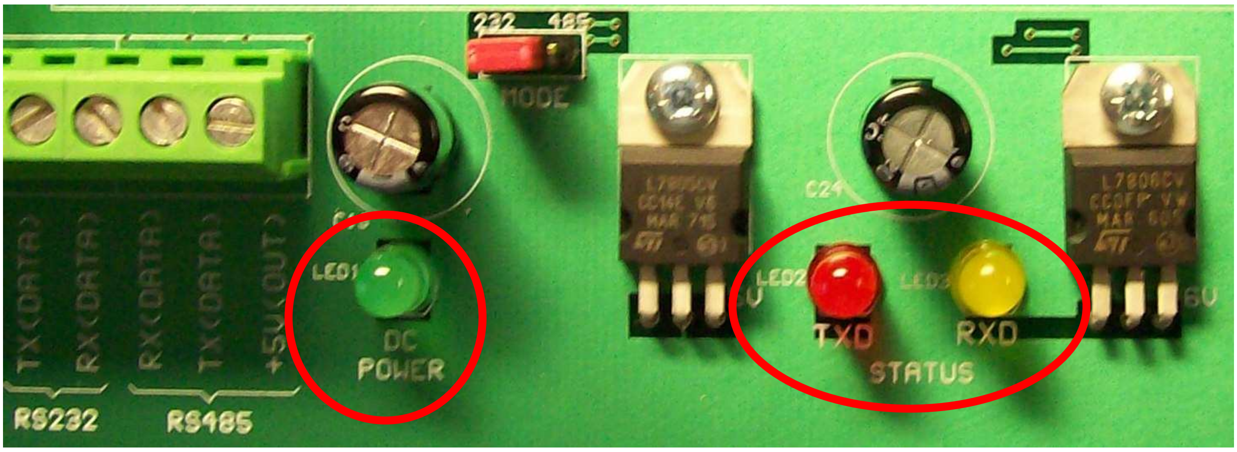
As standard, an N-type helical antenna is supplied with the Universal Receiver, contact the HWM Sales department for alternatives. If selecting a 3rd party antenna ensure the impedance is 50Ω.

3.1 Location

For indoor installations, it is recommended that the receiver is located away from sources of heat and electrical apparatus such as inverters. Care should be taken to minimise cable lengths both with respect to the antenna location and the attached terminal equipment. Generally, RS232 should be used for short distance links <10m and RS485 for longer links <300m.

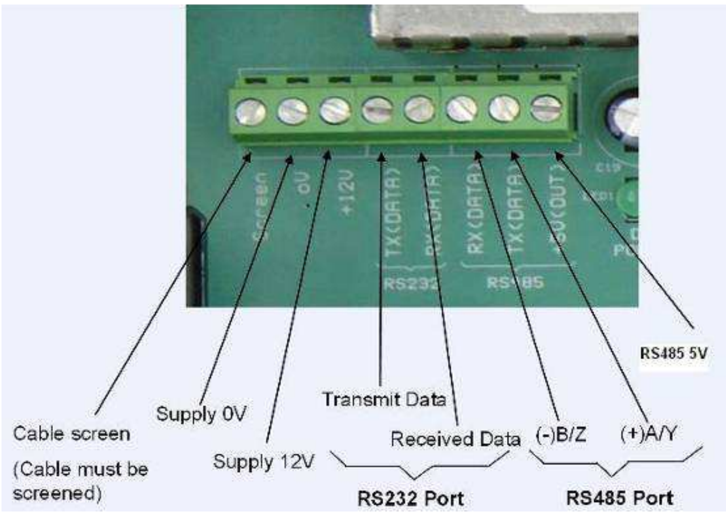
3.2 LED Indication

A number of visual indicators are present on the PCB to provide some level of diagnostics.



DC Power:	ON (Green)	– Receiver has power.
TxD LED:	ON (Red)	– Sending data on communications port.
RxD. LED:	ON (Amber)	– Receiving data from Receiver Module.

4 Connections



Data and power connections should always be made using screened cable. Using a screened cable will help to reject interference. You should always use a common ground point and avoid the formation of current loops.

Terminal	Description	Notes
Screen	Earth (Comms Cable Screen)	
0V	DC Power –ve	
+12V	DC Positive +ve	
TX(DATA)	RS232 Tx Output	
RX(DATA)	RS232 RX Input	
B/Z (-)	BZ - RS485 –ve	
A/Y (+)	AY - RS485 +ve	
+5V (OUT)	+5V Out	Can be used as pull-up for RS485 bus

Note: RS485 connections, it is the responsibility of the system builder to ensure that the connections are correctly terminated. Normally, cables with an impedance of greater than 100Ω should be used. Open ends may need terminating using 120Ω resistors between the ***AY*** and ***BZ*** terminals.

5 Configuration

5.1 Dip Switch Settings

In normal operation (Deported Mode) SW1 & SW2 are used to select the data format. The data is fed directly to the serial port without processing. This can be done using AMR receiver PC applications (V13 & V20/V21) – Contact HWM for details. Note, both switches should not be left in the ON position.

