

Obecný protokol komunikace s HW

Petr Novák (novakpe@fel.cvut.cz)

22.01.2018

Popis paketu stručně:

- Tvar paketu: StartZnak(je / není mezera)PaketType(není mezera)PaketName(není mezera)PaketIdent(je mezera)TěloPaketu(je/není mezera)StopZnak.
- Start znak '<', Stop znak '>'
- Paket Type
 - o Nic (položka není přítomna) – běžný paket (povel do zařízení)
 - o \$ - odpověď (z externího zařízení)
- Paket Name
 - o C = Command
 - Hodnota povelu (V = dekadické číslo).
 - Upřesnění povelu (X = dekadické číslo), pokud je použito.
 - o X = Nevyžádaná data (například kontinuální měření).
- Paket Ident – dekadické identifikační číslo v zasílaném paketu, zařízení toto číslo umístí do své odpovědi pro identifikaci, k jakému zaslanému paketu patří odpověď
- Tělo paketu – Písmeno následováno rovnítkem a (pokud není uvedeno jinak) dekadickým číslem / čísly. Více číslic odděleny čárkou, mezery pouze před písmeny, mezera určuje konec nějaké položky.
- Příklady (rozšíření formátu záleží na nastavení v externím zařízení):
 - o (Out) '<C V=1>' (nejzákladnější formát, povel Cmd=1), (In) '<C V=1>'
 - o (Out) '<C1 V=1>' (formát s identifikací, povel Cmd=1), (In) '<C1 V=1>'
 - o (Out) '<C1 V=1>' (formát s identifikací a typem paketu, povel Cmd=1), (In) ',<\$C1 V=1>'

Povely do zařízení (typ „Cmd“) a jejich odpovědi

// **Reset (Cmd=1)** nulování zařízení)

// (X=0 – HW Reset / X=1 – SW Reset)

(PC->HW) „**C V=1 X=1**“ – Reset HW (Hard reset)

(PC->HW) „**C V=1 X=2**“ – Reset SW (Soft reset)

// (odpověď je vyslána ještě před vykonáním povelu)

// (reset zařízení by měl nastat za 2s pro SW reset / 5s pro HW reset)

// (S – stav vykonání / zařízení)

(HW->PC) „**C V=1 S=?**“

// **Info (Cmd=2)** info o zařízení)

// (X=1 – DevInfo, základní informace o zařízení)

// (X=2 – HWInfo, informace o HW části zařízení)

// (X=3 – SWInfo, informace o SW části zařízení)

(PC->HW) „**C V=2 X=x**“

// (S – stav vykonání / zařízení)

(HW->PC) „**C V= 2 P=ProducentZarizeni N=NazevZarizeni S=?**“

// (T = string / V, R – celé číslo / U – AciHex číslo různé délky)

(HW->PC) „**C V= 2 T=TypHW V=VerzeHW R=RevizeHW U=UnikatnildHW S=?**“

(HW->PC) „**C V= 2 T=TypSW V=VerzeSW R=RevizeSW U=UnikatnildSW S=?**“

// **Init / Opem (Cmd=3** inicializace / otevření zařízení, ne spuštění)

(PC->HW) „**C V=3**“

// (... – možná data; S – stav vykonání / zařízení)

(HW->PC) „**C V=3 ... S=?**“

// **Close (Cmd=4** uzavření / ukončení zařízení, ne zastavení)

(PC->HW) „**C V=4**“

// (S – stav vykonání / zařízení)

(HW->PC) „**C V=4 S=?**“

// **Stop (Cmd=5** zastavení činnosti zařízení)

(PC->HW) „**C V=5**“

// (S – stav vykonání / zařízení)

(HW->PC) „**C V=5 S=?**“

// **Start (Cmd=6** spuštění činnosti zařízení)

(PC->HW) „**C V=6**“

// (... – možná data; S – stav vykonání / zařízení)

(HW->PC) „**C V=6 ... S=?**“

// **Check (Cmd=7** zjištění / kontrola stavu zařízení)

(PC->HW) „**C V=7**“

// (... – možná data;

// (N – identifikační číslo zařízení, 0-255)

// (R=0 – zařízení neběží / R=1 – zařízení běží)

// (S – stav vykonání / zařízení)

(HW->PC) „**C V=7 ... N=n R=? S=?**“

// **DevSet (Cmd=8** nastavení základních informací v zařízení)

