HASCOM International Pty Ltd

# Serial IO MKII

## Installation, Operation and Maintenance Manuals

| Revision | Date | Author |
|---|---|---|
| A | 14th February 2012 | Chris King, HASCOM International Pty Ltd |
| B | 24th March 2012 | Chris King, HASCOM International Pty Ltd |
| C | 11th June 2012 | Chris King, HASCOM International Pty Ltd |
| D | 8th July 2012 | Chris King, HASCOM International Pty Ltd |
| E | 3rd March 2014 | Chris King, HASCOM International Pty Ltd |
| F | 18th September 2014 | Chris King, HASCOM International Pty Ltd |
| | | |

HASCOM International
4 Lacrosse Rise
Sorrento
Western Australia, 6020

# Contents

# Illustration Index

# 1 Installation Manual

## *1.1 Introduction*

This manual is for installation staff and is designed to assist with the steps required to physically install the HASCOM Serial IO MKII unit.

## *1.2 Overview*

The HASCOM Serial IO MKII unit is designed to provide a powerful, yet cost effective field component in a railway remote control, monitoring and telemetry system. The unit provides support for many telemetry protocols used in railway systems and supports several varied carriers. It supports protocol conversion between any supported protocol & carrier combination as well as having 32 digital outputs and 32 digital inputs. Optionally it can support various analogue inputs and audio playback as well.

## *1.3 Installation*

### 1.3.1 Serial IO MKII unit

The HASCOM Serial IO MKII unit has a 19 inch, 1 rack unit (RU) form factor, making it ideal to mount in a communications cabinet, etc.


*Illustration 1: Front Panel*

The front panel has access to the LED indications as well as the SD card.


*Illustration 2: Rear panel*

The rear panel has access to all the connectors required during installation.

### 1.3.2 Power supply

The power connection is a removable terminal block type which accepts wires from 2 to 24 AWG, ie up to 4 $mm^2$ cross sectional area maximum.


*Illustration 3: Power connector*

The positive connection is on the left, the negative connection is on the right when viewed from the rear.

The HASCOM Serial IO MKII unit supply voltage should be in the range 8 to 28 volts DC and capable of supplying up to 12 watts (approximately 1 A at 12v) of power.

## 1.3.3 Modem port

The modem port is an RJ45 connector. Pin 1 is on the top left when viewed from the rear. The Tx and Rx audio signals are transformer isolated.


*Illustration 4: Modem connector*

The connector wiring attempts to be similar to the RJ48C standard.

| Pin | Signal | Common CAT5 wiring colour |
|---|---|---|
| 1 | Rx audio + | Orange/White |
| 2 | Rx audio - | White/Orange |
| 3 | Digital input, eg squelch input | White/Green |
| 4 | Tx audio + | Blue/White |
| 5 | Tx audio - | White/Blue |
| 6 | Digital output, eg press to talk output | Green/White |
| 7 | 0v / Ground / screen | White/Brown |
| 8 | 0v / Ground / screen | Brown/White |

The audio signal levels are adjusted using the configuration application. See section 2.5.4.19.

The digital output is an open collector transistor which sinks current to 0v / Ground when active. It has an on resistance of approximately 50 ohms and can sink up to 100 mA

The digital input has a 47 k ohm resistor to 0v / Ground and must be pulled up to at least 4v to activate the input.

### 1.3.4 Serial ports

The HASCOM Serial IO MKII unit has 3 serial RS232 data connections. Each port is transformer isolated. COM1 and COM3 are DTE connections with a 9 way D-type male connector, pin 1 being on the top left when viewed from the rear. COM2 is a DCE connection with a 9 way D-type female connector, pin 1 being on the top right when viewed from the rear.


*Illustration 5: Serial connectors*

COM1 and COM3, DTE

| Pin | Signal |
|---|---|
| 1 | n/c |
| 2 | Rx data (input) |
| 3 | Tx Data (output) |
| 4 | DTR (output) |
| 5 | 0v / Ground |
| 6 | n/c |
| 7 | RTS (output) |
| 8 | CTS (input) |
| 9 | n/c |

COM2, DCE

| Pin | Signal |
|---|---|
| 1 | n/c |
| 2 | Tx Data (output) |
| 3 | Rx data (input) |
| 4 | n/c |
| 5 | 0v / Ground |
| 6 | DTR (output) |
| 7 | CTS (input) |
| 8 | RTS (output) |
| 9 | n/c |

### 1.3.5 Ethernet

The HASCOM Serial IO MKII unit has one Ethernet connection capable of 10/100 Mb/s full duplex data rate with auto-crossover support, etc. The connector is an industry standard RJ45 type.

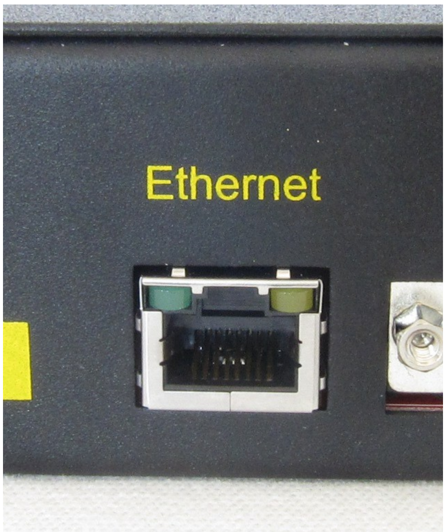



*Illustration 6: Ethernet connector*

The Ethernet / RJ45 standard is maintained by the IEEE. For more information see http://www.ieee802.org/.

The Green LED signifies a link is established. The yellow LED signifies a combination of transmit and/or receive activity.

### 1.3.6 Digital Input / Output ports

The standard HASCOM Serial IO MKII unit has 32 digital inputs and 32 digital outputs. The connectors are IDC 40 way male headers, with pin 1 at the top right when viewed from the rear.


*Illustration 7: Digital IO connectors*

The digital outputs are a relay contact closure type, so are therefore electrically isolated. They are capable of switching up to 1A at up to 110 Vdc or 125 Vac.

The digital inputs are of opto-isolator type. The opto-isolators LED input has a 2.2 k ohm series resistor on the internal PCB to limit the current. The LED current should be between 5 to 30 mA. Therefore the inputs can be directly driven with 12 to 60 volts.

If other voltages are required, the internal series resistor networks can be changed. On the digital IO board they are in a DIP 16 package mounted in an IC socket. RN1 is for inputs 1 to 8 or 17 to 24 while RN2 is for inputs 9 to 16 or 25 to 32.

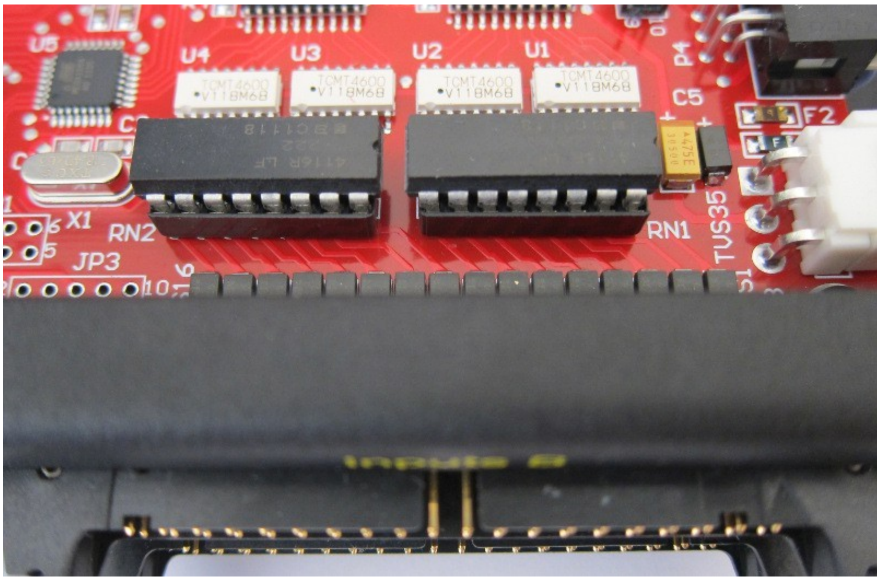
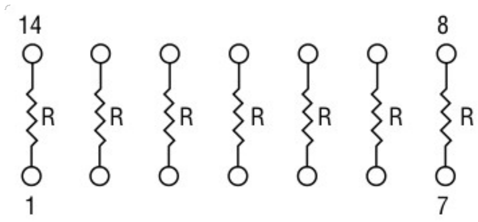



*Illustration 8: Digital input series resistors*

The resistor network must be of the type that has 8 isolated resistors in each 14 pin DIP package.




If external series resistors are being used, the current through the opto-isolators LED must be kept within the 5 to 30 mA range. The internal series resistor can be replaced with a smaller value if needed.

The pin configuration is as follows:

**Inputs**

| **Pin** | **Description** | **Pin** | **Description** |
|---|---|---|---|
| 1 | Input 1+ | 2 | Input 1- |
| 3 | Input 2+ | 4 | Input 2- |
| 5 | Input 3+ | 6 | Input 3- |
| 7 | Input 4+ | 8 | Input 4- |
| 9 | Input 5+ | 10 | Input 5- |
| 11 | Input 6+ | 12 | Input 6- |
| 13 | Input 7+ | 14 | Input 7- |
| 15 | Input 8+ | 16 | Input 8- |
| 17 | Input 9+ | 18 | Input 9- |
| 19 | Input 10+ | 20 | Input 10- |
| 21 | Input 11+ | 22 | Input 11- |
| 23 | Input 12+ | 24 | Input 12- |
| 25 | Input 13+ | 26 | Input 13- |
| 27 | Input 14+ | 28 | Input 14- |
| 29 | Input 15+ | 30 | Input 15- |
| 31 | Input 16+ | 32 | Input 16- |
| 33 | N/C | 34 | N/C |
| 35 | N/C | 36 | JP3 pin 9 |
| 37 | JP3 pin 7 | 38 | JP3 pin 5 |
| 39 | JP3 pin 3 | 40 | JP3 pin 1 |

Note: While the inputs are marked '+' and '-', they are actually bi-directional and will work either way around.

**Outputs**

| Pin | Description | Pin | Description |
|---|---|---|---|
| 1 | Output 1+ | 2 | Output 1- |
| 3 | Output 2+ | 4 | Output 2- |
| 5 | Output 3+ | 6 | Output 3- |
| 7 | Output 4+ | 8 | Output 4- |
| 9 | Output 5+ | 10 | Output 5- |
| 11 | Output 6+ | 12 | Output 6- |
| 13 | Output 7+ | 14 | Output 7- |
| 15 | Output 8+ | 16 | Output 8- |
| 17 | Output 9+ | 18 | Output 9- |
| 19 | Output 10+ | 20 | Output 10- |
| 21 | Output 11+ | 22 | Output 11- |
| 23 | Output 12+ | 24 | Output 12- |
| 25 | Output 13+ | 26 | Output 13- |
| 27 | Output 14+ | 28 | Output 14- |
| 29 | Output 15+ | 30 | Output 15- |
| 31 | Output 16+ | 32 | Output 16- |
| 33 | N/C | 34 | N/C |
| 35 | N/C | 36 | N/C |
| 37 | N/C | 38 | N/C |
| 39 | N/C | 40 | N/C |

Note: While the outputs are marked '+' and '-', they are relay “contact closure” outputs and will work either way around.

### 1.3.6 Analogue Input / Output ports

The HASCOM Serial IO MKII unit can optionally be supplied with analogue inputs cards. The connectors are IDC 26 way male headers, and are physically smaller than the usual 40 pin connector, with pin 1 at the top right when viewed from the rear. The upper row (Inputs/Outputs B) below are analogue ports.

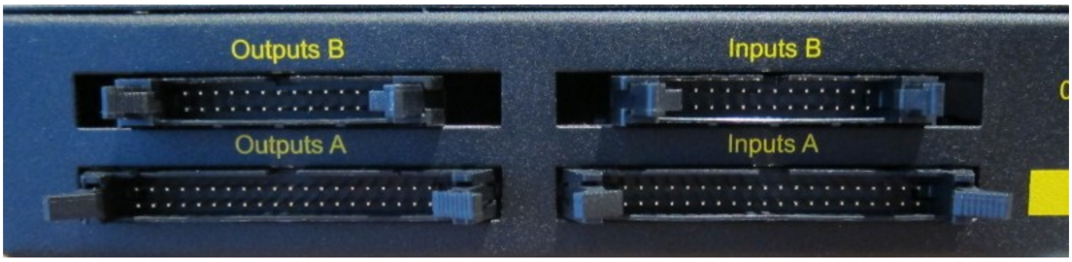


*Illustration 9: Analogue IO connectors*

The pin configuration is as follows:

**Inputs**

| Pin | Description | Pin | Description |
|---|---|---|---|
| 1 | Ch 1 current input | 2 | Ch 1 0v / return |
| 3 | Ch 1 voltage input | 4 | Ch 1 0v / return |
| 5 | N/C | 6 | N/C |
| 7 | Ch 2 current input | 8 | Ch 2 0v / return |
| 9 | Ch 2 voltage input | 10 | Ch 2 0v / return |
| 11 | N/C | 12 | N/C |
| 13 | Ch 3 current input | 14 | Ch 3 0v / return |
| 15 | Ch 3 voltage input | 16 | Ch 3 0v / return |
| 17 | N/C | 18 | N/C |
| 19 | Ch 4 current input | 20 | Ch 4 0v / return |
| 21 | Ch 4 voltage input | 22 | Ch 4 0v / return |
| 23 | N/C | 24 | N/C |
| 25 | Ground (non-isolated) | 26 | Ground (non-isolated) |

Input circuitry

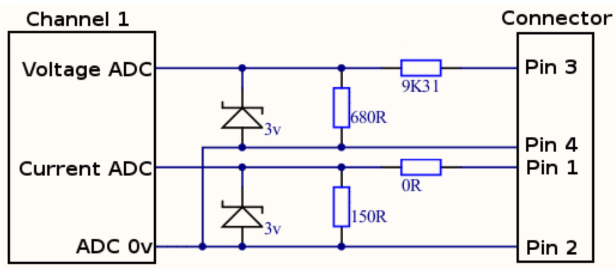


*Illustration 10: Analogue input circuit*

## Outputs

| Pin | Description | Pin | Description |
|---|---|---|---|
| 1 | Output 1, Opto emitter | 2 | Output 1, Opto collector |
| 3 | Output 2, Opto emitter | 4 | Output 2, Opto collector |
| 5 | Output 3, Opto emitter | 6 | Output 3, Opto collector |
| 7 | Output 4, Opto emitter | 8 | Output 4, Opto collector |
| 9 | N/C | 10 | N/C |
| 11 | Output 5, Relay common | 12 | Output 5, Relay NO |
| 13 | Output 5, Relay common | 14 | Output 5, Relay NC |
| 15 | Output 6, Relay common | 16 | Output 6, Relay NO |
| 17 | Output 6, Relay common | 18 | Output 6, Relay NC |
| 19 | Output 7, Relay common | 20 | Output 7, Relay NO |
| 21 | Output 7, Relay common | 22 | Output 7, Relay NC |
| 23 | Output 8, Relay common | 24 | Output 8, Relay NO |
| 25 | Output 8, Relay common | 26 | Output 8, Relay NC |

Output circuitry

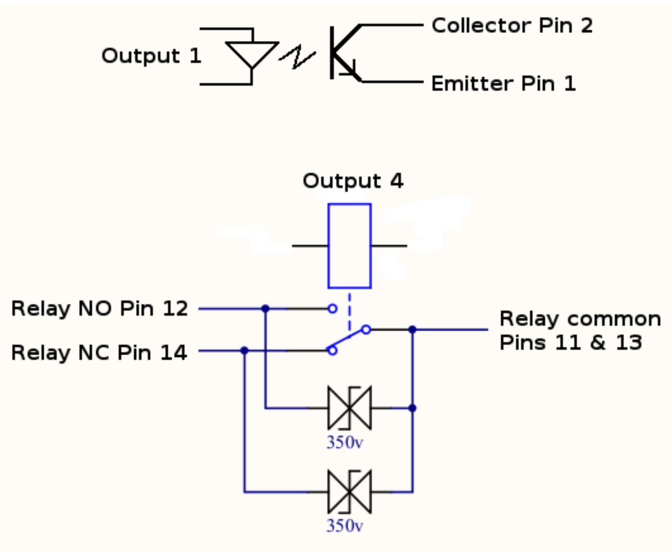



*Illustration 11: Analogue output circuit*

*Note: Opto-isolator collector-emitter is capable of handling 5mA and 70v maximum.*

## 1.3.7 Voice/Audio output ports

The HASCOM Serial IO MKII unit can optionally be supplied with voice/audio output cards. This card design is based on the analogue card detailed above, however the functionality of the output connector has been replaced with a stand alone trigger and audio playback circuit.

The input connector is an IDC 26 way male header, the output connector is an IDC 20 way male header, and they are physically smaller than the usual 40 pin connector, with pin 1 at the top right when viewed from the rear. The upper row (Inputs/Outputs B) below are audio ports. There is also a micro SD card slot next to the output connector.

*Illustration 12: Audio IO connectors*

The input pin configuration (which is exactly the same as the Analogue cards inputs) is as follows:

**Inputs**

| Pin | Description | Pin | Description |
|---|---|---|---|
| 1 | Ch 1 current input | 2 | Ch 1 0v / return |
| 3 | Ch 1 voltage input | 4 | Ch 1 0v / return |
| 5 | N/C | 6 | N/C |
| 7 | Ch 2 current input | 8 | Ch 2 0v / return |
| 9 | Ch 2 voltage input | 10 | Ch 2 0v / return |
| 11 | N/C | 12 | N/C |
| 13 | Ch 3 current input | 14 | Ch 3 0v / return |
| 15 | Ch 3 voltage input | 16 | Ch 3 0v / return |
| 17 | N/C | 18 | N/C |
| 19 | Ch 4 current input | 20 | Ch 4 0v / return |
| 21 | Ch 4 voltage input | 22 | Ch 4 0v / return |
| 23 | N/C | 24 | N/C |
| 25 | Ground (non-isolated) | 26 | Ground (non-isolated) |

Input circuitry

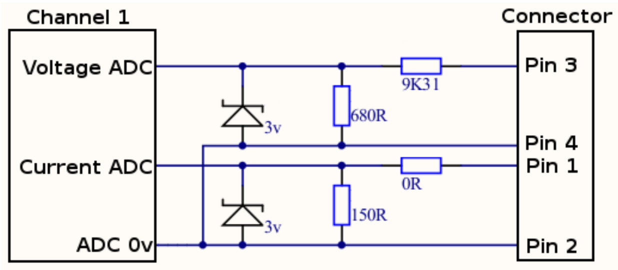



*Illustration 13: Analogue input circuit*

The output pin configuration is as follows:

**Outputs**

| Pin | Description | Pin | Description |
|---|---|---|---|
| 1 | Input 1+ | 2 | Input 1- |
| 3 | Input 2+ | 4 | Input 2- |
| 5 | Input 3+ | 6 | Input 3- |
| 7 | Input 4+ | 8 | Input 4- |
| 9 | Input 5+ | 10 | Input 5- |
| 11 | Input 6+ | 12 | Input 6- |
| 13 | Audio L+ | 14 | Audio L- |
| 15 | Audio R+ | 16 | Audio R- |
| 17 | PTT+ | 18 | PTT- |
| 19 | SQL+ | 20 | SQL- |

The input pins and SQL pins are opto-isolated inputs and operate in the same way as the standard digital input ports, see section 1.3.6.

