Card to Baseboard Interface

# Definitions

|  |  |
| --- | --- |
| SDR | Processing element capable of demodulating signals at the IF frequency |
| Card | General term for anything that plugs into the baseboard that is not an SDR |
| Transverter | Card capable of changing the carrier frequency of a modulated signal. This may capable of RX, TX or both. |
| PNT Card | TBD |
| Switch Card | TBD |
| PSU Card | TBD |
| RX | Receiving / Receiver |
| TX | Transmitting / Transmitter |

## Mechanical requirements

### Card (non-SDR)

This section applies to any card that is not an SDR.

Connection to the baseboard shall be implemented using DIN41612 type R connectors.

Connectors shall have 48 pins in 3 columns.

Suggested parts:

Baseboard: Amphenol FCI 86093487614755ELF

Card: Amphenol FCI 86093488613755E1LF

The connector shall be placed as follows:

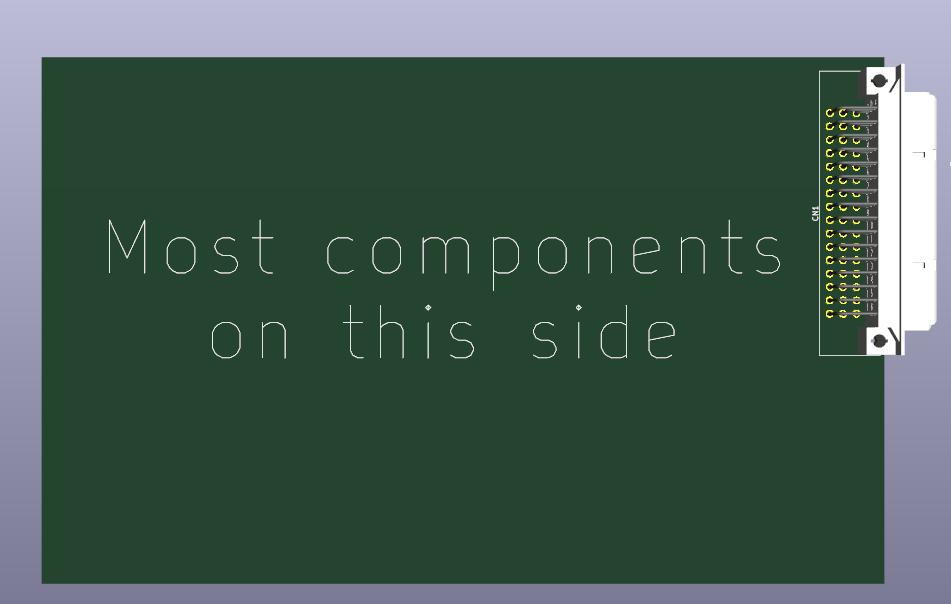


Figure 1 – Example PCB showing connector location

When looking from the component side of the card, a1 should be in the top right corner of the connector, dimensioned as per the Eurocard standard. This shown in Figure 2.

