

General Electric SRTP

Protokol General Electric SRTP (Service Request Transport Protocol)

[Podporované typy a verzie zariadení](#)
[Konfigurácia komunikačnej linky](#)
[Parametre protokolu linky](#)
[Konfigurácia komunikačnej stanice](#)
[Konfigurácia meraných bodov](#)
[Poznámky k Fanuc Robot R-30iA/R-30iB](#)
[Literatúra](#)
[Zmeny a úpravy](#)
[Revízie dokumentu](#)

Podporované typy a verzie zariadení

Protokol je implementáciou protokolu "Service Request Transport Protocol" vyvinutého firmou General Electric Automation and Controls (predtým GE Fanuc), slúžiaceho na komunikáciu s PLC. Takmer všetky GE zariadenia vybavené Ethernet portom podporujú protokol GE SRTP. Protokol GE SRTP je nástupcom sériových protokolov SNP a SNPX pre Ethernet médium.

Protokol bol otestovaný vo Fanuc Robot R-30iB:

- funkčné bolo ítanie hodnôt z typov pamäte %I, %IB, %Q, %QB, %M, %MB, %G, %GB, %AI, %AQ, %R, %RD, %RF
- funkčný bol zápis hodnôt do typov pamäte %R, %RD, %RF (ostatné, ktoré sa dali vyítať, síce vrátili úspech, pri ďalšom ítaní bola načítaná pôvodná hodnota)
- funkčné je ítanie aj zápis textových hodnôt (napr. string registers alebo komentáre k registrom - vi [Poznámky k Fanuc Robot R-30iA/R-30iB](#))

Konfigurácia komunikačnej linky

- Kategória komunikačnej linky: [TCP/IP-TCP](#).
- IP adresa podľa sieovej konfigurácie konkrétneho zariadenia GE.
Pozn: je možné nakonfigurovať aj viacero IP adries zariadenia (oddelených iarkou alebo bodkoiarkou).
- číslo portu je štandardne 18245.
- číslo linky je nepoužívané, nastavte hodnotu 1.

Pozn: ak sú všetky stanice v stave StOFF (alebo v simulácii), TCP spojenie bude zatvorené. Takto je možné riadiť TCP komunikáciu zo skriptu pomocou tell príkazu [STSTAT](#).

Parametre protokolu linky

Dialóg [konfigurácia linky](#) - záložka **Parametre protokolu**.

Ovplyvňujú niektoré voliteľné parametre protokolu. Môžu byť zadané nasledovné parametre protokolu linky:

Tab. . 1

| Parameter | Popis | Jednotka / rozmer | Náhradná hodnota |
|-----------------|--|-------------------|------------------|
| Maximum Payload | Maximálna veľkosť dát (v bajtoch) ítaných jednou výzvou. Merané body, ktorých adresy nasledujú za sebou, budú združované do požiadaviek na ítanie tak, aby veľkosť dát v odpovedi nepresiahla nakonfigurovanú hodnotu. | 1 až 1024 B | 100 B |
| Cycle Delay | akanie po vyítaní všetkých hodnôt stanice. Tento parameter umožňuje nastaviť menšie cykly ítaní ako 1 sekunda (asové parametre nakonfigurované na stanici majú rozlíšenie 1 sec). | ss.mss | 100 ms |
| Debug Values | Zapína ladiace informácie o načítaných a zapisovaných hodnotách meraných bodov. Odporúčame zapnúť iba v prípade nutnosti ladenia komunikácie, pretože výrazne zvyšuje záťaž CPU a spomaľuje komunikáciu. | YES/NO | NO |

Konfigurácia komunikačnej stanice

- Komunikovaný protokol: **General Electric SRTP**.
- Adresný parameter sa nepožaduje žiadny, na jednej komunikačnej linke je možné komunikovať iba s zariadením.
Pozn: na jednej linke môže byť viac staníc napr. kvôli rôznym asovým parametrom (rozdelenie na rýchle a pomalé body).

Konfigurácia meraných bodov

Možné typy hodnôt bodov: **Ai, Ci, Di, TiR,TxtI, Ao, Co, Dout, ToR, TxtO**

Adresa meraného bodu má formát `[+]MemoryArea Position[:StringLength] [Count]`

- Znak "+" sa používa pre pamäte obsahujúce celočíselné hodnoty a znamená, že hodnota bude interpretovaná ako Unsigned číslo (8, 16 alebo 32-bitové číslo bez znamienka).

Ak nie je znak "+" nájdený, hodnota je interpretovaná ako Signed číslo (8, 16 alebo 32-bitové číslo so znamienkom). Pre pamäte obsahujúce bity alebo float hodnoty je "+" ignorované.

