

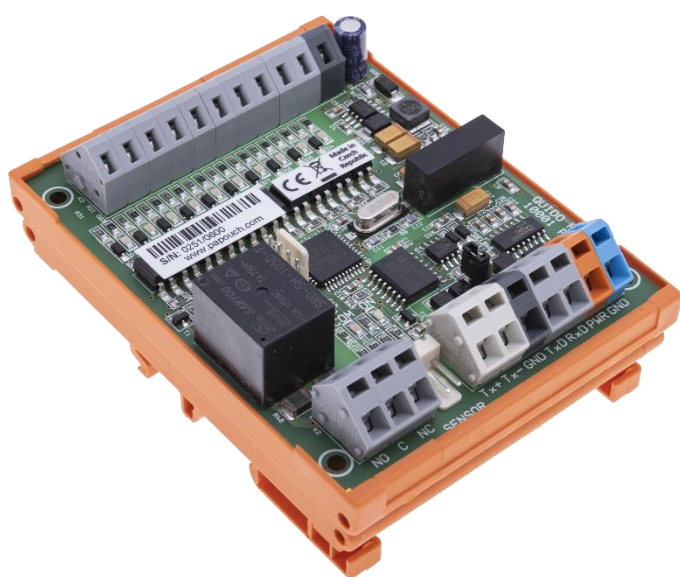
---

# Quido - MODBUS

---

Kompletní popis protokolů MODBUS RTU a TCP  
v I/O modulech Quido

---



# Quido - MODBUS

## Katalogový list

Vytvořen: 9.11.2009

Poslední aktualizace: 7.12.2022 15:15

Počet stran: 16

© 2022 Papouch s.r.o.

---

## Papouch s.r.o.

Adresa:

**Strašnická 3164/1a  
102 00 Praha 10**

Telefon:

**+420 267 314 268**

Internet:

**www.papouch.com**

E-mail:

**papouch@papouch.com**



**OBSAH**

Přehled změn v tomto dokumentu.....	3
Modbus v I/O modulech Quido.....	4
Komunikační parametry .....	4
Konfigurační propojky .....	4
Přepnutí protokolů.....	6
Spinel → MODBUS RTU .....	6
Povolení konfigurace .....	6
Přepnutí .....	6
MODBUS RTU → Spinel .....	6
Registry .....	7
Adresa .....	7
Přehled funkčních kódů .....	7
Identifikace zařízení .....	7
Holding Register .....	8
Input register.....	12
Discrete Inputs.....	12
Coils .....	13

**Přehled změn v tomto dokumentu****Verze 4.50<sup>1</sup>**

- Možnost [adresace sériovým číslem](#).
- Popis [konfiguračních propojek](#) na desce.
- Nový registr *Run time* s dobou běhu zařízení.
- Rozšířeny statusy měřené teploty.
- Přidány registry s počtem vstupů, výstupů a teploměrů.
- Teploměry, stavy vstupů a výstupů jsou nově k dispozici i v Holding registru.

<sup>1</sup> Verzí se má na mysli *[hw-version].[sw-version]* jak ji lze načíst pomocí funkce Identifikace zařízení (0x11). Tj. zařízení s identifikací Quido RS 4/4; v0209.04.50; f66 97; t1 je verze 4.50.

## MODBUS V I/O MODULECH QUIDO

Tento dokument popisuje komunikační protokoly MODBUS RTU a MODBUS TCP v I/O modulech Quido.

- Dokumentace hardwaru a popis funkcí I/O modulů je k dispozici na [papouch.com](http://papouch.com).
- Detailní informace o protokolu MODBUS jsou k dispozici na stránce [modbus.org](http://modbus.org).
- Pro vyzkoušení komunikace s Quidem přes Modbus můžete použít například [ModScan32](http://ModScan32).

### Komunikační parametry

#### Quida RS a Quida USB

Následující parametry se týkají Quid s rozhraním RS232, RS485 nebo USB.

