

Návrh komunikačního protokolu expo systému

4.12.2024

zpracoval:

Jan Stejskal
Voříškova 567/8, 623 00 Brno
IČ: 76299376, DIČ: CZ8502074273

Obsah

Popis komunikace.....	3
Identifikace a adresování.....	3
Datové věty.....	3
Čtení.....	4
Zápis.....	4
Hromadný zápis.....	5
Status.....	5
Sensor.....	5
Error.....	6
Složené datové věty.....	6
Katalogy.....	6

Popis komunikace

Komunikace mezi řídicí aplikací, řídicí elektronikou exponátu, aplikací exponátu a dalšími součástmi exponátního systému je založena na protokolu WebSocket. Komunikaci zabezpečuje websocket server, který může zprávy broadcastovat nebo adresovat na konkrétní klienty. Keep-alive připojení klientů je realizováno metodou ping/pong.

Komunikace zahrnuje diagnostická data, řídicí povely, data ze senzorů. Pro streamování multimediálního obsahu server umožňuje otevřít samostatné vlákno pro každý požadavek. Kontrola aktualizací aplikací a firmware probíhá přes websocket, samotný download pak přes http protokol.

Identifikace a adresování

Každé zařízení a aplikace se identifikuje pomocí unikátního ID. ID je nezávislé na MAC adrese i na přidělené IP adrese, což umožňuje zachování konfigurace síťové komunikace i v případě nutné výměny hardware. U aplikací, které mohou být spuštěné vícenásobně simultánně, se ID generuje dle přihlášeného uživatele.

Server umožňuje adresování dle ID, kdy doručuje zprávy mezi dvěma klienty na základě jejich ID. Dále umožňuje adresování dle skupin. Každé ID může být zahrnuto do jedné či více skupin a zpráva může být odeslána na celou skupinu nebo více skupin. Adresování zpráv zajišťuje WebSocket server, tím je zamezeno podvržení odesílatele či adresátů.

Datové věty

Datové věty jsou formátovány pomocí JSON objektů. Pro zrychlení komunikace a zmenšení datového objemu zpráv je možné použít MessagePack. V případě požadavku na šifrování komunikace s nutným doplněním délky lze použít byte 0xC1 pro doplnění délky před zašifrováním.

Každá datová věta musí obsahovat typ zprávy, unikátní identifikátor, informace o odesílateli a adresátovi, které zpracovává WebSocket server a payload.

Payload musí obsahovat index příkazu, který se má vykonat. Každé zařízení nebo aplikace může podporovat jiné indexy, podporované indexy jsou zahrnuty v katalogu daného zařízení/aplikace.

Všechny datové věty musí obsahovat povinné klíče:

- „t“ TYPE pro typ zprávy, hodnoty mohou být „q“ pro QUERY, „r“ pro RESPONSE, „s“ pro STATUS, „n“ pro SENSOR data, „e“ pro ERROR
- „u“ UID pro unikátní identifikátor asynchronní komunikace, použije se pro párování požadavku a odpovědi, typ uint32 (0 až 4294967295),
- „s“ SENDER pro ID odesílatele, odesílatelem je vždy jedno konkrétní ID, typ uint16 (0 až 65535),

- „p“ PAYLOAD pro data
 - „i“ INDEX pro číslo indexu příkazu, typ uint16 (0 až 65535).

Volitelné klíče:

- „r“ RECIPIENT pro ID příjemce, příjemce je vždy jedno konkrétní ID, typ uint16 (0 až 65535), v případě, že příjemce není definován, rozhoduje o příjemci WebSocket server

Payload odpovědi mimo klíč „i“ vždy obsahuje klíč „v“ VALUE s přečtenou, případně zapsanou hodnotou nebo klíč „e“ ERROR v případě chyby, kdy hodnotou je číslo chyby.

Čtení

Pro požadavek čtení indexu jsou povinné klíče „t“, „u“, „s“, „r“ a „p“, pro payload pak klíč „i“. Odpověď musí obsahovat klíče „t“, „s“, „u“, „p“, pro payload pak klíče „i“ a „v“ nebo „i“ a „e“.