- *MemoryArea* špecifikuje typ pamäte:

| MemoryArea | Popis typu pamäte | Vekos dát |
|-------------------|---|------------------|
| %IGNORE | Meraný bod bude ignorovaný. Pozn: za %IGNORE môžu nasledovať ľubovoľné znaky, napr. %IGNORE +%R1 | - |
| %I | InputBit (Discrete inputs, prístup po 1 bite) | 1 bit |
| %IB | InputByte (Discrete inputs, prístup po 1 byte) | 1 byte |
| %Q | OutputBit (Discrete outputs, prístup po 1 bite) | 1 bit |
| %QB | OutputByte (Discrete outputs, prístup po 1 byte) | 1 byte |
| %T | TemporaryBit (Temporary references, prístup po 1 bite) | 1 bit |
| %TB | TemporaryByte (Temporary references, prístup po 1 byte) | 1 byte |
| %M | MemoryBit (Internal references, prístup po 1 bite) | 1 bit |
| %MB | MemoryByte (Internal references, prístup po 1 byte) | 1 byte |
| %S | SBit (System status references, prístup po 1 bite) | 1 bit |
| %SSB | SByte (System status references, prístup po 1 byte) | 1 byte |
| %SA | SABit (System status references A, prístup po 1 bite) | 1 bit |
| %SAB | SABByte (System status references A, prístup po 1 byte) | 1 byte |
| %SB | SBBit (System status references B, prístup po 1 bite) | 1 bit |
| %SBB | SBBByte (System status references B, prístup po 1 byte) | 1 byte |
| %SC | SCBit (System status references C, prístup po 1 bite) | 1 bit |
| %SCB | SCByte (System status references C, prístup po 1 byte) | 1 byte |
| %G | GlobalBit (Discrete globals, prístup po 1 bite) | 1 bit |
| %GB | GlobalByte (Discrete globals, prístup po 1 byte) | 1 byte |
| %AI | AnalogInput (Analog input registers, prístup po 2 bajtoch) | 2 bytes |
| %AQ | AnalogInput (Analog output registers, prístup po 2 bajtoch) | 2 bytes |
| %R | Registers (System register reference, prístup po 2 bajtoch) | 2 bytes |
| %RD | Registers Double (System register reference, 2 registre ako double word) | 4 bytes |
| %RF | RegistersFloat (System register reference, 2 registre ako float) | 4 bytes |
| %L | LocalSubblock (Local registers, prístup po 2 bajtoch) | 2 bytes |
| %P | ProgramBlockData (Program registers, prístup po 2 bajtoch) | 2 bytes |

Poznámka: poradie bytov a slov v protokole je little-endian: B1 B2 pre 2-bajtové typy, B1 B2 B3 B4 pre 4-bajtové typy.

- *Position* udáva pozíciu objektu v rámci typu pamäte. Je to kladné 16-bitové číslo (1-65535).
Pozn: pre typy pamäte %RD a %RF sú platné pozície iba 1-65534, keďže sa jedná iba o celočíselnú/float interpretáciu dvoch za sebou idúcich registrov.
- *StringLength* udáva dĺžku reazca v znakoch. Tento parameter je povolený iba pre typ pamäte %R. Každý register je interpretovaný ako 2-znakový reazec (formát little endian). Tj. meraný bod s adresou %R1000:4 íta 4-znakový reazec z registrov 1000 a 1001.
- *Count* udáva počet objektov. Tento parameter má zmysel iba vtedy, ak je nakonfigurovaný [Cieový stpec štruktúry](#). Udáva počet objektov, ktoré budú naitané a zapísané do štruktúry.
- príklady adries:
%R 1 20
+%R 100
+%AI25
%AQ1 5
%IB 12
%R12111:80

Poznámky k Fanuc Robot R-30iA/R-30iB

Podľa dokumentácie FANUC Robot series R-J3/R-J3iB/R-30iA CONTROLLER SIMPLICITY HMI for Robots OPERATOR'S MANUAL, kapitola 6 - ADDRESS ASSIGNMENT TO POINTS, podkapitola 6.1 - READING AND WRITING I/O SIGNALS (%I, %Q, %M, %AI, %AQ) sú rôzne typy I/O signálov mapované na premenné SRTP protokolu takto:

