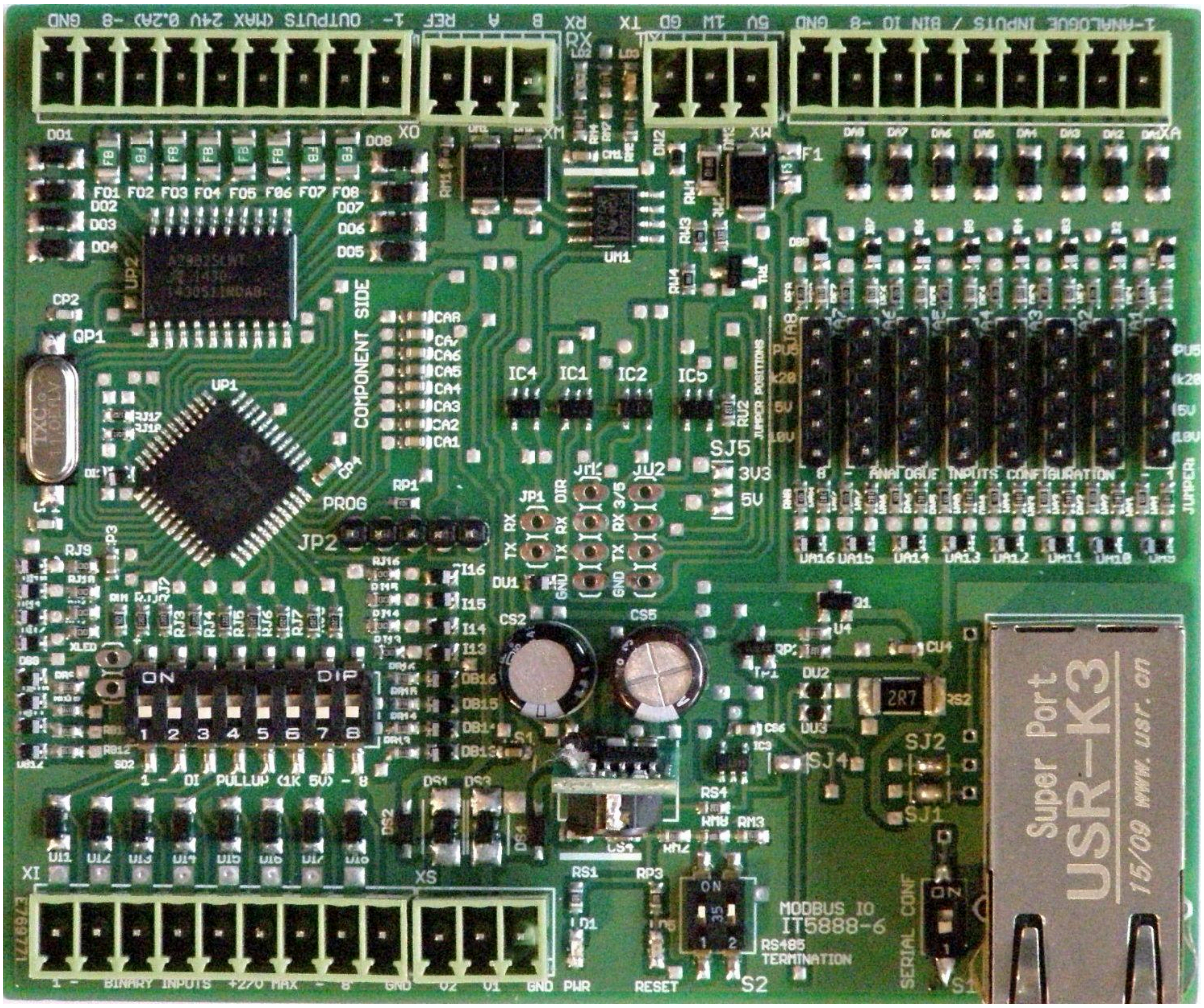
User Manual for Modbus-compatible I/O-module UF080808 and UF080808E



Manufacturer: *Uniflex Systems OÜ, www.uniflex.ee*

Table of Contents

1. Introduction

The I/O-module consists of a plastic DIN-rail mountable enclosure (106mm wide) and a PCB with connectors and electronics on it. To access the board, remove the bottom of the enclosure.

The input/output (I/O)-board described in this document is a slave device supporting Modbus-protocol, usable for interfacing with various analog or discrete field signals. The device is usable at home, office or in a light industrial environment, in various automation or data acquisition applications. For communication with the Modbus Master the board has one RS485 interface and one 3.3V (5V-tolerant) serial (UART) interface, both supporting communication speeds from 9600 to 115200 bps, none or even parity. The Ethernet interface is available with an optional Ethernet/serial converter installed.

The always supported communication protocol is ModbusRTU on the RS485 bus, available at connector XM. With the Ethernet interface module installed, another data channel supporting ModbusTCP becomes available. If the Ethernet module is not installed, the 3V3 (5V-tolerant) UART-port at the pin header JU2 becomes usable as the second communication channel, either for a master or wireless module oriented ModbusRTU communication.

The registers on the board can be accessed either from RS485 or UART (or Ethernet). A query made to one interface is always reflected on another, making the device usable as a ModbusTCP/RTU (or RTU/TCP) gateway.

There are 24 **protected discrete or analog I/O-channels** available on this I/O-board. Depending on the configuration, 8..16 discrete inputs, 8..16 discrete outputs and 0..8 analog inputs in various configurations can be used. Dallas 1-wire bus interface is available as well, supporting up to 18 DS18B20 temperature sensors.

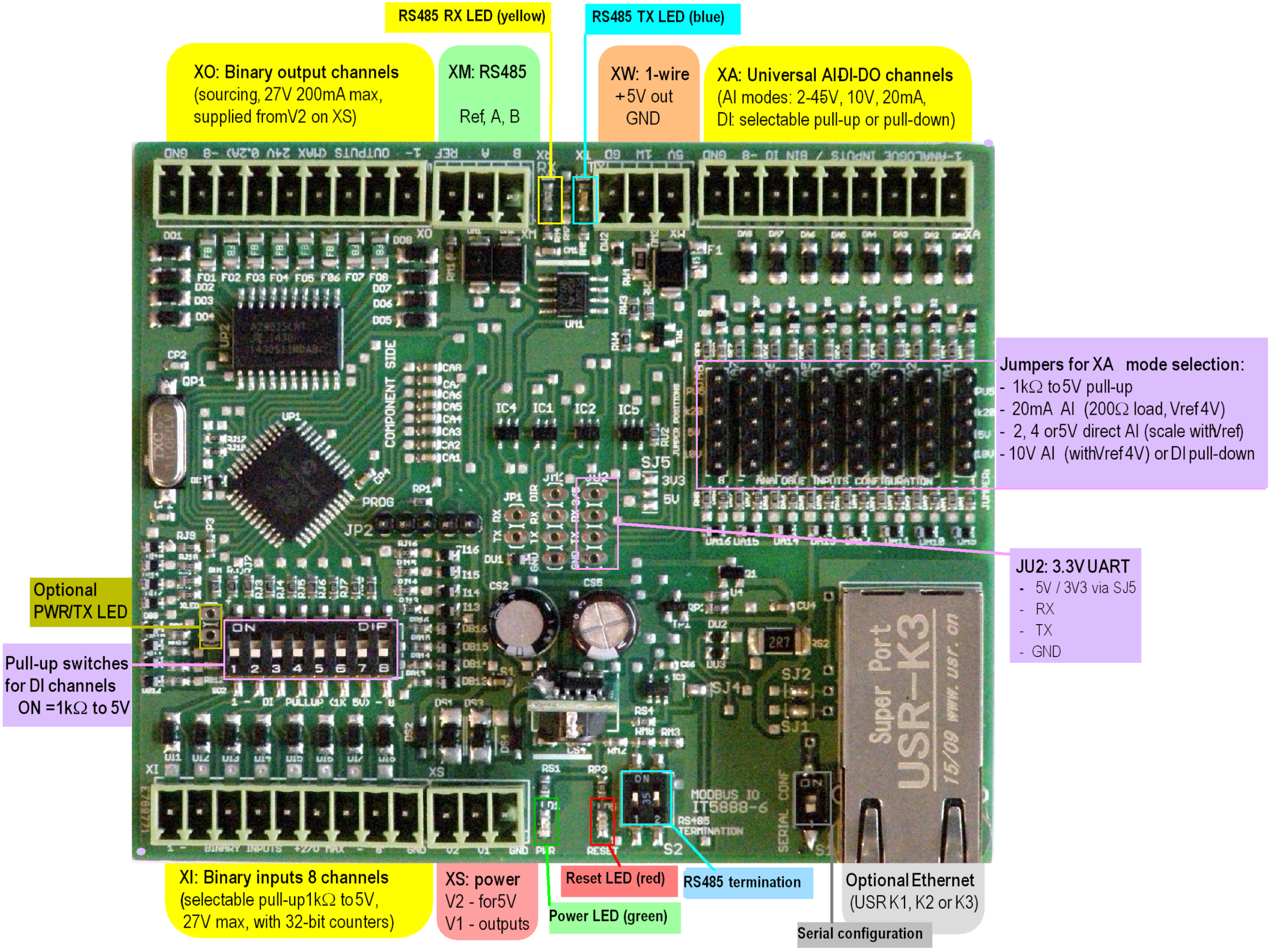
There are **three power inputs** on the device to be used in different situations. The powering options include either 5V DC or any DC voltage from 7 to 27V. The current consumption mainly depends on the load of the binary outputs. The outputs can directly drive resistive or inductive loads up to 2W with maximum current up to 200 mA per output channel (24V heating valves can be directly driven for example).