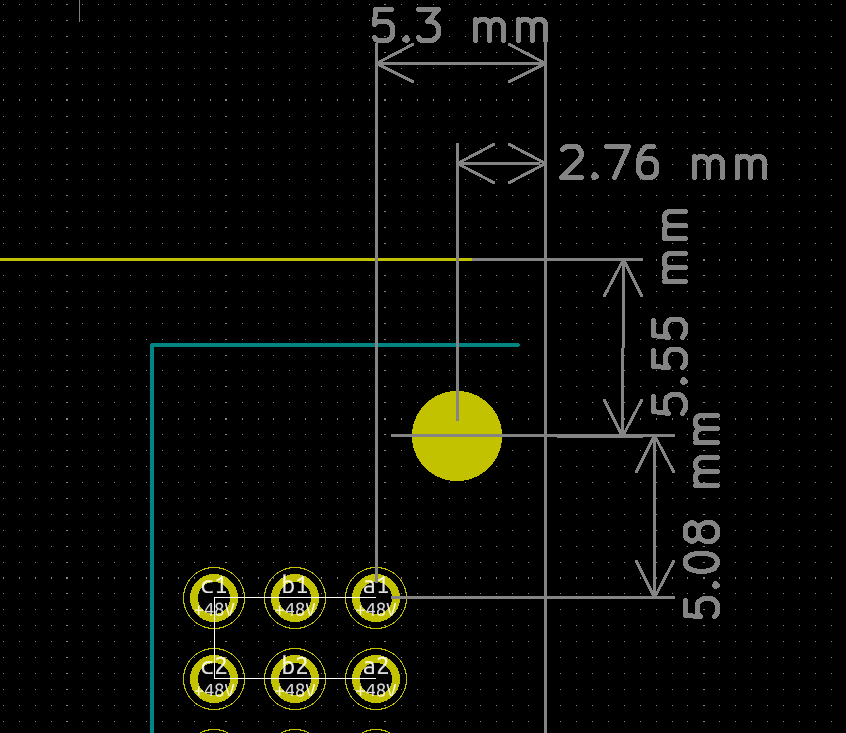


Figure 2 – Eurocard connector positioning requirements

## Pinout

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | a | b | c |  |  |  |
| 1 | +48V | +48V | +48V | 1 |  |  |
| 2 | +48V | +48V | +48V | 2 |  | Power |
| 3 | **0V** | **0V** | **0V** | 3 |  | **Ground** |
| 4 | +24V | +24V | +24V | 4 |  | RS485 |
| 5 | +24V | +24V | +24V | 5 |  | PSU EEPROM |
| 6 | **0V** | **0V** | **0V** | 6 |  | Address EEPROM |
| 7 | +12V | +12V | +12V | 7 |  | Spare |
| 8 | +12V | +12V | +12V | 8 |  | GPIO |
| 9 | **0V** | **0V** | **0V** | 9 |  |  |
| 10 | **0V** | **0V** | **0V** | 10 |  |  |
| 11 |  |  |  | 11 |  |  |
| 12 |  |  | TX\_EN | 12 |  |  |
| 13 | ADDR\_3V3 | ADDR\_SDA | RS485\_B | 13 |  |  |
| 14 | ADDR\_WP | ADDR\_SCL | RS485\_A | 14 |  |  |
| 15 | PSU\_3V3 | PSU\_SDA | PSU\_SCL | 15 |  |  |
| 16 | PSU\_ADDR2 | PSU\_ADDR1 | PSU\_ADDR0 | 16 |  |  |
|  | a | b | c |  |  |  |

Figure 3 – Non-SDR Card to Baseboard connector (using standard DIN41612 pin numbering)

### Description

|  |  |  |
| --- | --- | --- |
| Name | Direction | Description |
| +48V | Power Input | +48V input |
| 0V | Power Input | 0V input |
| +24V | Power Input | +24V input (may also be +28V depending on negotiation) |
| +12V | Power Input | +12V input |
| TX\_EN | Input | 3V3 logic signal. Card may only transmit if this is high. This must rely on an external logic high before TX is enabled i.e. the card should not have a pullup resistor on this pin |
| ADDR\_3V3 | Power Output | 3V3 output to supply the EEPROM on the baseboard. Capable of |
| ADDR\_SDA | Bidirectional | Data line for I2C bus to access the address EEPROM on the baseboard. 3V3 signalling, card is bus master and should provide pullup resistors if required. |
| ADDR\_WP | Input | Connected to the Write Protect (WP) pin on the address EEPROM |
| ADDR\_SCL | Output | Clock line for I2C bus to access the address EEPROM on the baseboard. 3V3 signalling, card is bus master should provide pullup resistors if required. |
| RS485\_A | Bidirectional | “A” signal of RS485 bus. 3V3, half-duplex signalling |
| RS485\_B | Bidirectional | “B” signal of RS485 bus. 3V3, half-duplex signalling |
| PSU\_3V3 | Power Input | 3V3 that must only be used for the EEPROM contains power supply requirements for the card (Power EEPROM) |
| PSU\_SDA | Bidirectional | Data line for I2C bus on the baseboard to access the power EEPROM. 3V3 signalling, card is slave and should not have pullup resistors connected to the I2C bus. Only the power EEPROM should be connected to this bus. |
| PSU\_SCL | Input | Clock line for I2C bus on the baseboard to access the power EEPROM. 3V3 signalling, card is slave and should not have pullup resistors connected to the I2C bus. Only the power EEPROM should be connected to this bus. |
| PSU\_ADDR0 | Input | Directly connected to ADDR0 pin on the Power EEPROM to allow baseboard to set Power EEPROM I2C address. |
| PSU\_ADDR1 | Input | Directly connected to ADDR0 pin on the Power EEPROM to allow baseboard to set Power EEPROM I2C address. |
| PSU\_ADDR2 | Input | Directly connected to ADDR0 pin on the Power EEPROM to allow baseboard to set Power EEPROM I2C address. |

Table 1 – Non-SDR Card to Baseboard connector pinout description

## Card – SDR

TBD

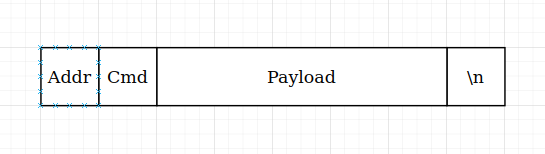
# RS485 Requirements

## Physical layer

Signalling is done with a 115200 baud 3V3 half-duplex RS485 signal. It is the responsibility of the SDR to provide pullup / pulldown resistors on the A/B pins to ensure that the bus is in a '1' state when the bus is not being driven. The RS485 termination on the master side shall be provided by the SDR. The termination at end of the bus furthest away from the master shall be provided by the baseboard. Transverters shall not have any termination resistors fitted.

