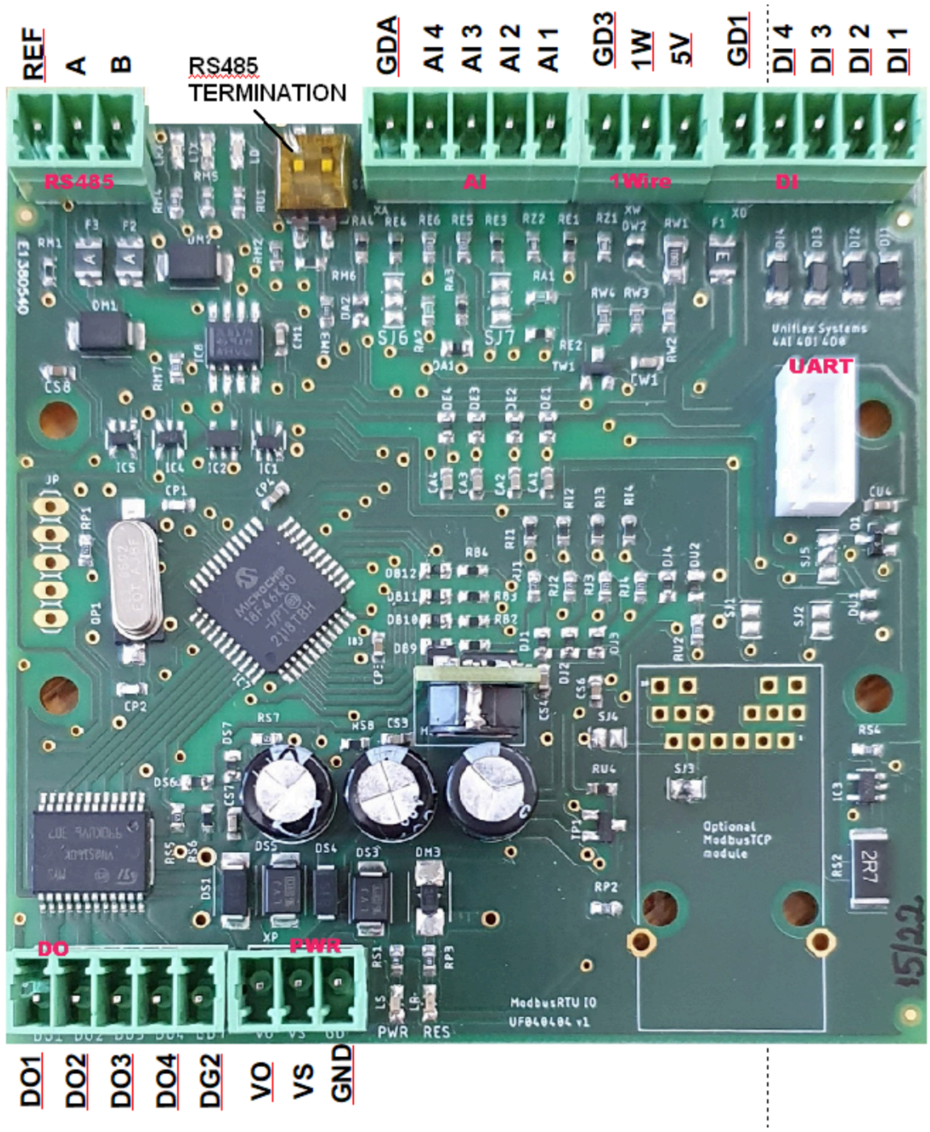
User Manual for Modbus-compatible I/O-modules UF040404 and UF040404E