The audio L & R pins are 600 ohm audio transfer coupled internal to the unit.

The PTT pins are a relay contact closure and operate in the same way as the standard digital output ports, see section 1.3.6.

### 1.4 Configuration

All configuration, including audio levels, is achieved with the configuration application. Please see section 2.5.

### 1.5 Part numbers

The part number for the Serial IO MKII, as marked on the rear of the unit is 025001.

A suffix might be added to describe the type of expansion boards fitted. There are currently 3 types of expansion board; 'D' for digital, 'A' for analogue & 'V' for voice. Examples of typical configurations are as follows:

| Part number | Board in lower slot | Board in upper slot |
|---|---|---|
| 025001-DD | Digital | Digital |
| 025001-DA | Digital | Analogue |
| 025001-DV | Digital | Voice/Audio |

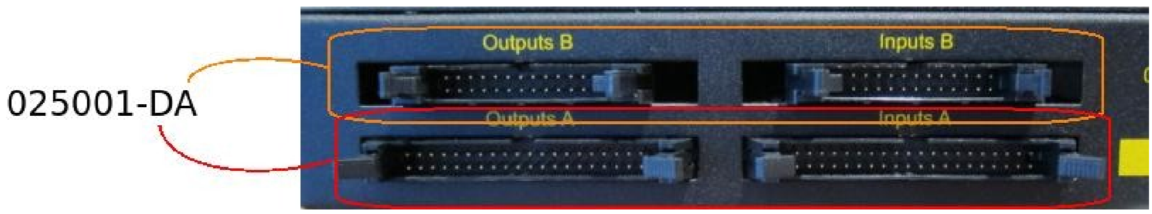



*Illustration 14: Part number suffix*

The “025001-DD” part is the standard unit configuration and this configuration is assumed if it is not specified otherwise.

# 2 Operation Manual

## 2.1 Introduction

This manual explains the operation of the Serial IO MKII and details the configuration options available.

## 2.2 Overview

The Serial IO MKII is intended to be used in railway telemetry systems. Primarily the device can convert between several different commonly used protocols enabling data communications between previously incompatible equipment. It is flexible enough to connect several different pieces of equipment in many different ways.

The standard Serial IO MKII also has 32 digital inputs and 32 digital outputs that can be monitored and controlled via any of these protocols. This digital IO can be used for general purpose control and monitoring, or it can be connected directly to track relays/circuits.

## 2.3 Description of operation

The Serial IO MKII is based on the concept of a central database of 512 bytes, each byte containing a block of 8 bits (making 4096 bits in total). Data is written to and read from this database by a combination of a protocol and a port, this combination is called a node.

This diagram shows two nodes both of which access the same bytes in the database, one reading, the other writing to each byte.

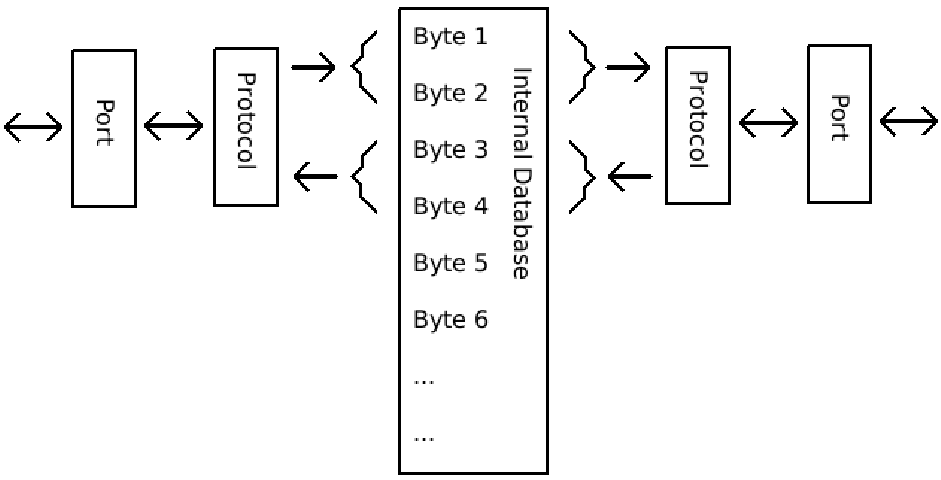



*Illustration 15: Operation 1*

The protocol in a node can be any available protocol and the port can be any available port. The Serial IO MKII supports up to 6 separate nodes simultaneously. For example the following diagram is an example, simple configuration.

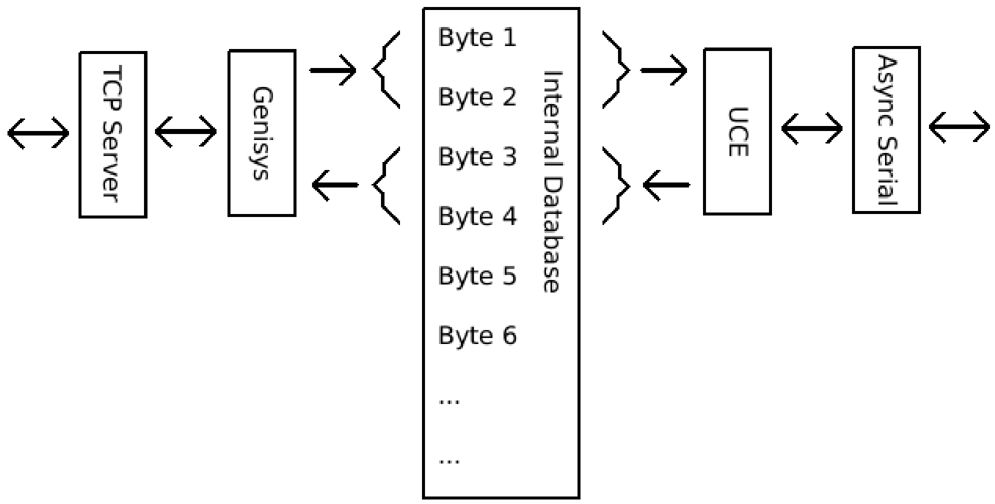



*Illustration 16: Operation 2*

The node configuration is not restricted to a matching pair, it can be any combination. For example this is a slightly more complex configuration.

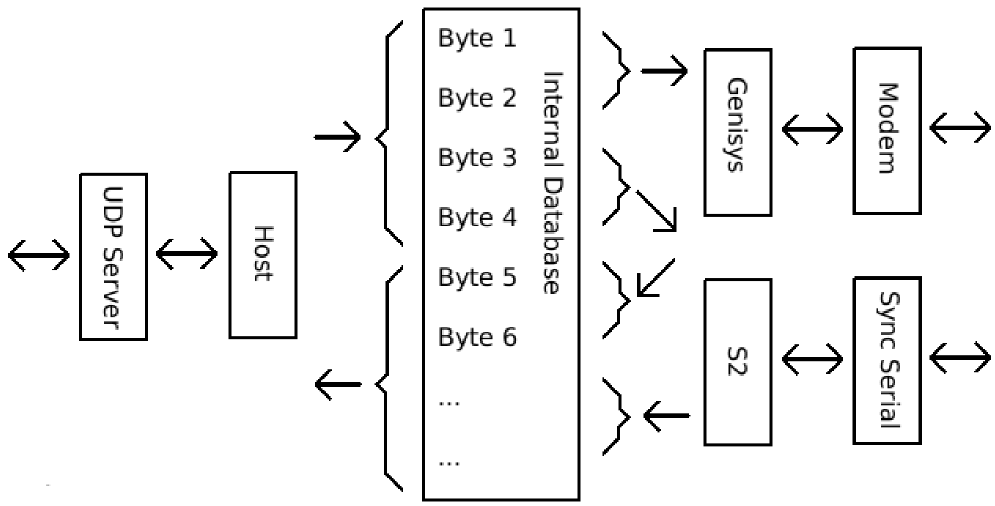



*Illustration 17: Operation 3*

Some protocols support point to multi-point operation, meaning there are slave IDs used within the protocol that are intended to direct data to different destinations depending on the slave ID. Typically in a telemetry system, each different field station will have a different ID to access it.

The Serial IO MKII supports up to 50 slave IDs within each point to multi-point protocol. For example, looking closer at a diagram of a node.

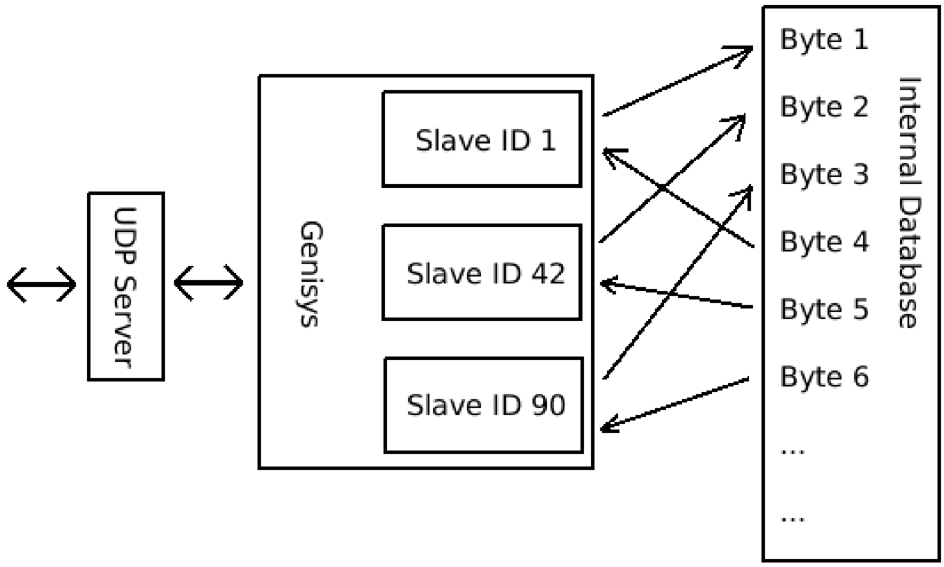



*Illustration 18: Operation 4*

An individual slave ID can write to a block of any number of bytes, at any address in the database.

Currently supported ports include

- Ethernet TCP server
- Ethernet TCP client
- Ethernet UDP server
- Ethernet UDP client
- Asynchronous serial port
- Synchronous serial port
- Modem port, supporting several modulation standards
- Internal SPI port, to communicate with internal expansion boards.

Currently supported protocols include

- Control, for monitoring and configuration of the Serial IO MKII
- IO master, to talk via the SPI port to internal expansion boards
- UCE master
- UCE slave
- S2 master
- S2 slave(s)
- Genisys master
- Genisys slave(s)
- Host master
- Host slave
- NTP client, to update the internal clock
- Analogue/Voice master
- Null

## 2.4 LED indications

There are several LED indications on the Serial IO MKII. These include the following.

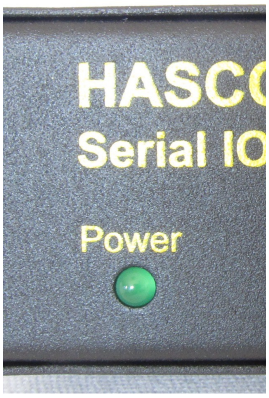



*Illustration 19: LED power indication*


*Illustration 20: LED general indications*


*Illustration 21: LED IO indications*

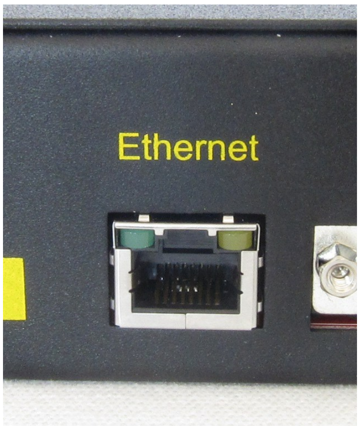



*Illustration 22: LED Ethernet indication*

These indications are explained below.

### 2.4.1 Power

This LED is lit steadily whenever the internal circuitry is powered (by the 5v line). This should always be lit for normal operation.

### 2.4.2 CPU

The LED flashes on and off once a second to show the main processor is running. During normal operation this LED should always be flashing.

### 2.4.3 SD

This LEDs flashes whenever the SD card is being accessed. When the LED is not lit, it is safe to remove the SD card from the SD card slot.

### 2.4.4 COM1

This LED flashes to show data activity on the serial port COM1. It will flash when valid data has been received according to the protocol using the port and when data is transmitted.

### 2.4.5 COM2

This LED flashes to show data activity on the serial port COM2. It will flash when valid data has been received according to the protocol using the port and when data is transmitted.

### 2.4.6 COM3

This LED flashes to show data activity on the serial port COM3. It will flash when valid data has been received according to the protocol using the port and when data is transmitted.

### 2.4.7 VF

This LED flashes to show data activity on the modem port. It will flash when valid data has been received according to the protocol using the port and when data is transmitted.

### 2.4.8 Eth

This LED flashes to show data activity on the Ethernet port. It will flash when valid data has been received according to the protocol using the port and when data is transmitted.

### 2.4.9 IO

This LED flashes to show data activity on the internal port that communicates with the Digital and/or Analogue IO boards within the unit. It will flash when valid data has been received according to the protocol using the port and when data is transmitted.

### 2.4.10 Digital inputs and outputs

These LEDs show the state of each of the digital inputs and outputs as follows.

| Inputs B | Outputs B |
| --- | --- |
| 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 | 17 18 19 20 21 22 23 24 25 26 72 28 29 30 31 32 |
| Inputs A | Outputs A |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 | 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 |

## 2.4.11 Analogue inputs and outputs

These LEDs show the state of each of the analogue inputs and outputs as follows.

| | |
|---|---|
| Inputs B | Outputs B |
| 9 10 11 12 13 14 15 16 | 9 10 11 12 13 14 15 16 |
| Inputs A | Outputs A |
| 1 2 3 4 5 6 7 8 | 1 2 3 4 5 6 7 8 |

The LED associated an input will flash at a speed that shows where in the measurement range the reading for that input is.

| | |
|---|---|
| Off | 0 – 19% |
| Slow flash | 20 – 39% |
| Medium flash | 40 – 59% |
| Fast flash | 60 – 79% |
| On | 80 – 100% |

The LED associated an output will light when the associated output goes active.

## 2.4.12 Voice/Audio inputs and outputs

These LEDs show the state of each of the voice/audio inputs and outputs as follows.

| Inputs B | Outputs B |
|---|---|
| 9 10 11 12 13 14 15 16 | 9 10 11 12 13 14 15 16 SD PTT |
| **Inputs A** | **Outputs A** |
| 1 2 3 4 5 6 7 8 | 1 2 3 4 5 6 7 8 SD PTT |

The LED associated an input will flash at a speed that shows where in the measurement range the reading for that input is.

| | |
|---|---|
| Off | 0 – 19% |
| Slow flash | 20 – 39% |
| Medium flash | 40 – 59% |
| Fast flash | 60 – 79% |
| On | 80 – 100% |

The LED associated an output will light when an event has been detected on the associated trigger input. The SD LED will light when there is reading/writing activity to the SD card. The PTT LED will light when the PTT output is activated.

## 2.5 Configuration

The configuration of a Serial IO MKII device is achieved with a software application called “Serial_IO_MKII_Config”. This application is available for Windows and Linux operating systems and can be downloaded from the HASCOM web site. The application main screen looks like the following:

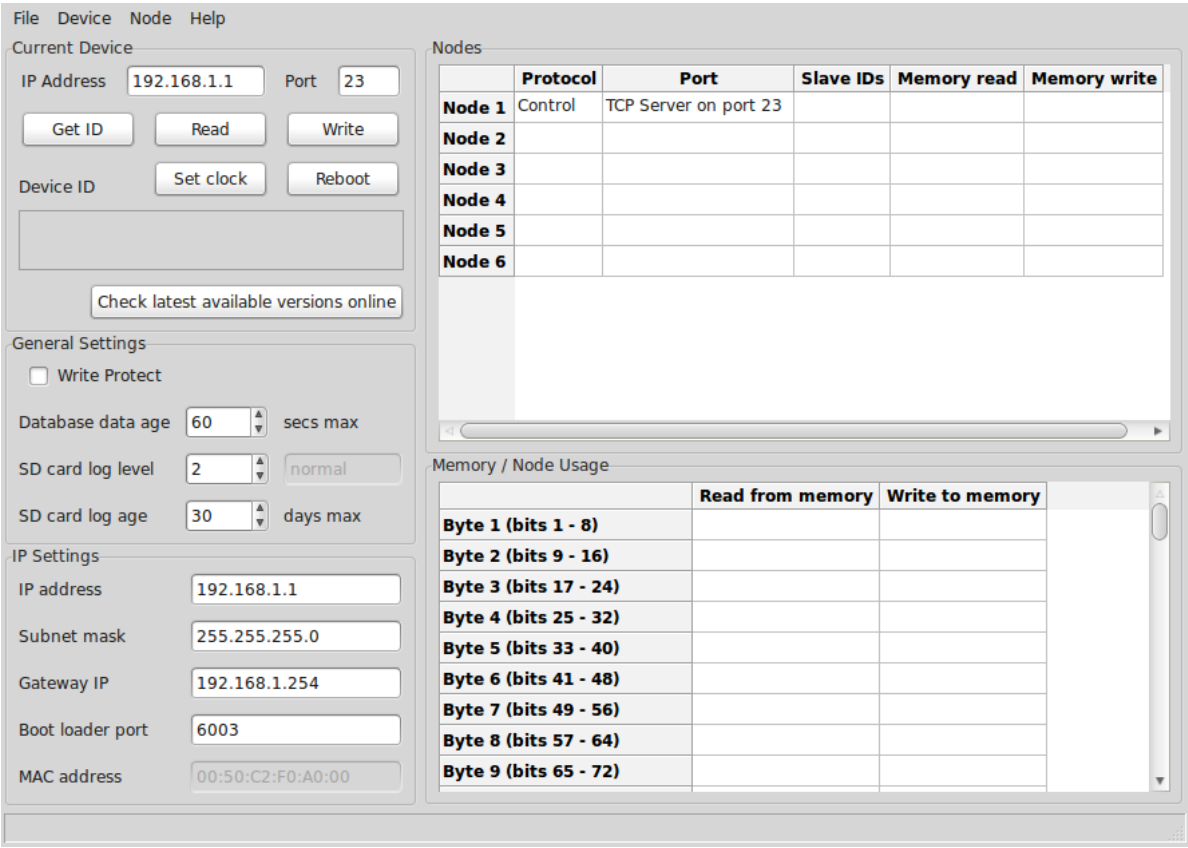


*Illustration 23: Config app main screen*

This application uses the Serial IO MKII Ethernet connection to communicate with the device, so a network connection to the device must exist for the application to work. The factory default static IP data is as follows:

| | |
|---|---|
| IP Address | 192.168.1.1 |
| Subnet mask | 255.255.255.0 |
| Gateway IP | 192.168.1.254 |
| Control/Configuration port | TCP 23 |

The Serial IO MKII stores it's configuration data in non-volatile memory and also optionally on an SD card. When the device boots it makes a copy of this configuration data and that copy is what it uses to run from.