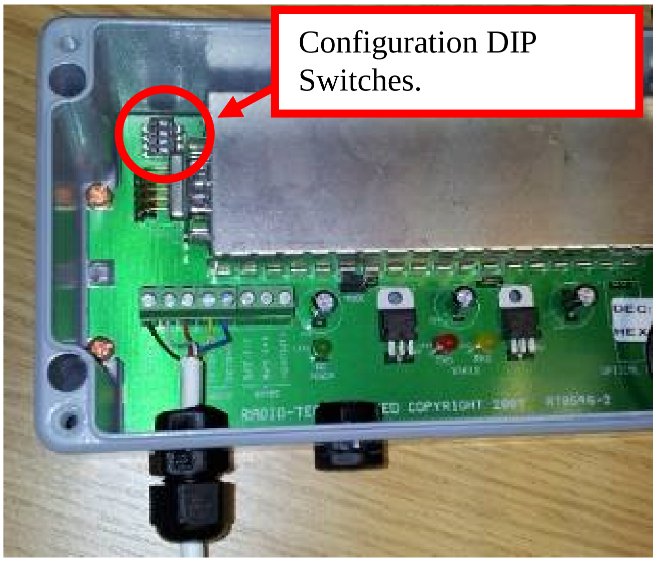
SW1 – Received Data Passed Through in Binary Format.
SW2 – Received Data Passed Through in ASCII Format.

In Modbus Mode SW1 and SW2 are both left OFF and SW3 & SW4 used to select the Baud rate.

SW3	SW4	Operation
OFF	OFF	2400Baud, No Parity, 1-Stop Bit
ON	OFF	4800Baud, No Parity, 1-Stop Bit
OFF	ON	9600Baud, No Parity, 1-Stop Bit
ON	ON	User Programmable (default 19200Baud, Even Parity, 1-Stop Bit

Default Settings for Modbus Rx:

SW1 – OFF
SW2 – OFF
SW3 – ON
SW4 – OFF



The communication mode selection jumper should set to either:
Left + Middle of header pins for RS232 mode.
Or ***Middle + Right*** of header pins for RS485 mode.

5.2 Programming

A simple programming application is available from HWM to program and validate the receiver. However, it is strongly recommended that this is built into the system.

The Universal Receiver supports the Modbus remote terminal unit (RTU) Slave Mode. The following function codes are supported.

- 03_{dec} Read Holding Registers
- 16_{dec} Write Multiple Registers
- 66_{dec} Special Function – Program Unit ID
- 68_{dec} Special Function – Set Serial Protocol

The receiver needs to be programmed with the addresses of all the transmitters expected to receive. A memory map is defined (***Appendix 1***) holding the transmitter IDs, status and data bytes. This data can be read back by the Modbus Master device.

From firmware version 10 each transmitter data byte is defaulted to FF_{hex} on power up of the Modbus Receiver to indicate invalid data. Reprogramming a transmitter address will also reset its data to FF_{hex}. The floating point registers in the alternate register maps will return $FFC00000_{hex}$ (NaN) until data is received.

5.3 Addressing Modes

The Modbus receiver has register maps in both 8-bit and 16-bit formats. These are shown in detail in Appendix 1.

5.3.1 Standard 8-Bit Register Map

The addresses of the transmitters need to be programmed into the memory map and read back with the data as verification. E.g. consider transmitter in memory slot 2:

Tx. No.	Modbus Register Address		Description
	Dec	Hex	
2	792	0318	Transmitter Address HHO
	793	0319	Transmitter Address HO
	794	031A	Transmitter Address LO
	795	031B	Transmitter Status
	796	031C	Data 1
	797	031D	Data 2
	798	031E	Data 3
	799	031F	Data 4

Registers $792\text{-}794_{dec}$ (0x0318) hold the 24-bit transmitter address, so if data needs to be received from transmitter number 54322_{dec} ($00D432_{hex}$), the registers will need to be programmed with the value $00D432_{hex}$, 00_{hex} into register 0x0318, $D4_{hex}$ into register 0x0319 and 32_{hex} into register 0x031A

When data is received from the transmitter, the rest of the registers will be filled with the data received. The transmitter status and data values vary in their content types depending on the transmitter type being received. Modbus supports 8 to 16-bit formats and many of the values implemented on the AMR transmitters vary between 8 to 24-bits.

The Universal Receiver sends all data in 16-bit format. When reading the 8-bit registers, the upper 8-bits are sent as 00_{hex}. E.g. data value 1122867_{dec} (112233_{hex}) stored in consecutive 8-bit registers would be sent as 0011_{hex}, 0022_{hex} and 0033_{hex}.

5.3.2 Standard 16-Bit Register Map

The standard 16-bit register map is available from firmware version 8 onwards. This offers the same data as the standard 8-bit map, but in 16-bit format.

Tx. No.	Modbus Register Address		Description	
	Dec	Hex	High byte	Low byte
N/A	4096	1000	Firmware ID	
	4097	1001	Firmware Version	
	4098	1002	Frequency Band	
…	…	…	…	
2	4118	1016	Transmitter Address HO 16bits	
	4119	1017	Transmitter Address LO 16bits	
	4120	1018	0	Transmitter Status
	4121	1019	0	Data 1
	4122	101A	Data 2	Data 3
	4123	101B	0	Data 4

Registers 4118 (0x1016) and 4119 (0x1017) hold the 24-bit transmitter address, so if data needs to be received from transmitter number 76322_{dec} ($012A22_{hex}$), the registers will need to be programmed with the value $012A22_{hex}$, 0001_{hex} into register 0x1016 and $2A22_{hex}$ into register 0x1017.

In addition to the channel registers there are a set of special registers at 4096 – 4098 (1000 – 1002_{hex}) that contain the following information.

- Firmware ID – This is a fixed value of 546.
- Firmware Version – Firmware version.
- Frequency Band – 0 for 153 MHz or 1 for 169 MHz.

5.3.3 Alternate 8-Bit Register Map

The alternate 8-bit register map is available from firmware version 10 onwards. This duplicates the registers in the standard 8-bit map and adds extra information.