The device has an on-board PIC-microcontroller taking care of all of the communication, multiplexing and calculations needed to correctly relate the I/O channels with the content of the according Modbus-registers. Some useful properties of the board include the built-in **32-bit counters** for each binary input and the ability to **generate single or recurring pulses** with controlled length and period on the binary outputs. The board is able to supply stable 5V power to other devices, with or without the **watchdog reset** functionality. The watchdog is able to cut the 5V power to the supplied devices for a predefined number of seconds, in the case of no Modbus communication with the I/O-board is detected for a predefined time.

The board can be fitted into a DIN-rail mountable enclosure CNMB-6 from CamdenBoss without any tools or screws. In that enclosure, there is also room for another (control or communication) board, which can be powered from the I/O-board connector JU2. The supply voltage for that unit is available on pin 1 of the JU2 and can be either 5V (1A max) or 3.3V (300 mA max), depending on the connection on the solder-pads next to the JU2.

The firmware of the board can be remotely updated using Modbus communication. The update tool (written in Python) is available for free, together with the latest firmware file.

In the next picture, the purpose of the connectors, jumpers, switches, and LED-indicators on the I/O board are shown.


RS485 RX LED (yellow)
RS485 TX LED (blue)
XO: Binary output channels
(sourcing, 27V 200mA max,
supplied fromV2 on XS)
XM: RS485
Ref, A, B
XW: 1-wire
+5V out
GND
XA: Universal AI-DI-DO channels
(AI modes: 2-45V, 10V, 20mA,
DI: selectable pull-up or pull-down)
Jumpers for XA mode selection:
- 1kΩ to5V pull-up
- 20mA AI (200Ω load, Vref4V)
- 2, 4 or5V direct AI (scale withVref)
- 10V AI (withVref4V) or DI pull-down
JU2: 3.3V UART
- 5V / 3V3 via SJ5
- RX
- TX
- GND
Optional PWR/TX LED
Pull-up switches for DI channels
ON =1kΩ to 5V
XI: Binary inputs 8 channels
(selectable pull-up1kΩ to5V,
27V max, with 32-bit counters)
XS: power
V2 - for5V
V1 - outputs
Reset LED (red)
Power LED (green)
RS485 termination
Optional Ethernet
(USR K1, K2 or K3)
Serial configuration

In the following pictures, **a few examples of cased products** where UF080808 is fitted into a plastic case for mounting on the 35mm DIN-rail can be seen.

I/O - extension module UF080808E with Ethernet and RS485 interfaces, for ModbusRTU or ModbusTCP - communication with the external Modbus Master.

Linux - based automation controller UFK080808 with one Ethernet, three USB interfaces and one RS485 interface for ModbusRTU-communication. WiFi and Bluetooth, HDMI video.

Usable as a ModbusRTU master or a gateway from any network protocol to the ModbusRTU (depends on the software).

2. Technical specifications

- Primary supply voltage 7..27V DC, current 30 mA max @12V (without an external load on the 5V output). To be connected to pin V1 of the connector XS.
  Alternatively, a 5V feed to the device is possible to replace the primary supply, bypassing the on-board DC/DC converter.

- Separate power supply voltage 7..27V for the output driver only, to pin V2 of the connector XS. The optimal voltage and maximum load for this power supply depend on the loads connected to the binary outputs (the loads will be supplied with the same voltage of and should not consume more than 1500mA total).

- 8 channels of discrete sourcing outputs (DO), available on connector XO.
  Main properties:
  - supply and output voltage 12..45V;
  - load current up to 700 mA (with self-recovering overload protection);
  - configurable power-on level;
  - single or periodical pulses with
    timing steps 0.25 ms and length/period up to 16.38 s, or
    timing steps 1 ms and length/period up to 65 s;

- 8 channels of discrete inputs (DI), available on connector XI. Main properties:
  - active level high or low (bitwise configurable via setup register),
  - 4 inputs usable for two Wiegand readers;
  - 32-bit counter on every input;
  - maximum input voltage 27V (due to overvoltage protection 28V);
  - manually controlled pull-up 1kΩ to 5V;
  - 100kΩ pull-down resistance (input impedance with deactivated pull-up).

- 8 universal channels (AI/DI), individually settable to one of the following:
  - 12-bit analog input 0..2V/0..4V/0..5V/0..10V or 0..20mA;
  - discrete input low or high active (bitwise configurable via setup register).

  The common properties of these channels:
  - 32-bit counter on every channel configured as binary input;
  - 3-level comparator output as 2 DI bits for AI voltage range determination;
  - maximum input voltage 27V (overvoltage protection at 28V);
  - jumper-controlled pull-up 1kΩ to 5V;
  - input impedance 200Ω, 25kΩ or >1MΩ depending on the selected channel mode.

- RS485 interface, with communication speeds from 9600 to 115200 bps, even or no parity, accessible via connector XM. ModbusRTU protocol supported by firmware

- Optional Ethernet 10/100BaseT interface with ModbusTCP protocol support

- 1-wire interface for up to 18 DS18B20 temperature sensors

- 5V 500 mA protected supply voltage output for external devices (via connector XW)

- 5V 1000 mA or 3.3V 300mA protected supply voltage for a master or communication device with UART interface (available on connector JU2)

- On-board LED-indicators for RS485 TX, RS485 RX, 5V power, 5V reset

- Board dimensions 86x103mm (3.39x4.06 inch), 21mm (0.29 inch) highest point (DC-DC module) fitting into a 6 DIN-rail modules wide plastic case from CamdenBoss (106mm width)

- Environment conditions for operation: temperatures ranging from -20 to +60 °C, humidity up to 95% (non-condensing)

- Standards to comply with: EN 55022 criterion B, EN 61000-4-2, EN 61000-4-3, EN 61000-4-4, EN 61000-4-5, EN 61000-4-6.

3. Power supply

The board can be supplied with two DC voltages, the second being for powering output channels only. The reason for separate supply feeds is that this way the power-hungry discrete outputs can be fed with 24 or 36 V supply voltage without battery backup, while the main supply is backed with a battery with a different voltage.

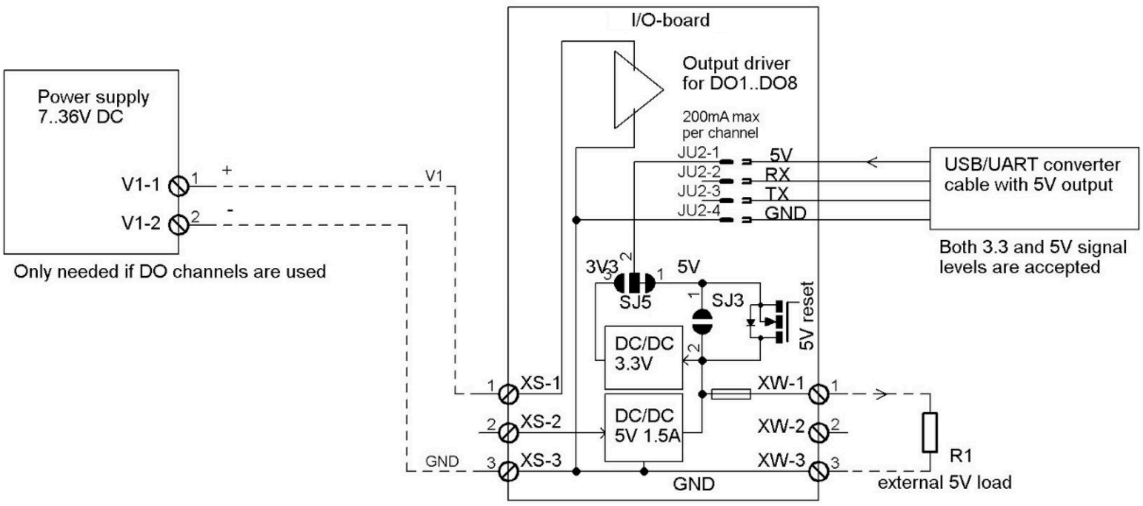
- The first supply voltage ranges from 7 to 27V is used to create a 5V supply for the board and the possible external loads. The current consumption without external loads is about 12mA@12V (via connector XS pin V1). The load will increase if any external consumers of 5V supply voltage are connected to connectors XW or JU2. The total power of external loads for 5V supply is limited to 7W.
- The supply voltage for the discrete outputs (via connector XS pin V2) ranges from 12 to 45V, consumed current being defined by the connected loads.The maximum possible load should not go above 1.5A, while the current for each individual output is limited to about 0,7A.