Komunikační rychlost Quid RS .....rozsah od 1200 Bd do 230,4 kBd (výchozí: 9,6 kBd)

Komunikační rychlost Quid USB .....fixní 115,2 kBd

Počet datových bitů .....8

Parita .....bez parity, sudá, lichá (výchozí: bez parity)

Počet stopbitů .....1, 2 (výchozí: 1)

Výchozí adresa .....0x31

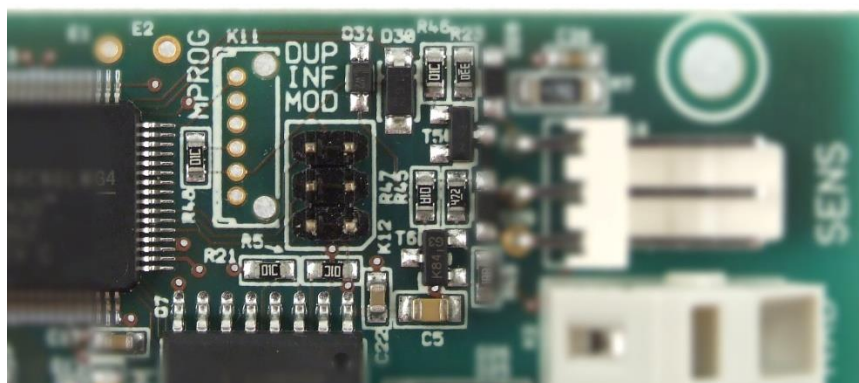
Výchozí protokol nastavený z výroby .....Spinel (jak [přepnout do Modbusu](#) viz str. 6)

#### Quida ETH

Quida s Ethernetovým rozhraním umí v režimu WEB komunikovat protokolem MODBUS TCP. Port pro komunikaci je nastavitelný na záložce *Ostatní*. Jako výchozí hodnota je nastaven port číslo 502.

### Konfigurační propojky

Na Quidech se od roku 2020 postupně po revizích hardwaru objevují konfigurační propojky, které usnadňují použití Quid v některých typických situacích.



obr. 1 - příklad propojek na Quido RS 2/2

Propojky jsou tři – Duplex, Info a Modbus. (Jak je vidět na obrázku, někdy je popisek v potisku mírně zkrácený.)

### Propojka Modbus

Pokud je v okamžiku zapnutí napájení tato propojka zkratována, komunikuje Quido protokolem Modbus bez ohledu na softwarovou konfiguraci.

### Propojka Info

Pokud je při připojeném napájení tato propojka krátce zkratována, Quido pošle na sériovou linku aktuální nastavení komunikačních parametrů.<sup>2</sup> Tato informace se posílá vždy v protokolu Spinel. RS verze zařízení posílá informaci na rychlosti 9600 Bd, USB a ETH verze na 115,2 kBd.

Quido odešle nejdříve odpověď na instrukci *Jméno a verze* a poté ještě paket, kde je v datech uvedena v ASCII formátu adresa, rychlost a protokol. Příklad:

```
*a?"4N?Address:34 Speed:6 Protocol:1ü?
```

Adresa je hexadecimální, rychlost je kód dle instrukce *Komunikační parametry* a protokol je číslo protokolu podle instrukce *Přepnutí komunikačního protokolu*.

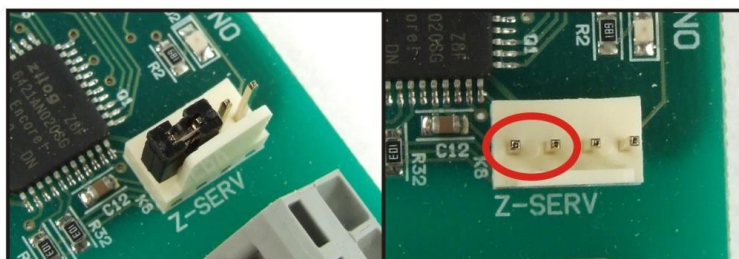
Propojka Info nesmí být zkratována při startu nebo restartu zařízení!

### Propojka Duplex

Propojkou se aktivuje režim obousměrného přenosu stavu vstupů a výstupů mezi dvěma Quidy 4/4 nebo 8/8. Sadu je možné objednat jako [QuidoDuplexRS](#) nebo [QuidoDuplexETH](#). V manuálech u těchto sad jsou další informace ohledně konfigurace tohoto režimu.

---

<sup>2</sup> Na dřívějších verzích Quid je možné stejnou akci vyvolat zkratováním dvou pinů na konektoru Z-SERV. Umístění pinů na starších Quidech je patrné z obrázku:



## PŘEPNUTÍ PROTOKOLŮ

Výchozím protokolem u Quid RS a USB je z výroby nastaven Spinel. Pro přepnutí do protokolu MODBUS RTU slouží následující instrukce z protokolu Spinel. Quido lze pohodlně přepnout do protokolu Modbus (nebo zpět) pomocí programu [Modbus configurator](#) (na stránce jsou i [příklady pro přepínání protokolů v Pythonu](#)) nebo zkratováním [jumperu Modbus](#) na desce elektroniky.