Tx. No.	Modbus Register Address		Description
	Dec	Hex	
N/A	5648	1610	Firmware ID HO
	5649	1611	Firmware ID LO
	5650	1612	Firmware Version
	5651	1613	Frequency Band
	5652	1614	Firmware ID2
	5653-5663	1615-161F	11 Bytes Spare
1	5664	1620	Transmitter Address HHO
	5665	1621	Transmitter Address HO
	5666	1622	Transmitter Address LO
	5667	1623	Transmitter Status
	5668	1624	Data 1
	5669	1625	Data 2
	5670	1626	Data 3
	5671	1627	Data 4
	5672	1628	Transmitter Type
	5673	1629	RSSI
	5674	162A	Tx Counter
	5675	162B	Tx Time HO
	5676	162C	Tx Time LO
	5677	162D	FP Analogue 1, Byte 1
	5678	162E	FP analogue 1, Byte 2
	5679	162F	FP analogue 1, Byte 3
	5680	1630	FP analogue 1, Byte 4
	5681	1631	FP analogue 2, Byte 1
	5682	1632	FP analogue 2, Byte 2
	5683	1633	FP analogue 2, Byte 3
	5684	1634	FP analogue 2, Byte 4
	5685-5695	1635-163F	11 Bytes Spare

Additional Information

- Transmitter Type — Transmitter type code received from the transmitter.
- RSSI — Signal strength indicator from the upper 4-bits of Data 4.
- Tx Counter — Transmitter message counter from the lower 4-bits of Data 4.
- Tx Time — Time in seconds since the last message received. This is set to 0 when the transmitter address is programmed.
- FP analogue 1 — Floating point (IEEE 754 FLOAT32) converted value from Data 1 and Data 2.
- FP analogue 2 — Floating point (IEEE 754 FLOAT32) converted value from Data 3 and Data 4.

Note: The data in these additional registers will only be valid for certain transmitter types. For more information see ***Section 6*** for details of transmitter types.

There is also an additional special register.

- Firmware ID2 – 1 for a 32 channel receiver.
  2 for a 255 channel receiver.

5.3.4 Alternate 16-Bit Register Map

The alternate 16-bit register map is available from firmware version 10 onwards. This duplicates the registers in the standard 16-bit map and adds the extra information listed in the alternate 8-bit map above.

Tx. No.	Modbus Register Address		Description	
	Dec	Hex	High byte	Low byte
N/A	3010	0BC2	0	Type ID
…	…	…	…	
N/A	21984	55E0	Firmware ID	
	21985	55E1	Firmware Version	
	21986	55E2	Frequency Band	
	21987	55E3	0	Firmware ID2
	21988-21999	55E4-55EF	12 Words Spare	
1	22000	55F0	Transmitter Address HO 16bits	
	22001	55F1	Transmitter Address LO 16bits	
	22002	55F2	0	Transmitter Status
	22003	55F3	0	Data 1
	22004	55F4	Data 2	Data 3
	22005	55F5	0	Data 4
	22006	55F6	0	Tx Type
	22007	55F7	0	RSSI
	22008	55F8	0	Tx Counter
	22009	55F9	Tx Time	
	22010	55FA	FP analogue 1, Bytes 1 & 2	
	22011	55FB	FP analogue 1, Bytes 3 & 4	
	22012	55FC	FP analogue 2, Bytes 1 & 2	
	22013	55FD	FP analogue 2, Bytes 3 & 4	
	22014-22015	55FE-55FF	2 Words Spare	

There is one more register listed in the map extract above.

- Type ID – This is a fixed value of 17140_{dec}.

5.4 Modbus Command Examples

Similarly, the host has to configure the addresses in the same way.
Below are some example messages:

Note: When using 8-bit addressing mode all data is transmitted as an 8-bit value and Modbus reads registers as 16-bit (2 registers). Therefore the high register is always set as 00.

5.4.1 Code 66 – Program Unit ID

To configure the receiver with unit ID 1001_{hex} using Modbus address 4

00 42 10 01 04 xx xx – all values are hexadecimal

Where	00	- Global Message
	42	- Message Code 66
	10 01	- 16-bit Receiver Address
	04	- Modbus Address (to be programmed into unit)
	xx xx	- CRC

To configure the receiver unit ID 1010 with a Modbus address 10

00 42 10 10 0A xx xx – all values are hexadecimal

5.4.2 Code 68 – Set Serial Protocol

To set receiver ID 1001_{hex} to 19200Baud, Even Parity & 1-Stop Bit

00 44 10 01 4B 00 01 01 xx xx – all values are hexadecimal

Where	00	- Global Message
	44	- Message Code 68
	10 01	- 16-bit Receiver Address
	4B 00	- 19200 Baud (Supported rates – 2400, 4800, 9600, 19200) $19200_{dec} = 4B00_{hex}$ $9600_{dec} = 2580_{hex}$ $4800_{dec} = 12C0_{hex}$ $1200_{dec} = 04B0_{hex}$
	01	- 00 = No Parity, 01 = Even Parity, 02 = Odd Parity
	01	- 1 Stop Bit
	xx xx	- CRC

5.4.3 Code 03 – Read Holding Registers (8-Bit Addressing)

Read Ch1 – Ch8 from Modbus Slave 10

0A 03 03 10 00 40 xx xx - all values are hexadecimal

where 0A - Slave Address 10
03 - Modbus Function Code
03 10 - First Register to Read
00 40 - 64 Registers to Read (***Appendix 1***)
xx xx - CRC