4. Digital communication

There are five digital data messaging channels available for simultaneous communication with the I/O-board. Two of them (RS485 and 3.3V UART (latter either directly or via Ethernet/serial converter module)) are transparent to each other and also to the PIC microcontroller, being used to read and write the I/O registers on the board using the ModbusRTU protocol (see Appendix 1 for detailed register information).

One digital communication interface is for 1-wire communication with DS18B20 temperature sensors. Two listen-only interfaces are available for the Wiegand-readers, their usage will reduce the number of available DI channels by 2 per Wiegand-reader.

Powering the I/O-board from USB/UART cable



The load to the supply source V1 is the sum of the loads connected to the binary outputs
Connect soldering pads SJ3 if load is planned for 5V output (XW-1)

4.1. Modbus-communication

The primary interface for Modbus-communication is the RS485 via connector XM. Depending on the product, an Ethernet-serial converter module (USR K1, K2, or K3) may be installed on the PCB. This module enables communication with the I/O-module over the LAN using TCP protocols while maintaining the ability for RS485 communication.

Note: the 3V3 UART interface (JU2) should not be used if the Ethernet-serial converter is installed (therefore no JU2-pins are installed for UF080808E board). When the Ethernet module is installed but the UART interface on JU2 has to be used, soldered jumpers SJ1 and SJ2 must be disconnected first. These jumpers must be shorted again to re-enable Ethernet access.

The basic communication protocol for the I/O-board is ModbusRTU. With the Ethernet/serial converter module USR K7 installed, ModbusTCP protocol becomes supported (TCP/RTU conversion is taken care of by the converter module).

The serial port of the onboard PIC-microcontroller, RS485 line and the Ethernet module (or UART, if Ethernet-module is not installed) share the same data for the communication. At all times, only one of these three parties should transmit, while the other two are listening. Without an active outbound transmission, all channels are listening.

All serial devices connected to the I/O board via either an RS485 or the 3.3V UART should share the same communication parameters (speed, parity) as the on-board PIC. The same parameters should be used to configure the serial port of the Ethernet module as well if used.

The RS485 line needs to be terminated at both ends. With the switch S2 on the board, termination can be activated (position ON for both sliders) or deactivated (position OFF for both sliders). **Important**: **The RS485 termination must be activated if no RS485 line is connected** (to the connector XM), to prevent line voltage floating.

4.2. Serial (ModbusRTU) communication

The default parameters for the Modbus communication with the on-board PIC-microcontroller are 19200 bps, 1 start bit, 1 stop bit, even parity (**19k2 8E1** in short). The speed and parity may be changed via the setup register with the address 273 (see Table 2).

The **default Modbus address** of the board is 1. One can **reset** the address, communication speed and parity to the factory default values (1, 19200 and EVEN respectively) by following the procedure mentioned below:

- Disconnect the supply power from the board
- Connect pins 4 and 5 of the header JP2 (using a jumper from any analog channel)
- Power the board for 5 seconds
- Disconnect the power and remove the jumper from JP2Now the values for the Modbus address and the serial communication are set to default.

To see and change the content of the on-board registers, connect a suitable Modbus Master device to the I/O-board via RS485, Ethernet or UART.

The legal Modbus address range is from 1 to 247**.** Address 0 is usable for a special broadcast purpose, meaning any write command to this address is fulfilled by all boards of this type connected to the RS485 bus, without a response to the master.

4.3. Ethernet (ModbusTCP) communication

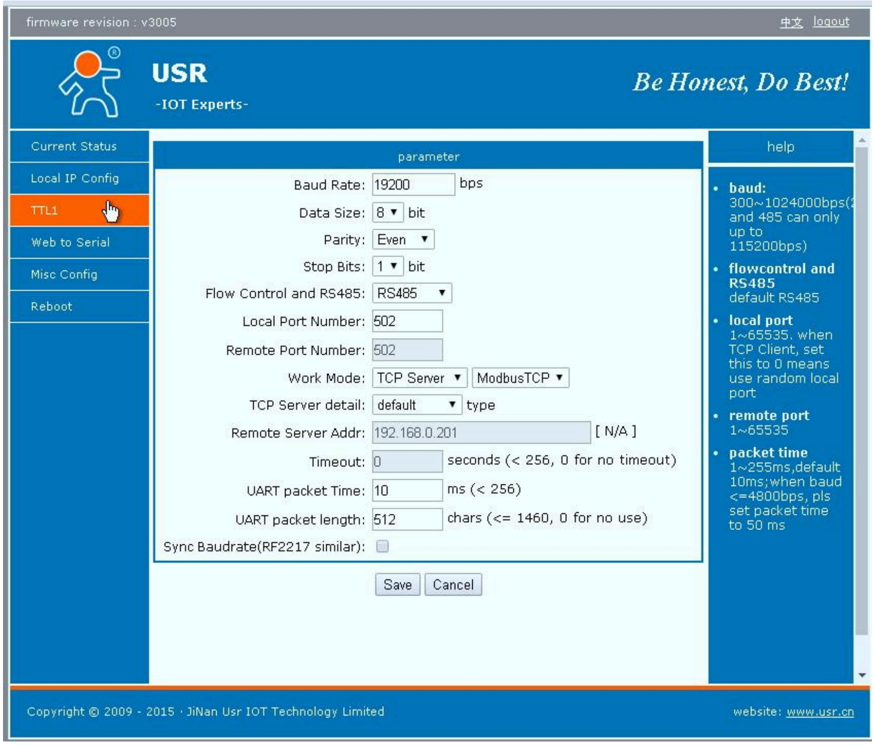
If the I/O-board has the Ethernet/serial converter installed (and solder-jumpers SJ1 and SJ2 closed), then instead of UART signals on JU2 the Ethernet connection can be used to communicate with the board. The RS485 port still remains functional, transparently reflecting the serial data from the Ethernet/serial converter. The communication protocol via Ethernet is ModbusTCP.

If it becomes necessary to use the UART connection via JU2 on the I/O board with Ethernet-port installed, then the soldered connections of SJ1 and SJ2 on the I/O board must be disconnected using a soldering iron. This disconnects the RX and TX signals from the Ethernet-serial converter from the according JU2 pins. Do not try to use UART and Ethernet communication at the same time.

If the current IP-address of the Ethernet/serial converter is not known, use the USR-Kx tool (downloadable from https://extinfo.uniscada.eu/support/tools/). This tool will discover the Ethernet-serial modules on the LAN even if your computer is not on the subnet of the converter (default IP-address being 192.168.0.7).

If the IP-address of the Ethernet module is known and the module is responsive (use the ping utility for testing), then simply open your web browser with this address (port 80) to configure the new network and/or serial port parameters.

However, the usage of the USR-Kx tool is preferred over the web setup to ensure a proper RS485 handshake, which the web setup seems to lack.



Baud Rate: 19200 bps
Data Size: 8 bit
Parity: Even
Stop Bits: 1 bit
Flow Control and RS485: RS485
Local Port Number: 502
Remote Port Number: 502
Work Mode: TCP Server ModbusTCP
TCP Server detail: default type
Remote Server Addr: 192.168.0.201 [N/A]
Timeout: 0 seconds (< 256, 0 for no timeout)
UART packet Time: 10 ms (< 256)
UART packet length: 512 chars (<= 1460, 0 for no use)
Sync Baudrate(RF2217 similar): □

- **baud:** 300~1024000bps(2 and 485 can only up to 115200bps)
- **flowcontrol and RS485** default RS485
- **local port** 1~65535. when TCP Client, set this to 0 means use random local port
- **remote port** 1~65535
- **packet time** 1~255ms,default 10ms;when baud <=4800bps, pls set packet time to 50 ms

Copyright © 2009 - 2015 · JiNan Usr IOT Technology Limited

website: www.usr.cn

The Ethernet-module on the PCB can be used both in **ModbusTCP server and client modes**.

In the **server** mode the module is listening to ModbusTCP queries from the LAN and forwards them as ModbusRTU queries to the RS485 interface and the onboard PIC microcontroller in parallel. No device on RS485 can have the same Modbus address (unit ID) as the onboard PIC. The responses to the incoming queries from the addressed slaves are converted to ModbusTCP responses and sent back to the ModbusTCP client on the LAN. The module must have a static IP-address in server mode. The local power for ModbusTCP is usually set to 502, but can be changed.

If the Ethernet-module is configured as a ModbusTCP **client**, then the address and port of the server must be defined in the module setup. The Modbus queries are initiated by the master on the RS485 bus in this case and sent to the server on the LAN. It is important that the unit ID of the onboard PIC and the server on the LAN do not match with each other, because the queries from RS485 are sent both to the onboard PIC and the server on the LAN. The Ethernet-module in client mode may have dynamic IP-address if preferred.

If the Ethernet interface will be the only communication channel to the I/O board in use, then it makes sense to increase the baud rate from the default 19k2 to the maximum level of 115k2. Change the content of the register defining the on-board microcontroller speed **before** you change the baud rate of the Ethernet-serial converter.