### Spinel → MODBUS RTU

#### Povolení konfigurace

Povoluje provedení servisní instrukce. Musí předcházet bezprostředně instrukci Přepnutí. Instrukci nelze použít s univerzální adresou nebo s adresou „broadcast“.

#### Zápis

Struktura:	→ <b>0xE4</b>
Příklad:	→ 2A 61 00 05 31 02 <b>E4</b> 58 0D ← 2A 61 00 05 31 02 <b>00</b> 3C 0D

#### Přepnutí

Přepnutí protokolu se provádí speciální instrukcí protokolu Spinel, formátu 97. Jako adresa musí být použita adresa konkrétního modulu (nelze použít tzv. „broadcast“ ani universální adresu). Instrukci musí bezprostředně předcházet instrukce „Povolení konfigurace“.

#### Parametry

protocol	1 byte	Identifikační číslo protokolu: <ul style="list-style-type: none"> <li>• 0x01: Spinel, formát 97 (binární) a 66 (ascii)</li> <li>• 0x02: Modbus RTU (jen pro Quida RS a Quida USB)</li> <li>• 0x0A: Spinel, pouze formát 97 (binární)</li> </ul>
----------	--------	---

#### Zápis

Struktura:	→ <b>0x2A</b> , protocol
Příklad:	→ 2A 61 00 06 31 02 <b>ED 02</b> 4C 0D <ul style="list-style-type: none"> <li>• 0x02: Přepnutí do protokolu Modbus RTU.</li> </ul> ← 2A 61 00 05 31 02 <b>00</b> 3C 0D

### MODBUS RTU → Spinel

Přepíná se zápisem do [Holding registru 0x0005](#) – viz str. 9.

## REGISTRY

### Adresa

- 0x31: Výchozí adresa zařízení (dekadicky 49). Adresu jde změnit v registru 1 (viz níže).
- 0x00: Univerzální adresa protokolu Modbus RTU (dekadicky 0). Pokud zařízení přijme tuto adresu, instrukce se provede, ale zařízení neodpoví.
- 0xF8: Univerzální adresa zařízení (dekadicky 248). Pokud zařízení přijme tuto adresu, instrukce se provede a zařízení odpoví. Toto lze prakticky využít jen pokud je připojeno pouze jedno zařízení!

### Jak změnit adresu pomocí sériového čísla? (Pouze pro Modbus RTU)

Díky následujícímu postupu je možné zapojit na linku RS485 více zařízení se stejnou adresou a pak jim jednotlivě adresu změnit:

- 1) Poznamenejte si sériové číslo zařízení. Je na štítku na zařízení ve formátu 1395/0069  
Číslo před lomítkem je Typ produktu a číslo za lomítkem je Číslo kusu.
- 2) Pomocí funkčního kódu 0x10 a univerzální adresy 0xF8 zapište do zařízení najednou tyto Holding registry:
  - a. *Typ produktu* (adr. 10) – zapište typ produktu ze štítku.
  - b. *Číslo kusu* (adr. 11) – zapište číslo kusu ze štítku.
  - c. *Adresa* (adr. 12) – zapište novou adresu, kterou chcete nastavit.
- 3) Od této chvíle komunikuje zařízení již s novou adresou.

### Přehled funkčních kódů

Zařízení umožňuje přistupovat ke své paměti – v závislosti na typu registru – těmito instrukcemi:

- 0x01 .....čtení coils
- 0x02 .....čtení diskretních vstupů
- 0x03 .....čtení holding registrů
- 0x04 .....čtení vstupních registrů
- 0x05 .....nastavení jednoho z diskretních výstupů <sup>3</sup>
- 0x06 .....nastavení jednoho holding registru <sup>3</sup>
- 0x0F .....nastavení několika diskretních výstupů najednou
- 0x10 .....zapsání do několika holding registrů
- 0x11 .....identifikace <sup>3</sup>

### Identifikace zařízení

Čtení identifikačního řetězce zařízení (Report slave ID).

#### Funkční kódy:

0x11 – Report slave ID

#### Parametry:

Počet bytů	1 Byte	dle řetězce
ID	1 Byte	ID je totožné s adresou zařízení
RI	1 Byte	Run Indikátor – zde vždy 0xFF (zapnuto)

<sup>3</sup> Tento funkční kód lze použít jen u MODBUS RTU.