The configuration application will edit the data stored in the non-volatile memory and optionally the SD card, but not the running copy, so therefore editing the configuration will not change any functionality of the device until the device is rebooted.

### 2.5.1 Current device

The current device section contains the settings of the device to be configured, and the buttons to initiate a data exchange with the device.

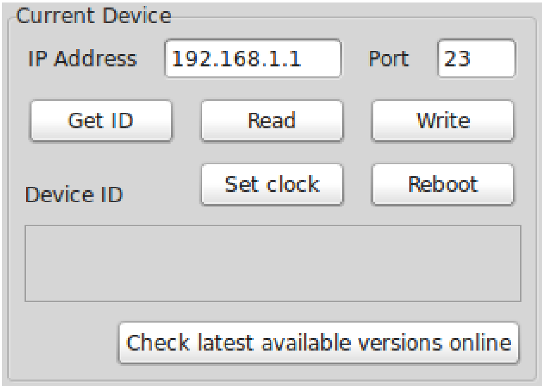



*Illustration 24: Config app current device*

The IP address and Port are the IP address and port that the Serial IO MKII device to be configured is currently using.

The Get ID button retrieves the ID, firmware version & serial number from the device. This can be useful to confirm communications is working to the device and that it is the device you expect it to be before making any configuration changes.

The Read and Write buttons reads and writes the configuration from/to the non-volatile memory in the device.

The Set clock button will use the local PCs time to set the Real Time Clock in the Serial IO MKII

The Reboot command causes the device to reboot and so any configuration changes to take effect.

The Check latest available versions online button will attempt to connect to the HASCOM website to retrieve and display the latest software and firmware version numbers.

### 2.5.2 General settings

These are general settings for the Serial IO MKII device being configured.

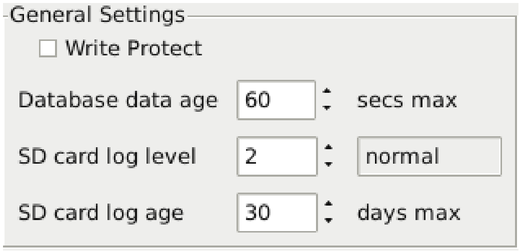



*Illustration 25: Config app current device*

Write Protect causes the configuration data once written to the device to be write protected. Once written and the device rebooted, no further changes can be made remotely and a factory reset is needed if you want to make changes.

The Database data age is the maximum age of data in the database for it to be considered valid. When data is received via any of the protocols it is considered to be fresh but it starts aging from that moment onwards. When data is too old, different actions may result depending on the configuration of the protocol accessing the data.

The SD card log level is the level of information written to any SD card that is inserted in to the SD card slot as follows:

| | |
|---|---|
| 0 | None |
| 1 | Important |
| 2 | Normal |
| 3 | Detailed |
| 4 | Debugging |
| 5 | Everything |

It is strongly recommended that levels 3 or 5 only be used on the test bench as they are very processor intensive, can take a long time to write to the SD card and can effect the normal operation of the unit.

The log data written to the SD will be written to a sub-directory called “/SERIALIO”. Within this directory files will be created with a naming convention of “yyyymmdd.txt” where yyyy is the year, mm the month and dd the day, eg 20120214.txt for 14th February 2012. The Serial IO MKII can use this naming convention to delete old log files so that the SD card does not fill up. SD Card Log Age sets the age after which old log files will be deleted.

## 2.5.3 IP settings

These are the IP settings the device shall use.

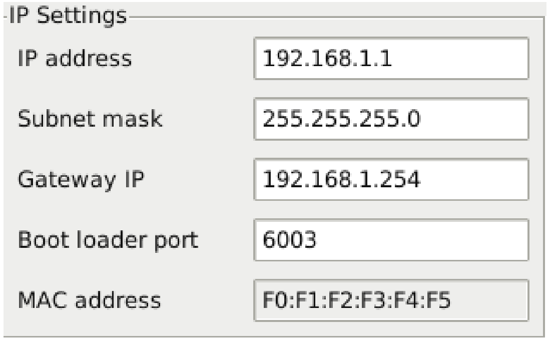



*Illustration 26: Config app IP settings*

The MAC address can not be edited and is displayed for informational purposes only.

## 2.5.4 Nodes

The Nodes and Memory / Node Usage areas are where a summary of the current node configuration is displayed. The Memory / Node Usage area is read only, however right clicking in the Nodes area will invoke a pop up menu to allow you to add, delete or edit a node.

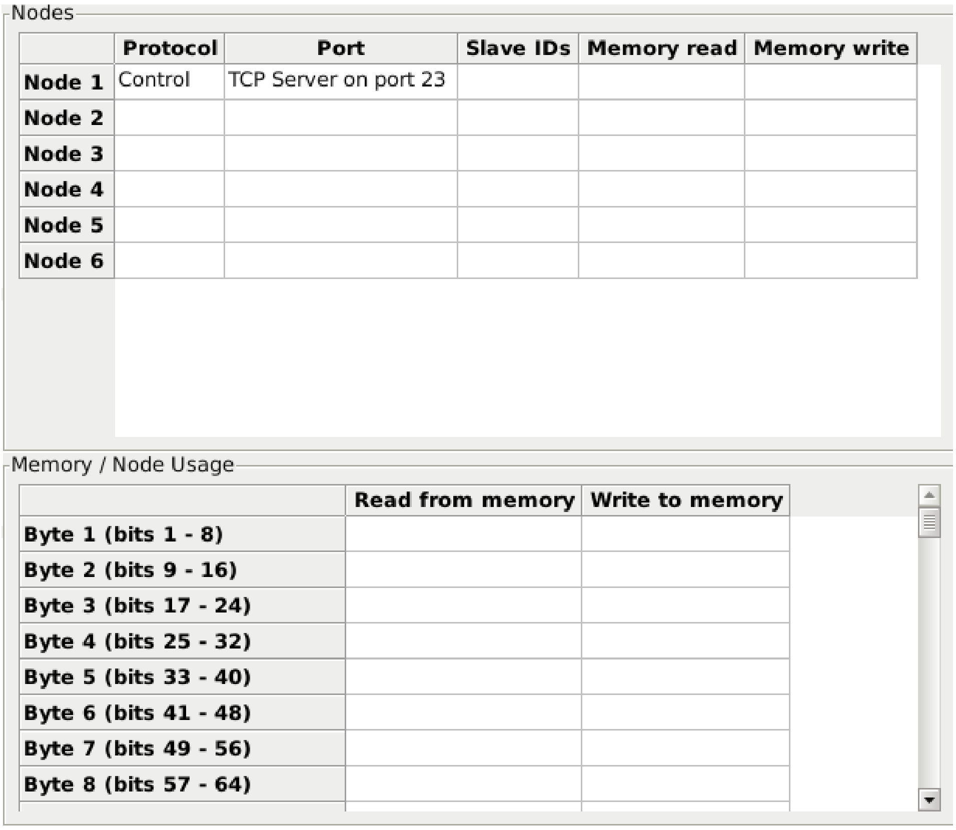



*Illustration 27: Config app Nodes*

When you add or edit a node, the edit node dialog box is displayed.

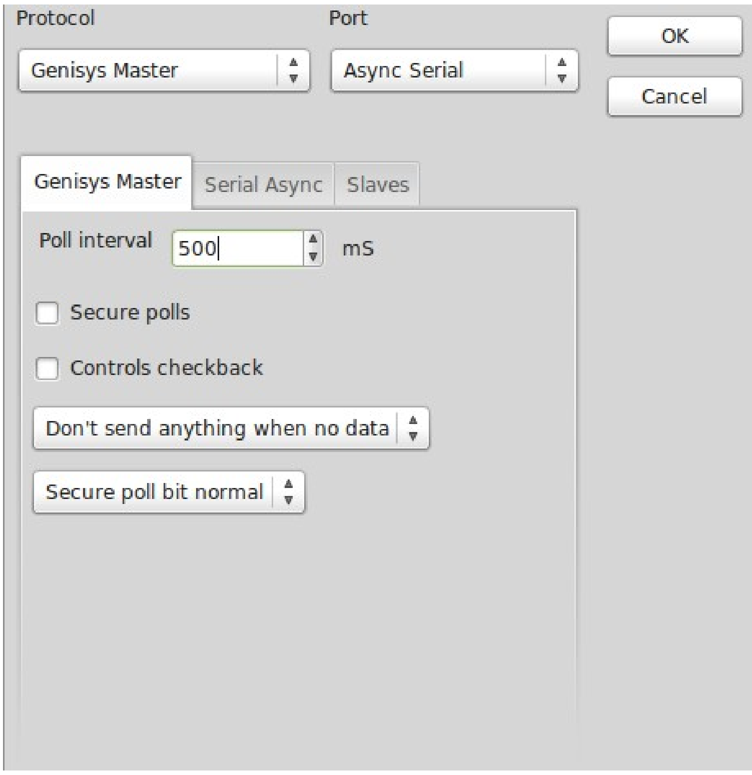


*Illustration 28: Config app edit node*

From the Protocol drop down selection you can select the protocol you want this node to use. Configuration specific to that protocol will then be displayed in the first tab.

From the Port drop down selection you can select the port you want this node to use. Configuration specific to that protocol will then be displayed in the second tab.

The slave configuration for this node is displayed in the 3rd tab.

### 2.5.4.1 Control

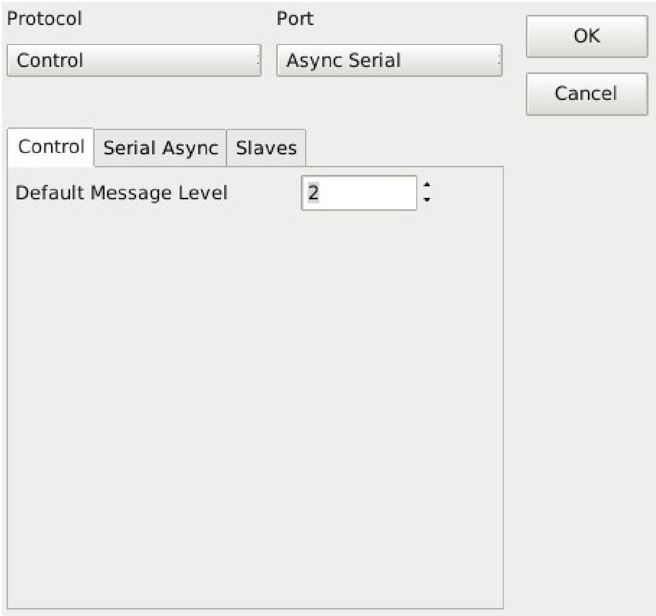



*Illustration 29: Config Control protocol*

Here you can set the default message level for when a Telnet connection is established.

### 2.5.4.2 IO master

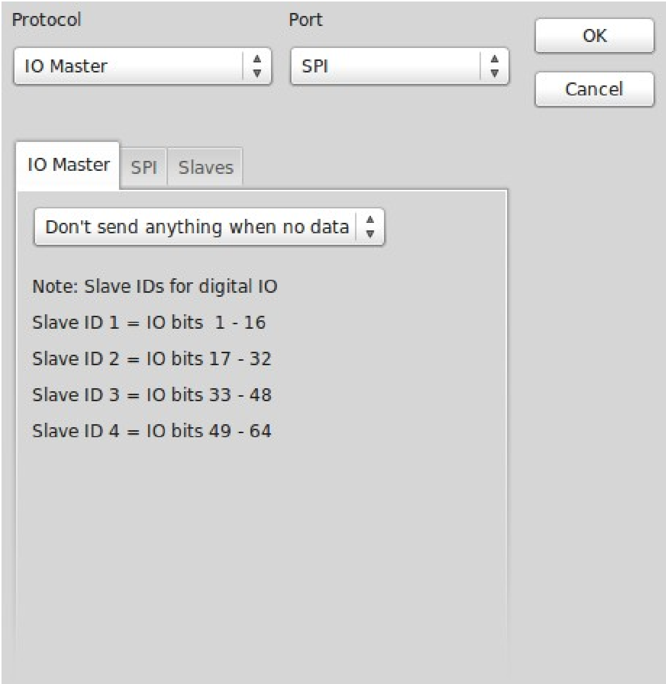



*Illustration 30: Config IO master protocol*

The "Don't send anything..." drop down allows selection of what action to take when the data being read from the database is too old.

### 2.5.4.3 Analogue/Audio master

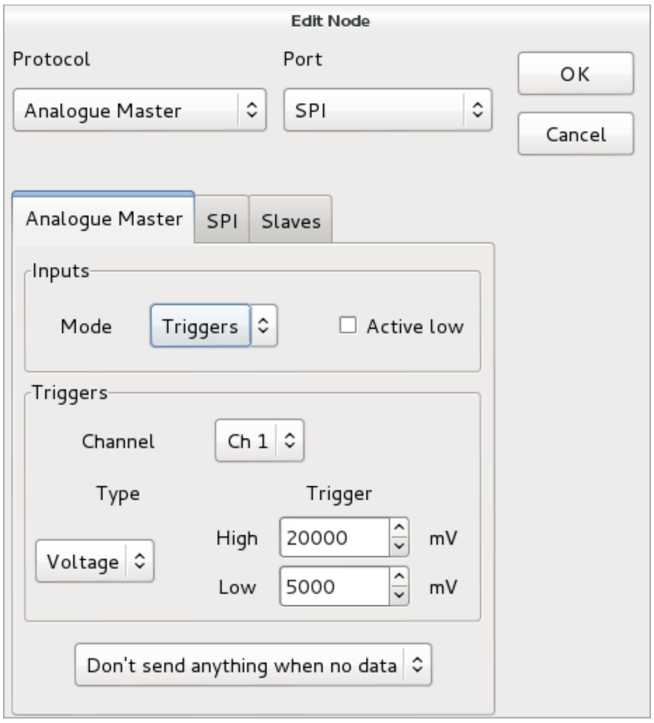



*Illustration 31: Config Analogue master protocol*

Inputs Mode selects the mode of operation of the analogue/audio master, this effects how the analogue measurements are processed.

Triggers Mode: In this mode of operation one data byte (8 bits) of information is produced. Each of the 4 input channels are compared against some defined trigger points. If the measured level is above or below a trigger point a bit is set in the resulting data byte (see “Active Low” below). The mapping of the data byte is as follows:

| Data bit | Signal |
|---|---|
| 1 | Channel 1 low |
| 2 | Channel 1 high |
| 3 | Channel 2 low |
| 4 | Channel 2 high |
| 5 | Channel 3 low |
| 6 | Channel 3 high |
| 7 | Channel 4 low |
| 8 | Channel 4 high |

Readings Mode: In this mode of operation sixteen data bytes (128 bits) of information is produced. Each of the 4 input channels are read for both voltage and current inputs and the reading for each input is stored in two data bytes (16 bits) as follows:

| Data bit | Signal |
|---|---|
| 1 | LSB |
| 2 | |
| 3 | |
| 4 | |
| 5 | |
| 6 | |
| 7 | 15 bit unsigned integer reading in either mV or uA |
| 8 | |
| 9 | |
| 10 | |
| 11 | |
| 12 | |
| 13 | |
| 14 | |
| 15 | MSB |
| 16 | Bit flag:<br>0 = voltage reading<br>1 = current reading |

| Data byte | Signal |
|---|---|
| 1<br>2 | Channel 1 voltage input |
| 3<br>4 | Channel 1 current input |
| 5<br>6 | Channel 2 voltage input |
| 7<br>8 | Channel 2 current input |
| 9<br>10 | Channel 3 voltage input |
| 11<br>12 | Channel 3 current input |
| 13<br>14 | Channel 4 voltage input |
| 15<br>16 | Channel 4 current input |

Active Low: When ticked, the trigger indication bits have reversed polarity, ie 1 for inactive and 0 for active.

Triggers: This section is only relevant when in trigger mode. It allows setting of the trigger levels.

Channel: Selects the channel for which the displayed settings apply.

Type: Selects either voltage or current as the type of measurement.

High / Low: Sets the trigger point in either mV (10000 mv = 10v) or uA (10000 uA = 10 mA).

The "Don't send anything..." drop down allows selection of what action to take when the data being read from the database is too old.

Outputs: the Analogue master protocol supports sending one byte (8 bits) of data. This byte is used to control the outputs on the Analogue IO Expansion board.

Note: For the Analogue Master protocol, the port should be set to SPI.

