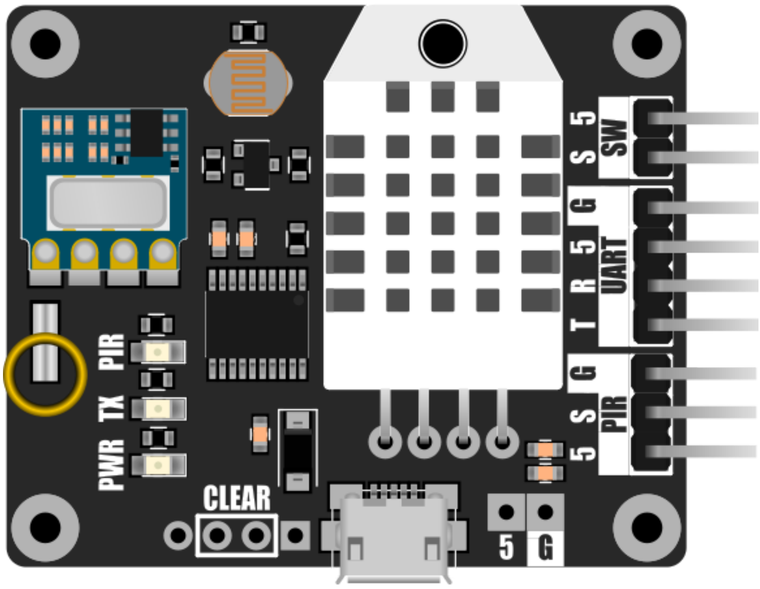
SmartRFy Sensor Module Manual



HCMODU0141

Revision 1.1.0

DISCLAIMER

SmartRFy modules are micro power short range devices (SRD) using the 433MHz frequency band. This frequency band is licensed exempt (within certain restrictions) in many countries including the UK, Europe, Asia, and the US. It is the end users responsibility to ensure that it is legal to operate devices in this frequency band within your own country before use.

As with all wireless devices, external factors such as range and interference may cause transmissions to be corrupted or blocked. Therefore these devices should not be used in applications where control or monitoring is a critical requirement.

This document is provided "as is". Hobby Components Ltd makes no warranties, whether express, implied or statutory, including, but not limited to, implied warranties of merchantability and fitness for a particular purpose, accuracy or lack of negligence. Hobby components Ltd shall not, in any circumstances, be liable for any damages, including, but not limited to, special, incidental or consequential damages for any reason whatsoever.

COPYRIGHT NOTICE

Table of Contents

SmartRFy system overview 4

SmartRFy sensor module 7

Features 8

Specification 9

Sensor Module Features 10

Power 10

DHT22 Temperature / Humidity sensor 10

PIR 11

LDR 12

Auxiliary switch output 12

LEDs 13

Serial UART 14

Command interface feature 14

Wireless serial port feature 14

Default configuration settings 15

Clear (Factory Reset) 15

SmartRFy commands 16

Command formats 16

Command format requirements 16

Local command format 17

SmartRFy generic commands – quick lookup table 18

Sensor module specific commands – quick lookup table 19

Generic SmartRFy commands 20

Firmware version (FWV?) 20

Get the modules zone (ZON?) 21

Get the modules address (ADD?) 21

Serial serial timeout time (STO=) 21

Set the baud rate (BUR=) 21

Set number of transmit resends (TRS=) 22

Set the serial verbose mode (VBM=) 23

Set the modules zone number (ZON=) 23

Set the modules address (ADD=) 24

Turn on/off repeater mode (TRP=) 24

Factory reset (FTRE) 25

SmartRFy sensor module commands 26

Get Humidity (HUM?) 26

Humidity transmit time (HTT=) 26

Humidity transmit name (HTN=) 27

Get Temperature (TMP?) 27

Temperature transmit time (TTT=) 27

Temperature transmit name (TTN=) 28

LDR transmit time (LTT=) 28

LDR transmit name (LTN=) 29

PIR transmit name (PTN=) 29

Set the state of the digital output pin (SW0=) 29

Set an alternative name for the digital output pin (SWN=) 30

Set the output pin high time (SWO=) 30

Set the threshold value to set the output pin high (STH=) 31

Set the threshold value to set a digital pin low (STL=) 31

Set the test condition for setting the output pin high (SMH=) 32

Set the test condition for setting the output pin low (SML=) 33

Appendix A: Example auxiliary output pin modes 34

Default trigger mode 34

Timer mode 35

Threshold mode 35

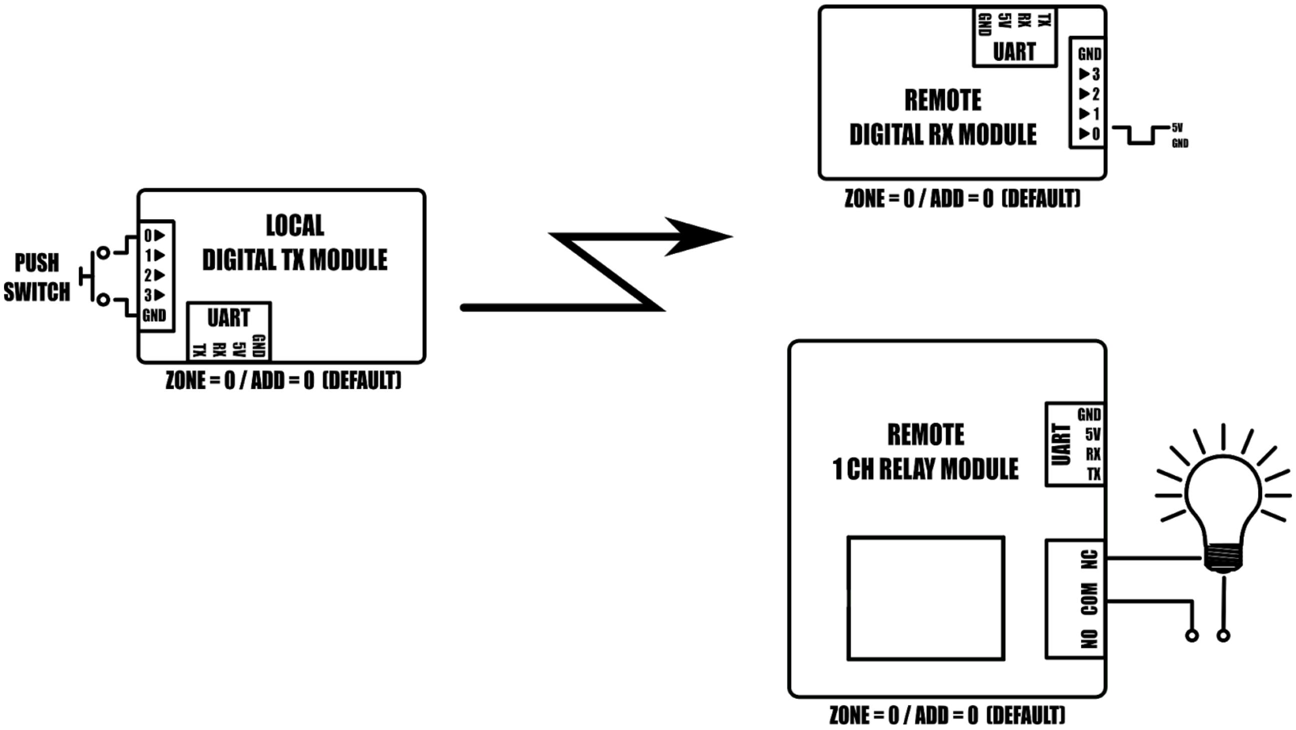
SmartRFy system overview

SmartRFy modules are a range of wireless modules that provide a simple low cost way of controlling and monitoring remote devices and sensors. They are designed to work at their basic level with no programming required. However, when reconfigured or controlled via their serial interface they can also provide more complex and even autonomous functions.

