UPZIO_0001_TP10_ MODBUS

Index

Presentation

- Glass Touch Panel with 10 touch buttons
- Communication via MODBUS® RTU interface
- White LED indication / feedback for each touch button (dimmable)
- RGB LED mode indication on glass plate (dimmable, 65636 colors)
- NFC (Near-Field Communication) for future applications

Integrated sensors:

Sensor	Range
Temperature sensor	-20..+120°C
Temperature accuracy	+/-0.3°C (0-65°C)
Relative humidity	0..100% (+/-3%)
VOC (air quality)	1..4095 PPB
eCO2 (carbon dioxide)	400..4095 PPM
Light	0..16383 LUX

Touch system:

The TP10_ModBus has 10 touch buttons. These buttons have auto calibration at startup and continuous adaptive tracking/calibration during operation. Disabling/enabling the TP10_ModBus via the Master control initiates a recalibration of the touch buttons. Each touch button can be disabled(masked) by register control. For a good functionality of the touch system in all situations, the PLC must be connected to Earth. Sensitivity can be adjusted via a Modbus register. After each adjustment, the system is recalibrated. During this recalibration (10s), the buttons will not respond and should not be touched to prevent incorrect calibration.

Specifications:

- Required voltage supply: 12-24VDC (30VDC Max),

-Inrush current: max 100mA.

- Power consumption: At 24VDC the power consumption is 0,75W max (32mA), and 0,5W typical (22mA). At 12VDC the power consumption is 0,6W max (50mA), and 0,36W typical (30mA)

- Visible dimensions: 87mm * 87mm * 6.7mm,

- Mounting:

    - Recessed depth: 8mm (in standard cavity wall mounting box or mounting box with screw fixing)
    - Use countersunk screws to mount the device into the mounting box
    - When making use of the integrated sensors, it is advised to not place the device above or close to a heating system or places where draught is possible
    - Install the device at a minimum distance of 20cm from electric cabling
    - Metal plates should be earthed
    - Cable entry through wall must be closed off to prevent false measurements through cavity air

- Cabling: 4 wires 0.8mm, (e.g. SVV / SGG 0.8mm 4 wires or UTP)

- Usage temperature 5 .. 40°C

- Storage temperature -10..85°C

- Relative humidity < 90% (non-condensing)

- Protection rating IP30 (indoor usage)

Installation

Wiring

The TP10/RA comes with a backplate that has a plug-in terminal block of 8 poles to establish a connection with the TP10. The TP10 requires 4 wires to be connected to connector 1-4 to work. Another set of wires can then be connected to connector 5-8 of the backplate to loop to the next Modbus device.

The GND of the TP10/RA should be properly grounded. If it is connected to the 0V line of the PLC, then this line should be grounded. A difference in electrical potential between the TP10 and its surroundings can cause the buttons to activate

The info on where to connect each wire can also be found on the backplate itself.

Connector	Function
1 and 5	+12VDC to +24V DC (from power supply)
2 and 6	GND 0V (from power supply)
3 and 7	Modbus A
4 and 8	Modbus B

Mounting the backplate

The backplate can be mounted on a wall via the 2 screw slots in the middle of the backplate, or via the 4 screw holes on each corner of the backplate. The backplate should still have a slight curve, if the backplate is completely flush with the wall, the screws are tightened too hard, and the TP10 can have problems making a good connection with the backplate.

Mounting the TP10

To mount the TP10 onto the backplate, just press it onto the backplate, making sure all 4 protrusions of the backplate are inside the frame of the TP10. The TP10 should then snap onto the backplate. To make sure it is attached properly, press onto all 4 corners of the TP10.

Removing the TP10

To remove the TP10 from the backplate. Grab the TP10 by its sides near the top of the TP10 and pull. The TP10 should snap clear of the backplate.

Visual example

A video example on how to mount a TP10 can be found on our site: https://www.upzio.com/

Modbus settings

Modbus dipswitches

The TP10 has a dipswitch block on the back that can be used to change the settings of the Modbus communication. The first 8 dipswitches can be used to determine the Modbus address. Dipswitch 9 is used to determine the baudrate. Dipswitch 10 switches between an even or an odd parity.