### 2.5.4.4 UCE master

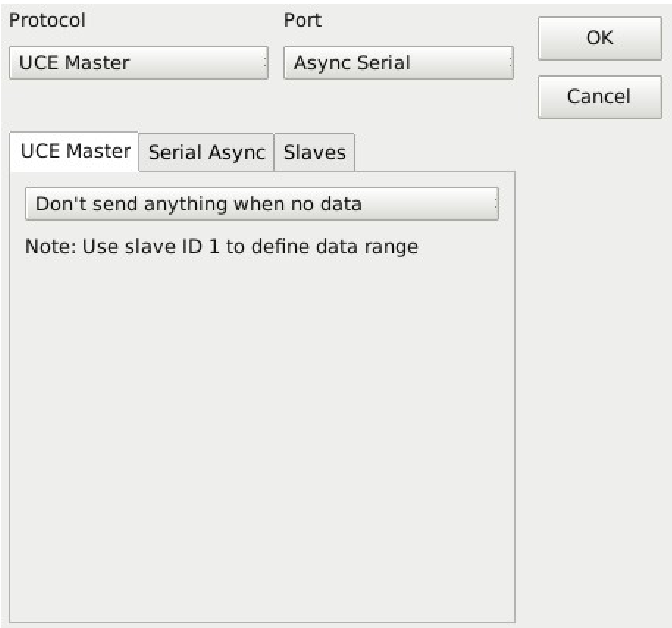



*Illustration 32: Config UCE master protocol*

The "Don't send anything..." drop down allows selection of what action to take when the data being read from the database is too old.

### 2.5.4.5 UCE slave

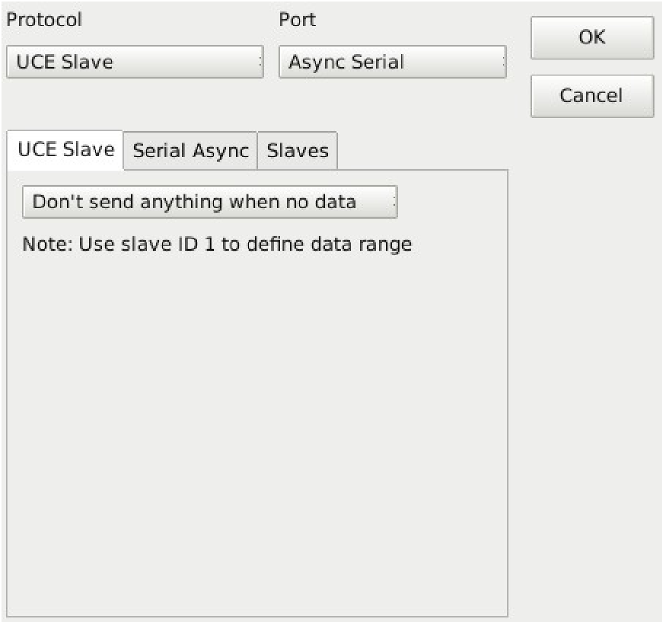



*Illustration 33: Config UCE slave protocol*

The ”Don't send anything...” drop down allows selection of what action to take when the data being read from the database is too old.

### 2.5.4.6 S2 master

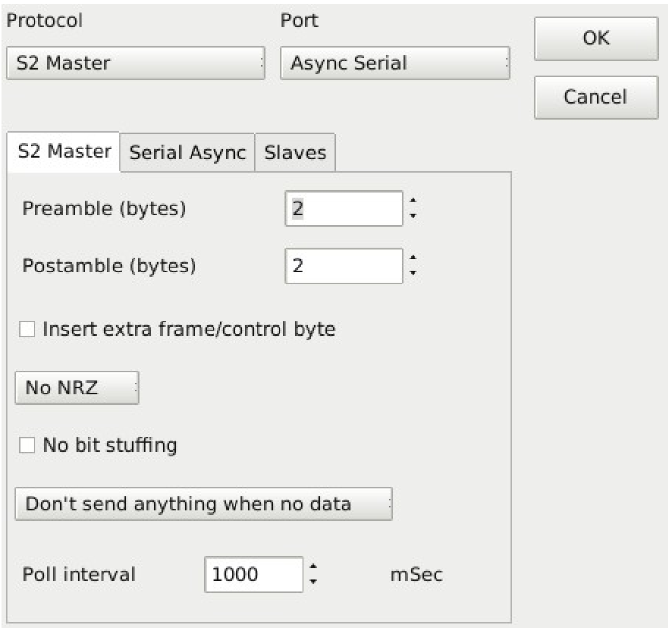



*Illustration 34: Config S2 master protocol*

The preamble and postamble is the number of 7E hex bytes that are transmitted before and after an S2 packet.

Insert extra frame/control byte causes an extra byte of data to be transmitted after the address byte in an S2 packet. This suits some implementations of the S2 protocol.

The NRZ drop down allows you to select if NRZ is applied at the protocol level. This suits some implementations of the S2 protocol.

S2 is a synchronous protocol, and therefore uses bit stuffing to differentiate between the 7E hex bytes sent as flags in between packets and 7E data bytes in packets. No bit stuffing allow this bit stuffing to be turned off and is useful using Ethernet protocols where one Ethernet packet is assumed to equate to one S2 packet.

The ”Don't send anything...” drop down allows selection of what action to take when the data being read from the database is too old.

The Poll interval is the time between each successive poll packet being sent.

### 2.5.4.7 S2 slave

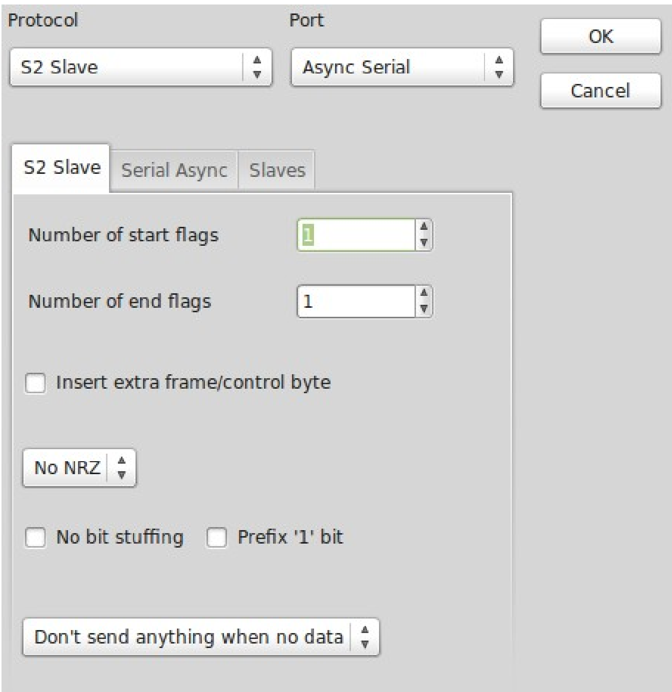



*Illustration 35: Config S2 slave protocol*

The preamble and postamble is the number of 7E hex bytes that are transmitted before and after an S2 packet.

Insert extra frame/control byte causes an extra byte of data to be transmitted after the address byte in an S2 packet. This suits some implementations of the S2 protocol.

The NRZ drop down allows you to select if NRZ is applied at the protocol level. This suits some implementations of the S2 protocol.

S2 is a synchronous protocol, and therefore uses bit stuffing to differentiate between the 7E hex bytes sent as flags in between packets and 7E data bytes in packets. No bit stuffing allow this bit stuffing to be turned off and is useful using Ethernet protocols where one Ethernet packet is assumed to equate to one S2 packet.

The Prefix '1' bit check box enables a work around to support incorrect 3rd party S2 master implementations. Contact HASCOM for more information.

The "Don't send anything..." drop down allows selection of what action to take when the data being read from the database is too old.

## 2.5.4.8 Genisys master

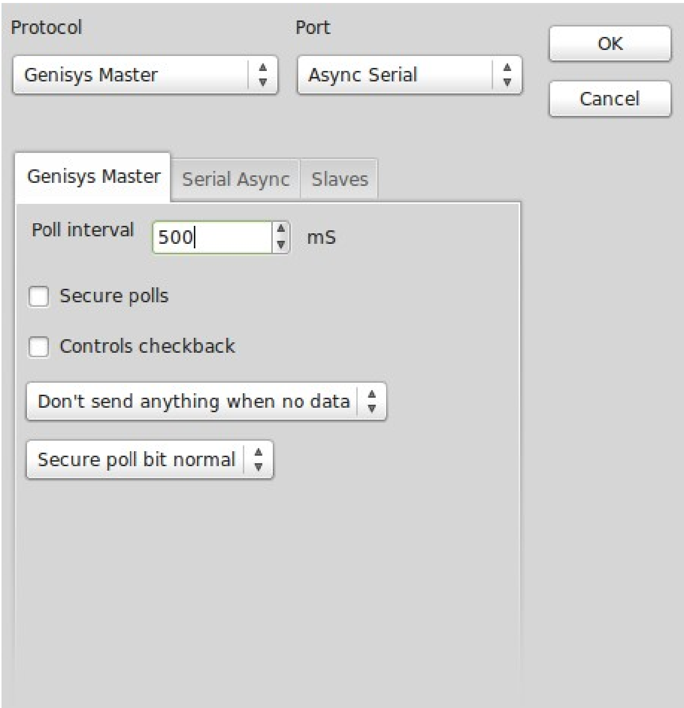



*Illustration 36: Config Genisys master protocol*

The Poll interval is the time between each successive poll packet being sent.

Secure polls and Controls checkback allow enabling of these protocol features.

The "Don't send anything..." drop down allows selection of what action to take when the data being read from the database is too old.

The Secure poll bit drop down allows selection of two different Genisys implementations. The difference effects if non-secure poll packets should be ignored, ie when not using secure polling, and the meaning of the bit flag that controls that setting is reversed on some 3rd party protocol implementations.

### 2.5.4.9 Genisys slave

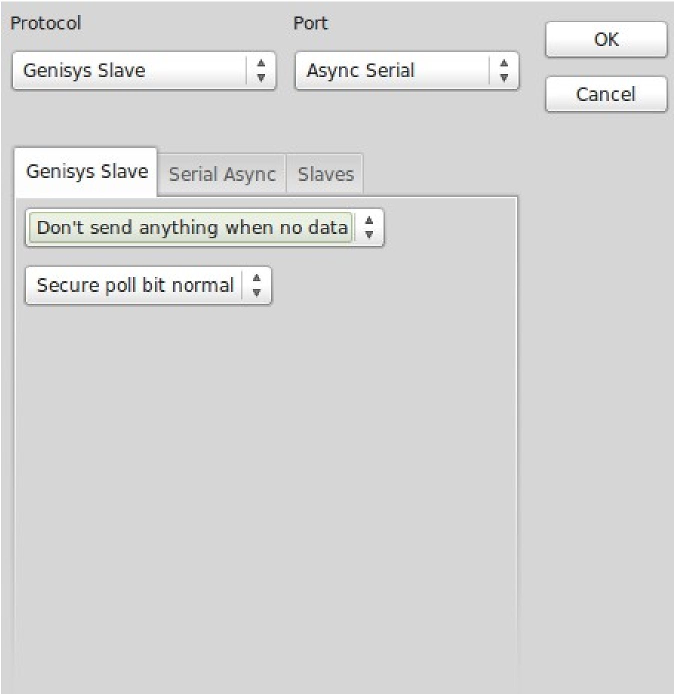



*Illustration 37: Config Genisys slave protocol*

The ”Don't send anything...” drop down allows selection of what action to take when the data being read from the database is too old.

The Secure poll bit drop down allows selection of two different Genisys implementations. The difference effects if non-secure poll packets should be ignored, ie when not using secure polling, and the meaning of the bit flag that controls that setting is reversed on some 3rd party protocol implementations.

### 2.5.4.10 Host master

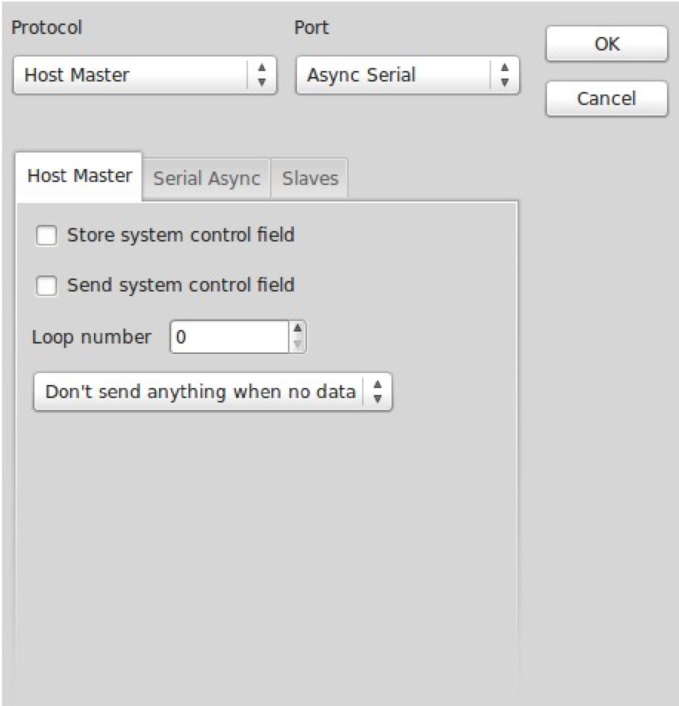



*Illustration 38: Config Host master protocol*

The Store system control field and Send system control field check boxes enable support of a very specific use case where the Serial IO is emulating a HASCOM Telemetry Server Rack and should normally be left unchecked. Contact HASCOM for more information.

The loop number is the identity of the loop being used.

The ”Don't send anything...” drop down allows selection of what action to take when the data being read from the database is too old.

### 2.5.4.11 Host slave

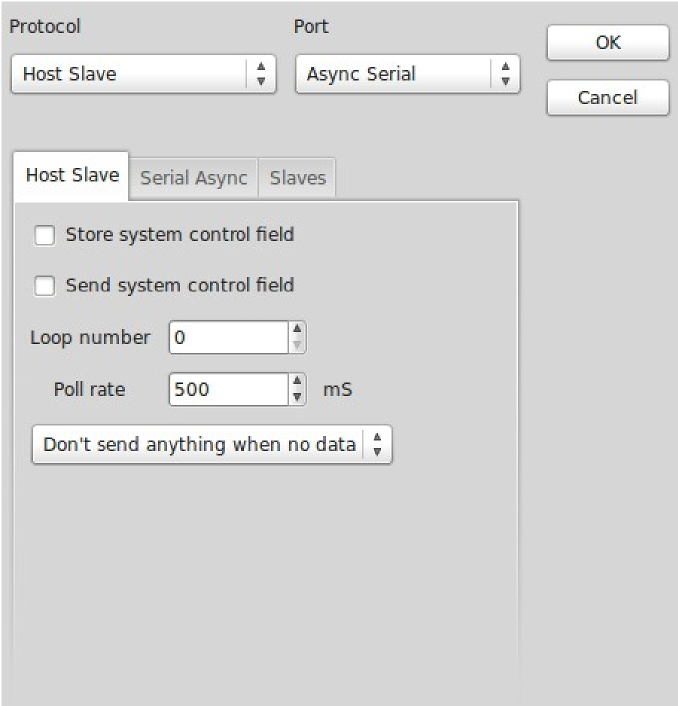



*Illustration 39: Config Host slave protocol*

The Store system control field and Send system control field check boxes enable support of a very specific use case where the Serial IO is emulating a HASCOM Telemetry Server Rack and should normally be left unchecked. Contact HASCOM for more information.

The loop number is the identity of the loop being used.

The Poll rate is the time between each successive poll packet being sent.

The ”Don't send anything...” drop down allows selection of what action to take when the data being read from the database is too old.

### 2.5.4.12 NTP client

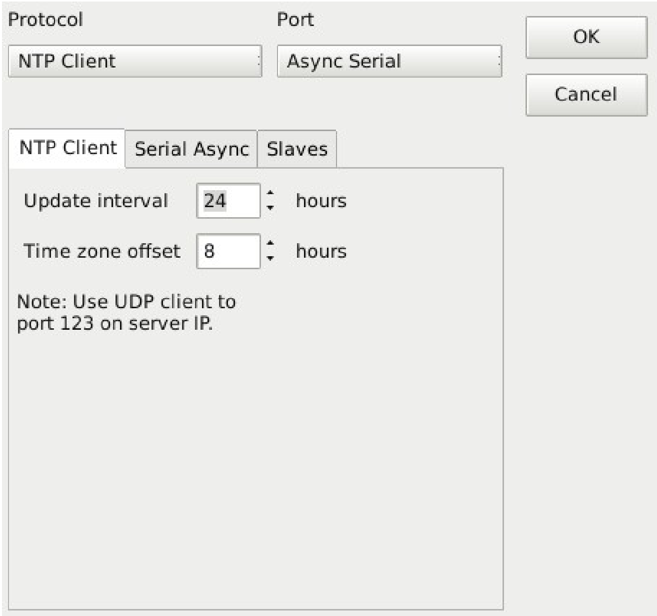



*Illustration 40: Config NTP client protocol*

The update interval is the time between successive requests being sent to the NTP server.

The timezone offset is the offset in hours of local time from the time returned from the server (typically UTC time).

### 2.5.4.13 Null

Sometimes when configuring nodes it may be useful for parts of the database to appear to contain valid data, when it is in fact not being written to by any node. This could be important when for example “Don't send anything when no data” is being used and there is a mismatch between the data sizes of nodes. The Null 'protocol' will write data to the database to make it appear valid.

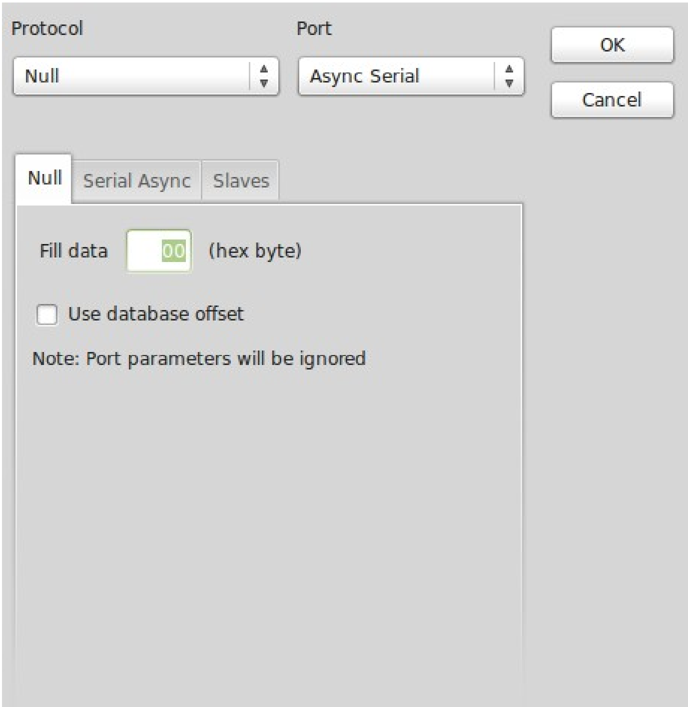



*Illustration 41: Config Null protocol*

The Fill data value is used to fill each byte (8 bits) of data that is filled in the database by the protocol. This is a hex value, so use “00” for all 0's, “FF” for all 1's, etc.