Read Ch9 – Ch16 from Modbus Slave 10

0A 03 03 50 00 40 xx xx - all values are hexadecimal

Read Ch17 – Ch24 from Modbus Slave 10

0A 03 03 90 00 40 xx xx - all values are hexadecimal

Read Ch25 – Ch32 from Modbus Slave 10

0A 03 03 D0 00 40 xx xx - all values are hexadecimal

5.4.4 Code 03 – Read Holding Registers (16-Bit Addressing)

Read Ch1 – Ch8 from Modbus Slave 10

0A 03 10 10 00 30 xx xx - all values are hexadecimal

where 0A - Slave address 10
03 - Modbus Function Code
10 10 - First Register to Read
00 30 - 48 Registers to read (***Appendix 1***)
xx xx - CRC

5.4.5 Code 16 – Write Multiple Register (8-bit Addressing)

Write address 51234_{dec} ($00C822_{hex}$) into Ch1 location and clear all other locations- to Slave 4

04 10 03 10 00 08 10 00 00 00 C8 00 22 00 00 00 00 00 00 00 00 00 00 xx xx

where 04 - Slave address 4
10 - Modbus Function Code 16
03 10 - First Register to Write to
00 08 - Number of Registers to Write to
10 - Number of Bytes following (16)
00 <u>00</u> 00 <u>C8</u> 00 <u>22</u> - address written into first 3 registers
00 00 - Register 4 – Status Byte
00 00 - Register 5 – Data Byte1
00 00 - Register 6 – Data Byte2
00 00 - Register 7 – Data Byte3
00 00 - Register 8 – Data Byte4
xx xx - CRC

5.4.6 Code 16 – Write Multiple Register (16-Bit Addressing)

Write address 51234_{dec} ($00C822_{hex}$) into Ch1 location and clear all other locations- to Slave 4

04 10 10 10 00 05 0C 00 00 C8 22 00 00 00 00 00 00 00 00 xx xx

where	04	- Slave address 4
	10	- Modbus Function Code 16
	10 10	- First Register to Write to
	00 05	- Number of Registers to Write to
	0C	- Number of Bytes Following (12)
	00 00	- Transmitter Address Hi 16 bits
	C8 22	- Transmitter Address Lo 16 bits
	00 00	- Register 3 (0, Transmitter Status)
	00 00	- Register 4 (0,Data 1)
	00 00	- Register 5 (Data 2,Data 3)
	00 00	- Register 6 (0,Data 4)
	xx xx	- CRC

6 Transmitter Types

This section describes a description of the data packet breakdowns for popular types of transmitter. A complete list of transmitter types is available – please contact HWM.

6.1 Pulse Transmitter – Type 81

Transmitter Status Byte	Bit 7 – Set for Low Battery. Bit 6 – Not Used. Bit 5 – Not Used. Bit 4 – Tamper (if available). Bits 0-3 – Firmware Revision.
Data 1 – Data 3	24-bit Pulse Count Value. For example, Value $234455_{hex} = 2311253_{dec}$
Data 4	High nibble is signal strength (between 0-10) for VHF transmitters. Cumulative Counter that increments for each transmission. Low nibble is incremental counter (between 0-15).

6.2 Digital Temperature Transmitter – Type 82

Transmitter Status Byte — Bit 7 – Set for Low Battery.
Bit 6 – Not Used.
Bit 5 – Not Used.
Bit 4 – Not Used.
Bits 0-3 – Firmware Revision.

Single Channel

Data 1 – Data 2	10-bit Temperature Channel 1 (High Byte First).
Data 3	Not Used.
Data 4	High nibble is signal strength (between 0-10) for VHF transmitters. Cumulative Counter that increments for each transmission. Low nibble is incremental counter (between 0-15).

Dual Channel

Data 1 – Data 2	10-bit Temperature Channel 1 (High Byte First).
Data 3 – Data 4	10-bit Temperature Channel 2 (High Byte First).

Positive Temperature readings are denoted by MSB clear.
Negative Temperature readings are denoted by MSB set.

- 0002_{hex} → +1.0°C
- 0001_{hex} → +0.5°C
- 0000_{hex} → 0°C
- $0FFF_{hex}$ → -0.5°C
- $0FFE_{hex}$ → -1.0°C

Examples for Dallas Sensors

8034_{hex} = MSB set - Invalid Temperature

$0032_{hex} = 50_{dec}$	Temperature = 50/2 = **+<u>25.0°C</u>**
$0FFE_{hex} = 4094_{dec}$	Temperature = (4096 - 4094)/2 = **-<u>1.0°C</u>**
$0FE0_{hex} = 4064_{dec}$	Temperature = (4096 - 4064)/2 = **-<u>16.0°C</u>**

16-bit Temperature Format (e.g. PT100 Temperature Transmitters)

Data 1 – Data 2	16-bit Temperature Channel 1 (High Byte First).
Data 3	Not Used.
Data 4	Cumulative Counter that increments for each transmission.

Positive Temperature readings are denoted by MSB clear.
Negative Temperature readings are denoted by MSB set.

- 0002_{hex} → +1.0°C
- 0001_{hex} → +0.5°C
- 0000_{hex} → 0°C
- $7FFF_{hex}$ → -0.5°C
- $7FFE_{hex}$ → -1.0°C

Examples for PT100 Sensor

8034_{hex} = MSB set - Invalid Temperature

$0034_{hex} = 52_{dec}$	Temperature = 52/2 = **+<u>26.0°C</u>**
$7FF0_{hex} = 32752_{dec}$	Temperature = (32768 - 32752)/2 = **-<u>8.0°C</u>**
$7FC3_{hex} = 32707_{dec}$	Temperature = (32768 - 32707)/2 = **-<u>30.5°C</u>**

Floating Point Registers

Version 10 firmware handles temperature as 16-bit floating point values. These are available in the floating point analogue registers (FP Analogue 1 and FP Analogue 2) of the alternate register maps.