Data	N Byte	Řetězec stejný jako v protokolu Spinel. Tedy například: Quido RS 4/4; v0209.04.50; f66 97; t1
------	--------	--

**Holding Register**

Adresa	Přístup	Funkce	Název																								
0 <sup>4</sup>	zápis	0x06	<p><b>Povolení konfigurace</b> Zápis hodnoty 0x00FF do tohoto paměťového místa musí předcházet všem instrukcím, zapisujícím do holding registru na adresy 0 až 15. Slouží k ochraně před nechtěnou změnou konfigurace. Není povoleno zapisovat Povolení konfigurace pomocí Multiply write zároveň s dalšími parametry!</p>																								
1	čtení, zápis	0x03, 0x06, 0x10	<p><b>Adresa (ID)<sup>5, 6</sup></b> Unikátní adresa zařízení v protokolu Modbus. Je očekáváno číslo z rozsahu 1 až 247. Adresa je unikátní pro protokol Modbus. <i>Výchozí adresou je 0x0031.</i> Pro nastavení adresy pomocí sériového čísla viz stranu 7.</p>																								
2	čtení, zápis	0x03, 0x06, 0x10	<p><b>Komunikační rychlost<sup>5, 6</sup></b> Rychlosti a jim odpovídající kódy: 1 200 Bd .....0x0003 2 400 Bd .....0x0004 4 800 Bd .....0x0005 9 600 Bd .....0x0006 (<i>výchozí nastavení</i>) 19 200 Bd .....0x0007 38 400 Bd .....0x0008 57 600 Bd .....0x0009 115 200 Bd .....0x000A 230 400 Bd .....0x000B</p>																								
3	čtení, zápis	0x03, 0x06, 0x10	<p><b>Datové slovo<sup>5, 6</sup></b> Datové slovo je vždy osmibitové.</p> <table border="1"> <thead> <tr> <th>Hodnota</th> <th>Parita</th> <th>Počet stopbitů</th> </tr> </thead> <tbody> <tr> <td>0x0000 (<i>výchozí</i>)</td> <td>není (N)</td> <td>1</td> </tr> <tr> <td>0x0001</td> <td>sudá (E)</td> <td>1</td> </tr> <tr> <td>0x0002</td> <td>lichá (O)</td> <td>1</td> </tr> <tr> <td>0x0003</td> <td>není (N)</td> <td>2</td> </tr> <tr> <td>0x0004</td> <td>sudá (E)</td> <td>2</td> </tr> <tr> <td>0x0005</td> <td>lichá (O)</td> <td>2</td> </tr> <tr> <td>0x0006 až 0x00FF</td> <td>není (N)</td> <td>1</td> </tr> </tbody> </table>	Hodnota	Parita	Počet stopbitů	0x0000 ( <i>výchozí</i> )	není (N)	1	0x0001	sudá (E)	1	0x0002	lichá (O)	1	0x0003	není (N)	2	0x0004	sudá (E)	2	0x0005	lichá (O)	2	0x0006 až 0x00FF	není (N)	1
Hodnota	Parita	Počet stopbitů																									
0x0000 ( <i>výchozí</i> )	není (N)	1																									
0x0001	sudá (E)	1																									
0x0002	lichá (O)	1																									
0x0003	není (N)	2																									
0x0004	sudá (E)	2																									
0x0005	lichá (O)	2																									
0x0006 až 0x00FF	není (N)	1																									

<sup>4</sup> Je možné setkat se i s číslováním registrů od jedničky, protože tento první registr má adresu 0.

<sup>5</sup> Zápisu do tohoto paměťového místa musí předcházet zápis hodnoty 0x00FF na adresu 0 do pozice Povolení konfigurace. Jde o ochranu před nechtěnou změnou konfigurace. Není povoleno zapisovat Povolení konfigurace pomocí Multiply write zároveň s dalšími parametry. **Po zápisu dojde k restartu zařízení a tím i k vynulování čítačů!**

<sup>6</sup> Toto nastavení se ukládá do paměti typu FLASH. To znamená, že (1) nastavení se pamatuje i po vypnutí napájení, a (2) počet zápisů do paměti je omezený (typicky jen několik desítek tisíc zápisů).