Požadavek na čtení indexu:

```
{"t": "q", "u": 4294967295, "s": 100, "r": 255, "p": {"i": 10001}}
```

Odpověď

```
{"t": "r", "u": 4294967295, "s": 255, "p": {"i": 10001, "v": 0.27}}
```

Odpověď s chybou

```
{"t": "r", "u": 4294967295, "s": 255, "p": {"i": 10001, "e": 404}}
```

Zápis

Pro požadavek zápisu indexu jsou povinné klíče „t“, „u“, „s“, „r“ a „p“, pro payload pak klíče „i“ a „v“. Zápisem se rozumí také požadavek na spuštění funkce. Odpověď musí obsahovat klíče „t“, „s“, „u“, „p“, pro payload pak klíče „i“ a „v“ nebo „i“ a „e“. „v“ VALUE v odpovědi obsahuje skutečně zapsanou hodnotu, případně hodnotu „true“ při úspěšném spuštění funkce.

Požadavek na zápis indexu:

```
{"t": "q", "u": 4294967294, "s": 100, "r": 255, "p": {"i": 10001, "v": 0.5}}
```

Odpověď

```
{"t": "r", "u": 4294967294, "s": 255, "p": {"i": 10001, "v": 0.5}}
```

Odpověď s chybou

```
{"t": "r", "u": 4294967294, "s": 255, "p": {"i": 10001, "e": 404}}
```

Hromadný zápis

Pro případy, kdy je třeba zapsat více hodnot nebo spustit více funkcí v definovaném pořadí, lze odeslat payload jako pole objektů. Povinné klíče jsou totožné jako u jednotlivého zápisu.

Požadavek na hromadný zápis:

```
{"t": "q", "u": 4294967293, "s": 100, "r": 255, "p": [{"i": 10002, "v": 2.5}, {"i": 10003, "v": 300}, {"i": 10004, "v": false}]}
```

Odpověď

```
{"t": "r", "u": 4294967293, "s": 255, "p": [{"i": 10002, "v": 2.5}, {"i": 10003, "v": 300}, {"i": 10004, "v": false}]}
```

Odpověď s chybou

```
{"t": "r", "u": 4294967293, "s": 255, "p": [{"i": 10002, "v": 2.5}, {"i": 10003, "v": 300}, {"i": 10004, "e": 400}]}
```

Status

Pro stavové informace a monitoring každé zařízení a aplikace odesílají zprávy typu STATUS. Tento typ zprávy nevyžaduje povinné klíče „u“ UID (u tohoto typu zpráv se neočekává žádná odpověď) a „r“ RECIPIENT, o adresátech těchto zpráv rozhoduje Websocket server.

Zpráva s jednou stavovou informací:

```
{"t": "s", "s": 255, "p": {"i": 50001, "v": false}}
```

Zpráva s více informacemi:

```
{"t": "s", "s": 255, "p": [{"i": 50001, "v": true}, {"i": 50002, "v": true}, {"i": 50003, "v": 230785}]}
```

Sensor

Senzory při změně stavu a nebo při změně měřené hodnoty odesílají zprávu typu „n“ SENSOR. Tento typ zprávy nevyžaduje povinné klíče „u“ UID (u tohoto typu zpráv se neočekává žádná odpověď) a „r“ RECIPIENT, o adresátech těchto zpráv rozhoduje Websocket server. Zprávy ze sensorů lze využít pro řízení dalších zařízení nebo aplikací.

Zpráva s jednou informací ze sensoru:

```
{"t": "n", "s": 65535, "p": {"i": 60001, "v": 0.5}}
```

Zpráva s více informacemi:

```
{"t": "n", "s": 65535, "p": [{"i": 60001, "v": 0.51}, {"i": 60002, "v": 0.9}, {"i": 60003, "v": -0.42}]}
```

Error

Zprávy typu „e“ ERROR obsahují informace o kritických chybách zařízení nebo aplikací exponátního systému. Tento typ zprávy nevyžaduje povinné klíče „u“ UID (u tohoto typu zpráv se neočekává žádná odpověď) a „r“ RECIPIENT, o adresátech těchto zpráv rozhoduje Websocket server. Payload může obsahovat klíč „e“ s kódem chyby, pokud daný index zahrnuje více možných chyb nebo více zdrojů chyb.

Zpráva s jednou chybovou informací:

```
{"t": "e", "s": 255, "p": {"i": 70001}}
```

Zpráva s jednou chybovou informací a chybovým kódem:

```
{"t": "e", "s": 255, "p": {"i": 70001, "e": 303}}
```

Zpráva s více chybami:

```
{"t": "e", "s": 255, "p": [{"i": 70001, "e": 303}, {"i": 70002}]}
```

Složené datové věty

Zprávy typu „q“ REQUEST mohou obsahovat sérii příkazů, které se mají vykonat v daném pořadí nebo se stejné příkazy mají odeslat více příjemcům. Pro tyto případy je možné využít složené datové věty, které jsou odeslány jako pole. Struktura jednotlivých datových vět a payload je totožná s jednotlivě zasílanými příkazy. Websocket server složené datové věty rozdělí na jednotlivé zprávy a odešle na požadovaná zařízení.