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

Table of Contents

1. Introduction....2
2. Technical specifications....3
3. Power supply....4
4. Digital communication.... 5
4.1. Modbus-communication.... 5
4.2. Serial (ModbusRTU) communication....5
4.3. Ethernet (ModbusTCP) communication.... 7
4.4. One-wire communication....8
4.5. Wiegand-communication....9
5. Using the discrete inputs.... 10
5.1. Introduction....10
5.2. Discrete level inputs.... 10
5.3. Pulse counters on discrete inputs.... 10
6. Using the discrete outputs.... 11
6.1. Introduction.... 11
6.2. Discrete level outputs.... 11
6.3. Pulse outputs.... 11
7. Using the analog inputs....12
7.1. Introduction....12
7.2. Converting analogue input values to discrete values.... 12
8. Working with 1-wire devices....13
9. Watchdog functionality.... 15
10. Upgrading the firmware.... 16
11. Usage tips....17
11.1. Watchdog registers....17
11.2 Loading the discrete outputs....17
Table 1: Register map.... 18
Table 2. Communication speed register (address 273) bit combinations....23
Table 3. PWM register (address 100..115) bit meanings....23
Table 4. Comparator output codes.... 24
Table 5. AI channel modes....25

1. Introduction

The I/O-board interfaces a number of real-world signals with the registers accessible for digital communication. The digital communication protocol always supported by the board (acting as a slave) is ModbusRTU or both Modbus RTU and TCP with an Ethernet module installed.

There are 4 AI (analog input), 4 DI (discrete input) and 4 DO (discrete output) channels available on this I/O-board. Additionally, the board supports up to 18 temperature sensors DS18B20.

The main supply voltage (socket XP, pin VS) can be any DC voltage in the range of 7 to 28V. The DO supply (socket XP, pin VO) can go up to 35V. The discrete outputs can directly drive resistive or inductive loads with a maximum current of up to 1A per output channel.

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 discrete input (DI) and the ability to **generate single or recurring DO pulses** with a 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 that 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-5 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 XU. The supply voltage for that unit is available on pin 1 of the XU and can be either 5V (1A max) or 3.3V (300 mA max), depending on the connection on the solder-pads SJ5 next to the XU.

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

If the optional Ethernet-module is installed (board type UF040404E), the ModbusTCP protocol becomes supported by the board (acting as a server) as well. The registers on the board can then be accessed either via RS485 or UART (or Ethernet). A query made to one interface is also reflected on another, making the device usable as a ModbusTCP/RTU (or TCP/RTU) gateway.

On the picture on the right, **an example of a cased product** (a Linux-based automation controller UFK04040 with UF040404 as the I/O-board) is presented.

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 VS 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 12..35V DC for the output driver only, to pin VO of the connector XS. The right voltage (with a maximum load of 1A per output channel) depends on the load.

- 4 channels of discrete outputs (DO), available on connector XO.
  Main properties:
  - sourcing with (supply and output) voltage of 12..35V;
  - load current up to 1 A per channel (protected);
  - configurable power-on level;
  - single or periodical pulses available, 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;

- 4 channels of discrete inputs (DI), available on connector XI.

  Main properties:
  - active level high or low (bitwise configurable via setup register);
  - inputs usable for two Wiegand readers;
  - 32-bit counter on every input;
  - maximum input voltage 27V;
  - register-controlled pull-up 1kΩ to 5V;
  - 33kΩ pull-down resistance (input impedance with deactivated pull-up).

- 4 analog input channels (AI) with 12-bit ADC

  - two current inputs 0..20mA;
  - two voltage inputs 0..10 or 0..4V DC

- 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

- OneWire interface for up to 18 DS18B20 temperature sensors

- 5V 1A protected supply voltage output for external devices (via connector XW)

- 5V 1A or 3.3V 300mA protected supply voltage for a master or a communication device with UART interface (available on connector XU)

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

- Board dimensions 84x86mm, 21mm highest point (DC-DC module) fitting into a 5 DIN-rail module wide plastic case from CamdenBoss

- Environment conditions for operation:

  - temperatures ranging from -20 to +60 °C;
  - humidity up to 95% (non-condensing)

- Standards complied 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 one or two DC voltages, the second being to power 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 28V. The current consumption is about 15 mA @12V (via connector XS, pin marked VS). The load will increase if any external consumers are connected to connectors XW or XU, the total power of external loads is limited to 7W.
- The supply voltage for the outputs (via connector XS pin marked VO) ranges from 7 to 36V, consumed current being defined by the DO loads. The maximum possible load is at least 1A per DO channel.