SmartRFy modules work by wirelessly communicating with one another using simple text based commands - for example, to turn on a remote relay a command such as RLY=1 can be sent, and to turn it back off RLY=0. These commands can be automatically sent from one module to another or can be sent by a user via a modules serial interface to manually control modules on the SmartRFy network.

They are addressable and can be grouped into one of 255 zones, with each zone capable of individually addressing 255 devices, giving a total of over 65,000 unique addresses. SmartRFy modules are designed to be both easy and flexible to use by providing three levels of control:

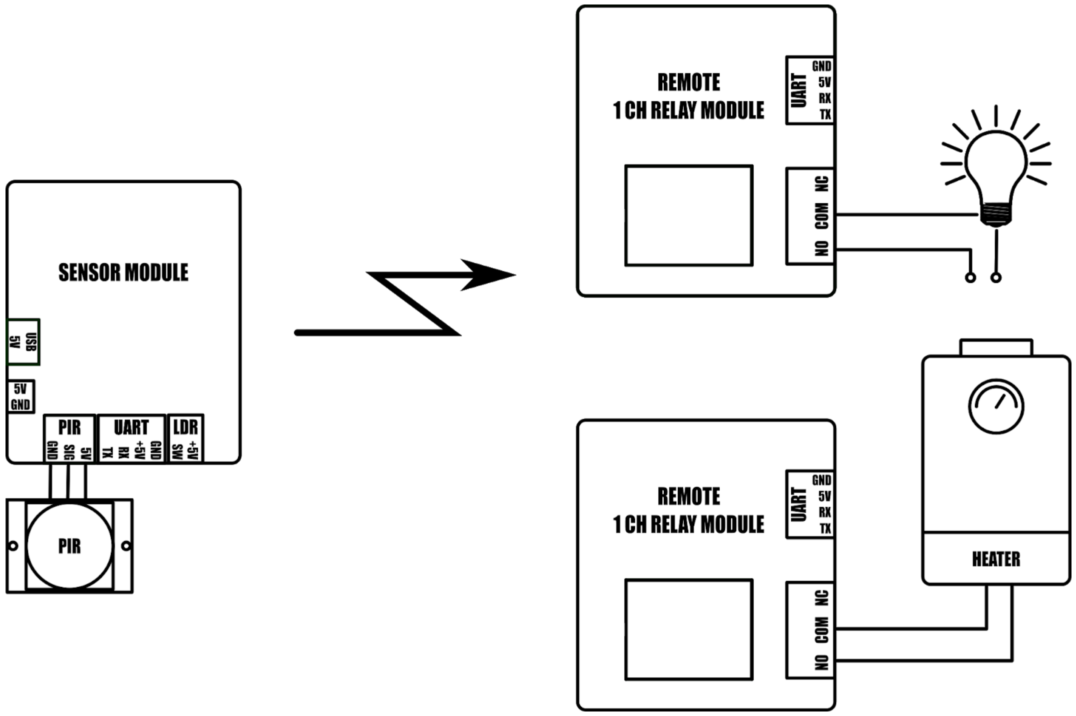
Zero configuration (out-of-the-box)



Example: Controlling a remote digital pin and light from a push switch.

With no configuration at all the SmartRFy range of modules allow basic functionality such as remotely switching relays, digital pins, or monitoring sensors, such as temperature, humidity, light, motion etc. All SmartRFy modules default to the same zone and address and so can respond to, or control, other modules without the need to modify any zone or address settings.

Serial port configuration

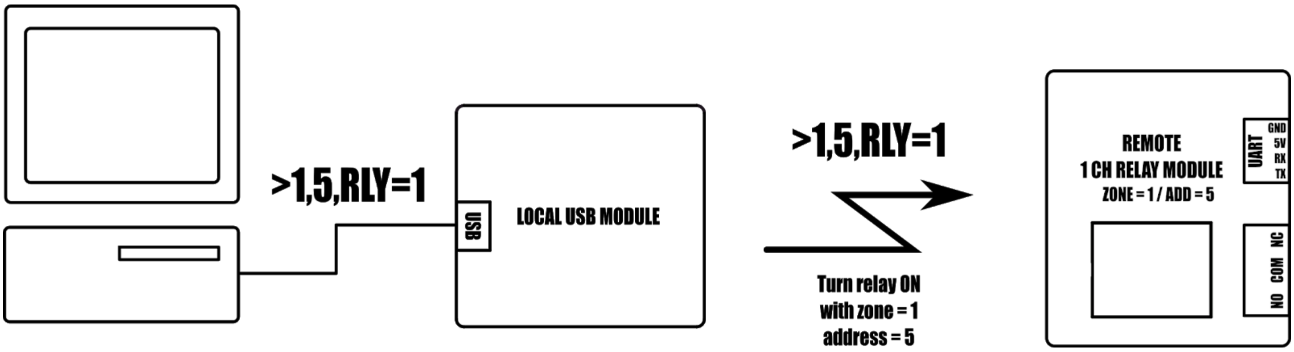


Example: Remote 1CH relay module configured to turn on light when local modules PIR is triggered and second remote relay module is configured to control a heating device based on local modules temperature sensor readings.

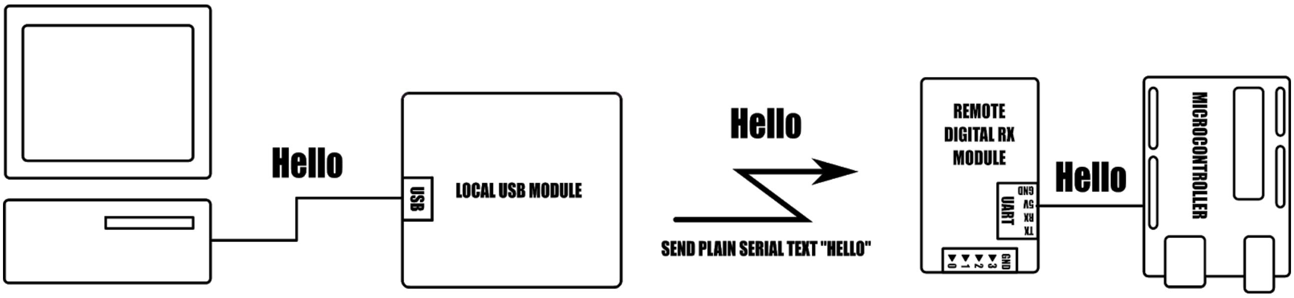
All smartRFy modules include a serial port which allows them to be re-configured using simple text based commands. These text commands allow changing of configuration settings and control of specific features of each module. Any setting changes are stored by the module's non-volatile memory and so are retained even when power is removed from the module. When reconfigured they are able to perform more complex functions and even some basic autonomous tasks. For example, a SmartRFy relay module can be reconfigured to control a heating system based on the temperature transmitted from a remote sensor or, turn on a flood light for a set amount of time when triggered by a remote

PIR – all without any additional hardware. Serial port configuration also allows a modules zone and address (all SmartRFy modules default to zone and address 0) to be changed. This provides the option for modules to be placed into groups, which will then only respond to other modules with a matching zone or address.