The Use database offset check box will instead fill the data with an incrementing number rather that all the same fill data.

### 2.5.4.14 TCP Server

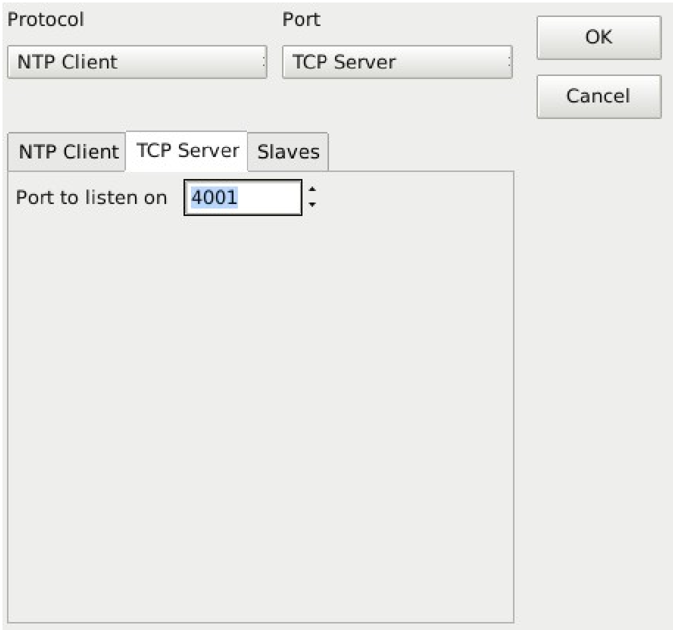


*Illustration 42: Config TCP server port*

The Port to listen on is the TCP port number the server shall run on.

### 2.5.4.15 TCP Client

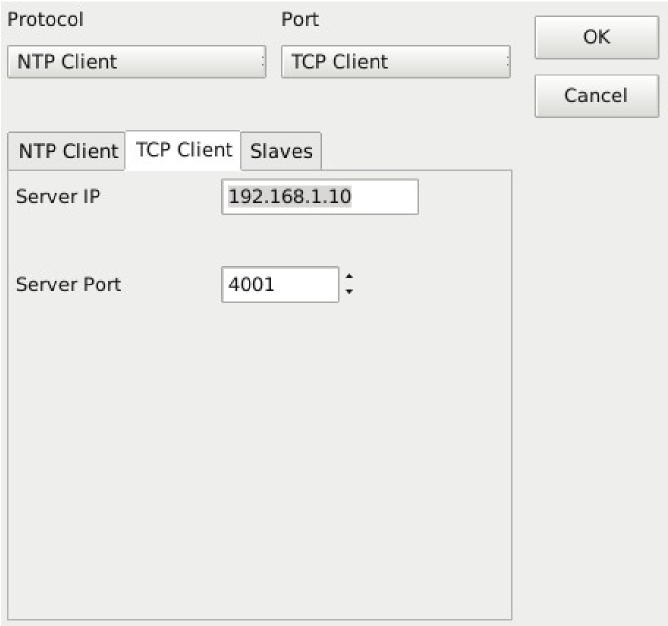


*Illustration 43: Config TCP client port*

The server IP is the IP address of the server PC. Note: the Serial IO MKII does not support DNS, so names can not be used.

Server port is the TCP port number the server is listening on.

#### 2.5.4.16 UDP Server

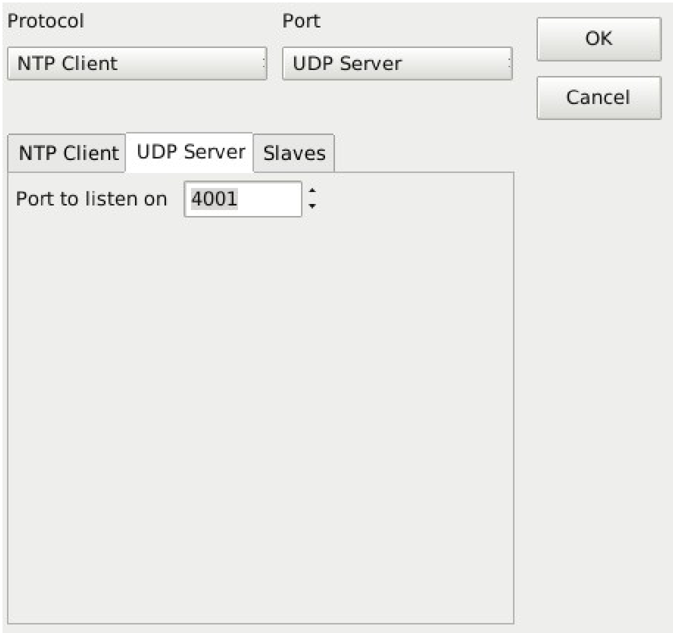


*Illustration 44: Config UDP server port*

The Port to listen on is the UDP port number the server shall run on.

### 2.5.4.17 UDP Client

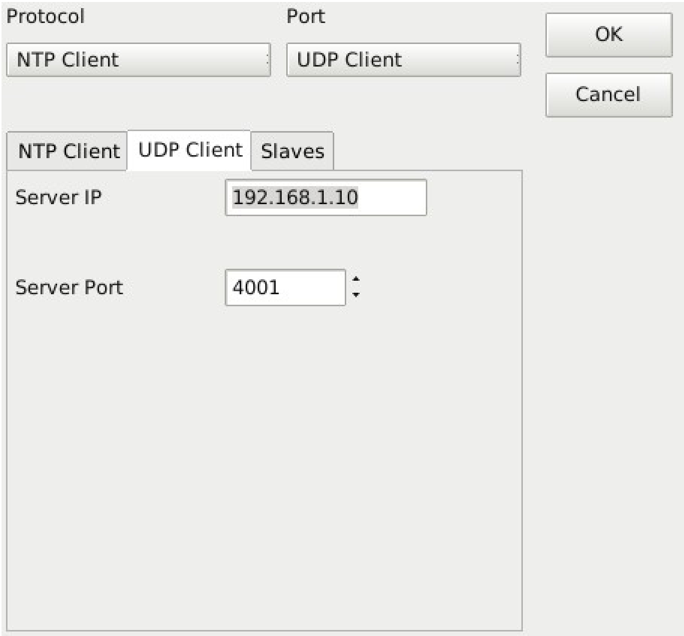



*Illustration 45: Config UDP client port*

The server IP is the IP address of the server PC. Note: the Serial IO MKII does not support DNS, so names can not be used.

Server port is the UDP port number the server is listening on.

### 2.5.4.18 Async serial

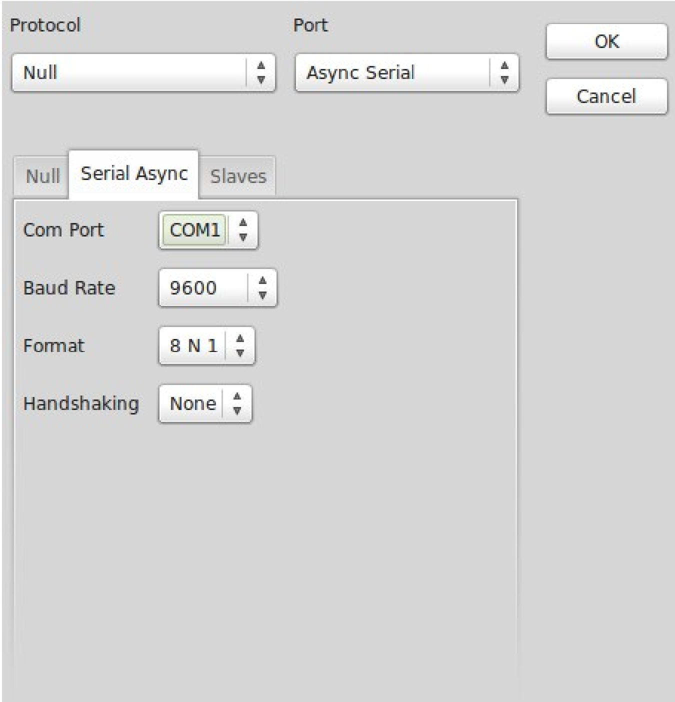



*Illustration 46: Config Async serial port*

Com port is the port to use on the Serial IO MKII.

The Baud rate and format allow selection of communications parameters.

The Format drop down list selects the character format, “8 N 1” means 8 data bits, No parity & 1 stop bit.

The Handshaking drop down list enables DTE, DCE or no hardware handshaking.

### 2.5.4.19 Sync serial

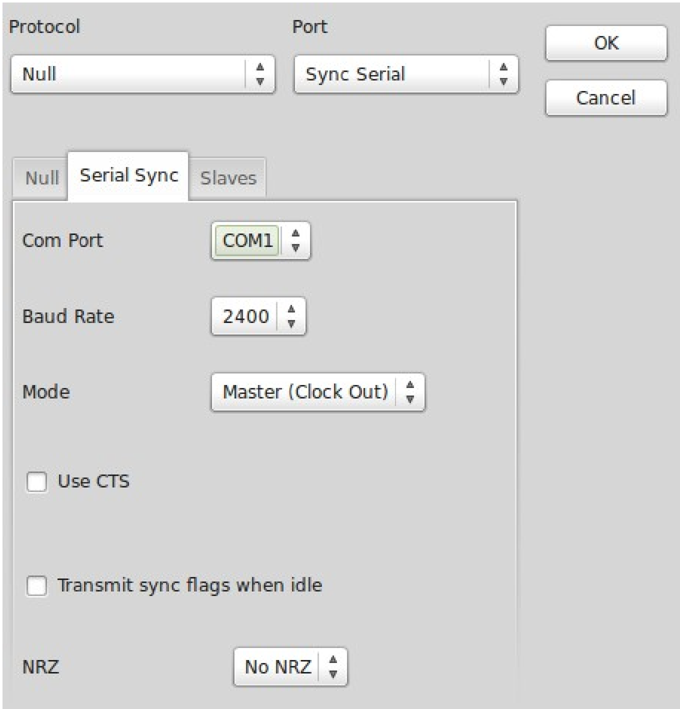



*Illustration 47: Config Sync serial port*

Com port is the port to use on the Serial IO MKII.

The baud rate and format allow selection of communications parameters.

The mode allows selection of the master or slave synchronous mode functionality.

Use CTS allows the CTS line to be used for flow control.

Use Transmit sync flags when idle causes 7E hex to be sent continuously when no data is being sent. This can be used for protocols like S2 master.

The NRZ drop down allows you to select if NRZ is applied at the port level.

Note: Using the Sync Serial port is very processor intensive. If the processor has other, relatively long tasks, such as writing to an SD card, the operation of the Sync Serial port may be effected and reliability of the connection could be impacted. It is recommended that when using Sync Serial, other task and the node count be kept to a minimum.

### 2.5.4.20 Modem

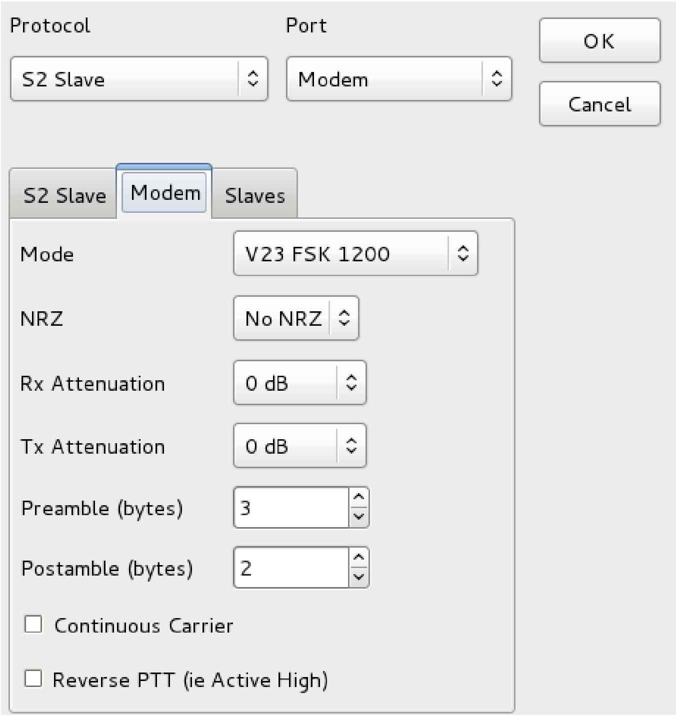



*Illustration 48: Config Modem port*

Mode allows selection of different modulation modes, such as V.22, V.23, Bell 103, etc.

The NRZ drop down allows you to select if NRZ is applied at the port level.

The attenuation drop downs allow both the receive and transmit signals to be attenuated.

The Preamble & Postamble selects the amount of time the modem will transmit continuous 1's before and after the data packet. The units are in bytes, this equates to “(bytes * 8) / baud” seconds, so 3 bytes at 1200 baud equates to 20 mS.

Continuous Carrier when selected causes the modem to continue transmitting in between data packets. A continuous stream of 1's are sent until the next data packet is ready to be sent. This can be useful when running as a master in a full duplex environment.

Reverse PTT allows the action of the PTT output to be reversed.

### 2.5.4.21 SPI

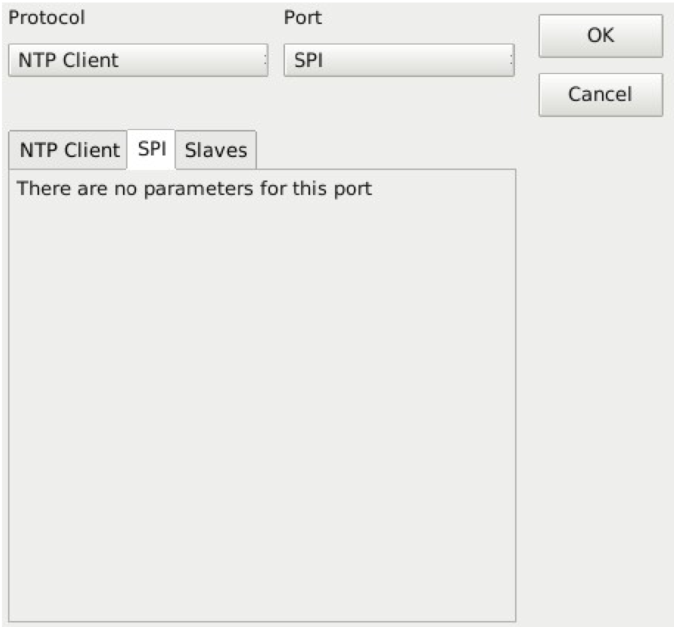



*Illustration 49: Config SPI port*

There are no parameters for the SPI port.

## 2.5.4.22 Slaves

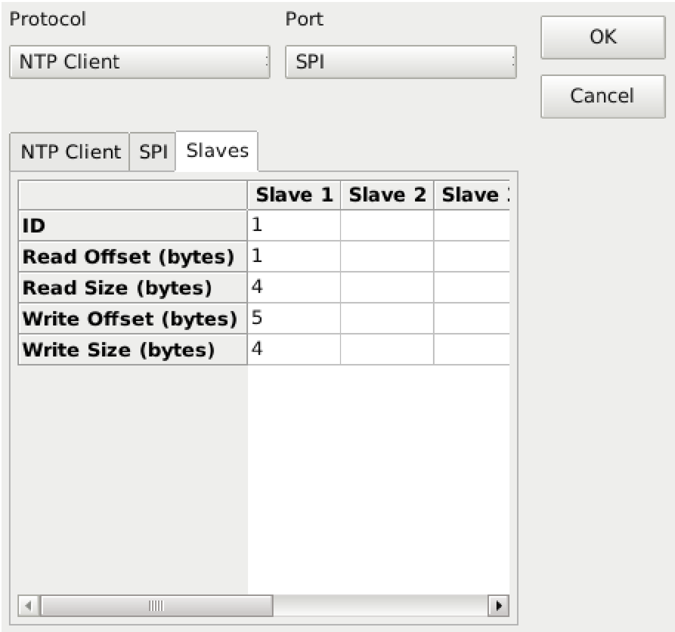



*Illustration 50: Config Slaves*

Slaves are edited by directly entering numeric information in to the grid displayed. Up to 50 slaves can be define for each protocol.

The ID is the slave ID to use within protocols supporting point to multi-point operation. For other protocols, a single slave with an ID of 1 should be used.

The Read offset and Read size define where in the database the slave will read data from when it wants to transmit data out the port.

The Write offset and Write size define where in the database the slave will write data to when it receives data via the port the port.

### 2.5.5 Load configuration from SD card

The Serial IO MKII will test for the presence of an SD card when it boots up. If it finds an SD card it will look for the following file “/SERIALIO/Config.sio”. If it finds that file, it will try to load it's configuration from it. If it succeeds it will store that configuration in internal non-volatile memory and use it, if it fails it will fall back to using the configuration it already has in internal non-volatile memory.

The file format of “Config.sio” file is the same as the file format that is written by the Serial_IO_MKII_Config application. Therefore it is possible to edit a configuration with the application, save it to a file on an SD card, then put that card in a Serial IO MKII device to reconfigure it at boot time.

Note that, if the configuration of a Serial IO MKII device is edited via the Ethernet port, if the file “/SERIALIO/Config.sio” exists on an SD card, that file will also be updated to match the updated configuration.

### 2.5.6 Factory reset button

If during normal operation the “Factory Reset” button is pressed for 5 seconds the configuration stored in non-volatile memory will be reset to factory defaults. The device will then reboot. Note however, the devices IP information will not be reset by this procedure.

However, if the “Factory Reset” button continues to be pressed for a further 2 seconds after the device reboots, the IP information will be reset back to factory defaults as well.

The default IP information is as follows:

| | |
|---|---|
| IP Address | 192.168.1.1 |
| Subnet mask | 255.255.255.0 |
| Gateway IP | 192.168.1.254 |

See section 3.7 for more information.

## 2.5.7 Voice/Audio operation & configuration

The voice/audio circuit is self contained and does not rely on the correct configuration or operation of the rest of the unit.