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 an 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 devices. 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.

4.1. Modbus-communication

The primary interface for Modbus-communication is RS485 via connector XM. Depending on the product, an Ethernet-serial converter module (USR K7) 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 (XU) should not be used if the Ethernet-serial converter is installed. When the Ethernet module is installed but the UART interface on socket XU has to be used, soldered jumpers SJ1 and SJ2 must be disconnected first. These jumpers must be short-circuited 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 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 either via 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 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 even if no RS485 line is connected** (to the connector XM), to prevent line voltage floating on the RS485 receiver input..

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 power supply 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 JP2

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.

4.3. Ethernet (ModbusTCP) communication

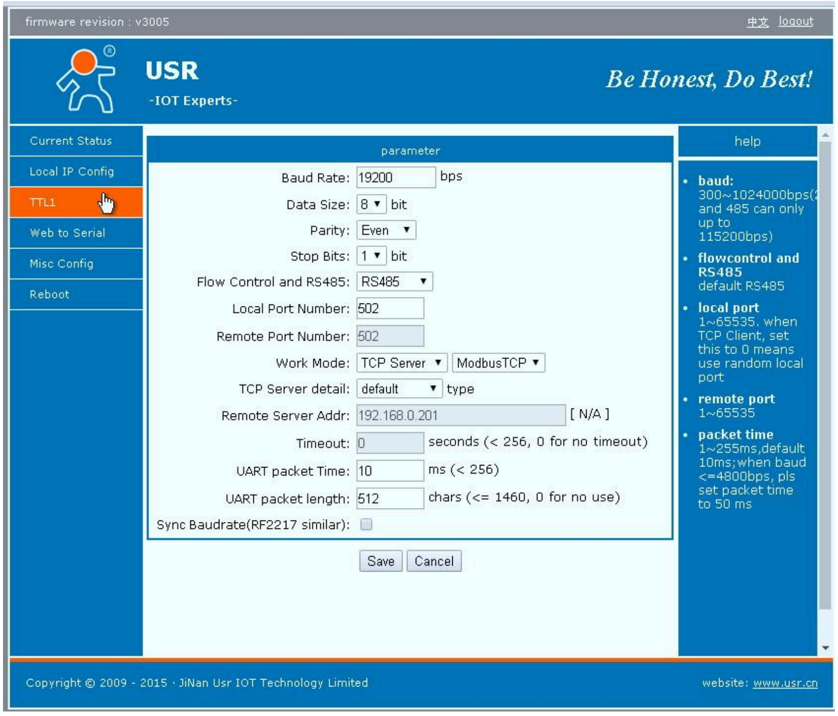
If the I/O-board has the Ethernet/serial converter installed (and solder-jumpers SJ1 and SJ2 closed), then instead of UART signals on XU the Ethernet connection can be used to communicate with the board. The RS485 port still remains functional, transparently reflecting the serial data from the Ethernet/serial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

If it becomes necessary to use the UART connection via the socket XU 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 XU 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.



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 an auto-resettable 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 Table 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 can be switched to the Wiegand mode by setting the according bits in the register with the address 273 (see Appendix 1).

The DI pull-up bits in register 0 must be activated for Wiegand communication. 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 number of discrete inputs is 4. The input data is bit-mapped to the holding register with address 1 to bits 12..15 (for DI1..DI4 respectively).

5.2. Discrete level inputs

The binary 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 number of pulses reaching the inputs are available at 32-bit counters (dedicated to each DI-channel, registers 408..415 for DI1..DI4).

If the number of changes is irrelevant, but the fact of change cannot be missed, a so-called “sticky bit” mode (valid for all inputs simultaneously if activated) can be used. In this case, the changed bit in the register will be fixed until the next register readout. This way we will be able to know which input has been changed since the last 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).

The DI-channels are equipped with a programmable pull-up to the 5V supply via a 1kΩ resistor. Pullups for DI1..DI4 are activated by bits 12..15 of the register 0. Thanks to the diode in the pull-up circuit, input voltages up to 27V are allowed even if the pull-up is unintentionally activated.
If the pull-up is not active, a 33 kΩ pull-down resistor is the load for the input channel. For the inputs used for Wiegand-readers, the pull-up must be activated.