Keep in mind that if both the RS485 and Ethernet interface are in use, then the communication between them is transparent. All slave devices (including the onboard PIC) on the serial side must share the same communication parameters and have a unique Modbus address (in the range from 1 to 247).

It is strongly advised to change the default admin password (admin) of the Ethernet-module in order to prevent the unauthorized change of parameters from the web browser or using the USR-Kx tool (the latter is able to discover the modules on the LAN, enabling easy access if the default password is in use).

4.4. One-wire communication

The I/O board supports up to **18** digital **DS18B20** temperature sensors**,** either in powered or parasite power mode. On the same connector, a 5V DC supply (protected with auto- resetting 500mA fuse) is available for the 1-wire (or other) devices.

The parasite power mode is usually preferred due to wiring simplicity. The I/O-board will provide “strong pullup” pulses to the 1-wire data line before reading, ensuring reliable communication also in the parasite power mode.

Using the default setup, the 1-wire sensors are discovered immediately after power-up or reset and assigned to their registers by the order of their ID-values. Therefore, the register assignment of the sensors may change if a sensor is added to or removed from the 1-wire bus and the board is restarted. In order to avoid unexpected changes in the sensor addressing, the sensor renumbering lock is activated by writing the value of 1 to the register with an address of 699. This will prevent sensor re-addressing during reset or power-up. With the lock activated, the lost sensor does not result in re-addressing of all other sensors and (by the value of 4096) it is easy to detect which sensor is lost or faulty. The default value for the register at address 699 is 0, allowing 1-wire sensor renumbering on each reset.

See also “Working with 1-wire devices” section and Appendix 1 for additional information regarding how to use the 1-wire sensors.

4.5. Wiegand-communication

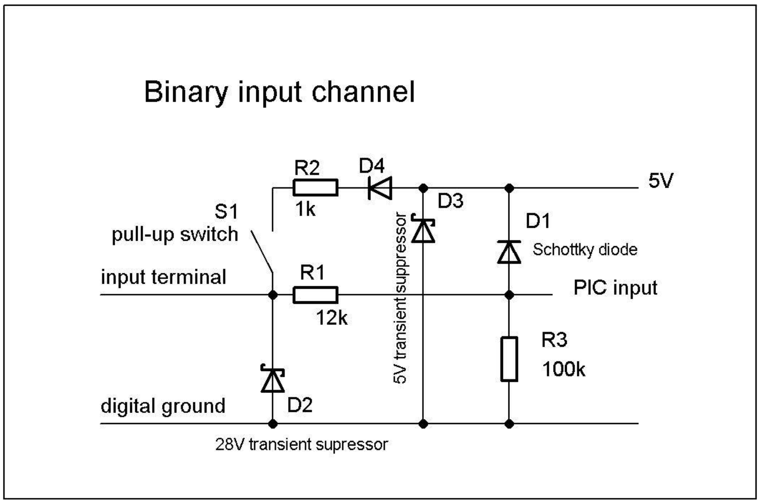
Up to two Wiegand-devices with message lengths up to 40 bits are supported. Each Wiegand-device needs two data lines to be connected. Two or four binary input channels (from the range of DI5..DI8) can be switched to the Wiegand mode by setting the according bits in the register with the address 273 (see Appendix 1).

The pull-up switches for the DI-channels used for Wiegand communication must be manually switched to ON to allow the Wiegand inputs to function. The length of the received message is available on the register with the address 10. The data bytes received from the Wiegand readers are stored into the registers with addresses ranging from 11 to 18. The stored data will not be overwritten before the existing value has been read by the Modbus Master.

5. Using the discrete inputs

5.1. Introduction

The maximum number of discrete inputs is 16. Eight of them are available in the DI port (via the connector DI) and the rest are accessible via the universal ANA-port (via the connector XA). The input data is bit-mapped to the holding register with address 1, where the eight most significant bits correspond to the channels DI1...DI8 and the least significant bits correspond to the channels marked as AI1 to AI8.



5.2. Discrete level inputs

The discrete inputs are equipped with a digital “debounce”-filter, cleaning the input signal from relay contact bounces (shorter than 4ms). Therefore do not use input pulses shorter than 5ms and repeating frequencies higher than 100 Hz on the DI-port. The ANA-port’s maximum speed is much lower due to the RC filters used.

On high pulse repeating frequencies the communication speed with the input register (with address 1), the Modbus-communication is slower than the input state change rate. To get the number of state changes (up and down) between the readouts, check the 32-bit counter (dedicated to each binary input).

If the number of changes is irrelevant, but the change cannot be missed, a so-called “sticky bit” mode (valid for all inputs simultaneously) can be used instead of counter checking. In this case, the changed bit in the register will be fixed until the next register readout from address 1.

This way we will be able to know which input has been changed since the last readout, but cannot know the number of changes or the current state of the changed input until the next readout. This enables faster changed input scanning compared to the counters usage (adding two registers to read for every input channel in use).

The active level of the input signal (high or low, default is high) can be changed by setting the mask in the input inversion register with the address 271 (see Appendix 1).

All input channels are equipped with either a switch or a jumper to activate the pull-up to the 5V supply via a 1kΩ resistor. Thanks to the diode in the pull-up circuit, input voltages up to 27V are allowed even if the pull-up resistor (R2 on the diagram above) is unintentionally connected.

If the pull-up switch is open, a 100kΩ pull-down is effective due to resistor R3. For the inputs used for Wiegand-readers the pull-up must be activated (switched ON).

5.3. Pulse counters on discrete inputs

The 32-bit counters are available for every input channel in the discrete input mode (DI1..DI8 and AI1..AI8). The address range of the counter registers (two holding registers per counter, starting from the most significant word) is from 384 to 399 for counters related to channels from AI1 to AI8 and from 400 to 415 for the counters related to the channels from DI1 to DI8 (see Appendix 1). The active edge (falling or raising) can be selected with the input inversion mask register with the address 271, the default is raising.

The initial state of the counters can be set by writing the initial value into the according registers, “write multiple registers”- command is also supported.

Do not switch off the “debounce” filter and do not use the “sticky bit” option if you are using counters. These options set by the certain bits in register with the address 273 (see Appendix 2) have an effect on all binary input channels of the I/O board. Always prefer DI-channels for counting short pulses, because there is a RC-filter on AI channels that can prevent shorter than 50 ms pulses from passing through.

6. Using the discrete outputs

6.1. Introduction

The eight sourcing discrete outputs with MOSFET drivers are capable of feeding up to 700 mA of current each, at maximum voltage of 45V (the high-state output voltage depends on supply voltage on pin V2 of the connector XS.

The output channels are individually protected against overload. If the overload condition is found, the overloaded channel will be switched off and the register with address 272 will have a value of 0 (the same value may also indicate the missing VO voltage). The output driver will periodically try to repower the overloaded channel to recover from the situation, with the simultaneous value of 1 in the register 272.

6.2. Discrete level outputs

The output control is done by writing the needed bitmap code to the holding register with the address 0. The MSB of the register content (bits 8..15) corresponds to the binary outputs DO1...DO8. The LSB (bits 0 to 7) are not used.

6.3. Pulse outputs

In addition to energizing the output channel as voltage level, it is also possible to generate voltage pulses with 0.25 - 16 ms resolution (depending on register 270’s content) on the discrete outputs, both **single** and **periodical** (continuous).

For the generation of single pulses, the registers in the address range 100..115 (corresponding to the binary outputs marked AI1..AI8 and DO1...DO8 accordingly) should be written into with the value of the length of the pulse in ms. The pulse starts immediately and ends when the time for the pulse length has elapsed. To generate an inverted pulse, one can write 1 to the level bit in the register with the address 0 and then start the pulse on the same channel via the register 1XY. The actual output is formed as XOR of the level value and the pulse generator output. The maximum pulse can be up to 65 s, if 0.25 ms timing step is used.

If periodic pulses with a stable length are needed on the output, then both period and pulse length must be set. The maximum value for the pulse period is 4095 ms, to be written into the register with an address of 150 (global for all DO-channels of the I/O-board). The value for pulse length must now have the most significant bit set in order to enable pulse recurrency (periodic pulse output).

An advanced property of the pulse outputs is the ability to set the phase of each of the outputs. The period is divided into 4 phases (with 90 degrees shift between them). Each discrete output can be tied to one of the phases, defined by 2 bits in the MSB of the channel data (see Table 3 at the end of the document).

7. Using the analog inputs

7.1. Introduction

The analog input channels (up to 8) are available on the ANA port. The channels must be configured correctly in order to use them as analog inputs. The channel direction and mode are defined by the MSB and LSB content of the setup registers with addresses 275.

The I/O-board has a 12-bit ADC. The reference voltage defines the input voltage range of ADC channels and can be set by writing value 0, 48 or 32 into the register with address 270. There are three reference voltages possible depending on the register value:

- value 0 activates the external reference voltage from the power supply (uncommon);
- value 48 activates the 4.096V internal reference voltage (this is the default value)
- value 32 activates the 2.048V internal reference (use this for maximum AI sensitivity).