Slave controlled



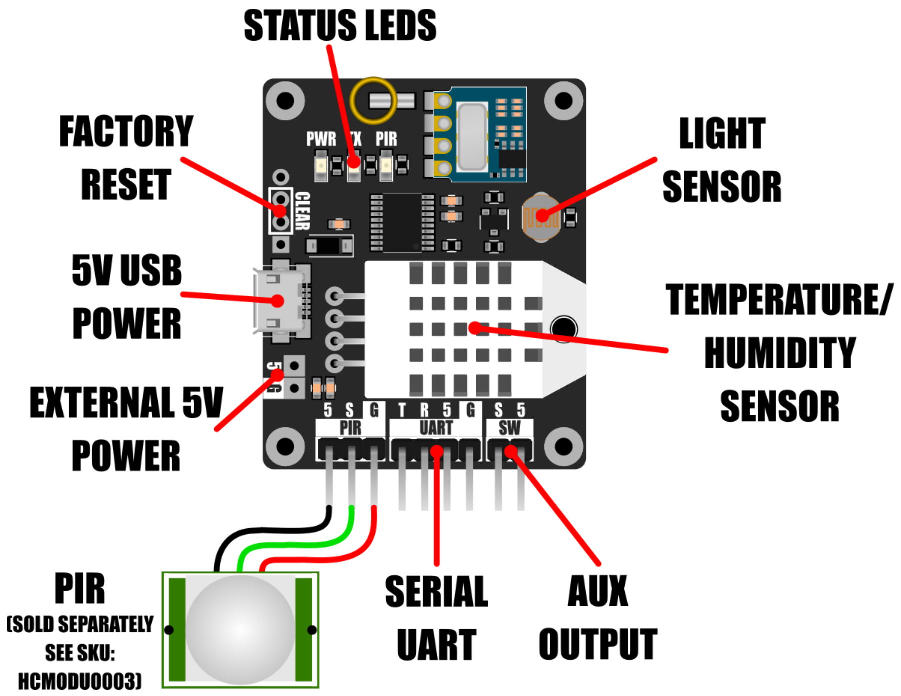
Example 1: Turning on a remote relay with zone =1 and address = 5



Example 2: Send plain serial text “Hello” to a remote microcontroller

SmartRFy modules can also be controlled via their serial port by other devices such as microcontrollers or computer based automation software such as Node-RED. Using a SmartRFy module as a slave device (must be capable of transmitting/receiving) gives a master device the ability to monitor and control any other SmartRFy devices on the network regardless of their zone and address. SmartRFy modules can even be used as a passive wireless serial port for passing non-SmartRFy data to other remotely connected devices.

SmartRFy sensor module



The SmartRFy sensor module adds the option to remotely monitor various sensors via a SmartRFy network connection. The sensor module includes a temperature and humidity sensor (DHT22) and has the ability to measure ambient light levels via an integrated photo-resistor. Out of the box the module will automatically transmit readings from these sensors once every 60 seconds (when reconfigured via the module serial port this can be changed from 10 to 65535 seconds) which can then be received by other SmartRFy devices with receive capability. For example remote sensor readings can be read directly into a computer via a SmartRFy USB module, or a SmartRFy relay module can be configured to trigger depending on sensor values transmitted from a remote sensor module.

The sensor module also includes a header for interfacing common types of 5V PIR motion detectors (see SKU: HCMODU0003). By default the sensor module will transmit the state of a connected PIR whenever its state changes. This transmitted state may then be used by any suitably configured remote SmartRFy module capable of receiving its data.

Additionally, the sensor module includes an auxiliary digital output (labelled SW). This pin can be reconfigured via the sensor module's serial interface to automatically trigger based on max and min values of one of the module's sensors.

Features

- Local and remote control via simple text based serial commands.
- Up to 255 zones and 255 addresses (default zone 0, address 0).
- Basic automation features (requires a USB to serial adaptor or microcontroller for configuration).
- Compatible with most microcontrollers featuring a serial UART interface.
- Wireless serial communication with 8 different baud rates (9600 default).
- Built-in temperature and humidity sensor (DHT22).
- Built-in light sensor (photoresistor).
- Automatically read and transmit temperature, humidity, and light levels (will transmit readings once every 60 seconds by default).
- Can automatically transmit state changes from a connected PIR sensor.
- Auxiliary digital output can be programmed to automatically trigger based on readings from any of the connected sensors and supports on/off, toggle, and timer modes.

Specification

Model number:	HCMODU0141
Supply Voltage:	4.5 to 5.5V
Supply current min:	9.5 mA (idle)
Supply current max:	21 mA (transmit)
Operating frequency:	433MHz (OOK)
Operating range:	30 Meters (unobstructed)
Temperature sensor range:	-40 to 80 oC
Temperature sensor resolution:	0.1oC
Humidity sensor range:	0 to 10 %RH
Humidity sensor resolution:	0.1 %RH
LDR sensor range	0 to 255
PIR input (S) pin characteristics	Input logic low voltage: -0.3V to Vdd x 0.3V Input high voltage Vdd x 0.7 to Vdd + 0.3V
Aux digital output (SW0):	Open collector with 10K pullup. 100mA max sink current
Interfaces:	RF Tx, serial UART, PIR, and relay Aux digital output
Module dimensions (WxDxH):	44.2 x 36.6 x 20mm

All measurements taken at Vdd=5V, 25ºC

Sensor Module Features

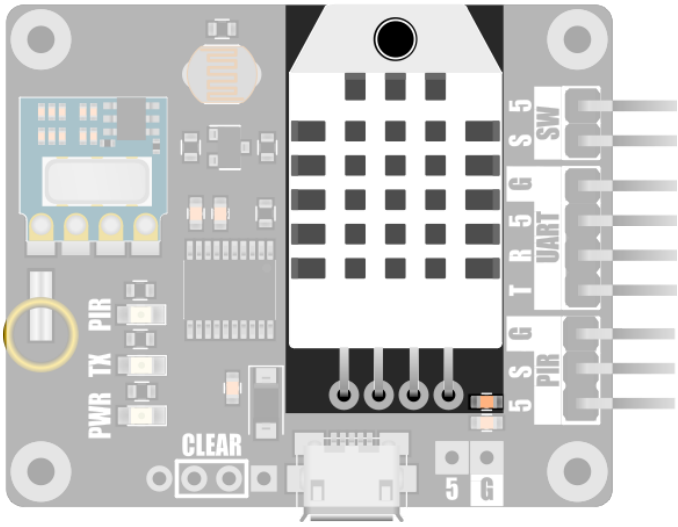
Power

Power can be applied to the module via the microUSB header. This allows the module to be conveniently powered via a USB port or suitable USB mains adapter.

Alternatively the module can be powered via an external power supply ranging from 4.5 to 5.5V with a minimum current capacity of 100mA. The power supply can either be connected to the module's external 5V power pads, or any of the header pins marked 5V and GND.

DHT22 Temperature / Humidity sensor

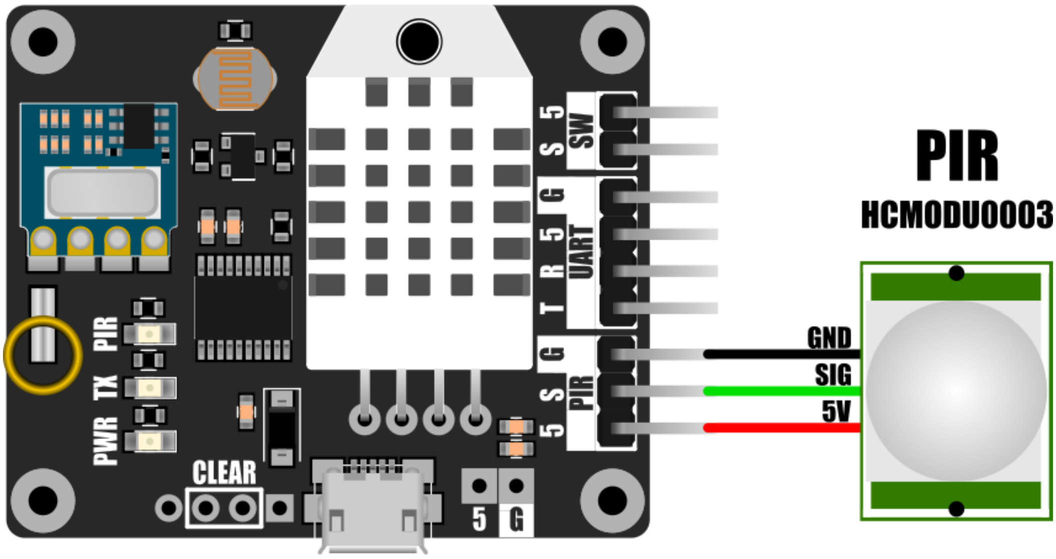
The sensor module includes a built-in DHT22 temperature and humidity sensor.