When all dipswitches are set to 0 or all dipswitches are set to 1, the modbus settings from the NFC are used.

Dipswitch	function
1	Address bit 1, adds 1 to the address when activated
2	Address bit 2, adds 2 to the address when activated
3	Address bit 3, adds 4 to the address when activated
4	Address bit 4, adds 8 to the address when activated
5	Address bit 5, adds 16 to the address when activated
6	Address bit 6, adds 32 to the address when activated
7	Address bit 7, adds 64 to the address when activated
8	Address bit 8, adds 128 to the address when activated
9	Baudrate: off = 9600, on = 19200
10	Parity: off = even parity, on = odd parity

Modbus NFC Settings

The NFC chip on the TP10 can be used to modify all modbus settings:

- Device address 1-247
- Baudrate 1200/2400/4800/9600/19200/28800/38400/57600/115200
- No parity (1-2 stopbit) / Odd parity (1 stopbit) / Even parity (1 stopbit)

Standard settings:

All dipswitches are off, so the TP10 uses these NFC standard settings:

- Address: 1
- Baudrate= 9600
- Stopbits: 1
- Parity: even

Modbus Registers

2's compliment

Some registers of the TP10 (temperature and dewpoint) are saved as a two's compliment value. The sum of a number and its two's complement is 2N. For instance, for the three-bit number 010, the two's complement is 110, because 010 + 110 = 1000. The two's complement is calculated by inverting the digits and adding one.

Example table:

Register value	Converted decimal value
0000 0000 0000 0000	0
0000 0000 0000 0001	1
0000 0000 0000 0010	2
0000 0000 0111 1110	126
0000 0000 0111 1111	127
1000 0000 0000 0000	-32768
1000 0000 0000 0001	-32767
1000 0000 0000 0010	-32766
1111 1111 1111 1110	−2
1111 1111 1111 1111	−1

Holding registers/ Input registers

All these register values can be found in both the holding registers and the input registers. They can only be read via input registers, but they can be read or written with holding registers.

register	Name	factor (value = register/factor)	read/write	description	example
1	Buttons	/	r	contains all button values, starting from the lowest bit. bit 0 = button 1, bit 9 = button 10	0000 0000 0000 0100 = button 3 is pushed
2	Temperature [°C]	10	r	Measured temperature, saved as two's compliment	223 = 22.3 °C
3	eCO2 [ppm]	10	r	estimated CO2 value	7800 = 780 ppm
4	VOC]ppb]	10	r	Volatile organic compounds value	15000 = 1500 ppb
5	Illuminance [Lux]	10	r	Illuminance value	10000 = 1000 lux
6	Humidity [%]	10	r	humidity	458 = 45.8 % humidity
7	dewpoint [°C]	10	r	The temperature at which air will condensate. Saved as two's compliment	223 = 22.3 °C
8	button LEDs	/	r/w	settings for all LEDs placed on the buttons, starting from the lowest. bit 0 = led on button 1, bit 9 = led on button 10	0000 0011 0000 0100 = led is active on buttons 3, 9 and 10
9	led masks (disables buttons)	/	r/w	disables the buttons, starting from the lowest. bit 0 = disables button 1, bit 9 = disables button 10	0000 0000 0011 0010 = buttons 2, 5 and 6 cannot be used.
10	RGB red(0-255)	/	r/w	Red color setting for all RGB LEDs on the TP10. 0 = lowest value, 255 = maximum value. This can be combined with the other RGB registers to create various colors	255 = TP10 gives of a bright red light
11	RGB green(0-255)	/	r/w	Green color setting for all RGB LEDs on the TP10. 0 = lowest value, 255 = maximum value. This can be combined with the other RGB registers to create various colors	0 = the TP10 gives of no green light