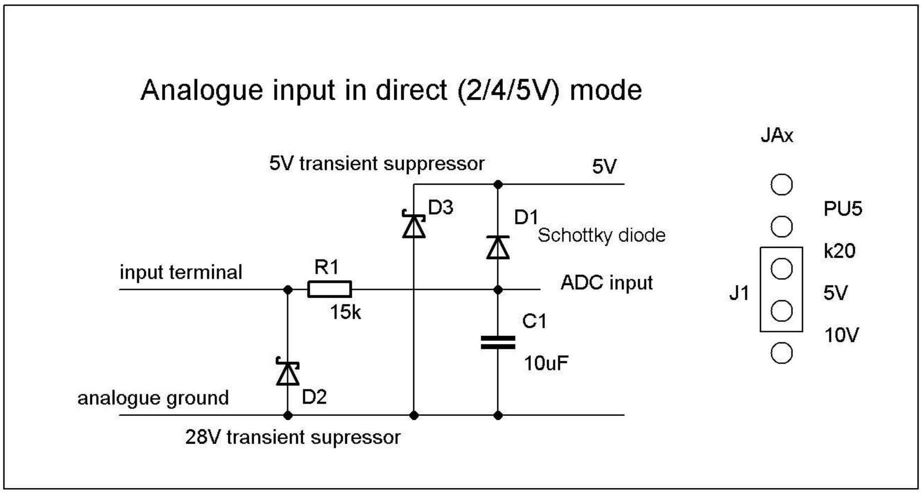
It is usually best to use the 4.096V reference voltage, as this provides (depending on the jumper positions) either direct mV readings or easily convertible values (like 2.50 mV or 5uA per ADC output code unit). The reference voltage is always the same for all analog inputs and can be set with the register at address 270. If the reference voltage is 4.096V (the default value), then the following full-scale analog measurement ranges are available depending on the jumper positions: **0..10V, 0..4V, 0..20mA**. The last position (pull-up 5V) may be needed in the discrete input mode or to power some analog sensors.

For the **analog measurements with maximum precision** it is suggested to calibrate the AI channels individually in 2 points (one point close to the beginning and another close to the end of the input range). This will also compensate for the ADC offset value at the low end of the measurement range. Correction to every measured value can be calculated using linear interpolation.

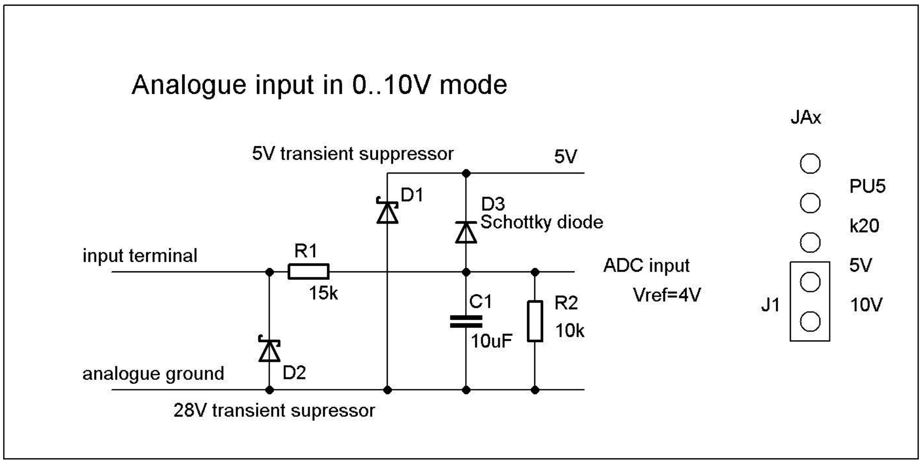
7.2. Analogue input modes

The following simplified schematic diagrams are helpful to understand the difference between the various modes that are possible for the input channels via the ANA port. The according jumper positions to be manually set on the pin header JAx (JA1..JA8) are shown as well.

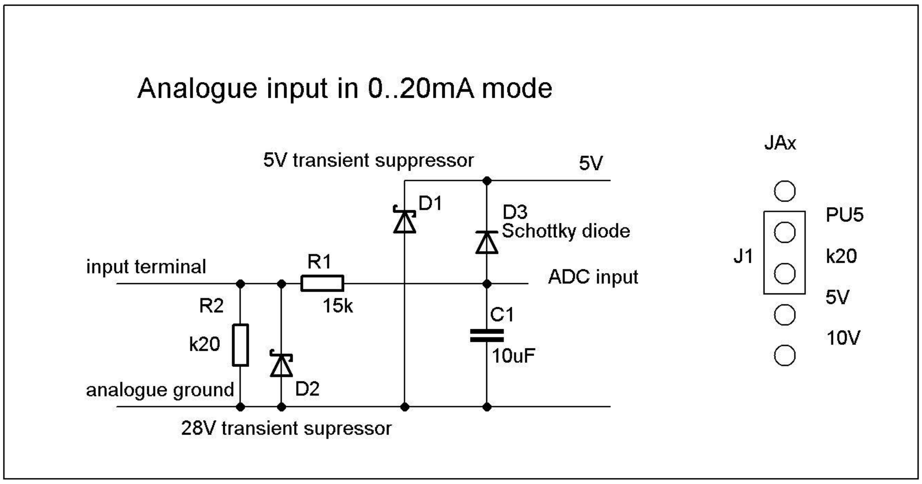
In the **direct AI mode**, the analog input range is determined by the reference voltage (via the register at address 270), being 2.048, 4.096 or 5.0 V. Jumper J1 must be in the position “5V”. ADC output code unit corresponds to 0.50, 1.00 or 1.22 mV depending on the reference voltage selection accordingly.



In the **10 V AI mode**, the input voltage is divided by resistors R1 and R2. Reference voltage should be set to 4.096 V. Then the ADC output code unit corresponds to the 2.50 mV voltage on input.

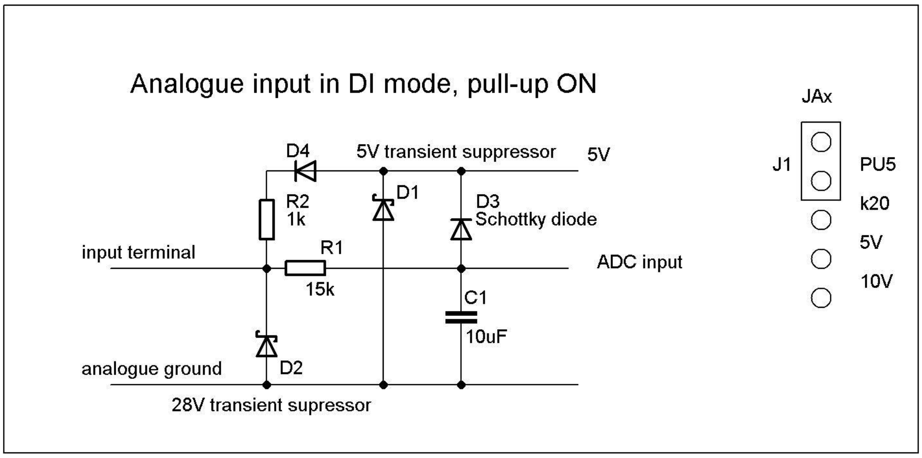


In the **0...20 mA AI mode**, the input voltage is determined by the current via R2. At 20 mA input current, the input voltage is 4 V, the reference voltage should be 4.096 V and the ADC output code unit corresponds to 5 μA (0.005 mA).



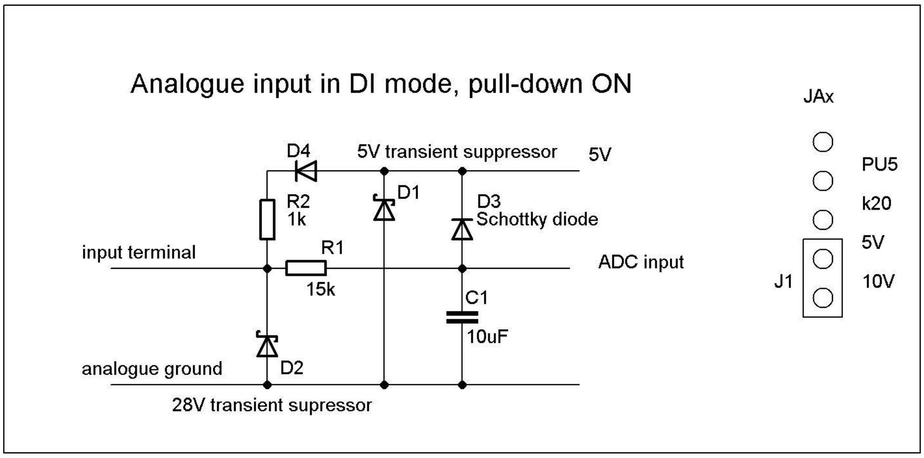
DI-mode with pull-up activated.