6.3 Relative Humidity & Temperature Transmitter – Type 83

Transmitter Status Byte	Bit 7 – Set for Low Battery.
	Bit 6 – Not Used.
	Bit 5 – Not Used.
	Bit 4 – Not Used.
	Bits 0-3 – Firmware Revision.
Data 1 – Data 2:	16-bit Raw Humidity Value.
Data 3 – Data 4:	16-bit Raw Temperature Value.

Temperature

Temperature (°C) = [Measured x 0.01] - 40

Temperature (Measured) = $17A9_{hex} \equiv 6057_{dec}$
Temperature (°C) = [6057 x 0.01] - 40
= +**<u>20.57°C</u>**

Humidity

Humidity (%) = [-0.0000028 x (Measured)^2] + [Measured x 0.0405] – 4

For example: Raw Value = $0544_{hex} \equiv 1348_{dec}$

Humidity (%) = [-0.0000028 x 1348 x 1348] + [1348 x 0.0405] - 4
= -5.08789 + 54.594 - 4
= **<u>45.50%</u>**

For a temperature compensated humidity value the following calculation will need to be applied:

Humidity = (Temperature - 25) x (0.01 + 0.00008 x Raw Value) + Calculated Humidity

For the example above:
Temperature Compensated Humidity (%) = (20.57-25) x (0.01 + 0.00008 x 1348) + 45.50
= **<u>44.98%</u>**

6.4 Analogue Current/Voltage Transmitter – Type 85

Transmitter Status:	Bit 7 – Set for Low Battery.
	Bit 6 – Not Used.
	Bit 5 – Not Used.
	Bit 4 – Not Used.
	Bits 0-3 – Firmware Revision.
Data 1 – Data 2:	16-bit Raw ADC Value (10-bit).
Data 3 – Data 4:	16-bit Not Defined (Currently used as supply indication).

Note: Refer to individual linearity chart supplied in the packaging carton of each transmitter for ADC values for input current or voltage references.

6.5 Contact (Open/Close) Transmitter – Type 87

Transmitter Status Byte	Bit 7 – Set for Low Battery. Bit 6 – Not Used. Bit 5 – Contact Status Ch2 (Set is Closed). Bit 4 – Contact Status Ch1 (Set is Closed). Bits 0-3 – Firmware Revision.
Data 1 – Data 3	24-bit Pulse Count Value. For example, Value $234455_{hex} = 2311253_{dec}$
Data 4	Cumulative Counter that increments for each data transmission. High nibble is signal strength (between 0-10) for VHF transmitters. Low nibble is incremental counter (between 0-15).

6.6 Alarm/Status Transmitter – Type 88

Transmitter Status Byte	Bit 7 – Set for Low Battery. Bit 6 – Not Used. Bit 5 – Alarm/Status Ch2 (Set is ON, Clear is OFF). Bit 4 – Alarm/Status Ch1 (Set is ON, Clear is OFF). Bits 0-3 – Firmware Revision.
Data 1 – Data 3	24-bit Pulse Count Value. For example, Value $234455_{hex} = 2311253_{dec}$
Data 4	Cumulative Counter that increments for each data transmission. High nibble is signal strength (between 0-10) for VHF transmitters. Low nibble is incremental counter (between 0-15).

6.7 Legionella Transmitter – Type 7A

Transmitter Status Byte	Bit 7 – Set for Low Battery. Bit 6 –Temperature Probe Fault. Bit 5 – Cold Water Valve Open. Bit 4 – Hot Water Valve Open. Bit 3 – Flow Fault. a) When the hot or cold water valve is open and if the bit is set, this indicates no water is detected. b) When the hot or cold water valve is closed and if the bit is set this indicates water is detected. Bits 0-2 – Firmware Revision.
Data 1 – Data 2	10-bit Temperature Channel 1 (High Byte First). ***Temperature (°C) = Value/10*** Positive Temperature readings are denoted by MSB (bit 15) clear. Negative Temperature readings are denoted by MSB (bit 15) set. 0002_{hex} → +1.0°C 0001_{hex} → +0.5°C 0000_{hex} → 0°C 8005_{hex} → -0.5°C 8010_{hex} → -1.0°C

Examples

$00F1_{hex} = 241_{dec}$	Temperature = 241/10 = **+<u>24.1°C</u>**
$8040_{hex} = 64_{dec}$	Temperature = 32832 - 32768 = 64_{dec} 64/10 = **-<u>6.4°C</u>**
$800B_{hex} = 11_{dec}$	Temperature = 32779 - 32768 = 11_{dec} 11/10 = **-<u>1.1°C</u>**
$8136_{hex} = 310_{dec}$	Temperature = 33078 - 32768 = 64_{dec} 310/10 = **-<u>31.0°C</u>**

Data 3 – Data 4	Not Used (Raw ADC) – Factory Use.

6.8 Analogue Transmitter – Type F1

There is a selection of different transmitters i.e. 0-10V Analogue and PT100 Temperature that use the F1 type. These are the most commonly used variants.

Single Channel Voltage 0 – 10V

Transmitter Status Byte	Bit 7 – Set for Low Battery.
	Bit 6 – ADC Type bit = 1.
	Bit 5 – Not Used.
	Bit 4 – Not Used.
	Bits 0-3 – Firmware Revision.
Data 1 – Data 2	10-bit Voltage Channel 1.
Data 3	Not Used.
Data 4	Not Used.