Adresa	Přístup	Funkce	Název
4	čtení, zápis	0x03, 0x06, 0x10	<b>Rozlišení konce paketu</b> <sup>5,6</sup> Konfiguruje, jak velká prodleva mezi byty bude považována za konec paketu. Prodleva se zadává v počtu bytů. Je možné zadat hodnotu 4 až 100. Výchozí hodnota je 10.
5	čtení, zápis	0x03, 0x06, 0x10	<b>Komunikační protokol</b> <sup>5,6</sup> Umožňuje přepnout zařízení do komunikace protokolem Spinel. Po odeslání odpovědi se zařízení přepne do zvoleného protokolu a dále komunikuje pouze jím. (V každém z protokolů existuje instrukce pro přepnutí protokolů.) Kód pro protokol <i>Spinel</i> : 0x0001 (výchozí) Kód pro protokol Modbus RTU: 0x0002 Pokud je zkratována <a href="#">propojka Modbus na desce elektroniky</a> , komunikuje zařízení vždy Modbusem, bez ohledu na stav tohoto registru!
10 pouze RTU	čtení, zápis	0x03, 0x10 <sup>7</sup>	<b>Typ produktu</b> Typové číslo produktu.
11 pouze RTU	čtení, zápis	0x03, 0x10 <sup>7</sup>	<b>Číslo kusu</b> Unikátní číslo kusu.
12 pouze RTU	čtení, zápis	0x03, 0x10 <sup>7</sup>	<b>Adresa</b> Viz nastavení adresy pomocí sériového čísla na straně 7.
13 pouze RTU	čtení	0x03	<b>Počet vstupů</b>
14 pouze RTU	čtení	0x03	<b>Počet výstupů</b>
15 pouze RTU	čtení	0x03	<b>Počet teploměrů</b>
100 – 160	<i>Modbus TCP:</i> jen čtení  <i>Modbus RTU:</i> čtení i zápis	<i>Modbus TCP:</i> 0x03  <i>Modbus RTU:</i> 0x03, 0x06, 0x10	<b>Stav čítačů</b> <i>Čítač umožňuje počítat jednotlivé změny stavu vstupu. Za změnu je považována změna logického stavu (nebo stavu připojeného kontaktu). Každý vstup má vlastní čítač. K hodnotě čítače je přičtena jednička při vybraných změnách na příslušném vstupu (změna z 1 do 0; změna z 0 do 1; případně obě změny).</i> Zde jsou uloženy aktuální stavy 16bitových čítačů na všech vstupech. (Počítání je z výroby vypnuto.) Celkový počet registrů odpovídá počtu vstupů Quida. U Quida s deseti vstupy tedy bude použitých deset 16bit registrů. Maximální počet čítačů je 60. (Na případných dalších vstupech nejsou čítače k dispozici.) Nulování čítačů se provádí zápisem nulové hodnoty. Čítače jsou vynulovány také při vypnutí nebo restartem zařízení.

<sup>7</sup> Registry 10 až 12 je nutné zapisovat najednou. Zápis nepřepíše hodnoty registrů Typ produktu a Číslo kusu. Zápis do těchto registrů slouží pouze pro funkci nastavení adresy zařízení pomocí sériového čísla (viz str. 6).

Adresa	Přístup	Funkce	Název				
			<p>Doporučený postup pro průběžné čtení aktuálního stavu čítačů:</p> <ol style="list-style-type: none"> <li>1) Přečtete hodnotu z tohoto registru Stav čítačů.</li> <li>2) Přečtenou hodnotu odečtete pomocí následujícího registru <i>Odečet od čítače</i>.</li> </ol> <p>Tímto postupem nepřijdete o žádný záznam změny na vstupu.</p>				
200 – 260	zápis	0x06, 0x10	<p><b>Odečet od čítačů</b>                      Adresa 200 patří ke vstupu IN1.                      Odečte zadanou hodnotu od aktuálního stavu čítače. (Hodnota, která má být odečtena, nesmí být větší než aktuální stav čítače. Při použití funkce 0x10 (Multiply write) nelze zapsat víc než 12 registrů najednou.)</p>				
300 – 360	čtení, zápis	0x03, 0x06, 0x10	<p><b>Konfigurace čítače</b> <sup>6</sup>                      Na adrese 300 je čítač vstupu IN1.</p> <ul style="list-style-type: none"> <li>• 0: čítač na tomto vstupu je <u>vypnutý</u></li> <li>• 1: čítač přičte jednotku při každé zaznamenané <u>náběžné hraně</u> na vstupu</li> <li>• 2: čítač přičte jednotku při každé zaznamenané <u>sestupné hraně</u> na vstupu</li> <li>• 3: čítač přičte jednotku <u>při každé</u> zaznamenané hraně (náběžné <u>i</u> sestupné) na vstupu</li> </ul>				
500 – 532	čtení, zápis	0x03, 0x06, 0x10  (i pomocí 0x10 pouze jeden výstup!)	<p><b>Nastavení jednoho výstupu na určitou dobu</b>                      Aktivuje vybraný výstup na určitou dobu – spustí na vybraném výstupu pulz zadané polarity na zadanou dobu. Pulz se spustí okamžitě po přijetí této instrukce. Opětovné spouštění pulzu, když ještě neskončil předchozí, je možné.</p> <p><b>Horní byte</b>                      0xFF → Sepnutí                      0x00 → Rozepnutí</p> <p><b>Dolní byte</b>                      0x00 až 0xFF → Doba, po kterou má být výstup sepnutý nebo rozepnutý (podle horního bytu). Jednotkou je 0,5 sec. Je tedy možné nastavit sepnutí na 0,5 až 127,5 sec.</p>				
600 – 728	čtení, zápis	0x03, 0x10 Jak při čtení, tak při zápisu je dovoleno zapisovat/číst pouze celou čtveřici registrů najednou!	<p><b>Termostat</b> <sup>6</sup>                      Po sobě následující čtveřice registrů se vztahují k jednotlivým výstupům a nastavení termostatu pro ně. Pro první výstup Quida se jedná o registry 600 (první) až 603 (čtvrtý).</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Registr</th> <th style="width: 75%;">Význam</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> </tr> </tbody> </table>	Registr	Význam		
Registr	Význam						