Set the holding register with the address 275 according to your needs. Pay attention to the RC-filter R1C1, making short-pulse sensing and counting impossible! Use channels DI1..DI8 for detecting pulses shorter than 500 ms.



DI mode with pull-down activated.

Set the holding register with address 275 according to your needs. Pay attention to the RC-filter R1C1, making shorter than 500 ms pulse sensing and counting impossible! Use channels DI1..DI8 to receive shorter pulses**.**



In addition to setting the channel jumpers, the correct values (bitmap codes) in the setup registers should be set accordingly as well (see Appendix 1) for the proper DI operation of channels in port ANA.

7.3. Converting analogue input values to discrete values

Often the exact analog values on the AI channels are not as important as their comparison result with some predefined threshold values. The comparison can always be done in the master device application software, but for that the analog values have to be read out first, 2 bytes per channel. If the comparison is done in the I/O-board firmware, then the comparison values (comparator outputs) for all analog inputs can be read as one register. This is much faster than reading all the analog channels and doing the range check in application software.

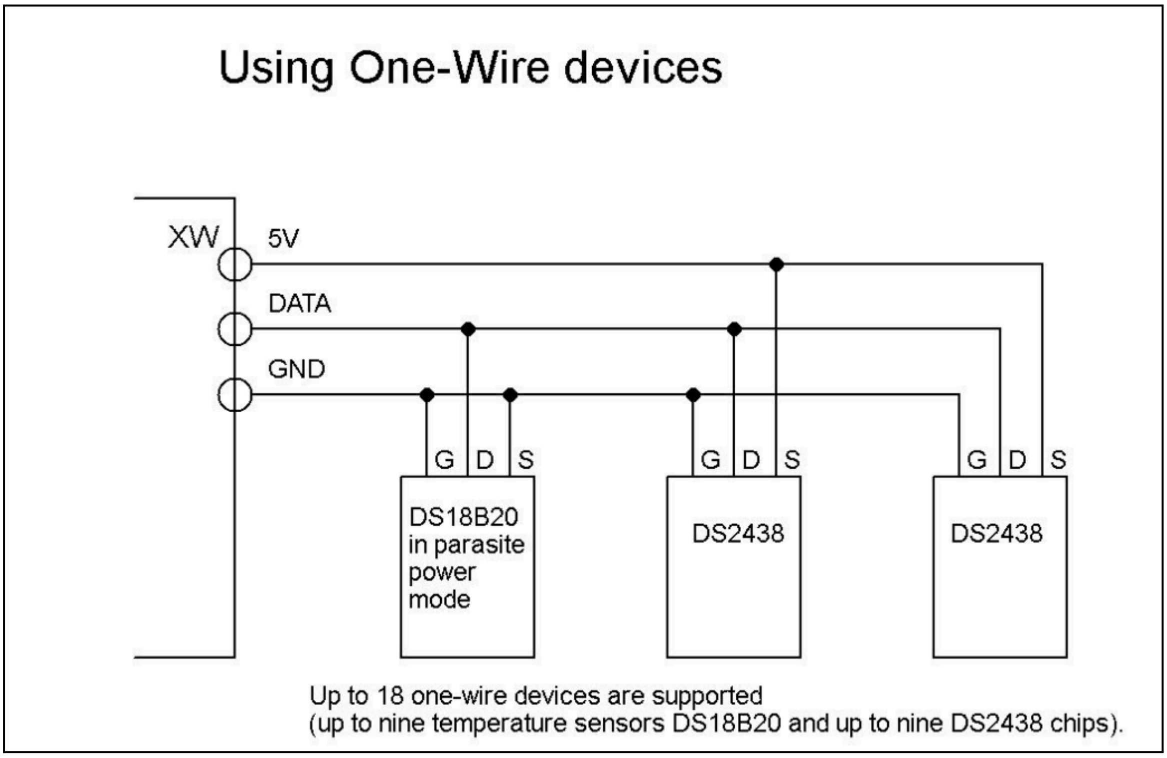
The IO-board is therefore equipped with **analog comparators** with 3 shared reference values (ref1...ref3 defined by the content of registers 501..503). Using the reference values it is possible to create up to 4 value ranges (0..ref1, ref1..ref2, ref2..ref3, ref3..4095) for mapping the actual analog value into a 2-bit binary code (see table A5 at the end of this document).

Using the comparator output is suggested when using movement or smoke detectors over the line with **dual end-of-line** resistors (enabling to detect line cut or short-circuiting). Note that the three reference values are shared for all AI channels, so the line resistors and pull-up voltages should be similar for them (set the channel **mode jumper to PU**-position to switch on the 5V pullup). The suggested ADC reference voltage is now 5V (not 4V as by default, the ADC reference voltage is set by register 270).

The lower bit of each of the comparator outputs in register 500 (bit0 for AI1, bit1 for AI2 and so on) is usable for signal level (state) detection relative to ref2. The upper bit of the comparator code per channel (bit8 for AI1, bit 9 for AI2 and so on) signals the line voltage “out of range” condition. To find out if the problem is either in line cut or short/circuiting, just take the lower bit into account.

8. Working with 1-wire devices

The digital Dallas 1-wire interface is provided on the connector XW. Up to nine DS18B20 temperature sensors plus up to nine DS2438 battery monitor devices (multi-channel differential ADC devices) are supported. Both data and the full 64-bit one-wire device ID can be read from the registers on the board.


Using One-Wire devices
XW
5V
DATA
GND
G
D
S
DS18B20 in parasite power mode
G
D
S
DS2438
G
D
S
DS2438
Up to 18 one-wire devices are supported
(up to nine temperature sensors DS18B20 and up to nine DS2438 chips).

The temperature readings from the **DS18B20** sensors are 12-bit signed values in 9 registers starting from the register address 600, and 9 sensors from address 700, one sensor per register. The values read must be divided by 16 to get the temperature value in Celsius.

Each 1-wire sensor has an unique 64-bit ID that is found and stored in the process of sensor auto-discovery. The discovery process normally starts at power-on but can also be started by a command (see below). The ID-s are stored in 4 sequential registers for each sensor starting from address 650 for the first 9 temperature sensors and from 750 for the rest.

In the default setup, the 1-wire sensors are discovered immediately after power-up or reset and assigned to their registers in the order of their ID-values. Thus the register assignment of the sensors may change, if a sensor is added or removed from the 1-wire bus and the board is restarted. In order to avoid unexpected changes in the sensor addressing, the sensor rediscovery lock is activated by writing the value 1 to the register with the address 699, This will prevent any 1-wire sensor addressing change during reset or power-up. With the lock activated, the disconnected or invalid sensor cannot cause the change in accessing other sensors on the next restarts of the I/O-board. The default value for the register at address 699 is 0, allowing 1-wire sensor discovery on each reset. Note: if data value for an onewire temperature sensor is 4096, then the sensor could not be read (can be missing or faulty).

It is also possible to write the sensor ID values in addition to the sensor auto-discovery process. This is handy if one sensor needs to be replaced or added but the registers of other 1-wire sensors should be kept unchanged. If the ID of the new sensor is not known, then the suggested list of actions to add it is the following:

1) read and store somewhere the existing sensor ID values from registers 650…, 4 words per one sensor;
2) write 9 to the register with the address 699 (remove the lock);
3) start auto-discovery by power break or writing 0xDC1A to the register with address 999;
4) read and store the sensor ID values from registers 600… (their order is likely different now);
5) write the new set of sensor ID-s into the registers starting from 650, in the necessary order, using the values stored earlier and the discovered value for the new sensor;
6) set the sensor discovery lock on again by writing 0x0001 to the register with the address 699.

Be aware that the onewire functionality is disabled, if register 279 has a value of zero.

In order to adjust the onewire related delays optimally according to the cabling in use, the **`mbusb`** helper tool (available at https://extinfo.uniscada.eu/support/tools/) can be used.

After starting an interactive Python session with that tool, you can find the optimal value for register 699. Use `tempdisc(mba),` where mba is the Modbus address (unit id) of the I/O board. The tempdisc() function runs through sensor discovery at various delays to find the best value for register 699. The order of the sensors can be changed by rewriting their ID-s after that. Any further changes to that order can be prevented by adding 1 to the suggested optimal value for register 699.

9. Watchdog functionality

If a controlling device (Embedded Linux board for example) is powered by a 5V (1000 mA max) supply from the board via pin1 in JU2, then it makes sense to set up a watchdog timer to reset the controlling device by cutting the 5V supply in case no Modbus-communication is detected by the on-board PIC for the time (in seconds) stored in the register with address 276. The length of the power cut pulse depends on the value in register address 277 (also in seconds). The default values for both registers are 0, meaning no power reset is in use.

It is also possible to initiate the power cut pulse by the Modbus Master, by writing 0xFEED to the register with address 999. The pulse length in seconds must be defined in the register with the address 277. This functionality has no effect if the controlling device is not supplied via pin1 of the on-board connector JU2.

Another time-out register is available on address 278. The content of this register defines the time without Modbus-communication until the I/O-module reset is done. This reset returns the outputs of the module to their initial values, defined by register with address 272.