Dual Channel Voltage 0 – 10V

Transmitter Status Byte	Bit 7 – Set for Low Battery.
	Bit 6 – ADC Type bit = 1.
	Bit 5 – Not Used.
	Bit 4 – Not Used.
	Bits 0-3 – Firmware Revision.
Data 1 – Data 2	10-bit Voltage Channel 1.
Data 3 – Data 4	10-bit Voltage Channel 2.

16-bit Voltage Format: Voltage = Value x 0.0122

Floating Point Registers

Version 10 firmware provides the ADC values as floating point analogue registers (FP Analogue 1 & FP Analogue 2) of the alternate register maps.

6.9 Pressure Transmitter – Type F2

Pressure Channel – 1st Transmission

Transmitter Status Byte	Bit 7 – Set for Low Battery. Bit 6 – Not Used. Bit 5 – Not Used. Bit 4 – Not Used. Bits 0-3 – Firmware Revision.
Data 1 – Data 2	16-bit Pressure Value in Deci-meters (dm).
Data 3 – Data 4	16-bit Pressure Value (Raw ADC Value).

Flow Channel – 2nd Transmission

Transmitter Status Byte	Bit 7 – Set for Low Battery. Bit 6 – Not Used. Bit 5 – Not Used. Bit 4 – Not Used. Bits 0-3 – Firmware Revision.
Data 1 – Data 3	24-bit Pulse/Flow.
Data 4	Cumulative Counter that increments for each transmission. High nibble is signal strength (between 0-10) – VHF Only. Low nibble is incremental counter (between 0-15).

Appendix 1

Standard 8-Bit Register Map

The standard 8-bit register map is as follows. The base address of a given channel can be calculated using the formula.

Channel Base Address = 784 + [(Channel - 1) x 8]

For example, channel 32 is 784 + [31 x 8] = 1032_{dec} or 0408_{hex}.

Tx. No.	Modbus Register Address		Description
	Dec	Hex	
1	784	0310	Transmitter Address HHO
	785	0311	Transmitter Address HO
	786	0312	Transmitter Address LO
	787	0313	Transmitter Status
	788	0314	Data 1
	789	0315	Data 2
	790	0316	Data 3
	791	0317	Data 4
2	792	0318	Transmitter Address HHO
	793	0319	Transmitter Address HO
	794	031A	Transmitter Address LO
	795	031B	Transmitter Status
	796	031C	Data 1
	797	031D	Data 2
	798	031E	Data 3
	799	031F	Data 4
…	…	…	…
32	1032	0408	Transmitter Address HHO
	1033	0409	Transmitter Address HO
	1034	040A	Transmitter Address LO
	1035	040B	Transmitter Status
	1036	040C	Data 1
	1037	040D	Data 2
	1038	040E	Data 3
	1039	040F	Data 4
…	…	…	…
255	2816	0B00	Transmitter Address HHO
	2817	0B01	Transmitter Address HO
	2818	0B02	Transmitter Address LO
	2819	0B03	Transmitter Status
	2820	0B04	Data 1
	2821	0B05	Data 2
	2822	0B06	Data 3
	2823	0B07	Data 4

The transmitter address and data values are 24-bit, Modbus works with 16-bit registers, all the above registers are sent as 16-bit addresses. The high byte will always be 00, e.g. a pulse count of 34562_{dec} (008702_{hex}), will be sent as 0x0000, 0x0087 & 0x0002 for Data 1, Data 2 and Data 3.

Standard 16-Bit Register Map

The alternate 16-bit map is available with firmware version 8 onwards.

The standard 16-bit register map is as follows. The base address of a given channel can be calculated using the formula.

Channel Base Address = 4112 + [(Channel - 1) x 6]

For example, channel 32 is 4112 + [31 x 6] = 4298_{dec} or $01CA_{hex}$.

Tx. No.	Modbus Register Address		Description	
	Dec	Hex	High byte	Low byte
N/A	4096	1000	Firmware ID	
	4097	1001	Firmware Version	
	4098	1002	Frequency Band	
1	4112	1010	Transmitter Address HO 16bits	
	4113	1011	Transmitter Address LO 16bits	
	4114	1012	0	Transmitter Status
	4115	1013	0	Data 1
	4116	1014	Data 2	Data 3
	4117	1015	0	Data 4
2	4118	1016	Transmitter Address HO 16bits	
	4119	1017	Transmitter Address LO 16bits	
	4120	1018	0	Transmitter Status
	4121	1019	0	Data 1
	4122	101A	Data 2	Data 3
	4123	101B	0	Data 4
…	…	…	…	
32	4298	01CA	Transmitter Address HO 16bits	
	4299	01CB	Transmitter Address LO 16bits	
	4300	01CC	0	Transmitter Status
	4301	01CD	0	Data 1
	4302	01CE	Data 2	Data 3
	4303	01CF	0	Data 4
…	…	…	…	
255	5636	1604	Transmitter Address HO 16bits	
	5637	1605	Transmitter Address LO 16bits	
	5638	1606	0	Transmitter Status
	5639	1607	0	Data 1
	5640	1608	Data 2	Data 3
	5641	1609	0	Data 4

Note: All the above registers are sent as 16 bit data with the high byte sent first, e.g. a pulse count of 34562_{dec} (8702_{hex}), will be sent as 0x87 & 0x02.

Alternate 8-Bit Register Map

The alternate 8-bit map is available with firmware version 10 onwards.

The alternate 8-bit register map is as follows. The base address of a given channel can be calculated using the formula.

Channel Base Address = 5664 + [(Channel - 1) x 32]

For example, channel 32 is 5664 + [31 x 32] = 6656_{dec} or $1A00_{hex}$.