Adresa	Přístup	Funkce	Název										
			<p>Formát <u>dolního</u> bytu: <b>FSSKTTTT</b></p> <p>„<b>F</b>“ – Zapnuta/vypnuta funkce hlídání teploty pro výstup (OUTx); (1 = zapnuto; 0 = vypnuto)</p> <p>„<b>SS</b>“ – Akce, která se má při nastavené teplotě provést  00 = sepnout výstup  01 = rozepnout výstup  10 = sepnout výstup na nastavenou dobu („kladný pulz“)  11 = rozepnout výstup na nastavenou dobu („záporný pulz“)</p> <p>„<b>K</b>“ – Kritická teplotní tendence – uplatní se pouze u sepnutí na nastavenou dobu:  0 – vzestup teploty  1 – pokles teploty</p> <p>„<b>TTTT</b>“ – Binární číslo teploměru, ke kterému se vztahují následující teplotní meze.</p>										
			Druhý Teplota ve formátu signed int – vyšší mez.										
			Třetí Teplota ve formátu signed int – nižší mez.										
			<p><u>Horní</u> byte:  Čas sepnutí relé ve vteřinách, pokud je nastaveno sepnutí na určitou dobu.</p> <p><u>Dolní</u> byte:  Určuje, co se má stát, pokud bude odpojen nebo přerušen kabel k teplotnímu senzoru.  0 – ponechat kontakt relé beze změny  1 – rozepnout kontakt relé  2 – sepnout kontakt relé</p>										
800 – 832 pouze TCP	čtení, zápis	0x03, 0x10  Jak při čtení, tak při zápisu je dovoleno zapisovat/číst pouze celou čtveřici registrů najednou!	<p><b>Hlídání teploty</b> <sup>6</sup></p> <p>Zařízení hlídá dvě teplotní meze. Při jejich opuštění pošle automatickou zprávu (např. HTTP GET). Po sobě následující čtveřice registrů se vztahují k jednotlivým teploměrům. Pro první teploměr se jedná o registry 800 (první) až 803 (čtvrtý).</p> <table border="1"> <thead> <tr> <th>Registr</th> <th>Význam</th> </tr> </thead> <tbody> <tr> <td>První</td> <td>Zapnutí (0x0001) nebo vypnutí (0x0000) hlídání teploty</td> </tr> <tr> <td>Druhý</td> <td>Pokud je teplota mimo meze a má se upozornění na tento stav odesílat opakovaně, zadejte sem periodu v sekundách, jak často se má zpráva odesílat.</td> </tr> <tr> <td>Třetí</td> <td>Horní mez teploty zadaná jako celé číslo (signed int). Jde o teplotu vynásobenou deseti. Příklad: Hodnotu 24.6 sem vložte jako číslo 246.</td> </tr> <tr> <td>Čtvrtý</td> <td>Dolní mez teploty zadaná jako celé číslo.</td> </tr> </tbody> </table>	Registr	Význam	První	Zapnutí (0x0001) nebo vypnutí (0x0000) hlídání teploty	Druhý	Pokud je teplota mimo meze a má se upozornění na tento stav odesílat opakovaně, zadejte sem periodu v sekundách, jak často se má zpráva odesílat.	Třetí	Horní mez teploty zadaná jako celé číslo (signed int). Jde o teplotu vynásobenou deseti. Příklad: Hodnotu 24.6 sem vložte jako číslo 246.	Čtvrtý	Dolní mez teploty zadaná jako celé číslo.
Registr	Význam												
První	Zapnutí (0x0001) nebo vypnutí (0x0000) hlídání teploty												
Druhý	Pokud je teplota mimo meze a má se upozornění na tento stav odesílat opakovaně, zadejte sem periodu v sekundách, jak často se má zpráva odesílat.												
Třetí	Horní mez teploty zadaná jako celé číslo (signed int). Jde o teplotu vynásobenou deseti. Příklad: Hodnotu 24.6 sem vložte jako číslo 246.												
Čtvrtý	Dolní mez teploty zadaná jako celé číslo.												
900 – 904	čtení	0x03	<b>Kopie input registrů 0 až 4</b>										