12	RGB Blue(0-255)	/	r/w	Blue color setting for all RGB LEDs on the TP10. 0 = lowest value, 255 = maximum value. This can be combined with the other RGB registers to create various colors	255 = TP10 gives of a bright blue light
13	Button count general	/	r	Number of times any button has been pressed	387 = there have been 387 button presses on the buttons
14	Button count 1	/	r	Number of times button 1 has been pressed	54 = button 1 has been pressed 54 times
15	Button count 2	/	r	Number of times button 2 has been pressed	31 = button 2 has been pressed 31 times
16	Button count 3	/	r	Number of times button 3 has been pressed	0 = button 3 has been pressed 0 times
17	Button count 4	/	r	Number of times button 4 has been pressed	47 = button 4 has been pressed 47 times
18	Button count 5	/	r	Number of times button 5 has been pressed	11 = button 5 has been pressed 11 times
19	Button count 6	/	r	Number of times button 6 has been pressed	158 = button 6 has been pressed 158 times
20	Button count 7	/	r	Number of times button 7 has been pressed	60 = button 7 has been pressed 60 times
21	Button count 8	/	r	Number of times button 8 has been pressed	1 = button 8 has been pressed 1 time
22	Button count 9	/	r	Number of times button 9 has been pressed	23 = button 9 has been pressed 23 times
23	Button count 10	/	r	Number of times button 10 has been pressed	2 = button 10 has been pressed 2 times
24	Button sound level (0-255)	/	r/w	feedback volume of sound played when pressing a button on the TP10. 0 = no sound, 255 = maximum volume	255 = standard value, the buttons make an audible noise when pressed.

25	button sensitivity (1-100)	/	r/w	sensitivity of the TP10. 10 = very sensitive buttons, 100 = the buttons react only when pressed firmly.	55 = standard value, the buttons react when lightly pressing the TP10
26	button led intensity (0-255)	/	r/w	intensity of the white LEDs on the buttons of the TP10. 0 = no light, 255 = maximum light emitted.	128 = standard value, the buttons light up brightly when activated.
…					
1000	Device identifier (major)	/	r	This register identifies the vendor. All upzio devices will have 0 on this register.	0 = upzio device
1001	Device identifier (minor)	/	r	This register identifies the device type. All upzio device types will have a different value in this register.	1 = TP10
1002	reset device	/	r/w	when bit 0 is set to 1, the device resets and reverts the value of this register to 0.	0000 0000 0000 0001 = device resets
1003	Locate device	/	r/w	when bit 0 is set to 1, the device lights up and changes colors rapidly, allowing it to be found quickly. (for installation purposes)	0000 0000 0000 0001 = device lights up and changes colors.
1004	Voltage level	10	r	The voltage the device receives, should be close to 24V	256 = 25.6V
1005	Version Hardware MSB	/	r	Most significant bits of the hardware version, when added to the least significant bits (shifted 16 bits) these form a date.	20200506 = 06/05/2020
1006	Version Hardware LSB	/	r	Least significant bits of the hardware version	
1007	Version firmware MSB	/	r	Most significant bits of the firmware version	20200709 = 09/07/2020
1008	Version firmware LSB	/	r	Least significant bits of the firmware version	
1009	Version registers MSB	/	r	Most significant bits of the registers	20200506 = 06/05/2020

1010	Version registers LSB	/	r	Least significant bits of the registers	
1011	UniqueID (byte 11 and 10)	/	r	These registers add up to form a Unique identifier number for the device.	Unique ID of example device: 16#0020, 16#003F, 16#5706, 16#4D41, 16#3937, 16#2031
1012	UniqueID (byte 9 and 8)	/	r		
1013	UniqueID (byte 7 and 6)	/	r		
1014	UniqueID (byte 5 and 4)	/	r		
1015	UniqueID (byte 3 and 2)	/	r		
1016	UniqueID (byte 1 and 0)	/	r		
1017	Error count	/	r	amount of errors that have occurred in the device	54 = there have been 54 errors since last startup of the TP10
1018	Error code	/	r	code of the last error that has occurred in the device	13 = error 13 was the last error that happened in the device

Coils/ Discrete input

All these bit values can be found in both the coils (00001 - 09999) and the discrete inputs (10001 - 19999). They can only be read via discrete inputs, but they can be read or written with coils.