## Message format

All communications shall have the following format



 Addr: 1 byte containing the destination address in raw binary

Cmd: 1 byte containing the command byte

Payload: Any number of bytes (including zero). Meaning is dependent on Cmd byte

\n: 1 byte set to 0x13 (Newline character)

## General Communication rules

Transverters shall only communicate in response to a request from the SDR.

SDR shall always be at address 0 – this may be assumed by any software running on the SDR.

Transverter shall not communicate with each other on the RS485 bus. I.e. all transmission from a transverter should be sent to address 0

All non-zero size payloads must be utf-8 encoded JSON.

### NamedValue type

Many of the measurements returned in the protocol are of type “NamedValue”. This has two compulsory elements.

{

“name”: “Dave”,

“value”: 23.0

}

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| name | String | Human Readable Measurement Name |
| value | Float | Measured Value |

Table 2: Fields in NamedValue object

### MAC Addresses

MAC addresses are used throughout as unique identifiers. Any MAC address should be a string with the MAC address formatted as 6 pairs of hexadecimal characters, separated by colons “:”. All letters shall be uppercase.

## Discovery Messages (Cmd 'D')

The contents of a discovery message should be the static information about a transverter which is not going to change during operation.

### SDR -> Card

This command should be sent with no payload to request a transverter's discovery information. Non-empty payloads are invalid.

### Transverter -> SDR

The response should include the following payload containing all of the following fields and no more. Example values are shown:

 {

"type": "Transverter"

"name": "Super Awesome Transverter",

"loFreq": 10,

"loAdd": True,

"minRfFreq": 10,

"maxRfFreq": 20,

"minPower": 10,

"maxPower": 20,

“supportsRX”: True

"supportsTX": True

}

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| type | String | Always "Transverter" for a transverter |
| name | String | Human Readable Transverter Name |
| loFreq | Int | (First two letters are lower case L and O) Local Oscillator frequency in Hz |
| loAdd | Boolean | (First two letters are lower case L and O) True if output frequency = LO + IF, False if output frequency = LO - IF |
| minRfFreq | Int | minimum RF freq of the transverter in Hz |
| maxRfFreq | Int | maximum RF frequency of the transverter in Hz |
| minPower | Int | minimum settable TX output power limit in dBm |
| maxPower | Int | maximum settable TX output power limit in dBm |
| supportsRx | Boolean | True if transverter supports RX else False. |
| supportsTx | Boolean | True if transverter supports TX else False. If this is False, the values for minPower and maxPower are irrelevant and can be set to anything |

Table 3: Fields for transverter discovery response

### Other -> SDR

TBD

## Status Messages (Cmd 'S')

The contents of a discovery message should be the static information about a card which is not going to change during operation. The response is dependant on the card type.

### SDR -> Card

This command should be sent with no payload to request a transverter's discovery information.

### Transverter -> SDR

The response should include the following payload containing all of the following fields and no more. Example values are shown:

{

“warnings”: {},

“errors”: {},

“temperatures”: {}

“state”: “RX”,

“rfPowerReadings”: {},

“dcPowerReadings”: {},

“controller”: “12:34:56:78:90:AB”

}

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| warnings | Array(Warning) | Array of warnings reported by the transverter. Can (and hopefully will be!) empty. |
| errors | Array(Error) | Array of errors reported by the transverter. Can (and hopefully will be!) empty. |
| temperatures | Array(NamedValue) | Array of temperatures read by the sensors on the transverter. |
| state | String / null | Indicates the operational state of the transverter. Allowed values are:  “rx”: Transverter is receiving  “tx”: Transverter is transmitting  “idle”: Transverter is not being used currently. “controller” should be null if this the case  “warmup”: Transverter is performing initial booting before being available e.g. PLL locking / OCXO warming up  “error”: Transverter has encountered error that means it requires the error manually clearing before continuing |
| rfPowerReadings | Array(NamedValue) | Contains RF power readings from various points in the transverter. Must contain a reading for “fwd” and “rev”. These readings can be null e.g. if the transverter doesn’t support TX |
| dcPowerReadings | Array(NamedValue) | Contains DC power readings from various points in the transverter. Can be empty. |
| controller | String / null | MAC address of the SDR controlling this transverter. If this transverter is not currently in use, this should be null |

Table 4: Field for transverter status response

### Other -> SDR

TBD