Tx. No.	Modbus Register Address		Description
	Dec	Hex	
N/A	5648	1610	Firmware ID HO
	5649	1611	Firmware ID LO
	5650	1612	Firmware Version
	5651	1613	Frequency Band
	5652	1614	Firmware ID2
	5653-5663	1615-161F	11 Bytes Spare
1	5664	1620	Transmitter Address HHO
	5665	1621	Transmitter Address HO
	5666	1622	Transmitter Address LO
	5667	1623	Transmitter Status
	5668	1624	Data 1
	5669	1625	Data 2
	5670	1626	Data 3
	5671	1627	Data 4
	5672	1628	Transmitter Type
	5673	1629	RSSI
	5674	162A	Tx Counter
	5675	162B	Tx Time HO
	5676	162C	Tx Time LO
	5677	162D	FP Analogue 1, Byte 1
	5678	162E	FP Analogue 1, Byte 2
	5679	162F	FP Analogue 1, Byte 3
	5680	1630	FP Analogue 1, Byte 4
	5681	1631	FP Analogue 2, Byte 1
	5682	1632	FP Analogue 2, Byte 2
	5683	1633	FP Analogue 2, Byte 3
	5684	1634	FP Analogue 2, Byte 4
	5685-5695	1635-163F	11 Bytes Spare

Continued over page…

Tx. No.	Modbus Register Address		Description
	Dec	Hex	
2	5696	1640	Transmitter Address HHO
	5697	1641	Transmitter Address HO
	5698	1642	Transmitter Address LO
	5699	1643	Transmitter Status
	5700	1644	Data 1
	5701	1645	Data 2
	5702	1646	Data 3
	5703	1647	Data 4
	5704	1648	Transmitter Type
	5705	1649	RSSI
	5706	164A	Tx Counter
	5707	164B	Tx Time HO
	5708	164C	Tx Time LO
	5709	164D	FP Analogue 1, Byte 1
	5710	164E	FP Analogue 1, Byte 2
	5711	164F	FP Analogue 1, Byte 3
	5712	1650	FP Analogue 1, Byte 4
	5713	1651	FP Analogue 2, Byte 1
	5714	1652	FP Analogue 2, Byte 2
	5715	1653	FP Analogue 2, Byte 3
	5716	1654	FP Analogue 2, Byte 4
	5717-5727	1655-165F	11 Bytes Spare
…	…	…	…
32	6656	1A00	Transmitter Address HHO
	6657	1A01	Transmitter Address HO
	6658	1A02	Transmitter Address LO
	6659	1A03	Transmitter Status
	6660	1A04	Data 1
	6661	1A05	Data 2
	6662	1A06	Data 3
	6663	1A07	Data 4
	6664	1A08	Transmitter Type
	6665	1A09	RSSI
	6666	1A0A	Tx Counter
	6667	1A0B	Tx Time HO
	6668	1A0C	Tx Time LO
	6669	1A0D	FP Analogue 1, Byte 1
	6670	1A0E	FP Analogue 1, Byte 2
	6671	1A0F	FP Analogue 1, Byte 3
	6672	1A10	FP Analogue 1, Byte 4
	6673	1A11	FP Analogue 2, Byte 1
	6674	1A12	FP Analogue 2, Byte 2
	6675	1A13	FP Analogue 2, Byte 3
	6676	1A14	FP Analogue 2, Byte 4
	6677-6687	1A15-1A1F	11 Bytes Spare

Continued over page…

Tx. No.	Modbus Register Address		Description
	Dec	Hex	
255	13792	35E0	Transmitter Address HHO
	13793	35E1	Transmitter Address HO
	13794	35E2	Transmitter Address LO
	13795	35E3	Transmitter Status
	13796	35E4	Data 1
	13797	35E5	Data 2
	13798	35E6	Data 3
	13799	35E7	Data 4
	13800	35E8	Transmitter Type
	13801	35E9	RSSI
	13802	35EA	Tx Counter
	13803	35EB	Tx Time HO
	13804	35EC	Tx Time LO
	13805	35ED	FP Analogue 1, Byte 1
	13806	35EE	FP Analogue 1, Byte 2
	13807	35EF	FP Analogue 1, Byte 3
	13808	35F0	FP Analogue 1, Byte 4
	13809	35F1	FP Analogue 2, Byte 1
	13810	35F2	FP Analogue 2, Byte 2
	13811	35F3	FP Analogue 2, Byte 3
	13812	35F4	FP Analogue 2, Byte 4
	13813-13823	35F5-35FF	11 Bytes Spare

Alternate 16-Bit Register Map

The alternate 16-bit map is available with firmware version 10 onwards.

The alternate 16-bit register map is as follows. The base address of a given channel can be calculated using the formula.

Channel Base Address = 22000 + [(Channel - 1) x 16]

For example, channel 32 is 22000 + [31 x 16] = 22496_{dec} or $57E0_{hex}$.