The operation is configured via a text file "config.ini" in the root directory on the micro SD card inserted next to the output connector. The audio files for playback should also be in the root directory on this micro SD card.

The config.ini file follows the same format as that used by Windows, which is well documented elsewhere. Please see http://en.wikipedia.org/wiki/INI_file.

Example config.ini file:

```
; This sets default parameters (optional)
[Defaults]
WaitSquelch=no          ; Wait for squelch before PTT (high, low, no)
AttenuationLeft=20      ; Attenuation in -dB for left channel (0-120)
AttenuationRight=20     ; Attenuation in -dB for right channel (0-120)
KeyUpDelay=2            ; Delay in seconds from PTT to playback start (1-60)
KeyDownDelay=1          ; Delay in seconds from playback end to PTT release (0-60)
ReplayCount=1           ; Number of replays in addition to the initial play (0-100)
ReplayDelay=30          ; Time to wait in seconds from end of one play to start of
the next (5-600)

; This defines any event specific parameters
[Input1_1]
FileName=in1_1.mp3      ; Filename (8.3 format only) to play, supports .wav & .mp3
AttenuationLeft=25      ; Optional parameters just for this event
AttenuationRight=25
KeyUpDelay=3
KeyDownDelay=1
ReplayCount=2
ReplayDelay=15

[Input1_0]
FileName=in1_0.mp3
AttenuationLeft=30
```

```
AttenuationRight=30
KeyUpDelay=1
KeyDownDelay=0
ReplayCount=0
ReplayDelay=15

[Input2_1]
FileName=in2_1.mp3
AttenuationLeft=20
AttenuationRight=120

[Input2_0]
FileName=in2_0.mp3
AttenuationLeft=120
AttenuationRight=20

[Input3_1]
FileName=in3_1.mp3

[Input3_0]
FileName=in3_0.mp3

[Input4_1]
FileName=in4_1.mp3

[Input4_0]
FileName=in4_0.mp3

[Input5_1]
FileName=in5_1.mp3

[Input5_0]
FileName=in5_0.mp3

[Input6_1]
FileName=in6_1.mp3

[Input6_0]
FileName=in6_0.mp3
```

```
[Digital1_1]
FileName=dig1_1.mp3

[Digital1_0]
FileName=dig1_0.mp3

[Digital2_1]
FileName=dig2_1.mp3

[Digital2_0]
FileName=dig2_0.mp3

[Digital3_1]
FileName=dig3_1.mp3

[Digital3_0]
FileName=dig3_0.mp3

[Digital4_1]
FileName=dig4_1.mp3

[Digital4_0]
FileName=dig4_0.mp3

[Digital5_1]
FileName=dig5_1.mp3

[Digital5_0]
FileName=dig5_0.mp3

[Digital6_1]
FileName=dig6_1.mp3

[Digital6_0]
FileName=dig6_0.mp3

[Digital7_1]
FileName=dig7_1.mp3
```

```
[Digital7_0]
FileName=dig7_0.mp3

[Digital8_1]
FileName=dig8_1.mp3

[Digital8_0]
FileName=dig8_0.mp3
```

The [Defaults] section defines the parameters to use when an event occurs if these parameters are not specifically defined for the event.

The [Input*n*_1] section defines the parameters to use when an event is triggered by input *n* going high.

The [Input*n*_0] section defines the parameters to use when an event is triggered by input *n* going low.

The [Digital*n*_1] section defines the parameters to use when an event is triggered by digital bit *n* going high. Digital bits are passed to the voice/audio board from the main unit and can be configured using the configuration app.

The [Digital*n*_0] section defines the parameters to use when an event is triggered by digital bit *n* going low. Digital bits are passed to the voice/audio board from the main unit and can be configured using the configuration app.

Up to 4 simultaneous alerts can be triggered at the same time.

## 2.6 Logging to SD card

If at any time during normal operation an SD card is inserted in to the SD card slot, the Serial IO MKII will automatically start writing data to a log file.

The SD card log level set in the configuration, see section 2.5.2, controls the level of information written to the SD card as follows:

| | |
|---|---|
| 0 | None |
| 1 | Important |
| 2 | Normal |
| 3 | Detailed |
| 4 | Debugging |
| 5 | Everything |

It is strongly recommended that levels 3 or 5 only be used on the test bench as they are very processor intensive, can take a long time to write to the SD card and can effect the normal operation of the unit.

The log data written to the SD will be written to a sub-directory called "/SERIALIO". Within this directory files will be created with a naming convention of "yyyymmdd.txt" where yyyy is the year, mm the month and dd the day, eg 20120214.txt for 14th February 2012. The Serial IO MKII can use this naming convention to delete old log files so that the SD card does not fill up. SD Card Log Age sets the age after which old log files will be deleted.

# 3 Maintenance Manual

## *3.1 Introduction*

The Serial IO MKII is a highly complex device, based on Atmel ATMega processors. This manual does not intend to explain the detailed workings of the device to component level, it is only to assist in maintenance and if repair by complete board replacement is being attempted.

## *3.2 Preventative Maintenance*

The units in operation should be regularly inspected for damage, corrosion, etc.

### 3.2.1 Environmental Considerations

The Serial IO MKII should be operated in a clean, dry, stable environment. Ambient temperature should be within 0°C -80°C, non-condensing.

### 3.2.2 Electrical Considerations

The Serial IO MKII should be operated within specifications as per the manual. For example the supply voltage must stay between 8 to 28 volts DC, and any external connections must not suffer from voltage spikes, etc.

### 3.2.3 Corrective Maintenance

A “repair by replacement” policy is suggested, with the faulty devices being returned to HASCOM for diagnosis and repair.

### 3.2.4 Spares and Configuration/Test Equipment

It is suggested to hold at least 2 complete spare Serial IO MKII device, or 1 device for every 5 in the field, which ever is greater. For critical applications a higher number or spares could be held.

Configuration is achieved with a Windows/Linux application which is freely available from HASCOM. The application connects to the unit via Ethernet. Therefore a PC running Windows or Linux with a spare Ethernet port & an Ethernet cable is required for configuration.

For configuration and testing, a bench power supply is needed to power the unit.

### *3.3 Main board*

The main board is responsible for almost all of the functionality of the Serial IO MKII.

### 3.3.1 Connectors

P1 - Expansion board data connector

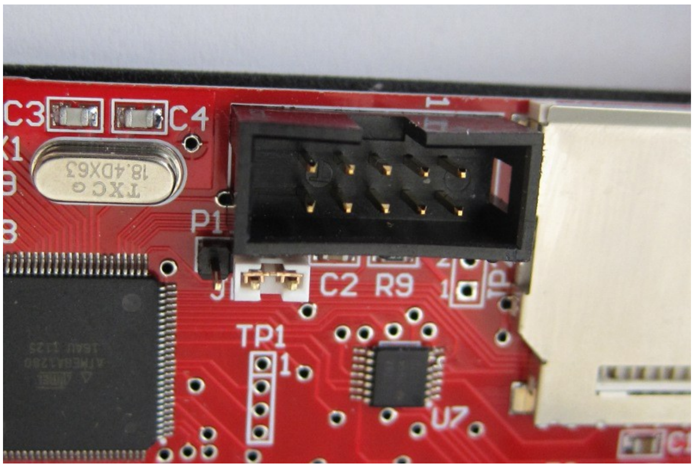



*Illustration 51: Main board P1 connector*

| | | | |
|---|---|---|---|
| 1 | SPI MOSI | 2 | +5v if JP2 linked |
| 3 | Gnd | 4 | Gnd |
| 5 | SPI ExtSS1 / Reset | 6 | SPI ExtSS4 |
| 7 | SPI SCK | 8 | SPI ExtSS3 |
| 9 | SPI MISO | 10 | SPI ExtSS2 |

P2 - Expansion board power connector

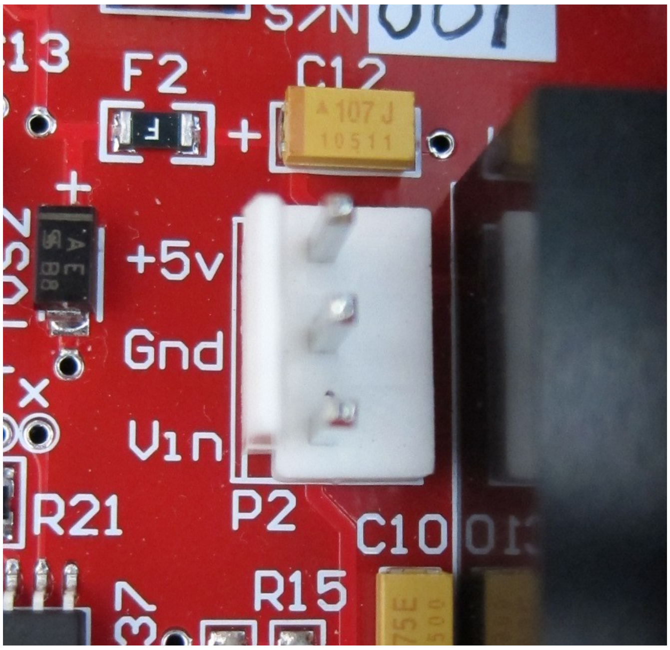


*Illustration 52: Main board P2 connector*

| | |
|---|---|
| 1 | +5V supply output |
| 2 | Gnd |
| 3 | Vin (fused supply voltage) output |

### 3.3.2 Jumpers

JP1 – Run / ISP

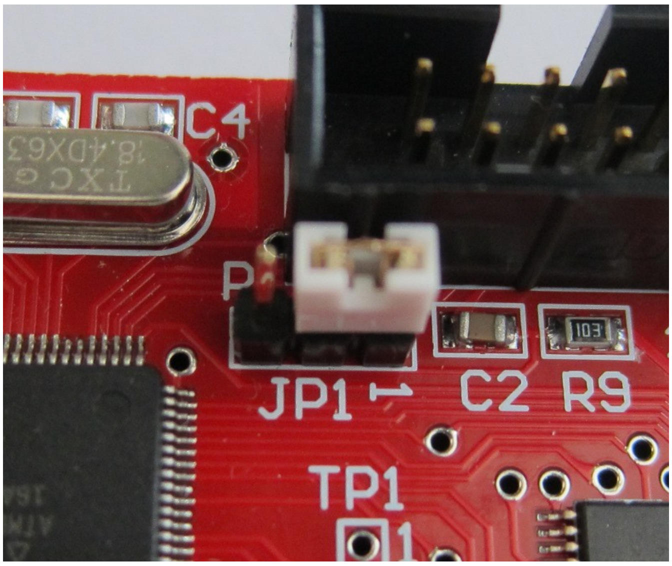


*Illustration 53: Main board JP1 jumper*

| | |
|---|---|
| Link 1-2 | SPI ExtSS1 to P1 pin 5 |
| Link 2-3 | Reset to P1 pin 5 |

JP1 should always be linked 1-2 for normal operation. Linking 2-3 is for factory programming the boot loader only.

JP2 – ISP power

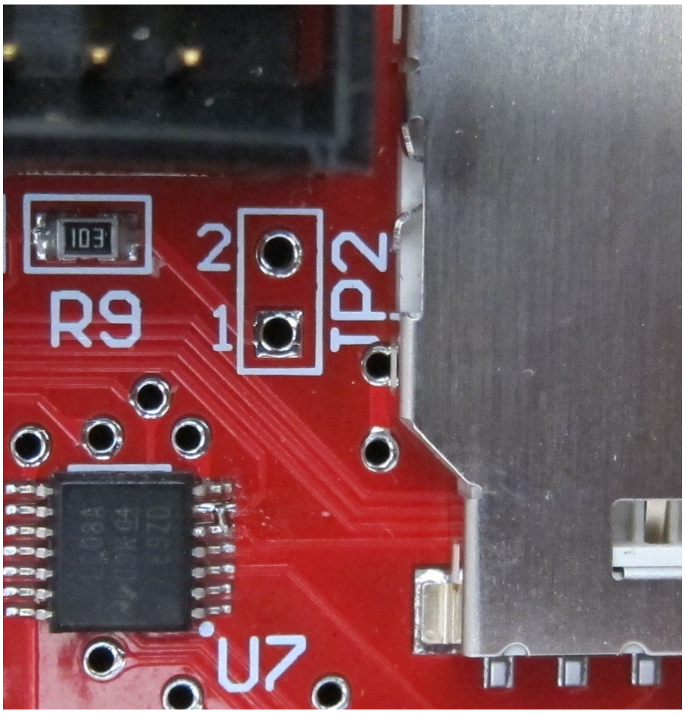



*Illustration 54: Main board JP2 jumper*

| | |
|---|---|
| Link 1-2 | Connect +5v to P1 pin 2 |

JP1 should never be linked for normal operation.

### 3.3.3 Test points

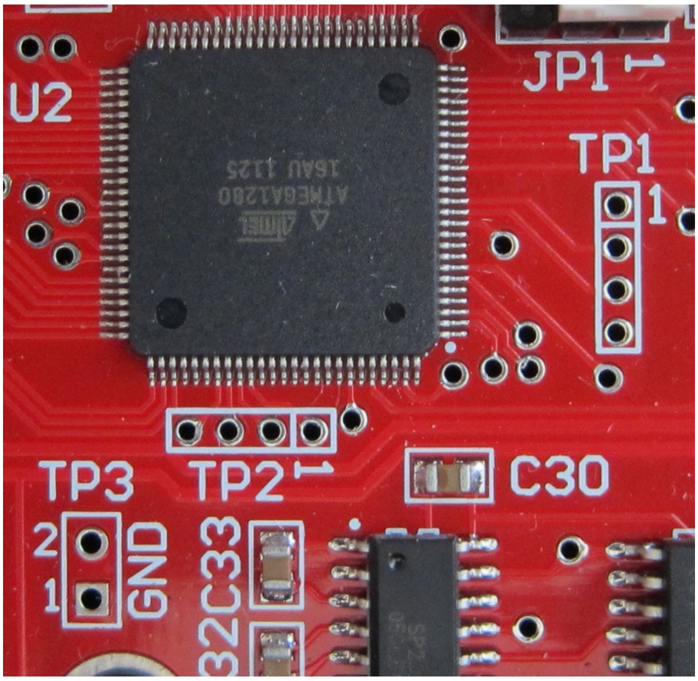


*Illustration 55: Main board test points*

TP1 – Serial debugger, not used in release firmware.

| | |
|---|---|
| 1 | +5v |
| 2 | Serial debug Tx, TTL levels |
| 3 | Serial debug Rx, TTL levels |
| 4 | Gnd |

TP2 – Spare IO, not used in release firmware.

| | |
|---|---|
| 1 | PK4 |
| 2 | PK5 |
| 3 | PK6 |
| 4 | PK7 |

TP3 – Ground points

| | |
|---|---|
| 1 | Gnd |
| 2 | Gnd |

## *3.4 Digital IO expansion board*

The Digital IO expansion board provides 16 opto-isolated inputs and 16 relay contact outputs. There are two boards in a standard Serial IO MKII, providing 32 inputs and 64 outputs in total.

### 3.4.1 Connectors

P3 – Power connector

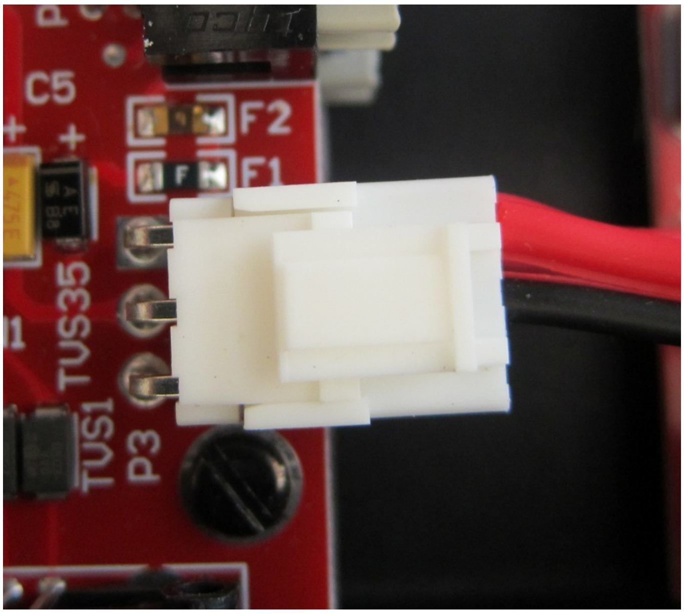


*Illustration 56: Digital IO board P3 connector*

| | |
|---|---|
| 1 | +5V supply input, fully regulated |
| 2 | Gnd |
| 3 | Not connected |

P4 – Data connector

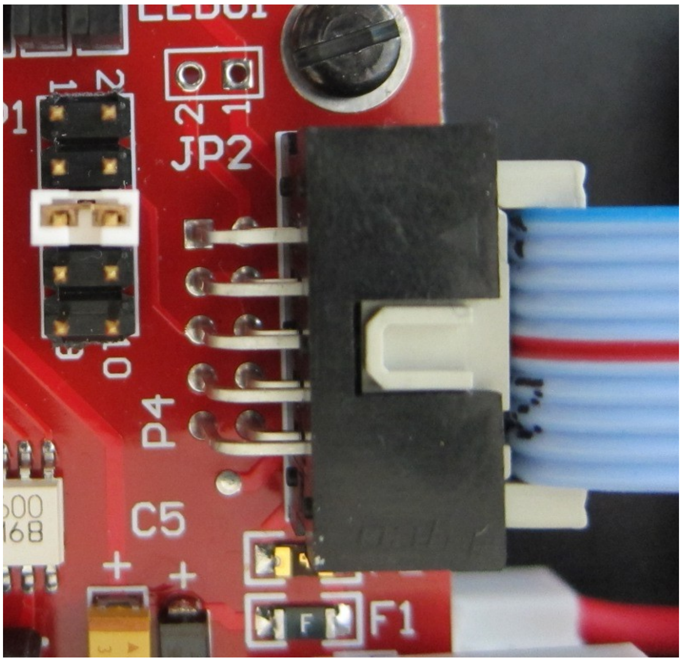



*Illustration 57: Digital IO board P4 connector*

| | | | |
|---|---|---|---|
| 1 | SPI MOSI | 2 | +5v if JP2 linked |
| 3 | Gnd | 4 | Gnd |
| 5 | SPI ExtSS1 / Reset | 6 | SPI ExtSS4 |
| 7 | SPI SCK | 8 | SPI ExtSS3 |
| 9 | SPI MISO | 10 | SPI ExtSS2 |

### 3.4.2 Jumpers

JP1 – Slave select / ISP

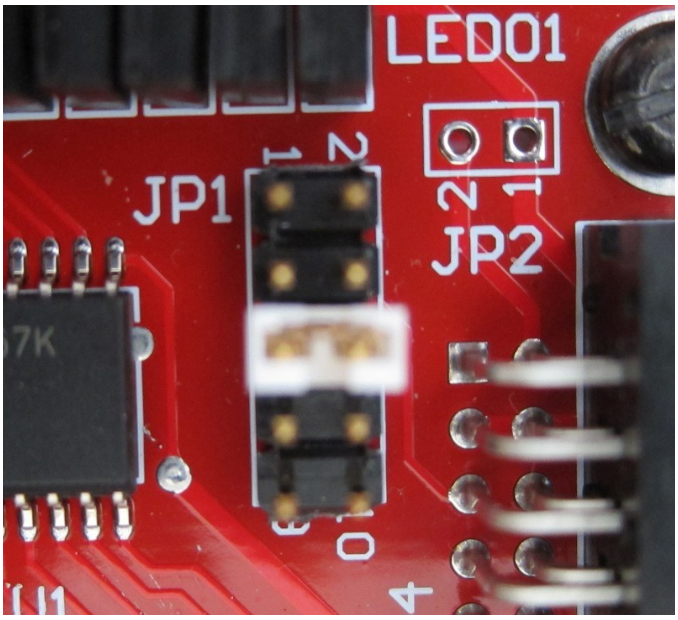



*Illustration 58: Digital IO board JP1 jumper*

| | |
|---|---|
| Link 1-2 | Factory programming only |
| Link 3-4 | Slave select 1, eg Inputs/Outputs A |
| Link 5-6 | Slave select 2, eg Inputs/Outputs B |
| Link 7-8 | Slave select 3, eg Inputs/Outputs C |
| Link 9-10 | Slave select 4, eg Inputs/Outputs D |

For normal operation, only one link should be made at a time. Linking 1-2 is for factory programming only.