Register 279 is related to onewire temperature sensors. The sensors are normally read one after another between Modbus queries. If no Modbus queries are arriving, then the temperature readings are refreshed after the number in register 279 of seconds delay between the individual readings. No onewire temperature readings will be done if register 279 contains value zero!

10. Upgrading the firmware

It is possible to update the firmware of the I/O module remotely via Modbus-communication, should the reason for the update appear. If the firmware change becomes necessary, the according hex file becomes available on the website of Uniflex Systems.

The update process is based on uploading the firmware in Intel HEX format using multiple register write commands according to the ModbusRTU protocol. The suggested way to carry through the update is to use the software written in Python, also available from the manufacturer.

The update process does not affect the configuration data of the I/O-module, meaning the serial communication baud rate, parity, Modbus-address and all other parameters will be preserved. During the update process, the on-board microcontroller will be switched into so-called bootloader mode, where the normal operation of the I/O-module is stopped. Depending on the communication speed in use, the whole update process takes less than a minute.

If the update process fails for some reason (like a power cut in the middle of the upload), the I/O module will remain in bootloader mode. There are two ways to recover from the bootloader mode:

- To retry the firmware update process, using the same Modbus address and communication parameters. The software tool is able to resume the failed update.
- To reprogram the PIC microcontroller using a suitable programmer (PRESTO from ASIX for example). This will reset all setup data as well.

11. Usage tips

11.1. Watchdog registers

If the I/O-module is used with Linux-board (powered with 5V via JU2), then the suggested values for watchdog registers are as follows:

- 276: 512,
- 277: 9,
- 278: 768 or 0

If no serial communication on UART or RS485 has been detected for 512 s, then 5V cutoff for 9s will “cold” reboot the connected CPU module (powered via the UART socket).

11.2 Loading the discrete outputs

Each output channel has the maximum load capability of 700 mA with individual thermal protection. The overloaded channel will be temporarily shut down and restored without affecting other channels. The shutdown condition can be (globally) detected via register 116 (see the description in the table A1). Due to the FET-driver in use, the voltage drop on the output compared to the supply voltage V2 is small (less than 0.5V and caused mainly by the Schottky diode in the supply circuit).

If the total load current is expected to be higher than 3A, then an additional wire must be soldered to the PCB bottom side between the two 2 x 5 mm pads, to increase the cross-sectional area of the output supply circuit.

The maximum supply voltage for DO is 45V.

Table 1: Register summary

Register Address	**R** (readable) **W** (writeable) **F** (write to flash)	**Description**
0	RW	**DO register**, MSB contains the channel states of the DO port, LSB contains the states of the AI port (if used as outputs, see register 271and 275 usage)
1	R	**DI register**, MSW=DI port, LSB=AI port (regarding the input channels, see the setup registers 271 and 275 descriptions)
2	R	**analog input AI1**, 12-bit ADC
3	R	**analog input AI2**, 12-bit ADC
4	R	**analog input AI3**, 12-bit ADC
5	R	**analog input AI4**, 12-bit ADC
6	R	**analog input AI5**, 12-bit ADC
7	R	**analog input AI6**, 12-bit ADC
8	R	**analog input AI7**, 12-bit ADC
9	R	**analog input AI8**, 12-bit ADC
10	R	**The length of the received (but unread) Wiegand-data** in bits, MSB for reader 1 (di channels 5, 6), LSB for the reader 2 (di 7, 8)
11	R	The most significant bits of the Wiegand data from the reader 1
12	R	The following bits of the Wiegand data from the reader 1
13	R	The following bits of the Wiegand data from the reader 1
14	R	The least significant bits of the Wiegand data from the reader 1
15	R	The most significant bits of the Wiegand data from the reader 2
16	R	The following bits of the Wiegand data from the reader 2
17	R	The following bits of the Wiegand data from the reader 2
18	R	The least significant bits of the Wiegand data from the reader 2
100..115	RW	Pulse output (PWM) registers for DO channels, see table 3 for bit value meanings. For single pulses, the last used pulse length value will remain readable in the registers.
116	R	bit0: DO supply (pin VO on socket XS) voltage test, value 1 if the voltage is 5V or more and no DO overload bit15: flips on 1wire cycle end (since v690)
149	RW	The synchronization register for the periodical pulse generation (PWM) on output. On write the delay to period start (in ms) can be given. On read the time in ms since the beginning of the current PWM period length in defined units is returned.
150	RWF	**PWM period** for DO channels in periodic pulse output mode (global) LSB: period length 0..4095 units. Keep 0 in bits 12..15. See the unit definition by the register 270. Note: this register write resets the period counter.

151	R	**Actual state of the DO channels** (being the result of XOR of DO and PWM values)
256	R	**Device type** (0xF1)
257	R	**Firmware version** MSB.LSB
258	RWF*	Serial number 1, in LSB (writable once as a single register after fw programming)
259	RWF*	Serial number 2, in LSB (writable once as a single register after fw programming)
270	RWF	**MSB: PWM unit** (global) 0: 1 ms (default value) 1: 0.25 ms 2: 0.5 ms 3: 1 ms 4: 2 ms 5: 4 ms 6: 8 ms 7: 16 ms LSB: AI (analog input) reference voltage selection. Use 0x00 for reference voltage from 5V supply 0x30 for internal Vref=4.096V (the default value) 0x20 for internal Vref=2.048V Example: code 0x0330 stands for 1 ms PWM time unit, 4096 mV reference voltage
271	RWF	**DI inversion bitmap** (DI active level for a channel the according bit is high, . MSB bits (8..15) relate to port DI, LSB bits (0..7) relate to port ANA channels in DI mode. See also register 275's description for more port ANA related information.
272	RWF	**DO power-on states** as bitmask (active high). MSB: for DO channels (DO MSB) LSB: for ANA channels used as binary outputs (DO LSB), (see register with address 275 for port ANA mode information).
273	RWF	**serial/Wiegand configuration,** LSB bit 0 - speed bit 0 bit 1 - speed bit 1 bit 2 - speed bit 2 bit 3 - debounce OFF (0=on, effective for all DI channels) bit 4 - sticky bit ON (0=off, effective on all DI channels) bit 5 - Wiegand reader 1 enable for inputs DI 5 and 6 bit 6 - Wiegand reader 2 enable for inputs DI 7 and 8 bit 7 - serial parity (1=none, 0=even) See table A2 below for baudrate bit details. A few example values: 2 for 19k2 8e1, for 38k4 8N1, 5 for 115k2 8E1, 1 for 9k6 8E1, 129 for 9k6 8N1. 130 for 19k2 8N1, 18 for 19k2 8E1 (all with debounce filter on).
274	RWF	**Modbus address** of the board, must be within the range from 1 to 247 for normal communication). See Note 4 below this table regarding address 0.
275	RWF	**ANA-port mode control** (bitmap, one bit from MSB and one from LSB for each ANA-port channel): MSB bits: 0=AI, 1=DI; LSB bits: 1=output (DO), 0=input. for DO, the according DI bit must also be active! use 0x0000 to put all 8 ANA channels into AI mode, use 0xFF00 to put all 8 ANA channels into DI mode, use 0xF000 to put ai5..ai8 into DI mode, but ai1..ai4 to AI mode

		use 0xFFF0 to put ANA channels 5..8 into DO mode and channels 1..4 to DI mode
276	RWF	**5V power reset timeout** in seconds, activated by missing or invalid serial communication. Use value 0 (default) for this or for the next register in order to not use this reset. The reset pulse (see the next register for pulse length) cuts 5V power from the first pins of the connectors JU2 and XW.
277	RWF	**5V power reset pulse length** (see also above) to define the 5V power cut pulse length in seconds. Cuts the 5V power from JU2 and XW. Use value 0 (default) for this or the previous register in order to not use this reset.
278	RWF	**IO-board reset timeout** in seconds, activated by missing or invalid serial communication. Reset zeroes the counters and returns the DO-values to their power-up values defined by the content of the register with the address 272. Use value 0 (default) not to use this reset.
384, 385	RW	Counter on input AI1 in DI mode, 32 bits
386, 387	RW	Counter on input AI2 in DI mode, 32 bits
388, 389	RW	Counter on input AI3 in DI mode, 32 bits
390, 391	RW	Counter on input AI4 in DI mode, 32 bits
392, 393	RW	Counter on input AI5 in DI mode, 32 bits
394, 395	RW	Counter on input AI6 in DI mode, 32 bits
396, 397	RW	Counter on input AI7 in DI mode, 32 bits
398, 399	RW	Counter on input AI8 in DI mode, 32 bits
400, 401	RW	Counter on input DI1, 32 bits
402, 403	RW	Counter on input DI2, 32 bits
404, 405	RW	Counter on input DI3, 32 bits
406, 407	RW	Counter on input DI4, 32 bits
408, 409	RW	Counter on input DI5, 32 bits
410, 411	RW	Counter on input DI6, 32 bits
412, 413	RW	Counter on input DI7, 32 bits
414, 415	RW	Counter on input DI8, 32 bits
496,497	R	Time in seconds since the last 5V power break on JU2 (see also registers with addresses 276 and 277)
498, 499	R	**PIC uptime** in seconds (32 bit value), counted from power-up or reset. Read 2 registers together!
500	R	**analog comparator** outputs, line fault bits in MBS, signal bits in LSB. See table 4 for further explanation.
501	RWF	Reference level 1 for the comparators (shared for all 8 channels), see table 4 for further explanation.
502	RWF	Reference level 2 for the comparators (shared for all 8 channels), see table 4 for further explanation.
503	RWF	Reference level 3 for the comparators (shared for all 8 channels), see table 4 for further explanation.
504	RWF	Hysteresis for the comparators (shared for all 8 channels), see table A5. Maximum allowed value 29.