Tx. No.	Modbus Register Address		Description	
	Dec	Hex	High byte	Low byte
N/A	3010	0BC2	0	Type ID
…	…	…	…	
N/A	21984	55E0	Firmware ID	
	21985	55E1	Firmware Version	
	21986	55E2	Frequency Band	
	21987	55E3	0	Firmware ID2
	21988-21999	55E4-55EF	12 Words Spare	
1	22000	55F0	Transmitter Address HO 16bits	
	22001	55F1	Transmitter Address LO 16bits	
	22002	55F2	0	Transmitter Status
	22003	55F3	0	Data 1
	22004	55F4	Data 2	Data 3
	22005	55F5	0	Data 4
	22006	55F6	0	Tx Type
	22007	55F7	0	RSSI
	22008	55F8	0	Tx Counter
	22009	55F9	Tx Time	
	22010	55FA	FP Analogue 1, Bytes 1 & 2	
	22011	55FB	FP Analogue 1, Bytes 3 & 4	
	22012	55FC	FP Analogue 2, Bytes 1 & 2	
	22013	55FD	FP Analogue 2, Bytes 3 & 4	
	22014-22015	55FE-55FF	2 Words Spare	
2	22016	5600	Transmitter Address HO 16bits	
	22017	5601	Transmitter Address LO 16bits	
	22018	5602	0	Transmitter Status
	22019	5603	0	Data 1
	22020	5604	Data 2	Data 3
	22021	5605	0	Data 4
	22022	5606	0	Tx Type
	22023	5607	0	RSSI
	22024	5608	0	Tx Counter
	22025	5609	Tx Time	
	22026	560A	FP Analogue 1, Bytes 1 & 2	
	22027	560B	FP Analogue 1, Bytes 3 & 4	
	22028	560C	FP Analogue 2, Bytes 1 & 2	
	22029	560D	FP Analogue 2, Bytes 3 & 4	
	22030-22031	560E-560F	2 Words Spare	

Continued over page…

Tx. No.	Modbus Register Address		Description	
	Dec	Hex	High byte	Low byte
32	22496	57E0	Transmitter Address HO 16bits	
	22497	57E1	Transmitter Address LO 16bits	
	22498	57E2	0	Transmitter Status
	22499	57E3	0	Data 1
	22500	57E4	Data 2	Data 3
	22501	57E5	0	Data 4
	22502	57E6	0	Tx Type
	22503	57E7	0	RSSI
	22504	57E8	0	Tx Counter
	22505	57E9	Tx Time	
	22506	57EA	FP Analogue 1, Bytes 1 & 2	
	22507	57EB	FP Analogue 1, Bytes 3 & 4	
	22508	57EC	FP Analogue 2, Bytes 1 & 2	
	22509	57ED	FP Analogue 2, Bytes 3 & 4	
	22510-22511	57EE-57EF	2 Words Spare	
…	…	…	…	
255	26064	65D0	Transmitter Address HO 16bits	
	26065	65D1	Transmitter Address LO 16bits	
	26066	65D2	0	Transmitter Status
	26067	65D3	0	Data 1
	26068	65D4	Data 2	Data 3
	26069	65D5	0	Data 4
	26070	65D6	0	Tx Type
	26071	65D7	0	RSSI
	26072	65D8	0	Tx Counter
	26073	65D9	Tx Time	
	26074	65DA	FP Analogue 1, Bytes 1 & 2	
	26075	65DB	FP Analogue 1, Bytes 3 & 4	
	26076	65DC	FP Analogue 2, Bytes 1 & 2	
	26077	65DD	FP Analogue 2, Bytes 3 & 4	
	26078-26079	65DE-65DF	2 Words Spare	

Note: All the above registers are sent as 16-bit data with the high byte sent first, e.g. for a pulse count of 34562_{dec} (8702_{hex}), will be sent as 0x87 & 0x02.

Appendix 2

Modbus CRC Algorithm

A 16-bit CRC checksum is implemented on every message to detect any bit errors in the message. The checksum calculation is only used to detect errors only.

The CRC generating polynomial used is: $x^{16} + x^{15} + x^2 + 1$

CRC Algorithm:

1. Load a 16-bit register with FFFF.
2. Exclusively OR the first 8-bit byte with the high order byte of the 16 bit register. The result put into the 16-bit register.
3. Shift the 16-bit register one bit to the right.
4. If the bit shifted to the right is a 0, return to step 3 otherwise Exclusively OR the generating polynomial 1010 0000 0000 0001 with the 16-bit register.
5. Repeat steps 3 & 4 until 8 shift have been performed.
6. Exclusively OR the next 8-bit byte with the 16-bit register.
7. Repeat step 3 through 6 until all bytes of the message have been exclusively OR'd with the 16-bit register and shifted 8 times.
8. The contents of the 16-bit register are the 2 byte CRC error check and is added to the message MSB first.

Code Example

The following example shows how the CRC can be verified in ‘C’

```
#include <stdio.h>

/**************************************************************
 Sample code to check CRC data from a Wi5

 The sample data starts from the transmitter type, it does not
 include the Date Time Stamp.  The transmitter type is
 included in the data packet sent to the crc_check routine
 even though the type byte is not included in the CRC check.
 This is to maintain compatibility with the User guide.

**************************************************************/

typedef enum{
  CRC_OK,
  CRC_NG
}ENUM_CRC_CHECK;

ENUM_CRC_CHECK crc_check(unsigned char *d)
  {
    unsigned short CRC;
    int count, count2;

    CRC = 0xffff;
    d++;

    for(count2 = 0; count2 < 8 ; count2++)
      {
      CRC ^= *d++;
      for(count = 0; count < 8; count++)
        {
          if(CRC & 1)
            CRC = (CRC >> 1) ^ 0xa001;
          else
            CRC = (CRC >> 1);
        }
      }

    if((((CRC>>8) & 0xff)==d[1]) && ((CRC & 0xff)==d[0]))
      return CRC_OK;


   return CRC_NG;

  }


int main(void)
{
  /*Test example which passes CRC check*/
  unsigned char data[]={0x81,0x02,0xB4,0x69,0x02,0x00,0x00,0x15,0x94,0x4A,0x1F};

  if(crc_check(data)==CRC_OK)
     printf("CRC OK\n");
  else
     printf("CRC NG\n");
}
```