By default the module is configured to take temperature and humidity measurements from this sensor every 60 seconds. These measurements are then automatically transmitted to all other SmartRFy devices within range and also output to the modules serial interface. This sample rate can be changed via the serial interface using the temperature (TTT) and humidity (HTT) transmit rate commands and can be set anywhere between 10 and 65535 seconds. Setting either transmit rate to 0 will disable reading and transmitting of that measurement.

PIR

An optional PIR sensor (see item HCMODU0003) can be connected to the module via the PIR interface header to allow passive infrared motion sensing. To interface to the sensor module the PIR sensor must have a digital TTL output with a logic high level of 2.4 to 3.6V.

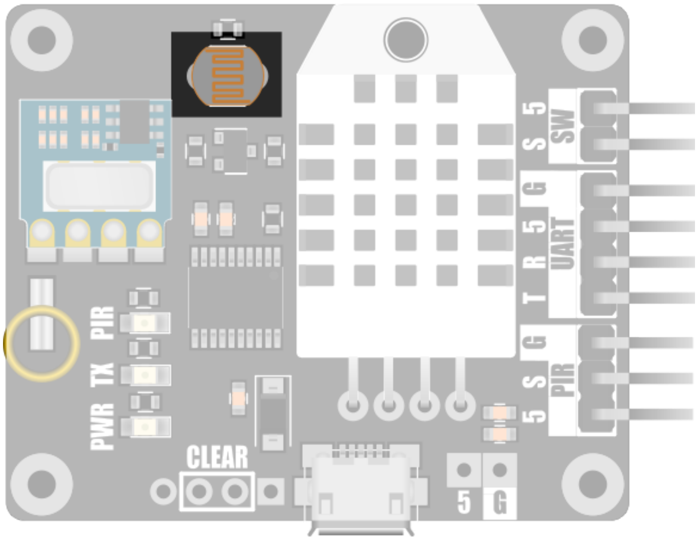


When connected, the sensor module will transmit the state of the PIR to all other SmartRFy devices within range and will also output the state to the modules serial port whenever the logic state of the PIR changes.

Note: There is no pull-up resistor on the S input pin and so it will float between a logic high and low when disconnected from the sensor. Therefore it is recommended to connect this pin to 5V using the supplied jumper when a PIR sensor is not connected.

LDR

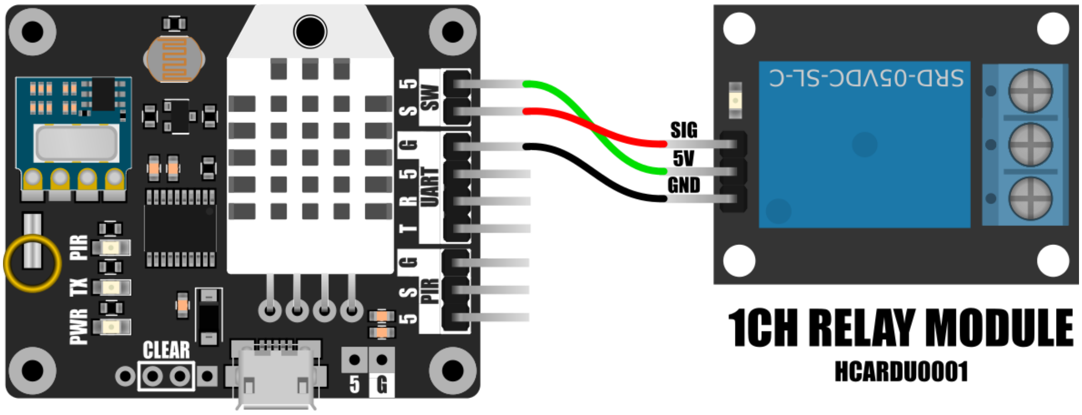
The sensor module also includes a built-in light dependent photoresistor (LDR) which gives the module the ability to measure ambient light levels.



By default the module is configured to take analogue measurements from the LDR sensor every 60 seconds. These measurements are then transmitted to all other SmartRFy devices within range and are also output to the modules serial interface. This 60 second sample rate can be changed via the serial interface using the LDR transmit rate (LTT) commands and can be set from anywhere between 10 and 65535 seconds. Setting the transmit rate to 0 will disable reading and transmitting of the sensor.

Auxiliary switch output

The sensor module includes an auxiliary digital output pin which can be used to automatically drive external devices based on the current state of one of the module's sensors. It consists of an open-collector output with a 10K pullup resistor and is capable of sinking up to 100mA of current.

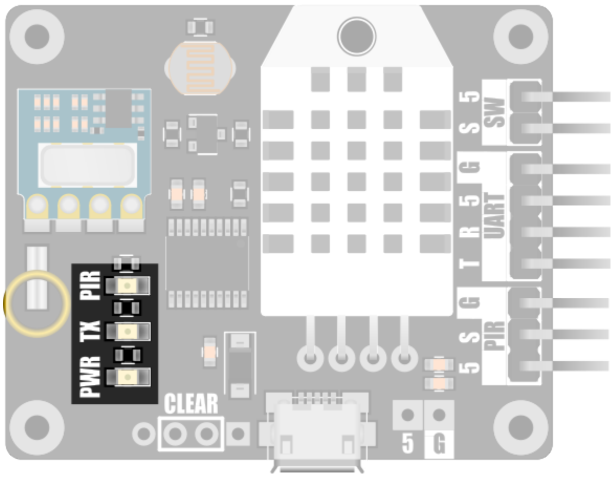


The aux output can be configured via the modules serial interface to change state based on any of the module's sensors. For example, it can be configured to pull high when the temperature sensed by the temperature sensor exceeds a set level and pull low when the temperature drops below a different level.

The aux output also supports a timer mode function. When timer mode is enabled the aux output will remain high for a user specified amount of seconds (1 to 65535 seconds) whenever it is triggered. For example, the aux output can be configured to turn on a flood light whenever a connected PIR is triggered for a set amount of time.