Depending where the card is installed in the Serial IO MKII effects which link should be made. The board installed nearest to the base of the enclosure should use Slave select 1, the next board up should use Slave select 2, etc.

JP2 – ISP power

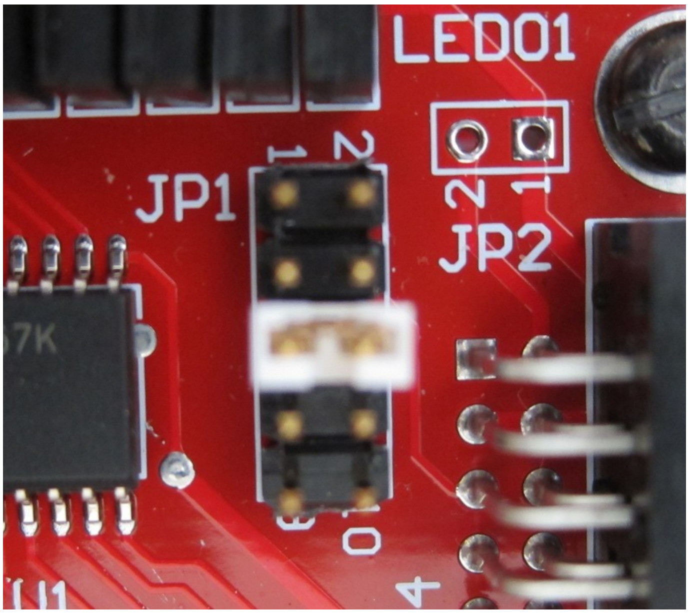


*Illustration 59: Digital IO board JP2 jumper*

| | |
|---|---|
| Link 1-2 | Connect +5v to P4 pin 2 |

JP2 should never be linked for normal operation.

JP3 – Inputs connector configurable pins

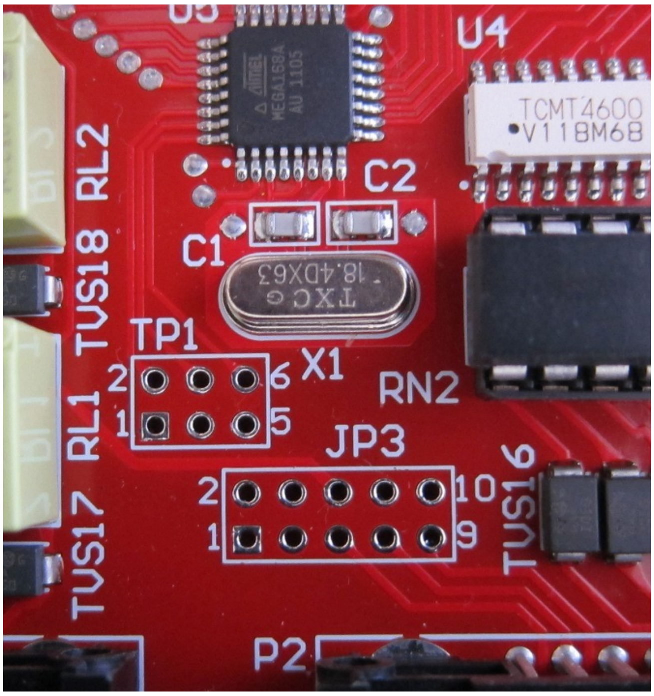



*Illustration 60: Digital IO board JP3 jumper*

JP3 allows for custom configuration of some of the spare pins on the output connector P2 as follows:

| | | | |
|---|---|---|---|
| 1 | Wired to P2 pin 40 | 2 | Gnd |
| 3 | Wired to P2 pin 39 | 4 | Gnd |
| 5 | Wired to P2 pin 38 | 6 | +5v |
| 7 | Wired to P2 pin 37 | 8 | +5v |
| 9 | Wired to P2 pin 36 | 10 | +5v |

### 3.4.3 Test points

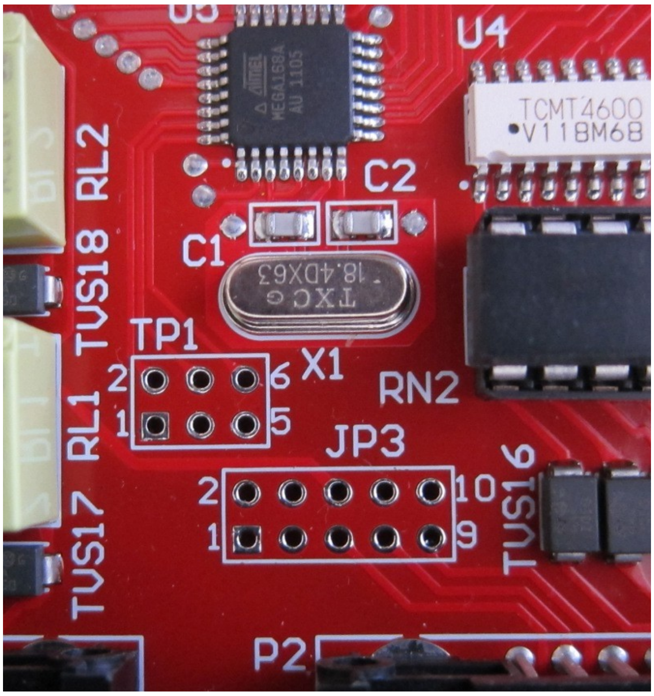



*Illustration 61: Digital IO board TP3 test point*

TP1 – Serial debugger, not used in release firmware.

| | | | |
|---|---|---|---|
| 1 | Gnd | 2 | Serial debug Rx, TTL levels |
| 3 | Gnd | 4 | Serial debug Tx, TTL levels |
| 5 | Gnd | 6 | Spare IO, PD2 |

### *3.5 Analogue/Voice input expansion board*

*The Analogue/Voice IO expansion board provides four 12 bit analogue to digital converter input channels, which are individually electrically isolated. Each channel has two inputs, one which can measure voltage between 0v to 44v with approximately 10mV resolution, and the other can measure current between 0mA to 20 mA with approximately 5 uA resolution.*

*The board also has 4 opto-isolator outputs for small signal applications only, and 4 relay outputs with both NO & NC pins available.*

*There are two digital IO boards in a standard Serial IO MKII, either or both of them can optionally be replaced by analogue IO boards.*

#### 3.5.1 Connectors

P1 – Data connector

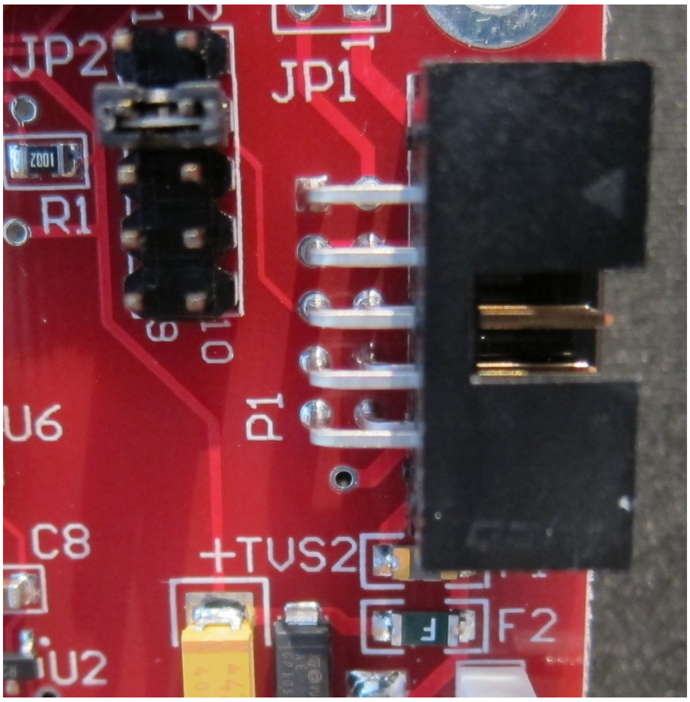


*Illustration 62: Analogue IO board P1 connector*

| | | | |
|---|---|---|---|
| 1 | SPI MOSI | 2 | +5v if JP2 linked |
| 3 | Gnd | 4 | Gnd |
| 5 | SPI ExtSS1 / Reset | 6 | SPI ExtSS4 |

| | | | |
|---|---|---|---|
| 7 | SPI SCK | 8 | SPI ExtSS3 |
| 9 | SPI MISO | 10 | SPI ExtSS2 |

P2 – Power connector

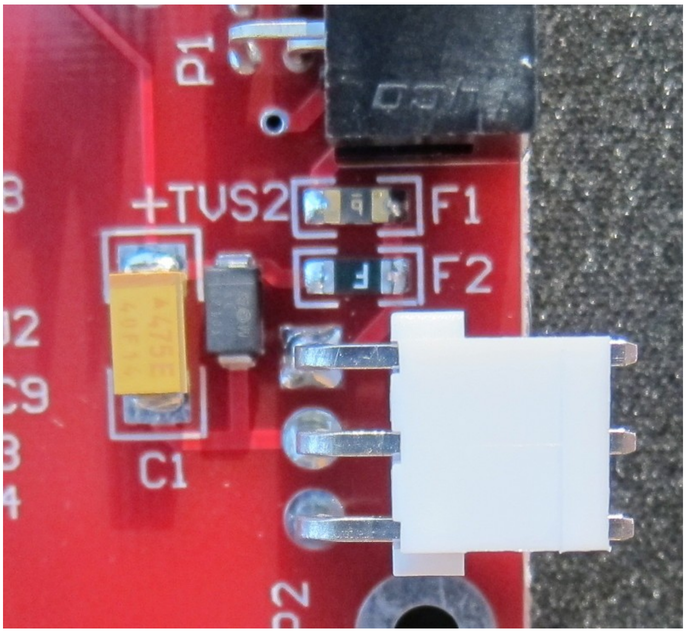



*Illustration 63: Analogue IO board P2 connector*

| | |
|---|---|
| 1 | +5V supply input, fully regulated |
| 2 | Gnd |
| 3 | Not connected |

### 3.5.2 Jumpers

JP1 – ISP power

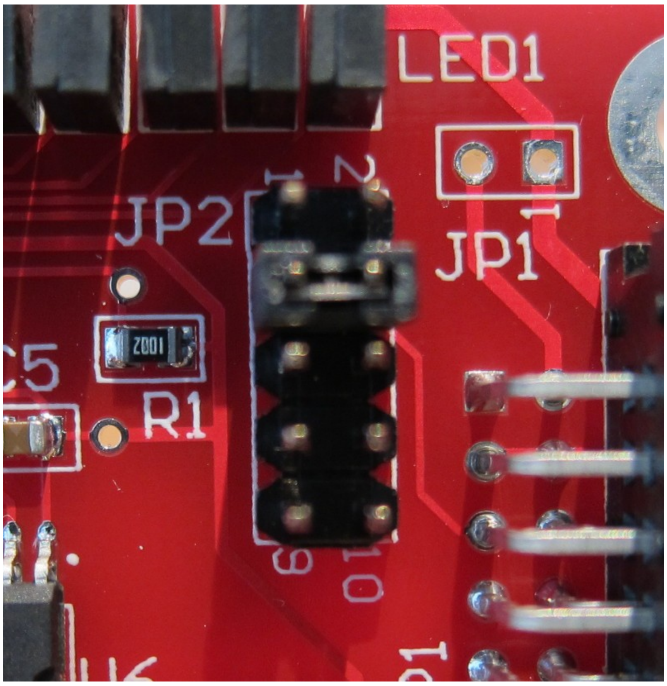



*Illustration 64: Analogue IO board JP1 jumper*

| | |
|---|---|
| Link 1-2 | Connect +5v to P4 pin 2 |

*JP1 should never be linked for normal operation.*

JP2 – Slave select / ISP

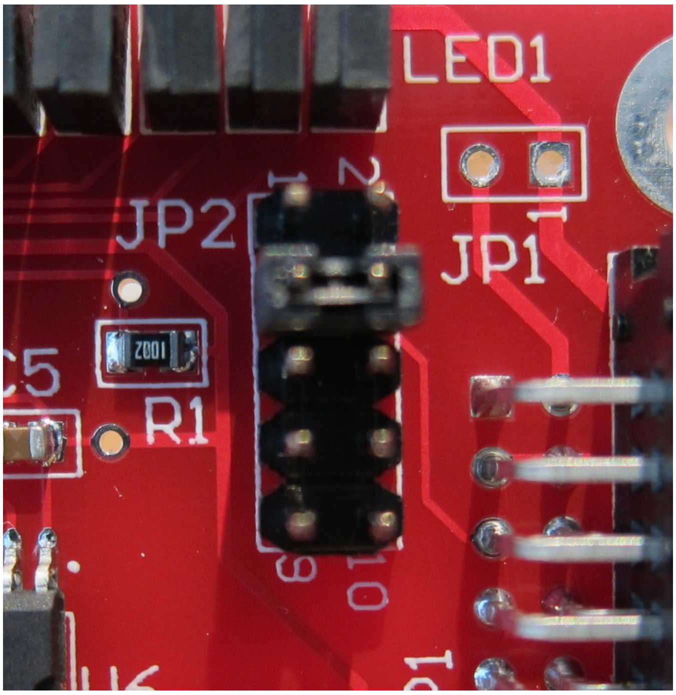



*Illustration 65: Analogue IO board JP2 jumper*

| | |
|---|---|
| Link 1-2 | Factory programming only |
| Link 3-4 | Slave select 1, eg Inputs/Outputs A |
| Link 5-6 | Slave select 2, eg Inputs/Outputs B |
| Link 7-8 | Slave select 3, eg Inputs/Outputs C |
| Link 9-10 | Slave select 4, eg Inputs/Outputs D |

For normal operation, only one link should be made at a time. Linking 1-2 is for factory programming only.

Depending where the card is installed in the Serial IO MKII effects which link should be made. The board installed nearest to the base of the enclosure should use Slave select 1, the next board up should use Slave select 2, etc.

JP3 – Serial data

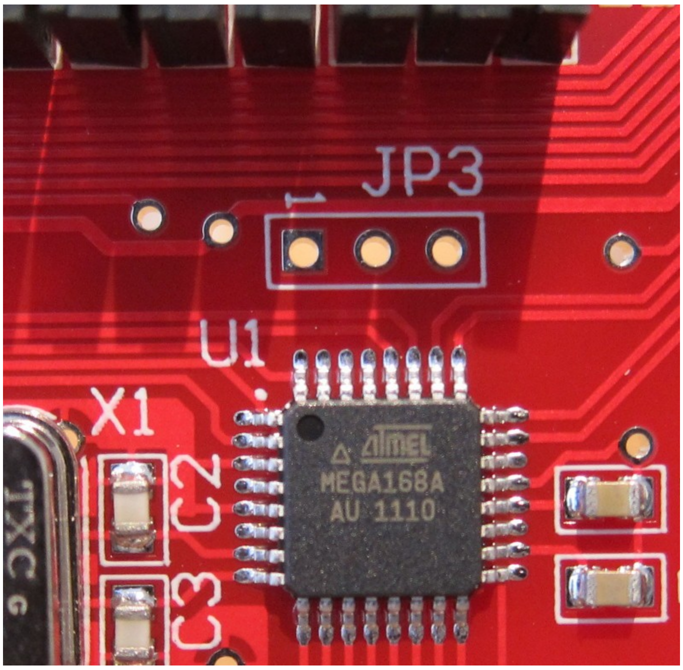



*Illustration 66: Analogue IO board JP3 jumper*

| | |
|---|---|
| Pin 1 | Tx data at TTL levels |
| Pin 2 | Rx data at TTL levels |
| Pin 3 | Ground / 0v |

*JP3 is for factory use only.*

## *3.6 Protocol overview*

### 3.6.1 UCE protocol

The UCE protocol is defined by Harmon Industries. The following is only a summary of the protocol. For full details, contact Harmon Industries.

The protocol is ASCII packet based with each packet starting with a new line character (0A hex) followed by a message type character and a terminating carriage return character (0D hex). Two packets, the controls and indications, contain data bytes and a CRC after the message type character, all presented in ASCII .

| Message type | Contents |
|---|---|
| ACK | <lf>+<cr> |
| NACK | <lf>-<cr> |
| Available | <lf>A<cr> |
| Busy | <lf>B<cr> |
| Recall | <lf>R<cr> |
| Controls | <lf>C004nnnnnnnncccc<cr> |
| Indications | <lf>I004nnnnnnnncccc<cr> |

Where nnnnnnnn is up to 64 bits of controls or indications and cccc is a check sum.