Adresa	Přístup	Funkce	Název
od 1000	čtení	0x03	<b>Stavy vstupů</b> Kopie stavů vstupů z Discrete Inputs. Počet registrů podle počtu vstupů. První je IN1. 0x0001 znamená aktivní vstup.
od 1200	čtení	0x03	<b>Stavy výstupů</b> Kopie stavů výstupů z Coils. Počet registrů podle počtu výstupů. První je OUT1. 0x0001 znamená aktivní výstup.
1400, 1401	čtení	0x03	<b>Run time</b> Doba od zapnutí nebo resetu zařízení v sekundách.

## Input register

Adresa	Přístup	Funkce	Název
0 <sup>8</sup>	čtení	0x04	<b>Status naměřené teploty</b> 0..... hodnota je platná 1..... chyba senzoru nebo odpojený senzor 2..... teplota je nad horní mezí 3..... teplota je pod dolní mezí Chyba senzoru nebo odpojený senzor se zde projeví po cca deseti sekundách.
1	čtení	0x04	<b>Naměřená hodnota – celé číslo</b> Naměřená teplota jako celé číslo se znaménkem (signed integer). Číslo znamená teplotu vynásobenou deseti. Pokud teplota není platná (dle statusu), je zde číslo -9999. <i>Příklad:</i> Teplota 23,4 °C je v tomto registru reprezentována hodnotou 234.
2, 3	čtení	0x04	<b>Naměřená hodnota – float</b> Naměřená teplota jako 32 bit float podle IEEE 754. <sup>9</sup> Pokud teplota není platná (dle statusu), je zde číslo -9999.
4 pouze RTU	čtení	0x04	<b>Teplotní jednotka</b> 0..... stupně Celsia 1..... stupně Fahrenheita 2..... Kelvin

## Discrete Inputs

Funkční kód **0x02** je určen pro **čtení stavu vstupů**. Čte 1 až X vstupů (maximum dle počtu vstupů na Quidu). V dotazu je specifikováno číslo prvního čteného vstupů i počet vstupů, které se mají přečíst. Vstupy jsou číslovány od nuly. Tedy například vstupy 1 až 10 mají pořadová čísla 0 až 9.

V odpovědi jsou stavy vstupů představovány jednotlivými bity. Hodnota 1 znamená aktivní vstup (připojeno napětí nebo sepnutý kontakt), hodnota 0 neaktivní vstup. Nejnižší bit v prvním bytu odpovědi představuje stav prvního vstupu, který byl adresován v dotazu.

<sup>8</sup> Je možné se setkat s číslováním registrů od jedničky nebo od nuly, protože tento první registr má adresu 0.

<sup>9</sup> Popis normy IEEE 754 je k dispozici například zde: [http://en.wikipedia.org/wiki/IEEE\\_754](http://en.wikipedia.org/wiki/IEEE_754)

Pokud není počet vstupů násobkem osmi, jsou přebytečné bity vyplněny nulami. V proměnné *Počet hodnot* je specifikován počet bytů se stavem vstupů.

Adresa	Přístup	Funkce	Obsah
0	čtení	0x02	<b>Stav prvního</b> požadovaného vstupu
1	čtení	0x02	<b>Stav druhého</b> požadovaného vstupu
...			

#### Příklad:

Příklad čtení vstupů 1 až 8.