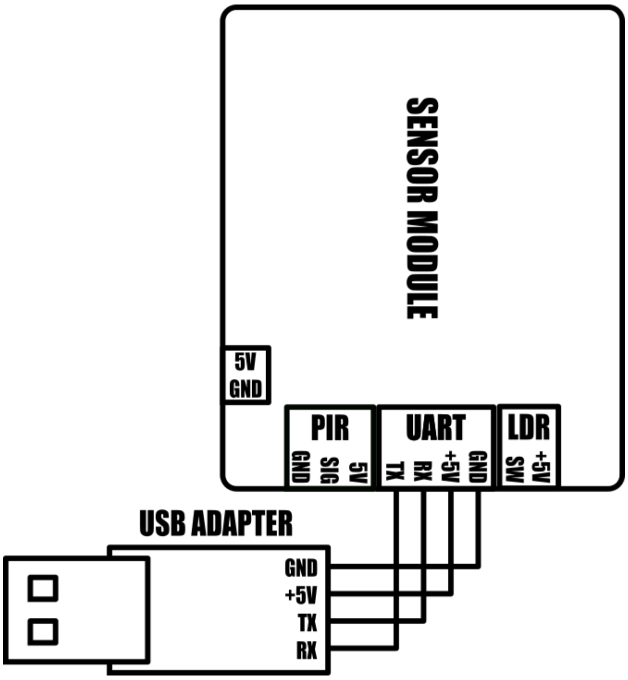
LEDs

The sensor module includes a set of 3 LEDs which indicate the current state of the module.



PWR LED (green)	Indicates the module is currently powered.
TX LED (red)	Indicates the module is currently transmitting.
PIR LED (amber)	Indicates a connected PIR sensor is currently triggered (S = high).

Serial UART



Command interface feature

The module includes a full duplex serial UART interface which can be used for issuing local commands and settings directly to the module (local commands), or transmitting commands to other remote SmartRFy devices (remote commands).

By default the serial interface will output any sensor data from any of the built-in sensors, or the optionally connected PIR sensor. The type of information that the serial interface outputs can be controlled using the verbose command (VBM).

Wireless serial port feature

Additionally, the serial interface can also act as a passive wireless serial port for receiving non-SmartRFy related data from other devices connected to remote SmartRFy modules.

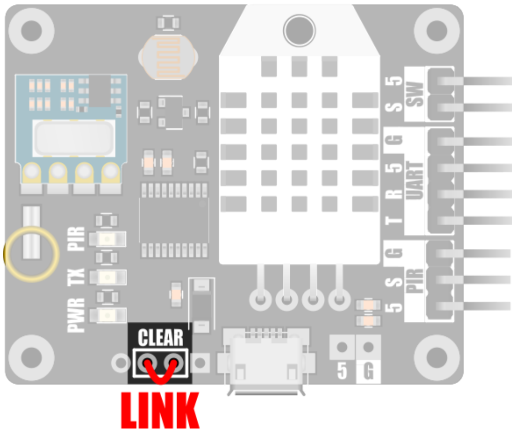
Note that when receiving passive serial data from a remote SmartRFy device, the modules receive serial buffer is limited to 25 bytes and the maximum data transfer rate from a remote device is limited by the RF transfer rate and not the actual serial port connection speed.

Default configuration settings

By default the serial interface is configured to 9600 baud, 8 data bits, no parity, and 1 stop bit. See the serial interface control section for more information on setting the baud rate, verbose mode, and issuing commands.

Note: Connecting the module to a PC via its serial port will require an additional USB to serial adapter (see Hobby Components items HCARDU0011 or HCMODU0076).

Clear (Factory Reset)



The module is capable of storing a number of user configuration settings in its non-volatile memory. This allows the module to retain these settings even after power is removed. The two pads marked ‘CLEAR’ provide a means of restoring the module back to its original factory default settings. Should you wish to reset the module, connect power to the module whilst shorting the two pads together.

Alternatively the module can be restored back to factory default settings by issuing the factory reset (FTRE) command via the serial interface – see the SmartRFy commands section for more information.

SmartRFy commands

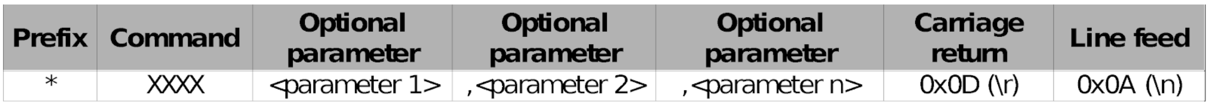
Command formats

SmartRFy modules are capable of responding to an array of text based commands sent to the module via its serial interface. These commands can be either module configuration settings, commands to control features of the locally connected module, or to be transmitted on to a remote module via the RF interface.

Command format requirements

- In all cases the command must always be terminated with a carriage return (ASCII code 13) and a line feed (ASCII code 10).
- A command sent to a module via its serial port can either be processed by the module itself (local module), or be transmitted to another module within range (remote module) – assuming the local module has transmit capability. To tell the local module if the command is intended for itself, or to be transmitted to a remote module, the command must be prefixed with either an asterisk (*), for a local command, or a greater than symbol (>) for a remote module. For example *RLY=1 will be processed by the local module only, >RLY=1 will be transmitted to any module within range.
- Commands must always be in uppercase and no part of the command should include white space characters, including spaces.
- Some commands require more than one parameter. In this case parameters should be separated by commas (,) with no spaces.
- Commands are always 4 characters in length. Commands that require one or more parameters will have an equals sign (=) as the last character (e.g. XXX=), and commands that are querying information will have a question mark (?) as the last character (e.g. XXX?).
- The device will respond to a successful local command with the response ‘OK’. An unsuccessful command will respond with an ‘ERROR’. There is no response by the local device when issuing remote commands. All responses are terminated with a carriage return and line feed.
- No serial data sent to the local device, command or otherwise, may exceed 25 characters in length (including carriage return / linefeed termination).

Local command format



Prefix	Command	Optional parameter	Optional parameter	Optional parameter	Carriage return	Line feed
*	XXXX	<parameter 1>	,<parameter 2>	,<parameter n>	0x0D (\r)	0x0A (\n)

Example: *Setting and confirming the zone of the local device*

*ZON=1<CR><LF>

Sets the zone of the connected module to 1.

Response:

OK<CR><LF>

*ZON?<CR><LF>

Requests the current zone of the local device.

Response:

1<CR><LF>

SmartRFy generic commands – quick lookup table

Command	Description	Parameter(s)	Response
FWV?	Get firmware version	None	Vx.x (C) HobbyComponents.com
ZON?	Get zone number (default: 0)	None	Zone number (0 to 255)
ADD?	Get address (default: 0)	None	Address (0 to 255)
STO=Time	Sets the timeout time for the serial interface (default: 2000ms)	Time: Timeout time in milliseconds (0 to 65535)	OK
BUR=Baud	Sets the BAUD rate for the serial interface	Baud: 0 = 1200 BAUD Baud: 1 = 2400 BAUD Baud: 2 = 4800 BAUD Baud: 3 = 9600 BAUD (default) Baud: 4 = 19200 BAUD Baud: 5 = 38400 BAUD Baud: 6 = 57600 BAUD Baud: 7 = 115200 BAUD	OK
TRS=Resends	Sets the number of times any Tx data will be retransmitted	Resends: Number of resends (0 to 5)	OK
VBM=Mode	Sets the verbose mode which specifies what received data is output to the serial interface (default: 0)	Mode: 0 = Both passive serial & commands from all zones and addresses Mode: 1 = Only passive serial data from all zones and addresses Mode: 2 = Only control from all zones and addresses Mode: 3 = Both passive serial & commands with matching zone and address Mode: 4 = Only passive serial data with matching zone and address Mode: 5 = Only commands with matching Zone and address	OK
ZON=Zone	Sets the devices zone number	Zone: Zone number (0 to 255)	OK
ADD=Address	Sets the devices address	Address: Devices address (0 to 255)	OK
TRP=State	Turns on or off repeater mode (default: off)	State: Repeater state (0 = off, 1 = on) Note: Only available on modules with both RF Tx & Rx capability	OK
FTRE	Performs a factory reset – all settings a restored to their default values	None	OK