| Robot controller I/O signal | PLC address | Príklad |
|--|------------------------------|---|
| Digital Input DI[x] | %Qx | DI[1] <=> %Q1 |
| Digital Output DO[x] | %Ix | DO[1] <=> %I1 |
| Robot Input RI[x] | %Q(5000+x) | RI[1] <=> %Q5001 |
| Robot Output RO[x] | %I(5000+x) | RO[1] <=> %I5001 |
| UOP Input UI[x] | %Q(6000+x) | UI[1] <=> %Q6001 |
| UOP Output UO[x] | %I(6000+x) | UO[1] <=> %I6001 |
| SOP Input SI[x] | %Q(7000+x) | SI[0] <=> %Q7001 |
| SOP Output SO[x] | %I(7000+x) | SO[0] <=> %I7001 |
| Weld Input WI[x] | %Q(8000+x) | WI[0] <=> %Q8001 |
| Weld Output WO[x] | %I(8000+x) | WO[0] <=> %Q8001 |
| Wire Stick Input WSI[x] | %Q(8400+x) | WSI[0] <=> %Q8400 |
| Wire Stick Output WSO[x] | %I(8400+x) | WSO[0] <=> %Q8400 |
| Group Input GI[x] | %AQx | GI[1] <=> %AQ1 |
| Group Output GO[x] | %AIx | GO[1] <=> %AI1 |
| Analog Input AI[x] | %AQ(1000+x) | AI[1] <=> %AQ1001 |
| Analog Output AO[x] | %AI(1000+x) | AO[1] <=> %AI1001 |
| PMC keep relay DO[x] (x : 10001 – 10144) Ka.b | %Ix %I((a*8)+b+10001) | DO[10001] <=> %I10001 K2.5 <=> %I10022 |
| PMC internal relay DO[x] (x : 11001 – 23000) Ra.b | %M(x-11000) %M((a*8)+b+1) | DO[11001] <=> %M1 R2.5 <=> %M22 |
| PMCdata table GO[x] (x : 10001 – 12000) D(a*2), D((a*2)+1) | %AI(x-6000) %AI(a+4001) | GO[10001] <=> %AI4001 D4, D5 <=> %AI4003 |

Podľa rovnakého dokumentu, podkapitola 6.2 - READING FROM AND WRITING TO REGISTERS (%R) je štandardné mapovanie registrov z radia robota na adresy v PLC :

| Robot controller data | PLC address | Príklad |
|-----------------------|-------------|--------------|
| Register | %Rx | R[1] <=> %R1 |

Hodnoty v registroch sú 16-bitové čísla so znamienkom, desatinné časti sú odrezané.

Dokumentácia alej uvádza, že toto štandardné mapovanie je možné zmeniť pomocou systémovej premennej **\$SNPX_ASG**. Táto umožňuje nastaviť multiplikátor pri prevode a zmeniť mapovanie registrov. Jeden register môže byť mapovaný do jednej premennej %R (16-bitové číslo so znamienkom), do dvoch za sebou idúcich premenných %R (32-bitové číslo so znamienkom, použité v adrese meraného bodu %RD) alebo do dvoch za sebou idúcich premenných %R ako reálne číslo (32-bitové float číslo, použité v adrese meraného bodu %RF).

Štandardné mapovanie registrov R zabezpečuje nastavenie \$SNPX_ASG[1]:

| Položka | Hodnota |
|-----------|---------|
| \$ADDRESS | 1 |

| | |
|------------|------------|
| \$SIZE | 10000 |
| \$VAR_NAME | 'R[1]@1.1' |
| \$MULTIPLY | 1.000 |

Pri tomto nastavení budú registre R[1] .. R[10000] mapované na registre %R1 .. %R10000 ako 16-bitové ísla so znamienkom.

Štandardné mapovanie poziných registrov PR (urených na ukladanie pozícií robota) zabezpeuje nastavenie \$SNPX_ASG[2]:

| Položka | Hodnota |
|------------|---------|
| \$ADDRESS | 11000 |
| \$SIZE | 100 |
| \$VAR_NAME | 'PR[1]' |
| \$MULTIPLY | 0.000 |

Pri tomto nastavení budú jednotlivé zložky polohy (X, Y, Z, W, P, R at) uložené ako 32-bitové float ísla na adresách %RF11000, %RF11002, %RF11004 at (50 registrov na jeden PR, celkove 100 registrov na PR[1] a PR[2]. Viac detailov vi podkapitola 6.3 *READING FROM AND WRITING TO POSITION REGISTERS (%R)*.

Štandardné mapovanie registra aktuálnej pozície POS zabezpeuje nastavenie \$SNPX_ASG[3]:

| Položka | Hodnota |
|------------|----------|
| \$ADDRESS | 12000 |
| \$SIZE | 100 |
| \$VAR_NAME | 'POS[0]' |
| \$MULTIPLY | 0.000 |

Pri tomto nastavení budú jednotlivé zložky aktuálnej polohy (X, Y, Z, W, P, R at) uložené ako 32-bitové float ísla na adresách %RF12000, %RF12002, %RF12004 at (50 registrov na jeden POS, celkove 100 registrov na POS[0] a POS[1]. Viac detailov vi podkapitola 6.4 *READING AND WRITING THE CURRENT POSITION (%R)*.