Dotaz:		Odpověď:	
Funkční kód	0x02	Funkční kód	0x02
Adresa MSB	0x00	Počet bytů	0x01
Adresa LSB	0x00	Stav vstupů	0xA7
Počet vstupů MSB	0x00		
Počet vstupů LSB	0x08		

Výsledkem dotazu je byte 0xA7, což je v binárním vyjádření 1010 0111. Jednotlivé bity odpovídají stavům vstupů. Nejnižší bit představuje vstup číslo 1.

## Coils

Přístup k aktuálním stavům a ovládání výstupních relé.

### Funkční kód 0x01

Tento funkční kód je určen pro **čtení stavu výstupů**. Čte 1 až X výstupů (maximum dle počtu výstupů na Quidu). V dotazu je specifikováno číslo prvního čteného výstupů i počet výstupů, které se mají přečíst. Výstupy jsou číslovány od nuly. Tedy například výstupy 1 až 10 mají pořadová čísla 0 až 9.

V odpovědi jsou stavy výstupů představovány jednotlivými bity. Hodnota 1 znamená sepnutý výstup, hodnota 0 rozepnutý výstup. Nejnižší bit v prvním bytu odpovědi představuje stav prvního výstupu, který byl adresován v dotazu.

Pokud není počet výstupů násobkem osmi, jsou přebytečné bity vyplněny nulami.

### Funkční kódy 0x05 a 0x0F

Tyto funkční kódy jsou určeny pro **ovládání výstupů**. V dotazu je specifikováno, které výstupy mají být nastaveny. Výstupy jsou číslovány od nuly. Tedy například výstup 5 má pořadové číslo 4.

Logická 1 znamená sepnutí výstupu, logická 0 rozepnutí výstupu.

V odpovědi je uveden funkční kód, adresa a počet výstupů, které byly změněny.

Adresa	Přístup	Funkce	Výstup
0	čtení, zápis	0x01, 0x05, 0x0F	<b>První</b> adresovaný výstup
1	čtení, zápis	0x01, 0x05, 0x0F	<b>Druhý</b> adresovaný výstup
...			

#### Příklad čtení:

Příklad čtení výstupů 1 a 2.

<i>Dotaz:</i>	
Funkční kód	0x01
Adresa MSB	0x00
Adresa LSB	0x00
Počet výstupů MSB	0x00
Počet výstupů LSB	0x02

<i>Odpověď:</i>	
Funkční kód	0x01
Počet bytů	0x01
Stav výstupů	0x02

Výsledkem dotazu je byte 0x02, což je v binárním vyjádření 0000 0010. Je nastaven druhý nejnižší bit. To znamená, že výstup 1 je rozepnutý a 2 je sepnutý. (Zbylé bity jsou vyplněny nulami.)

**Příklad zápisu:**

Příklad zápisu stavu výstupů 20 až 29 (celkem deset výstupů):

Data pro výstupy, jsou uložena ve dvou Bytech: 0xCD a 0x01 (1100 1101 0000 0001 binárně)

*Bit:* 1 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1  
*Číslo výstupu:* 27 26 25 24 23 22 21 20 - - - - - 29 28

Jako první je odeslán byte 0xCD se stavem výstupů 27 až 20. Nejnižší bit představuje nejnižší výstup 20. Další byte (0x01) obsahuje zbylé bity 28 a 29. Ostatní bity jsou vyplněny nulami.

<i>Dotaz:</i>	
Funkční kód	0x0F
Adresa MSB	0x00
Adresa LSB	0x13
Počet výstupů MSB	0x00
Počet výstupů LSB	0x0A
Počet bytů	0x02
Hodnoty MSB	0xCD
Hodnoty LSB	0x01

<i>Odpověď:</i>	
Funkční kód	0x0F
Adresa MSB	0x00
Adresa LSB	0x13
Počet výstupů MSB	0x00
Počet výstupů LSB	0x0A

Výsledkem příkazu je změna stavu některých výstupů na Quidu.



# Papouch s.r.o.

Přenosy dat v průmyslu, převodníky linek a protokolů, RS232/485/422/USB/Ethernet/GPRS/WiFi, měřicí moduly, inteligentní teplotní čidla, I/O moduly, elektronické aplikace dle požadavků.

Adresa:

**Strašnická 3164/1a  
102 00 Praha 10**

Telefon:

**+420 267 314 268**

Internet:

**[www.papouch.com](http://www.papouch.com)**

E-mail:

**[papouch@papouch.com](mailto:papouch@papouch.com)**