Sensor module specific commands – quick lookup table

Command	Description	Parameter(s)	Response
HTT=Time	Sets the transmit time interval for DHT22 humidity measurements (default: 60)	Time: Transmit interval time in seconds (0 to 65535)	OK
HTN=Name	Changes the command name used when transmitting humidity measurements (default: HUM)	Name: Three letter name of the command	OK
HUM?	Gets the current humidity and both transmits and outputs the result to the serial port	None	Returns the current humidity in RH%
TTT=Time	Sets the transmit time interval for DHT22 temperature measurements (default: 60)	Time: Transmit interval time in seconds (0 to 65535)	OK
TTN=Name	Changes the command name used when transmitting temperature measurements (default: TMP)	Name: Three letter name of the command	OK
TMP?	Gets the current temperature and both transmits and outputs the result to the serial port	None	Returns the current temperature in oC
LTT=Time	Sets the transmit time interval for LDR measurements (default: 60)	Time: Transmit interval time in seconds (0 to 65535)	OK
LTN=Name	Changes the command name used when transmitting light measurements (default: LDR)	Name: Three letter name of the command	OK
LDR?	Gets the current light level from the LDR sensor and both transmits and outputs the result to the serial port	None	Returns the current light level (0 to 255)
PTN=Name	Changes the command name used when transmitting the PIR state (default: PIR)	Name: Three letter name of the command	OK
PIR?	Gets the current state of the PIR input and both transmits and outputs the result to the serial port	None	Returns the current digital state of the PIR input pin
SW0=State	Sets the state of digital output pin (SW0)	State: Floating point value which determines the state of the output depending on STH, STL, SMH, & SML settings. Default: 0 = output low, 1 = output high	OK
SWN=Name	Sets an alternative three letter command name the digital output pin (SW0) will respond to (default SW0=)	Name: Three letter name of the alternate command	OK
SWO=Time	Sets the amount of time the digital output pin (SW0) will stay high when triggered. The pin will automatically go low when time has elapsed	Time: Output pin high time in seconds (0 to 65525). Default = 0 (disabled).	OK
STH=Threshold	Sets the threshold value required to pull the output pin SW0 high (default: 1)	Threshold: A decimal number which sets the level required to trigger the output pin	OK

STL=Threshold	Sets the threshold value required to pull the output pin SW0 low (default: 0)	Threshold: A decimal number which sets the level required to de-energise the relay	OK
SMH=Mode	Sets the condition under which the output pin SW0 will go high (default: =)	Mode: > The value passed by the SW0= command must be greater than the Threshold level set by the STH= command Mode: = The value passed by the SW0= command must equal the Threshold level set by the STH= command Mode: < The value passed by the SW0= command must be less than the Threshold level set by the STH= command	OK
SML=Mode	Sets the condition under which the output pin SW0 will go low (default: =)	Mode: > The value passed by the SW0= command must be greater than the Threshold level set by the STH= command Mode: = The value passed by the SW0= command must equal the Threshold level set by the STH= command Mode: < The value passed by the SW0= command must be less than the Threshold level set by the STH= command	OK

Generic SmartRFy commands

Commands listed below are common to all SmartRFy modules, including the sensor module. Note that all commands must be proceeded with a carriage return and line feed, but for clarity purposes this is omitted from the examples in this section.

Firmware version (FWV?)

Gets the firmware version of the module.

Example: *FWV?

Returns: Vx.x

(C) HobbyComponents.com

Where Vx.x is the version number

Get the modules zone (ZON?)

Gets the modules current zone number.

Example: *ZON?

Returns: The current zone number as a decimal value (0 to 255)

Get the modules address (ADD?)

Gets the modules current address.

Example: *ADD?

Returns: The current address as a decimal value (0 to 255)

Serial serial timeout time (STO=)

Sets the timeout time for the serial interface in milliseconds. Maximum timeout time is 65535ms (65.535 seconds). If a complete command (including CR + LF termination) is not received within this time the command will be ignored and an 'ERROR' will be returned.

Default: 2000 (2 seconds)

Example: *STO=1000

The above example will set the serial port timeout time to 1000ms (1 second).

Returns: OK

Set the baud rate (BUR=)

Sets the communication baud rate for the serial UART interface. There are 8 (0 to 7) possible settings for the baud rate:

0 = 1200 BAUD
1 = 2400 BAUD

2 = 4800 BAUD
3 = 9600 BAUD (default)
4 = 19200 BAUD
5 = 38400 BAUD
6 = 57600 BAUD
7 = 115200 BAUD

NOTE: This command sets the serial communication speed between the connected device and the SmartRFy module – it does not set the wireless RF communication speed between modules.

Default: 3

Example: *BUR=5

The above example will set the serial port baud rate to 38400 baud.

Returns: OK

Set number of transmit resends (TRS=)

Sets the number of times the module will automatically re-transmit data. Increasing the number of times data is retransmitted will reduce the chances of the data not being received due to interference but will also increase transmission times and network traffic. The number of resends can be set from 0 (no resends) to 5.

Default: 2

Example: *TRS=3

The above example will set the number of resends to 3. Therefore any data wirelessly transmitted by the module will be transmitted a total of 4 times.

Returns: OK

Set the serial verbose mode (VBM=)

Sets what type of data received by the module is automatically output to its serial port. There are 6 possible modes (0 to 5):

0 = Both passive serial and commands from all zones and addresses
1 = Only passive serial data from all zones and addresses
2 = Only control from all zones and addresses
3 = Both passive serial and commands with matching zone and address
4 = Only passive serial data with matching zone and address
5 = Only commands with matching zone and address

Default: 0

Example: *VBM=5

The above example will set the module to only output received commands that have been transmitted from a module with the same zone and address as itself.

Returns: OK

Set the modules zone number (ZON=)

Sets the modules zone number. The zone number can be between 0 and 255.

Note that zone 255 is a special zone number. Setting the module to this zone number means that it will treat received data from any zone as if it were in the same zone as itself. Therefore this zone number can be used when you wish the module to respond to commands from modules in multiple zones. Additionally, when transmitting data with a zone of 255 all modules within range will respond to the data as if it is in the same zone as itself. Therefore this zone number can also be used when you require the module to control multiple remote modules in different zones.

Default: 0

Example:	*ZON=2

The above example will set the modules zone number to 2.

Returns: OK

Set the modules address (ADD=)

Sets the module address. The address can be between 0 and 255.

Note that address 255 is a special address. Setting the module to this address means that it will treat received data from any address as if it were the same address as itself. Therefore, this address can be used when you wish the module to respond to commands from modules with different addresses. Additionally, when transmitting data with an address of 255 all modules within range will respond to the data as if it has the same address as itself. Therefore this address can also be used when you require the module to control multiple remote modules with different addresses.

Default:		0

Example:	*ADD=5

The above example will set the modules address to 5.

Returns: OK

Turn on/off repeater mode (TRP=)

Turns repeater mode on or off. When turned on the module will retransmit any data it receives regardless of its zone and address. This mode can be used when a remote module is outside the range of a module that needs to communicate with it but the repeating module is within range of both modules. Setting the repeater mode to 1 will turn on repeater mode and setting it to 0 will turn it off.

Note: Turning on the repeater mode will double network traffic for any modules within range.

Default: 0 (off)

Example: *TRP=1

The above example will turn on repeater mode.

Returns: OK

Factory reset (FTRE)

Performs a factory reset of the module. All module settings will be restored to their factory defaults.

Example: *FTRE

The above example will restore all settings to their defaults.

Returns: OK

SmartRFy sensor module commands

Commands below are specific to the SmartRFy sensor module and can be used to configure parameters relating to the hardware features of this module.

Get Humidity (HUM?)

Gets the current relative humidity in percent from the DHT22 sensor and both transmits and outputs the result to the serial port.