550	RW	Copy of register 0 (DO)
551	R	Copy of register 1 (DI)
552	R	Copy of register 116 (pwr VO presence)
553	R	Copy of register 500 (comparator)
554	R	Copy of register 10 (Wiegand data length in buffer)
555	R	Copy of register 151 (actual DO, since fw 6.41, see note 6)
600	R	1-wire thermosensor DS18B20 #1 data, 12 bits
601	R	1-wire thermosensor DS18B20 #2 data, 12 bits
602	R	1-wire thermosensor DS18B20 #3 data, 12 bits
603	R	1-wire thermosensor DS18B20 #4 data, 12 bits
604	R	1-wire thermosensor DS18B20 #5 data, 12 bits
605	R	1-wire thermosensor DS18B20 #6 data, 12 bits
606	R	1-wire thermosensor DS18B20 #7 data, 12 bits
607	R	1-wire thermosensor DS18B20 #8 data, 12 bits
608	R	1-wire thermosensor DS18B20 #9 data, 12 bits
650..653	RWF	1-wire thermosensor DS18B20 #1 ID (64 bits, 653=LSW)
654..657	RWF	1-wire thermosensor DS18B20 #2 ID (64 bits, 657=LSW)
658..661	RWF	1-wire thermosensor DS18B20 #3 ID (64 bits, 661=LSW)
662..665	RWF	1-wire thermosensor DS18B20 #4 ID (64 bits, 665=LSW)
666..669	RWF	1-wire thermosensor DS18B20 #5 ID (64 bits, 669=LSW)
670..673	RWF	1-wire thermosensor DS18B20 #6 ID (64 bits, 673=LSW)
674..677	RWF	1-wire thermosensor DS18B20 #7 ID (64 bits, 677=LSW)
678..681	RWF	1-wire thermosensor DS18B20 #8 ID (64 bits, 681=LSW)
682..685	RWF	1-wire thermosensor DS18B20 #9 ID (64 bits, 685=LSW)
696	R	since v725: RO copy of r699
697	R	**1-wire quality** indication value 0...1000 based on averaged number of CRC-errors found. Value 1000 means no errors, value 0 means no success in 1 wire queries at all. The initial quality value after module power-up, reset or 1-wire discovery is 500.
699	RWF	**1-wire control** register, to adapt the 1-wire interface with various line lengths and weights. Use only if the default behavior brings unsatisfying results. Bit 0 active disables 1-wire auto-discovery on power-up (keeping the order and values of currently stored sensor ID-s (located in the registers 650....685 and 750...785)). Bit 1 active disables adaptive delay control in 1-wire communication (may adjust the value for the delay2, see below). Bits 2..7 contain delay1 value. Bits 8..12 contain delay2 value, changed adaptively (if bit1 enabled) according to the line capacitancy. Default value 286 with autodiscovery on power-on. Use 287 to fix sensor ID-s for added reliability.Try 540 for adaptive autodiscovery.

700	R	1-wire thermosensor DS18B20 #10 data, 12 bits
701	R	1-wire thermosensor DS18B20 #11 data, 12 bits
702	R	1-wire thermosensor DS18B20 #12 data, 12 bits
703	R	1-wire thermosensor DS18B20 #13 data, 12 bits
704	R	1-wire thermosensor DS18B20 #14 data, 12 bits
705	R	1-wire thermosensor DS18B20 #15 data, 12 bits
706	R	1-wire thermosensor DS18B20 #16 data, 12 bits
707	R	1-wire thermosensor DS18B20 #17 data, 12 bits
708	R	1-wire thermosensor DS18B20 #18 data, 12 bits
750..753	RWF	1-wire thermosensor DS18B20 #10 ID (64 bits, 753=LSW)
754..757	RWF	1-wire thermosensor DS18B20 #11 ID (64 bits, 757=LSW)
758..761	RWF	1-wire thermosensor DS18B20 #12 ID (64 bits, 761=LSW)
762..765	RWF	1-wire thermosensor DS18B20 #13 ID (64 bits, 765=LSW)
766..769	RWF	1-wire thermosensor DS18B20 #14 ID (64 bits, 769=LSW)
770..773	RWF	1-wire thermosensor DS18B20 #15 ID (64 bits, 773=LSW)
774..777	RWF	1-wire thermosensor DS18B20 #16 ID (64 bits, 777=LSW)
778..781	RWF	1-wire thermosensor DS18B20 #17 ID (64 bits, 781=LSW)
782..785	RWF	1-wire thermosensor DS18B20 #18 ID (64 bits, 785=LSW)
998	W	**Firmware upload** register

Notes

1. The real register addresses (starting from 0) are shown.
2. All registers can be accessed as holding registers, even if they are read-only.
3. The allowed range to set the Modbus address is from 1 to 247.
4. Modbus address 0 is recognized as a special broadcast address 0 to fulfill write commands without a response to the master (can be used for resetting all DO channels on all UF080808 devices on the bus).
5. Allow 0.1 s delay after each write to register(s) to be stored into the flash (marked as RWF) before sending another Modbus command. Any query following within 80 ms after a flash write related command may fail.
6. Registers in the address range 550..555 enable sequential access to often-needed holding registers (with discrete information) through a single query (with sequential addresses).

Table 2. Communication speed register (address 273) bit combinations

Bit	19200	9600	19200	38400	57600	115200
0	0	1	0	1	0	1
1	0	0	1	1	0	0
2	0	0	0	0	1	1

Table 3. PWM register (address 100..115) bit meanings

Bit	Meaning
15	Periodical pulse output if set to 1
14	Phase lock (set to 1 to follow the phase)
13..12	Phase (0..3) selection
11..0	Pulse length (1..4095 ms)

Note.
It is possible to synchronize the PWM-operation between multiple I/O-boards, writing the period register 150 of all similar boards using broadcast address 0 (no response will be given).

Registers in the address range from 100 to 115 are also writable using the command “Write Multiple Registers” in addition to single register writes.

Table 4. Comparator output codes

Comparator output	**aicode** **<** **ref1 - hyst**	**ref1 + hyst** **<** **aicode** **<** **ref2 - hyst**	**ref2 + hyst** **<** **aicode** **<** **ref3 - hyst**	**aicode** **>** **ref3 + hyst**
MSB bit	1	0	0	1
LSB bit	0	0	1	1

Notes

- Comparator outputs for each AI channels are are as line health bit in MSB and status bit in LSB of the 16-bit register 500; for example, AI1 state is expressed as bit0 and line health as bit 8 in the register 500, AI2 state is expressed as bit1 and line health as bit 9 in the register 500 and so on;
- aicode is the ADC output code (12 bit) for the AI channel;
  codes ref1..ref3 are the comparator reference (threshold) values, stored in the registers 501..503;
- hyst is the (hysteresis) value used in level checking, stored in the register with the address 504;
- MSB bits with value 1 signal line fault (out of normal voltage range) condition. The normal range is defined by threshold values ref1 and ref3 (stored in registers 501 and 503);
- LSB bits represent the signal value, compared to ref2; equals to 1 if aicode > ref2.
- Use analog reference value (see register 270) higher than or the same as the pull-up voltage on the line. If line pullup is set by the analog channel configuration jumper in position PU5, set register 270 to 0 in order to use the 5V reference from the board power supply.
- Comparator outputs usage instead of reading analog values is especially useful if AI channels are connected to the access control sensors with DEOL (dual end of line) resistors. By reading one register (500) instead of 8 (registers 2...9), all AI channels are checked related to signal level, line cut or short-circuiting. Comparator output examples for line connected to AI1 (read from register 500) follow:
  - 0x0000 No sensor signal, line OK *
  - 0x0001 Sensor signal active. line OK *
  - 0x0100 Line short-circuited
  - 0x0101 Line cut
  - * signal values may be swapped, depending on sensor output contact (NC/NO) usage
- This comparator output bit placement (failure bit in the upper byte and the signal bit in the lower byte of register 500) is valid for firmware version 0x281 (641) and up. In the earlier firmware versions with the comparator (since 0x27E), the comparator output bits were placed sequentially in the same register - for example, AI1 occupying bits 0 and 1 (signal and line failure bits accordingly).
- The maximum value for hysteresis is 29 (reg 500 will have value 0xff00 if the hysteresis is above that).