The sequence of a typical message transaction is as follows:

| Master | Slave |
|---|---|
| Send Controls | |
| | Send ACK |
| Request Indications | |
| | Send Indications |
| Send ACK | |
| Send Available | |

At any time, the slave can arbitrarily send an indications packet to the master. Typically this will be as a result of a change in state.

### 3.6.2 S2 protocol

The S2 protocol is defined by Westinghouse. The following is only a summary of the protocol. For full details, contact Westinghouse.

The S2 protocol is based on the SDLC (serial data link control) standard, which is a bit based synchronous protocol. There is only 1 packet type:

| **Byte number** | | **Description** |
|---|---|---|
| 1 | Bit 7 | Direction bit<br>(1 for controls, 0 for indications) |
| | Bit 6 | System bit (not used) |
| | Bits 5 - 0 | 6 bit address (bit 0 is LSB) |
| 2 to *n* | | Control or Indication bits |
| *n*+1<br>*n*+2 | | CRC |

Each packet is preceded by, and followed by the SDLC flag sequence (7E hex), which may be continuously transmitted while the data line is idle. To ensure data transparency, if a sequence of 5 consecutive 1's is transmitted and it is not part of a flag sequence, a 0 is inserted to avoid generating false flag sequences.

In the some implementations of the S2 protocol, non-return to zero (NRZi) encoding is used, whereby a 0 is shown by a change in state of the data line and a 1 is shown by no change in state. Also in some implementations an additional byte, typically all 0s, is inserted after the 1st byte in the packet.

The sequence of a typical message transaction is as follows:

| **Master** | **Slave** |
|---|---|
| Send Controls | |
| | Respond with Indications |

The master will continuously send controls with a define wait period between each packet transmission, known as the inter scan delay. This is to allow the slave to respond in simplex systems.

### 3.6.3 Genisys protocol

The Genisys protocol is defined by a number of different companies, not all implementations are compatible. The following is only a summary of the protocol.

A packet consists of the following data sequence:

| Control Character | Station Address | Data Bytes | Security Checksum | Termination Character |
|---|---|---|---|---|

The control character defines the packet type:

| | |
|---|---|
| F0 hex | Escape |
| F1 hex | Acknowledge master |
| F2 hex | Indication data |
| F3 hex | Control data check back |
| F4 hex | Reserved |
| F5 hex | Reserved |
| F6 hex | Termination character |
| F7 hex | Reserved |
| F8 hex | Reserved |
| F9 hex | Common control data |
| FA hex | Acknowledge indication data and poll |
| FB hex | Poll |
| FC hex | Control data |
| FD hex | Recall |
| FE hex | Execute check back controls |
| FF hex | Reserved |

The station address is the station the packet is either addressed to, or sent from.

All data bytes that are F0 hex and higher are modified so that the framing of the packet is not effected. The data byte is changed from F*n* hex to two bytes of F0 0*n* hex. The data bytes are arranged in pairs, the first byte is a memory address and the second byte is the data for/from that memory address.

The checksum uses the standard CRC-16 polynomial.

### 3.6.4 Host protocol

The Host protocol is defined by EDMI. The following is only a summary of the protocol. For full details, contact EDMI.

This protocol operates over an RS232 serial link at 19200 baud with 7 data bits, odd parity and one stop bit.

There are 2 packet types:

| Message type | Contents |
|---|---|
| Controls request message | <ACK><LoopNumber><CR> |
| Controls/Indications message | <ESC><Address><ControlField><Data><LoopNumber><CR> |

Where the “ Address” byte is defined as:

| Value | Bits |
|---|---|
| Direction (1 for controls, 0 for indications) | 7 (MSB) |
| System bit (normally 0) | 6 |
| Field station address | 5 - 1 |

The system control field is usually 0.

The sequence of a typical message transaction is as follows:

| Master | Slave |
|---|---|
| | Send controls request message |
| Responds with controls message | |
| | Send indications message |

### 3.6.5 NTP client protocol

The Network Time Protocol (NTP) is a protocol for synchronising the clocks of computer systems over packet-switched, variable-latency data networks. The protocol documentation is publicly available, so is not covered here. See http://www.ntp.org/ for more information.

It is normal practice that a server will listen on UDP port 123, so you should configure the Serial IO MKII as a UDP client connecting to the server IP on port 123.

### 3.6.6 Control protocol

The control protocol is used to interact with the Serial IO MKII itself, for example to write configuration data to the device.

The configuration application expects the Control protocol to be available on TCP port 23. If you delete or change the Control protocol node so that it is no longer listening on TCP port 23 you will no longer be able to reconfigure the unit. You will have to do a factory reset to recover the unit.

It is also possible to use the Control protocol for monitoring and debugging. You can use any standard Telnet application to connect to the unit in the standard Telnet port of TCP 23. Once connected you will see messages displayed and be able to enter a few basic commands.

#### 3.6.6.1 Control protocol messages

The Control protocol messages are the same messages that are written to the SD card when in use. They give some indication of what the Serial IO MKII is doing.

You can vary the message level with the “level” command below to see more of less messages. The message levels are:

| | |
|---|---|
| 0 | None |
| 1 | Important |
| 2 | Normal |
| 3 | Detailed |
| 4 | Debugging |
| 5 | Everything |

It is strongly recommended that levels 4 or 5 only be used on the test bench as they are very processor intensive and can effect the normal operation of the unit.

### 3.6.6.2 Control protocol commands

The Control protocol supports some basic commands:

| | |
|---|---|
| clear count | Clear packet counts |
| cls | Clear screen |
| firmware update | Reboot and stay in boot loader |
| id | Unit ID |
| level *n* | Set message level to *n* |
| mem | Peak memory usage |
| reset/reboot | Reboot |
| status | Show node/slave status |
| time | Show current date & time |
| ver | Protocol version |

There are also a number of command used by the configuration application. These contain binary data and can not be entered on a Telnet terminal. The content of the command is beyond the scope of this manual.

| | |
|---|---|
| get config | Get the device configuration |
| set config | Set the device configuration |
| set time | Set the device real time clock |

### *3.7 Boot loader and firmware updates*

When the Serial IO MKII first turns on, it will be in boot loader mode for approximately 5 seconds. This is shown by the front panel LEDs flashing, bottom row first, then top row.

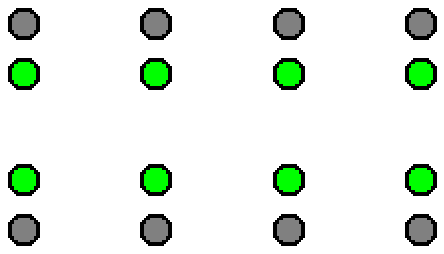


*Illustration 67: Boot loader LEDs, normal boot*

In this mode a firmware upload can be initiated via either the Ethernet port or the COM2 serial port.

The unit will have the IP address that it has been configured with. If you do not know the IP address, you can hold down the “Factory Reset” button for approximately 2 seconds at power up and the LEDs will flash in a different pattern.

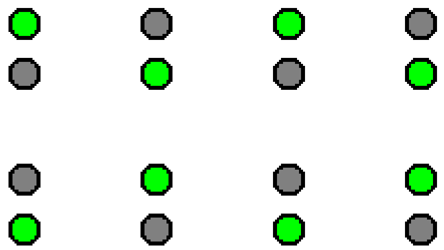


*Illustration 68: Boot loader LEDs, Ethernet reset boot*

The stored IP data will be reset to defaults as follows:

| | |
|---|---|
| IP Address | 192.168.1.1 |
| Subnet mask | 255.255.255.0 |
| Gateway IP | 192.168.1.254 |
| Bootloader port | UDP 6003 |

If the LEDs flash in the following sequence, the device has detected an internal hardware error and is not continuing operation until the error is fixed. Try toggling power to the device, if the error continues, it will need to be returned to the factory for repair.

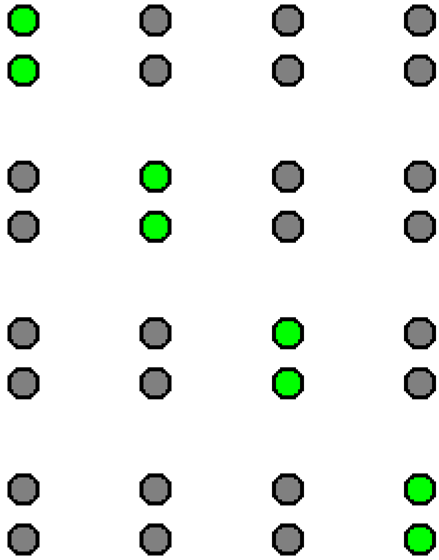


*Illustration 69: Boot loader LEDs, Internal hardware error*

The firmware update procedure is currently beyond the scope of this document.

# Appendix A – Railmaster compatibility

The  Serial IO MKII can be operated in a mode that makes it compatible with systems running EDMI Railmaster Compact units.

## *Connections*

The digital IO connections are directly compatible with the digital IO connections on a Railmaster Compact, the ribbon cables can simply be swapped over.

The Serial IO MKII VF connection has a different connector type to a Railmaster, and so needs an adapter cable. For typical 4 wire, full duplex connections the following adapter cable is sufficient.

| **Serial IO MKII VF RJ45 connector** | | **15 way male D-Type connector** | |
|---|---|---|---|
| 1 | Rx+ | 1 | Rx+ |
| 2 | Rx- | 9 | Rx- |
| 3 | Squelch | | |
| 4 | Tx+ | 8 | Tx+ |
| 5 | Tx- | 15 | Tx- |
| 6 | Press to talk | | |
| 7 | Ground | | |
| 8 | Ground | | |

Note: The Ground pins are not electrically isolated, they are the same as power ground.

When operating in place of a Railmaster in Front End Processor (FEP) mode, the “Host Slave” protocol should be used on any of COM1 – COM3 to replicate the Railmaster “Office” protocol. Standard RS232 serial cables can be used.

### *Configuration - slave*

The following configurations are suggested for typical slave applications.

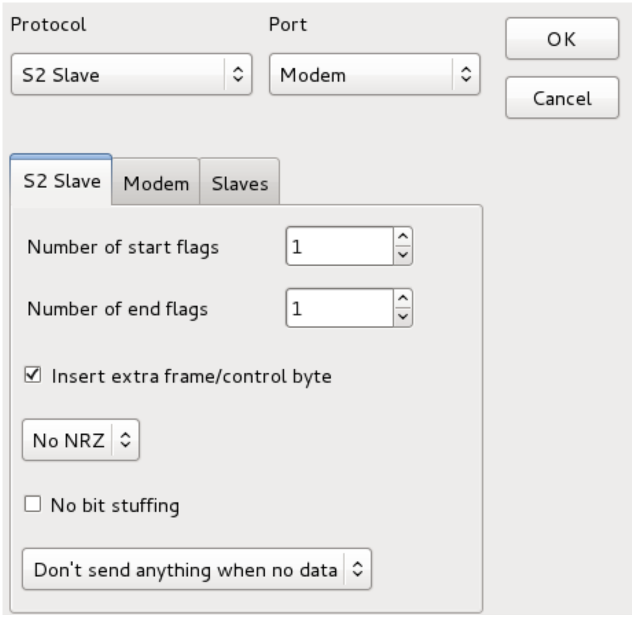



*Illustration 70: Config S2 slave as Railmaster*

Note: "No NRZ" may need to be set to "NRZi" depending on the Railmaster configuration.

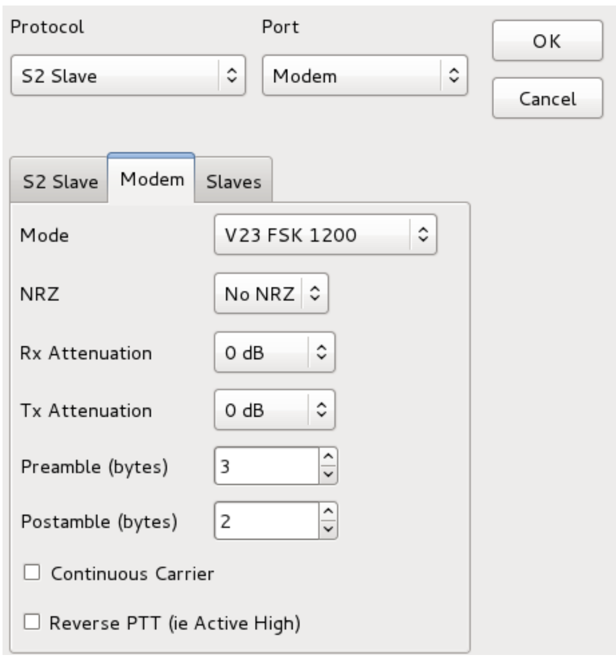



*Illustration 71: Config Modem as Railmaster*

Note: "Attenuation", "Preamble" and "Postamble" may need to be adjusted in some environments.

### *Configuration - master*

The following configurations are suggested for typical master applications.

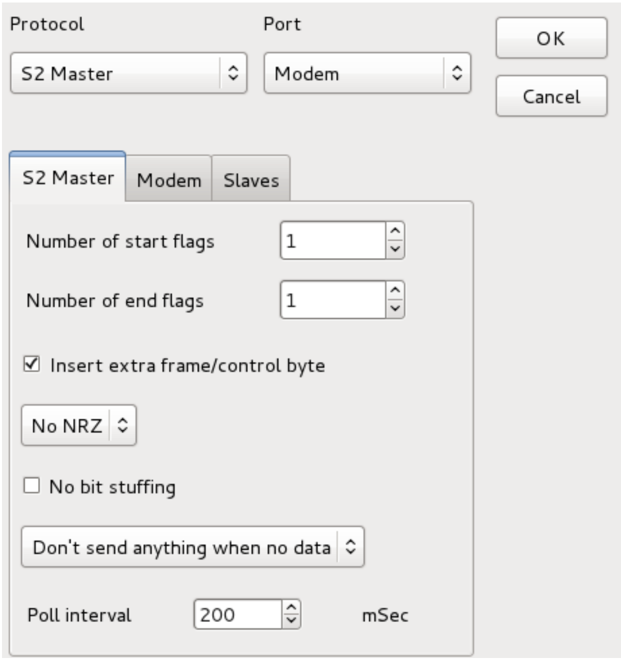



*Illustration 72: Config S2 master as Railmaster*

Note: “No NRZ” may need to be set to “NRZi” depending on the Railmaster configuration.

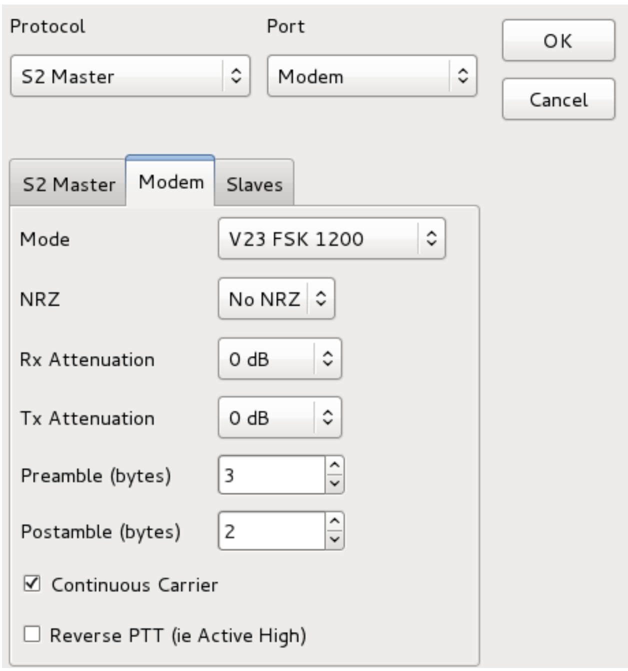



*Illustration 73: Config Modem as Railmaster*

Note: “Attenuation”, “Preamble” and “Postamble” may need to be adjusted in some environments.

# Appendix B - Specifications

| | |
|---|---|
| Supply voltage | 8 to 28 volts DC |
| Supply power | All output relays inactive: 1.5 watts, approximately 125 mA at 12v<br>All output relays active: 12 watts, approximately 1 A at 12v |
| Standard connections | Power supply<br>Modem VF, 600 ohm interface, 1200/2400/4800 baud<br>3 x serial port, sync/async, RS232 levels<br>Ethernet, 10/100 Mb/s, full duplex, auto-crossover<br>SD card slot, memory cards up to 32 GB<br>32 x digital inputs, opto-isolated<br>32 x digital outputs, relay contact closure |
| Optional connections | 4 x 12 bit isolated ADC, each capable of measuring<br>voltage (0-44v) and current (0-20mA)<br>4 x digital output, low current transistor, opto-isolated<br>4 x digital output, relay, NO, NC & common contacts available |
| Main board | CPU: Atmel ATMega1280<br>18.432 MHz, 256k flash, 8k RAM, 4k EEPROM<br>RAM: Cypress CY62256N , 32k<br>Ethernet: WIZnet W5300<br>Modem: CML CMX868<br>Real time clock: STMicroelectronics M41T81<br>PSU: RECOM R-745.0P |
| Digital IO board | CPU: Atmel ATMega168<br>18.432 MHz, 16k flash, 1k RAM, 512 bytes EEPROM |
| Analogue IO board | CPU: Atmel ATMega168<br>18.432 MHz, 16k flash, 1k RAM, 512 bytes EEPROM |
| Voice/Audio IO board | CPU: Atmel ATMega168<br>18.432 MHz, 16k flash, 1k RAM, 512 bytes EEPROM<br>CPU: Atmel ATMega324<br>18.432 MHz, 32k flash, 2k RAM, 1024 bytes EEPROM<br>Audio: VLSI VS1053B |
| Temperature range | -40°C to +85°C, non-condensing |
| Dimensions | 482.6 (W) x 43.7 (H) x 75 (D)<br><br>19 inch rack mountable<br>1 rack unit (RU) high<br>75mm deep |
| Weight | 2 KG |

# Appendix C - Contact information


| | |
|---|---|
| Company name: | **HASCOM International Pty Ltd** |
| ABN: | **18 150 118 755** |
| Address: | **4 Lacrosse Rise**<br>**Sorrento**<br>**Western Australia  6020** |
| Phone: | **08 9415 1311** |
| Fax: | **08 9415 1312** |
| Email: | **hascom@hascom.com.au** |
| Website: | **http://www.hascom.com.au** |