Example: *HUM?

The module will now simultaneously both transmit and output the current humidity to the serial port.

Returns: HUM=xx.x *Where xx.x is the current relative humidity in %*

Humidity transmit time (HTT=)

Sets the time interval in seconds for automatically reading and transmitting the current humidity from the connected DHT22 sensor. The transmit time can be set anywhere between 10 and 65535 seconds. Setting the transmit time to 0 will disable automatic reading and transmitting of humidity measurements.

Default: 60 (1 minute)

Example: *HTT=600

The module will now automatically read and transmit the current humidity every 600 seconds (10 minutes).

Returns: OK

Humidity transmit name (HTN=)

Sets the 3 letter command name the module will use when transmitting humidity readings.

Default: HUM

Example: *HTN=ABC

The module will now transmit humidity measurements in the format ABC=xx.x instead of the default HUM=xx.x

Returns: OK

Get Temperature (TMP?)

Gets the current temperature in °C from the DHT22 sensor and both transmits and outputs the result to the serial port.

Example: *TMP?

The module will now simultaneously both transmit and output the current temperature to the serial port.

Returns: TMP=xx.x Where xx.x is the current temperature in °C (-40 to 80)

Temperature transmit time (TTT=)

Sets the time interval in seconds for automatically reading and transmitting the current temperature from the connected DHT22 sensor. The transmit time can be set anywhere between 10 and 65535 seconds. Setting the transmit time to 0 will disable automatic reading and transmitting of temperature measurements.

Default: 60 (1 minute)

Example: *TTT=600

The module will now automatically read and transmit the current temperature every 600 seconds (10 minutes).

Returns: OK

Temperature transmit name (TTN=)

Sets the 3 letter command name the module will use when transmitting temperature readings.

Default: TMP

Example: *TTN=ABC

The module will now transmit temperature measurements in the format ABC=xxx.x instead of the default TMP=xxx.x

Returns: OK

LDR transmit time (LTT=)

Sets the time interval in seconds for automatically reading and transmitting the current light level from the connected light dependent resistor (LDR). The transmit time can be set anywhere between 10 and 65535 seconds. Setting the transmit time to 0 will disable automatic reading and transmitting of LDR measurements.

Default: 60 (1 minute)

Example: *LTT=600

The module will now automatically read and transmit the current light level every 600 seconds (10 minutes).

Returns: OK

LDR transmit name (LTN=)

Sets the 3 letter command name the module will use when transmitting LDR readings.

Default: LDR

Example: *LTN=ABC

The module will now transmit LDR measurements in the format ABC=xxx instead of the default LDR=xxx

Returns: OK

PIR transmit name (PTN=)

Sets the 3 letter command name the module will use when transmitting the PIR state.

Default: PIR

Example: *PTN=ABC

The module will now transmit the PIR state in the format ABC=x instead of the default PIR=x

Returns: OK

Set the state of the digital output pin (SW0=)

Sets the state of the module's auxiliary digital output pin. The command can be used to pull high (5V) or pull low (0V) the on-board digital output pin.

Default: SW0=0 *(off)* / SW0=1 *(on)*

Example: *SW0=1

The above example will pull the output pin high

Returns: OK

Set an alternative name for the digital output pin (SWN=)

Sets an alternative 3 letter command name which the auxiliary digital output pin will respond to. Setting an alternative command name allows the pin to be controlled by other devices such as temperature sensors, PIR's, switches etc. (Also see pin connmands SWO, SWH, SWL, SWH, SWL, and Appendix A for more options and examples of configuring this pin).

Default: SW0

Example: *SWN=ABC

The output pin will now respond to a command in the format of ABC=xxxx.

Returns: OK

Set the output pin high time (SWO=)

Sets the amount of time in seconds that the output pin will be pulled high for. Setting the pin high time to 0 (default) will disable the pin high timer and the pin will not return low unless it receives a command to do so. Setting the pin high time to anything between 1 and 65535 will cause the pin to stay high until that amount of time (in seconds) has passed.

Default: 0 (disabled)

Example: *SWO=60

In the above example the modules' output pin will automatically return low after 60 seconds from receiving a command to set it high.

Returns: OK

Set the threshold value to set the output pin high (STH=)

Sets the threshold value which will be used to determine if the output pin should be pulled high when a SW0 command is received. The pin high threshold can be a positive or negative decimal number. The module will compare this value with the value received from an SW0 command and together with the pin high mode (SMH) determine if the pin should be pulled high. (Also see pin commands SWO, STH, STL, SMH, SML, and Appendix A for more options and examples of configuring the output pins).

Default: 1

Example: *STH=100

In the above example the output pin will be pulled high if the module receives the pin command SW0=100 (assuming SMH is set to default).

Returns: OK

Set the threshold value to set a digital pin low (STL=)

Sets the threshold value which will be used to determine if the output pin should be pulled low when a SW0 command is received. The pin high threshold can be a positive or negative decimal number. The module will compare this value with the value received from an SW0 command and together with the pin low mode (SML) determine if the pin should be pulled low. (Also see pin commands SWO, STH, STL, SMH, SML, and Appendix A for more options and examples of configuring the output pins).

Default: 0

Example: *STL=-50

In the above example pin 0 will be pulled low if the module receives the pin command SW0=-50 (assuming SML is set to default).

Returns: OK

Set the test condition for setting the output pin high (SMH=)

Sets the threshold condition used to determine if the output pin should be pulled high. When a pin command (SW0) is received its parameter value is tested against the pin threshold high (STH) value, and this threshold condition determines if the pin should be pulled high. The SMH threshold condition can be set to one of three values:

SMH=< The output pin is pulled high if the value received by the SW0 command is *less than* the pin threshold high (STH) parameter.

SMH== The output pin is pulled high if the value received by the SW0 command is *equal to* the pin threshold high (STH) parameter.

SMH=> The output pin is pulled high if the value received by the SW0 command is *greater than* the pin threshold high (STH) parameter.

Default: =

Example: *SMH=>

In the above example the output pin will be pulled high if the module receives the pin command SW0 with a value greater than the current pin threshold high (STH) level.

Returns: OK

Set the test condition for setting the output pin low (SML=)

Sets the threshold condition used to determine if the output pin should be pulled low. When a pin command (SW0) is received its parameter value is tested against the appropriate pin threshold low (TxL) value, and this threshold condition determines if the pin should be pulled low. The SML threshold condition can be set to one of three values:

SML=< The output pin is pulled low if the value received by the SW0 command is *less than* the pin threshold low (STL) parameter.

SML== The output pin is pulled low if the value received by the SW0 command is *equal to* the pin threshold low (STL) parameter.

SML=> The output pin is pulled low if the value received by the SW0 command is *greater than* the pin threshold low (STL) parameter.

Default: =

Example: *SML=<

In the above example the output pin will be pulled low if the module receives the pin command SW0 with a value less than the current pin threshold low (STL) level.