5.3. Pulse counters on discrete inputs

The 32-bit counters are available for every discrete input channel (DI1..DI4). The address range of the counter registers (two holding registers per counter, starting from the most significant word) is from 408 to 415. 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 certain bits in a register with the address 273 (see Appendix 2) have an effect on all binary input channels of the I/O board.

6. Using the discrete outputs

6.1. Introduction

The four sourcing discrete outputs with MOSFET drivers are capable of feeding up to 1 A of current each, at maximum voltage of 35V (the high-state output voltage depends on the supply voltage on pin 2 of the connector XS).

The output channels are individually protected against overload.

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..11) corresponds to the binary outputs DO1...DO4. Note that the LSB (bits 0 to 7) of this register are not used, but the bits 12..15 define the pull-up mode of the discrete inputs DI1..DI4.

6.3. Pulse outputs

Instead of activating the output channel to output a steady voltage, it is also possible to generate voltage pulses on the discrete outputs, both **single** and **periodical** (continuous).

For the generation of single pulses, the registers in the address range 108..111 (corresponding to the binary outputs marked DO1...DO4 accordingly) should be written the pulse length value. The pulse starts immediately on register write and ends when the time for the pulse length has elapsed. To generate an inverted pulse, one can write 1 to the corresponding bit in the register with the address 0 and then start the pulse on the same channel via the corresponding register in the range of 108..111. The actual output is formed as XOR of the bit value in register 0 and the pulse generator output. The maximum pulse length can be up to 4095 timing units. The timing unit can be set from 0.25ms to 16s (using setup register 270).

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 timing units, to be written into the register 150 (this variable is global for all DO-channels of the I/O-board). The value for pulse length must have the most significant bit set in order to enable pulse recurrency (multivibrator mode).

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 possible phases (with a 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 A3 at the end of the document).

7. Using the analog inputs

7.1. Introduction

Four analog input channels are available. All of them can function as 0..4V voltage inputs (given that 4.06 V reference voltage is used), two also support 0..20mA current signal. The rest of the inputs can also accept 0..10V voltage input if the mode is accordingly set via solder jumpers on the PCB.

The default modes via jumper connections when the module is supplied is 0..4V for all channels. Consult Table 5 at the end of this document for the jumper positions setting the AI channel modes.

Keep register 275 content as 0 to use AI-channels properly. The suggested (and default) reference voltage for the ADC is 4.096V, set by the register 270 content. Other options for the reference voltages are 2.048 and 5V (the power supply voltage), having a direct effect on the input ranges.

7.2. 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 the 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 18 DS18B20 temperature sensors are supported. Both data and the full 64-bit one-wire device ID can be read from the registers on the board.

Usually the DS18B20 sensors are used in parasite powering mode, meaning their power pin is connected to the ground. Only ground (pin GD) and data (pin 1W) of the terminal block XW are used in this case. If however the sensors are measuring temperatures more than 60 C, their supply demand is increased and it may be needed to use an active powering scheme for the high-temperature sensors. On the terminal block XW (pin 5V) the 5V supply is available for this.

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 0 to the lowest bit of the register (remove the possible lock), keeping the rest of content;
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 setting bit0 of the register 699 content (the higher bits in this register are for defining the delays during onewire read process).

Be aware that value 0 in the register 279 disables onewire functionality.

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 XU, 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 the 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 XU.

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 and also resets the DI-related counters.

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 Modbus protocol. The suggested way to carry through the update is to use the software written in Python, 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), connected to programming pads JP. This will reset all setup data however.

11. Usage tips

11.1. Watchdog registers

If the I/O-module is used with Linux-board (powered with 5V via XU), then the suggested values for watchdog registers are as follows (see the explanation of the values effect further below):

- 276: 512,
- 277: 9,
- 278: 0,
- 279:10

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

Due to register 278 containing 0, no reset of the IO-board itself will occur in the case of missing Modbus communication.