Čtení z více zařízení současně:

```
[{"t": "q", "u": 4294967292, "s": 100, "r": 255, "p": {"i": 10001}}, {"t": "q", "u": 4294967291, "s": 100, "r": 254, "p": {"i": 10001}}, {"t": "q", "u": 4294967290, "s": 100, "r": 253, "p": {"i": 10001}}]
```

Katalogy

Každé zařízení a aplikace exponátního systému vyžaduje katalog ve formátu JSON, který definuje všechny indexy, které má zařízení nebo aplikace implementované a lze je tak číst, zapisovat nebo obojí. Katalogy jsou verzované dle verze firmware zařízení nebo verze aplikace.

Každý katalog obsahuje číslo verze a seznam indexů, u každého indexu pak jeho číslo, název, příznak (pro čtení, zápis, obojí), datový typ, výchozí hodnotu, minimální a maximální hodnotu, způsob zobrazení.

Dále obsahuje seznam implementovaných chybových kódů.

Vzorový katalog:

```
{  "device": "Expo app",  "version": 0.1,  "groups": [    {"id": 1, "name": "{\\\"Menu: Control\\\"}", "title": "{\\\"Title: Control\\\"}"},    {"id": 2, "name": "{\\\"Menu: Diagnostics\\\"}", "title": "{\\\"Title: Diagnostics\\\"}"}
```

```

    ],
    "indexes": {
        "1000": {
            "g": 1, // group
            "n": "{\\"Index: Play\\"}", // name
            "f": "w", // flag // "r", "w", "rw"
            "t": "b", // data type // i=integer, f=float, b=bool, s=string
            "i": "b", // input type // p=print, b=button, n=number, pass=password,
                                r=range, t=text, s=selectbox, d=date, ti=time
            "s": true // show
        },
        "1001": {
            "g": 1, // group
            "n": "{\\"Index: Stop\\"}", // name
            "f": "w", // flag // "r", "w", "rw"
            "t": "b", // data type // i=integer, f=float, b=bool, s=string
            "i": "b", // input type // p=print, b=button, n=number, pass=password,
                                r=range, t=text, s=selectbox, d=date, ti=time
            "s": true // show
        },
        "1002": {
            "g": 1, // group
            "n": "{\\"Index: Seek to\\"}", // name
            "f": "w", // flag // "r", "w", "rw"
            "t": "f", // data type // i=integer, f=float, b=bool, s=string
            "i": "n", // input type // p=print, b=button, n=number, pass=password,
                                r=range, t=text, s=selectbox, d=date, ti=time
            "d": 0, // default
            "min": 0, // min
            "max": 65535, // max
            "s": true // show
        },
        "2000": {
            "g": 2, // group
            "n": "{\\"Index: Version\\"}", // name
            "f": "r", // flag // "r", "w", "rw"
            "t": "f", // data type // i=integer, f=float, b=bool, s=string
            "i": "p", // input type // p=print, b=button, n=number, pass=password,
                                r=range, t=text, s=selectbox, d=date, ti=time
            "d": 0, // default
            "min": 0.1, // min
            "max": 99.99, // max
            "s": true // show
        },
        "65535": {
            "g": 2, // group
            "n": "{\\"Index: Example\\"}", // name
            "f": "rw", // flag // "r", "w", "rw"
            "t": "f", // data type // i=integer, f=float, b=bool, s=string
            "i": "p", // input type // p=print, b=button, n=number, pass=password,
                                r=range, t=text, s=selectbox, d=date, ti=time
            "u": "°C", // units
            "d": 0, // default
            "min": 0, // min
            "max": 140, // max
            "s": true, // show
            "val": {
                "6": "{\\"Index: Value 6\\"}",
                "1": "{\\"Index: Value 1\\"}",
                "2": "{\\"Index: Value 2\\"}",
                "3": "{\\"Index: Value 3\\"}",
                "4": "{\\"Index: Value 4\\"}",
                "5": "{\\"Index: Value 5\\"}",
            },
            "rc": function(){ // read callback
            },
            "wc": function(){ // write callback
            }
        },
    },
    "errors": {
        "100": "{\\"Error: Example error\\"}"
    }
}

```