Coil/ discrete input	Name	read/write	description	example
1	Button 1	r	Contains the state of button 1	0 = button not pressed, 1 = button pressed
2	Button 2	r	Contains the state of button 2	
3	Button 3	r	Contains the state of button 3	
4	Button 4	r	Contains the state of button 4	
5	Button 5	r	Contains the state of button 5	
6	Button 6	r	Contains the state of button 6	
7	Button 7	r	Contains the state of button 7	
8	Button 8	r	Contains the state of button 8	
9	Button 9	r	Contains the state of button 9	
10	Button 10	r	Contains the state of button 10	
11	Button led 1	r/w	Setting for led placed on button 1	0 = led on button off, 1 = led on button on
12	Button led 2	r/w	Setting for led placed on button 2	
13	Button led 3	r/w	Setting for led placed on button 3	
14	Button led 4	r/w	Setting for led placed on button 4	
15	Button led 5	r/w	Setting for led placed on button 5	
16	Button led 6	r/w	Setting for led placed on button 6	
17	Button led 7	r/w	Setting for led placed on button 7	
18	Button led 8	r/w	Setting for led placed on button 8	
19	Button led 9	r/w	Setting for led placed on button 9	
20	Button led 10	r/w	Setting for led placed on button 10	
21	led mask 1	r/w	disables button 1	

22	led mask 2	r/w	disables button 2	0 = button active, 1 = button disabled, the button won’t activate or give feedback
23	led mask 3	r/w	disables button 3	
24	led mask 4	r/w	disables button 4	
25	led mask 5	r/w	disables button 5	
26	led mask 6	r/w	disables button 6	
27	led mask 7	r/w	disables button 7	
28	led mask 8	r/w	disables button 8	
29	led mask 9	r/w	disables button 9	
30	led mask 10	r/w	disables button 10	
...				
1002	reset TP10	r/w	when set to 1, the TP10 resets and reverts the value of this register to 0.	1 = TP10 resets
1003	Locate TP10	r/w	when bit 0 is set to 1, the TP10 lights up and changes colors rapidly, allowing it to be found quickly. (for installation purposes)	1 = TP10 lights up and changes colors.

Recommended communication method

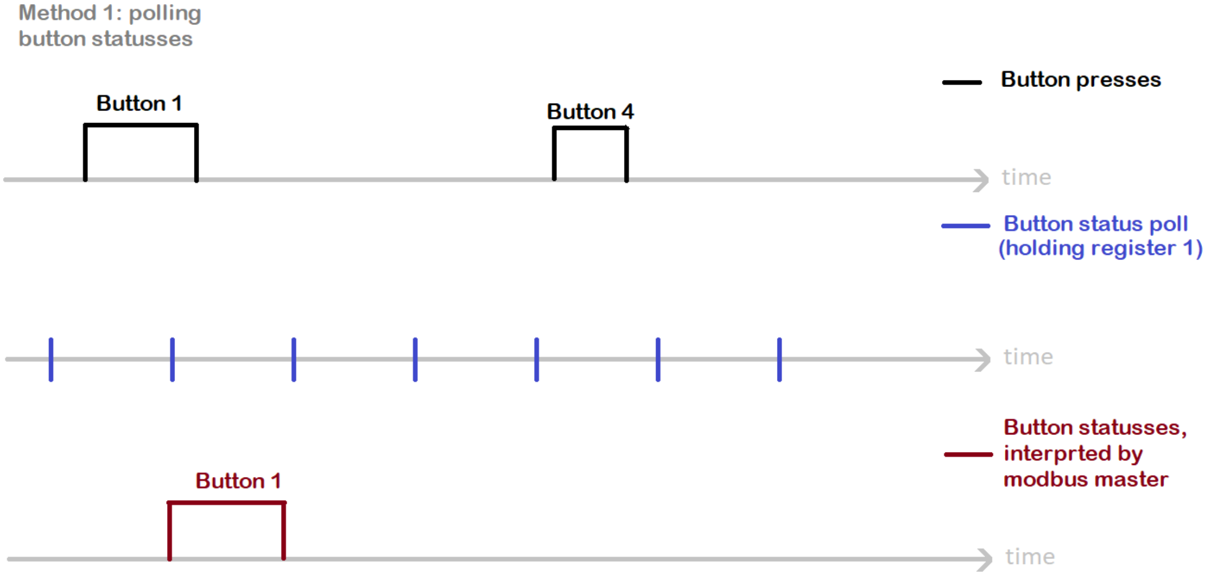
Modbus is a polling protocol, the modbus slaves cannot send messages if the modbus mast did not ask the slave to send a message. Because of this, the buttons must be polled frequently to get a decent reaction time. All other registers can be polled less frequently because a fast response time is not required for the other registers. To implement the TP10 with a Beckhoff PLC, it is recommended to use the libraries available on our website www.upzio.com.