// (N – přidělené identifikační číslo zařízení, 0-255)

(PC->HW) „**C V=8 N=n**“

// (S – stav vykonání / zařízení)

(HW->PC) „**C V=8 S=?**“

// **Cmd=9 Setup** (uložení / načtení / nulování nastavení)

// (X=1 / Store – uložit aktuální konfiguraci do EEPROM / FLASH, tedy zachovat pro vypnutí)

// (X=2 / Load – načíst uloženou konfiguraci z EEPROM / FLASH, obnovit ze zálohy)

// (X=3 / Clear – nulovat uloženou konfiguraci, vymazat EEPROM / FLASH uchovávající konfiguraci)
(PC->HW) „C V=9 X=x“

// (S – stav vykonání / zařízení)
(HW->PC) „C V=9 S=?“

// **Update (Cmd=10** aktualizace zařízení)
// (X – způsob aktualizace, zatím nepoužito)
(PC->HW) „C V=10 X=x“

// (odpověď je vyslána ještě před vykonáním povelu)
// (přepnutí do režimu aktivace je za 5s)
// (S – stav vykonání / zařízení)
(HW->PC) „C V=10 S=?“

// **SetParams (Cmd=11** nastavení parametrů pro zařízení)
// (A – od adresy / pozice; C – kolik / počet; D - data)
(PC->HW) „C V=11 A=a C=c D=...”

// (S – stav vykonání / zařízení)
(HW->PC) „C V=11 S=?“

// **GetParams (Cmd=12** čtení parametrů ze zařízení)
// (A – od adresy / pozice; C – kolik / počet)
(PC->HW) „C V=12 A=a C=c“

// (A – od adresy / pozice; C – kolik / počet; D – data; S – stav vykonání / zařízení)
(HW->PC) „C V=12 A=... C=... D=... S=?“

// **SetSpeed (Cmd=15** přímé / snadné nastavení rychlosti zařízení)
// (D - data)
(PC->HW) „C V=15 D=...”

// (S – stav vykonání / zařízení)
(HW->PC) „C V=15 S=?“

// **SetGain (Cmd=16** přímé / snadné nastavení zesílení v zařízení)
// (D - data)
(PC->HW) „C V=16 D=...”

// (S – stav vykonání / zařízení)
(HW->PC) „C V=16 S=?“

// **SetData (Cmd=17** přímé / snadné nastavení jaká data zařízení zpracovává / vysílá)
// (D - data)
(PC->HW) „C V=17 D=...”

// (S – stav vykonání / zařízení)

(HW->PC) „C V=17 S=?“

// **GetValue** (Cmd=19 vyčtení jedné sady dat / měření ze zařízení)
(PC->HW) „C V=19“

// (D – data; S – stav vykonání / zařízení)
(HW->PC) „C V=19 D=... S=?“

// **DevSubCmd** (Cmd=21 pod-povely specifické pro zařízení)
// (X – pod-povel specifický pro zařízení)
(PC->HW) „C V=10 X=x“

// (... – další data; S – stav vykonání / zařízení)
(HW->PC) „C V=7 ... S=?“

// speciální povely pro konkrétní zařízení jsou od čísla **Cmd=51**
...

Nevyžádaná data ze zařízení

Data v nestandardním / speciálním paketu podle typu zařízení:

// X – značí nestandardní (ale nějak textový) paket
// hned následují datové položky
// přenášeny jako ASCII-HEX hodnoty (1byte = 2znaky AsciiHex), pro komprimaci
// 16bit a více hodnoty, nejprve nejvyšší byte (2znaky), poté nižší byte (2znaky), ...
// (S je status: 0 – OK (vše v pořádku) / 1,2,3 ... jiný stav)
„XH1L1H2L2...HnLnS“ (příklad pro 16bit přenos)

Poznámky:

- V řetězci nesmí být mezera. Řetězec musí obsahovat pouze tisknutelné znaky. Místo mezery se doporučuje používat podtržítko (_).
- Verze HW a SW je vždy složena ze dvou čísel a to VERze a REVize. Každé číslo by mělo být „byte / 8 bitů“. Celé číslo verze je tedy VERze.REVize, například „1.2“. VERze se používá pro indikaci změn nebo přidání schopností zařízení. REVize se používá zejména pro indikaci menších vylepšení / oprav schopností zařízení.