Mapovanie ALM registrov obsahujúcich históriu alarmov je možné nastavením \$SNPX_ASG[x]:

| Položka | Hodnota |
|------------|----------|
| \$ADDRESS | 12100 |
| \$SIZE | 300 |
| \$VAR_NAME | 'ALM[1]' |
| \$MULTIPLY | 1.000 |

Pri tomto nastavení budú jednotlivé zložky histórie alarmov (AlarmID, Alarm number, Alarm severity, Alarm message) prístupné na nakonfigurovaných adresách (100 registrov na jeden ALM, tj. 300 registrov pokryje 3 alarmy). Viac detailov vi podkapitola 6.5 *READING ALARM HISTORY (%R)*.

Mapovanie PRG registrov obsahujúcich stav vykonávania programu (program execution status) je možné nastavením \$SNPX_ASG[x]:

| Položka | Hodnota |
|------------|----------|
| \$ADDRESS | 12400 |
| \$SIZE | 18 |
| \$VAR_NAME | 'PRG[1]' |
| \$MULTIPLY | 1.000 |

Pri tomto nastavení budú jednotlivé zložky stavu vykonávania programu (Program name, Line number, Execution status, Calling program name) prístupné na nakonfigurovaných adresách (18 registrov na jeden PRG). Viac detailov vi podkapitola 6.6 *READING THE PROGRAM EXECUTION STATUS (%R)*.

Mapovanie rôznych systémových premenných je takisto možné nastavením \$SNPX_ASG[x]. Príklad:

| Položka | Hodnota |
|------------|-----------------------|
| \$ADDRESS | 12420 |
| \$SIZE | 2 |
| \$VAR_NAME | '\$mcr.\$genoverride' |
| \$MULTIPLY | 1.000 |

Pri tomto nastavení bude v registroch 12420 a 12421 systémová premenná OVERRIDE (položka \$genoverride systémovej premennej \$mcr) ako 32-bitový integer (adresa %RD12420, prístup na čítanie aj zápis). Viac detailov vi podkapitola 6.7 *READING FROM AND WRITING INTO SYSTEM VARIABLES (%R)*.

Mapovanie komentárov k registrom, pozíciám registrov a I/O je takisto možné nastavením \$SNPX_ASG[x]. Príklad:

| Položka | Hodnota |
|------------|---------|
| \$ADDRESS | 12440 |
| \$SIZE | 160 |
| \$VAR_NAME | 'R[C1]' |
| \$MULTIPLY | 1.000 |

Pri tomto nastavení budú v registroch %R12420 až %R12599 komentáre k registrom R1 až R4 (40 registrov, t.j. 80 znakov na komentár - adresy %R12440:40, %R12480:40, %R12520:40 a %R12560:40). Podobne je možné nakonfigurovať komentáre k iným typom objektov (napr. 'PR[C1]', 'DI[C2]', 'GI[C1]', 'GO[C1]' at). Viac detailov vi podkapitola 6.8 *READING AND WRITING THE COMMENT OF REGISTERS, POSITION REGISTERS, AND I/O (%R)*.

Mapovanie hodnôt I/O a stavu simulácie je takisto možné nastavením \$SNPX_ASG[x]. Príklad:

| Položka | Hodnota |
|------------|----------|
| \$ADDRESS | 12430 |
| \$SIZE | 4 |
| \$VAR_NAME | 'RO[S1]' |
| \$MULTIPLY | 1.000 |

Pri tomto nastavení budú v registroch %R12430 až %R12433 stavy simulácie výstupov RO[1] až RO[4]. Podobne je možné nakonfigurovať stavy simulácie iných typov objektov (napr. 'DI[S1]', 'DO[C2]', 'DO[S1]', 'RI[S1]' at) ako aj hodnoty týchto objektov (napr. 'DI[1]', 'DI[2]', 'UO[1]', 'SI[1]' at). Viac detailov vi podkapitola 6.9 *READING AND WRITING THE VALUE AND SIM STATUS OF I/O (%R)*.

Literatúra

https://en.wikipedia.org/wiki/Service_Request_Transport_Protocol

FANUC Robot series R-J3/R-J3iB/R-30iA CONTROLLER CIMPLICITY HMI for Robots OPERATOR'S MANUAL (R-30iA CONTROLLER CIMPLICITY HMI for Robots_[B-82604EN_01].pdf)



Blog

O protokole General Electric SRTP si môžete prečítať aj blogy

- [Komunikácia - roboty Fanuc](#)
- [Komunikácia - roboty Fanuc - druhá časť](#)

Zmeny a úpravy

-

Revízie dokumentu

- Ver. 1.0 - 12. marec 2020 - Vytvorenie dokumentu.



Súvisiace stránky:

[Komunikané protokoly](#)