There two kinds of registers that can be used to read out the button statuses:

1. Holding/input register 13 – 23 are button counters
2. Holding/input register 1 and coil/discrete register 1-10 are the button statuses.

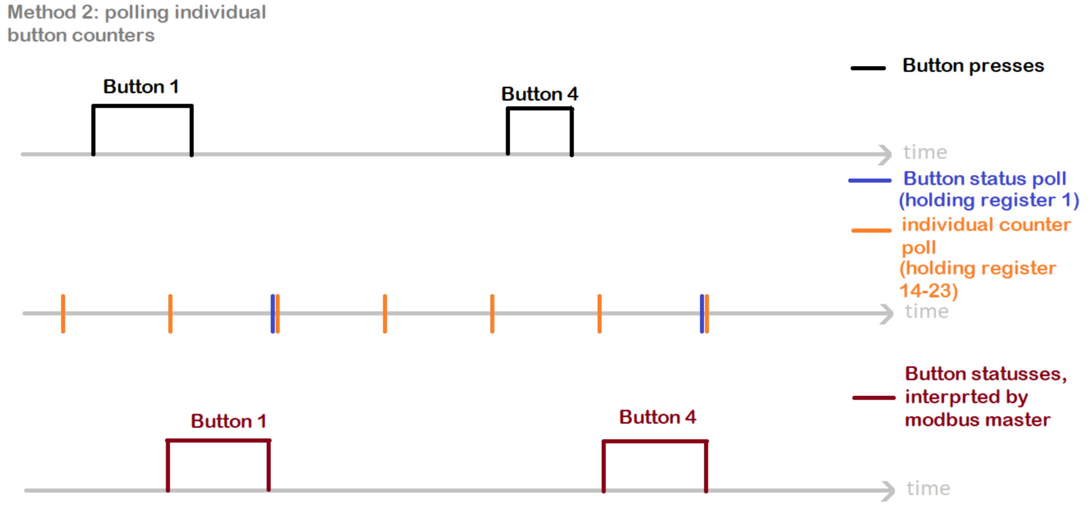
Reading out the buttons can be done in three ways:

1. A simple implementation can be done by simply polling the button statuses. But this method might miss a button if the button press is shorter than the polling time. The image below shows the principle of this method, and the problem of missing a button.



2. A better but more advanced implementation can be done by polling the individual button counters. A change on one of the button counters can be interpreted as a button press. If a button is pressed, the button statuses can be polled to know how

long a button is being pressed. The image below shows the principle of this method.



3. The best but most advanced implementation is to only poll the general button counter if no buttons are being pressed. Polling one register will make your messages shorter and faster, which will allow for a faster response time or more TP10s on one modbus line. If the general button counter changed, the individual button counters can be polled once. A change in the individual button counters can be interpreted as a button press. If a button is pressed, the button statuses can be polled to know how long a button is being pressed.