Returns: OK

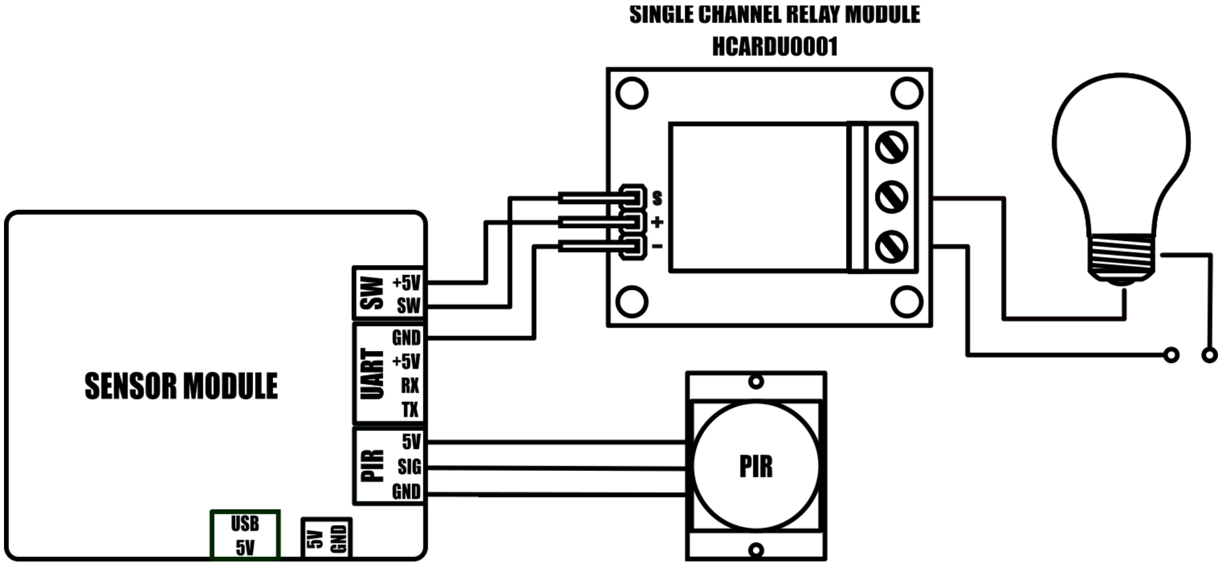
Appendix A: Example auxiliary output pin modes

The state of the auxiliary digital output pin can be configured to toggle based on the current state of any of the sensor modules' sensors, including the externally connected PIR. This output pin can then be used to drive external devices such as LED’s, relay modules, etc.

The following section shows example applications which demonstrate how the output pin can be reconfigured.

Default trigger mode

In this example the output pin will be configured to mimic the state of a PIR sensor connected to the PIR header of the sensor module and will be used to control a light bulb (via a relay module).



Change the following settings via the modules serial interface (see serial uart section for information about interfacing to the modules serial interface):

*SWN=PIR	Output pin will now listen to the PIR sensor
*SWO=0	Turn off timer mode (default)
*STH=1	Threshold high = 1 (default)
*STL=0	Threshold low = 0 (default)
*SMH==	Threshold high mode = trigger value must be equal to STH (default)
*SML==	Threshold low mode = trigger value must be equal to STL (default)

Timer mode

In timer mode the output pin can be configured to remain asserted for a set amount of time in seconds whenever it has been triggered. This example builds on the previous example to allow the light bulb to remain illuminated for 1 minute after the PIR has been triggered.

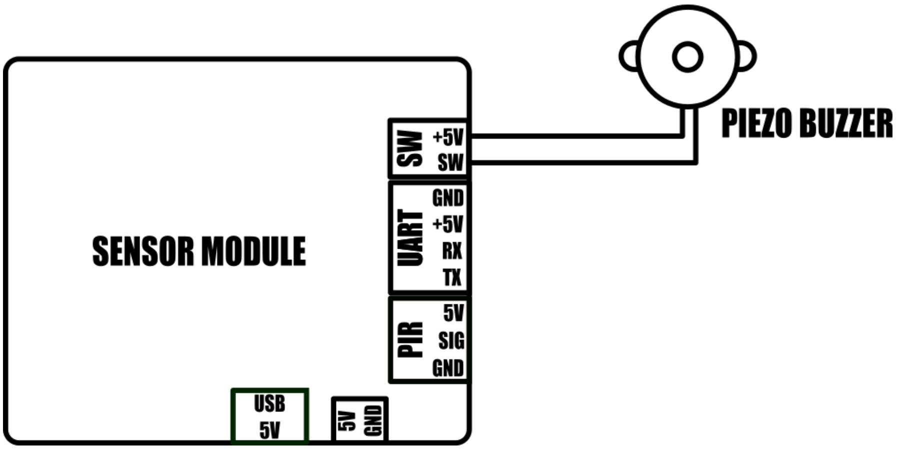
After following the default trigger mode example, change the following setting via the modules serial interface (see serial uart section for information about interfacing to the modules serial interface):

*SWO=60	Set output pin on time to 60 seconds

The output pin will now remain high for 60 seconds after being triggered by the PIR sensor.

Threshold mode

In threshold mode the output pin can be programmed to automatically pull high or low when a value from one of the sensor modules sensors goes above or below a predefined pair of thresholds.

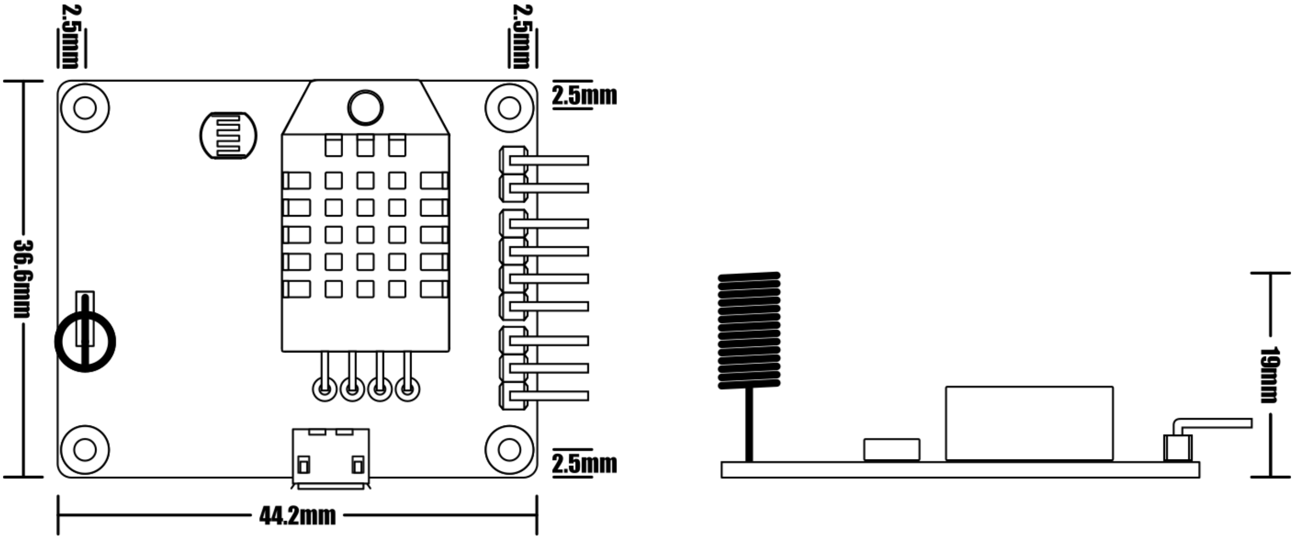


In this example the output pin will be configured to automatically sound an alarm (by pulling the output pin low) whenever the temperature sensed by the DHT22 sensor goes above 25ºC and will turn the alarm off when it drops back below 22.5ºC.

Change the following settings via the modules serial interface (see serial uart section for information about interfacing to the modules serial interface):

*SWN=TMP	Output pin will now listen to the temperature sensor
*SWO=0	Turn off timer mode (default)
*STH=22.5	Threshold high = 22.5 (default)
*STL=25	Threshold low = 25 (default)
*SMH=<	Threshold high mode = trigger value must be less than STH (default)
*SML=>	Threshold low mode = trigger value must be greater than STL (default)

Dimensions


2.5mm
2.5mm
2.5mm
36.6mm
2.5mm
44.2mm
19mm