If no Modbus queries are arriving, then the temperature readings are refreshed after 10 s delay between the individual readings. No onewire temperature readings will be done if register 179 contains value zero!

11.2 Loading the discrete outputs

Each output channel has a maximum load capability of at least 1A with individual thermal protection. The overloaded channel will be temporarily shut down and restored without affecting other channels. The maximum supply voltage for DO is 36 V and may well be different from the main supply voltage (from 9 to 24V).

Table 1: Register map

Register Address	**R** (readable) **W** (writeable) **F** (write to flash)	Description
0	RW	**DO register**, DO1..DO4 controlled by output bits 8..11 (see also register 272 usage) bits 12..15 usable for pull-up activation for DI1..DI4 respectively
1	R	**DI register**, DI1..DI4 are represented by bits 12..15, bit 11 is 1 if supply on XP contact VS is present, bits 0..10 are not in use
2	R	**analog input AI1**, 12-bit ADC with input range 0..20mA
3	R	**analog input AI2**, 12-bit ADC with input range 0..20mA
4	R	**analog input AI3**, 12-bit ADC with input range 0..4 or 0..10V (depends on solder-jumper SJ7)
5	R	**analog input AI4**, 12-bit ADCwith input range 0..4V or 0..10V (depends on solder-jumper SJ6)
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
108..111	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 VO is present (should be 12 V or more)
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 MSB. See the unit definition by the register 270. Note: this register write resets the period counter.
151	R	**Actual state of the DO channels** (bit8..bit11, being the result of XOR of DO and PWM values)

256	R	**Device type** (0xF1)
257	R	**Firmware version**
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 (use with caution) 0x30 for internal Vref=4.096V (the default value) 0x20 for internal Vref=2.048V (use with caution) 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). bit8..bit11 for DO1..DO4
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 to enable max 100 Hz pulse rates and 5 ms minimal pulse length (0=on, effective for all DI channels, max pulse rate 50 Hz, minimal pulse length 10 ms) bit 4 - sticky bit ON (0=off, effective on all DI channels) bit 5 - Wiegand reader 1 enable for inputs DI 1 and 2 bit 6 - Wiegand reader 2 enable for inputs DI 3 and 4 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 with sticky bit 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** (use 0x0000, do not modify)
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 XU 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 XU 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.
279	RWF	Onewire temperature sensors metering period s, value 0 stops all onewire operations. Note: all temperature sensors are read once during this period, to ensure low self-heating.
408, 409	RW	Counter on input DI1, 32 bits
410, 411	RW	Counter on input DI2, 32 bits
412, 413	RW	Counter on input DI3, 32 bits
414, 415	RW	Counter on input DI4, 32 bits
496, 497	R	Time in seconds since the last 5V power break on XU (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 ()
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)
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, to be used when temperature sensor 700 is in use (699 and 700 must not be read in the same query!)
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. Suggested initial value 540. Bit 0 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 enables 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 enabled) according to the line capacitance.
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
999	W	**PIC control register**: PIC reset if the data written to this register is **0xDEAD**, 1-wire discovery starts if the data value is **0xDC1A** (also needs the 1-wire discovery to be enabled by the lowest bit = 0 in the register with the address 699), 5V power cut pulse starts if data is **0xFEED** (pulse length is defined by the content of the register 277).

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. 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.

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 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 output 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).

Table 5. AI channel modes

Jumper	Pads	Open	Connected
SJ8	1-2	AI2 is voltage input (0..4V)	AI2 is current input (0..20mA)
	2-3	AI1 is voltage input (0..4V)	AI1 is current input (0..20mA)
SJ7	1-2	if 2-3 is closed	if 2-3 is open
	2-3	AI3 is voltage input (0..4V)	AI3 is voltage input (0..10V)
SJ6	1-2	if 2-3 is closed	if 2-3 is open
	2-3	AI4 is voltage input (0..4V)	AI4 is voltage input (0..10V)
Note: the current and voltage ranges are valid for the (default) reference voltage of 4.096V for the ADC.			