

KomUniVal

Štruktúra KomUniVal

```
struct KomUniVal {
    unsigned short uvStatus;
    TLimitStatus   uvLimitStatus;
    unsigned int   uvProcAlarmStatus;
    TValueType     uvValType;
    unsigned short uvFlags;
    D2Time         uvValTime;
    D2Time         uvProcAlarmTime;
    TBVal          uvBoval;
    int            uvIntval;
    double         uvRealval;
    TStVal         uvStationval;
    D2Time         uvTmAval;
    double         uvTmRval;
    TQVal          uvQval;
    char           *uvTxtval;
};
```

Štruktúra *KomUniVal* prenáša hodnotu a stav objektu systému D2000 ([meraný bod](#), [stanica](#), [linka](#)). Jej jednotlivé položky majú nasledovný význam:

uvStatus

Môže nadobúda kombináciu nasledovných hodnôt:

```
#define SB_Val_Valid      0x0001
#define SB_Pa_Alarm      0x0002
#define SB_Pa_NoAck      0x0004
#define SB_Pa_Blocked    0x0008
#define SB_Val_Weak      0x0010
#define SB_Val_NoAck     0x0020
#define SB_Val_Transient 0x0040
#define SB_Val_Default   0x0080
#define SB_Val_Manual    0x0100
#define SB_Pa_Critical   0x0200
```

Implementácia komunikačného protokolu môže pre objekty typu [Meraný bod](#) meniť iba hodnoty:

- SB_Val_Valid – označuje platnosť hodnoty meraného bodu
- SB_Val_Weak – hodnota meraného bodu je nedôveryhodná (weak)

Ostatné hodnoty sú menené interne komunikaným procesom.

uvLimitStatus

```
typedef enum {LS_InLimit, LS_VL_Limit, LS_L_Limit, LS_H_Limit, LS_VH_Limit, LS_LimitsProblem} TLimitStatus;
```

Parameter označuje limitný stav hodnoty meraného bodu. Implementácia protokolu nastavuje tento parameter iba v prípade hodnoty ST_SOURCE_LIMITS parametra *Stat* call-back funkcie [PointNewValue](#).

Možné hodnoty sú:

- LS_InLimit - hodnota je v definovaných medziach
- LS_VL_Limit - hodnota klesla pod najnižšiu medzu (len pre body typu Ai, Ao, Ci, Co)
- LS_L_Limit - hodnota klesla pod dolnú medzu (len pre body typu Ai, Ao, Ci, Co)
- LS_H_Limit - hodnota prekroila hornú medzu (len pre body typu Ai, Ao, Ci, Co)
- LS_VH_Limit - hodnota prekroila najvyššiu medzu (len pre body typu Ai, Ao, Ci, Co)
- LS_LimitsProblem - problém s hodnotami medzí meraného bodu (prekriženie hodnôt alebo neplatnosť hodnoty dynamickej medze)

uvProcAlarmStatus

Príznaky procesných alarmov. Implementácia protokolu tento parameter nesmie meniť.

uvValType

```
typedef enum {VT_NAN, VT_Bo, VT_Int, VT_Re, VT_Di, VT_Do, VT_De, VT_Ai, VT_Ao, VT_Ae, VT_Ci, VT_Co, VT_Ce, VT_St, VT_Li, VT_Al, VT_Pr, VT_TmA, VT_TmR, VT_TiA, VT_ToA, VT_TiR, VT_ToR, VT_Txt, VT_Arr, VT_Qi, VT_Unused1, VT_TxtI, VT_TxtO} TValueType;
```

Typ hodnoty objektu. Význam pre komunikáciu majú iba nasledovné typy, ktoré:

- VT_Ai - reálny vstup
- VT_Ao - reálny výstup
- VT_Ci - celoíselný vstup
- VT_Co - celoíselný výstup
- VT_Di - dvojstavový vstup
- VT_Do - dvojstavový výstup
- VT_TiA - absolútny as vstup
- VT_ToA - absolútny as výstup
- VT_TiR - relatívny as vstup
- VT_ToR - relatívny as výstup
- VT_Qi - štvorstavový vstup
- VT_TxtI - textový vstup
- VT_TxtO - textový výstup

Táto hodnota sa nesmie meniť!

uvFlags

```
#define VF_A 0x0001
#define VF_B 0x0002
#define VF_C 0x0004
#define VF_D 0x0008
#define VF_E 0x0010
#define VF_F 0x0020
#define VF_G 0x0040
#define VF_H 0x0080
#define VF_I 0x0100
#define VF_J 0x0200
#define VF_K 0x0400
#define VF_L 0x0800
#define VF_M 0x1000
#define VF_N 0x2000
#define VF_O 0x4000
#define VF_P 0x8000
```

Implementácia môže ubovone nastavova kombinácie 16 flagov ABCDEFGHIJKLMNOP hodnoty meraného bodu. Hodnoty flagov sú VF_A až VF_P.

uvValTime

Reálny as aktuálnej hodnoty.

uvProcAlarmTime

Reálny as poslednej zmeny príznaku procesných alarmov. Nemeni.

uvBoval

Aktuálna hodnota dvojstavového vstupu alebo výstupu Di a Do.

Hodnoty:

- D_False - FALSE
- D_True - TRUE
- D_Oscillate - oscilácia hodnoty – nenastavova, analyzuje a prípadne nastavuje proces [D2000 KOM](#).

uvIntval

Aktuálna hodnota celoíselného vstupu alebo výstupu Ci a Co.

uvRealval

Aktuálna hodnota reálneho vstupu alebo výstupu Ai a Ao.

uvStationval

```
typedef enum {ST_ON, ST_OFF, ST_COMERR, ST_HARDERR, ST_SIMUL, ST_WAIT} TStVal;
```

Stav stanice. Nikdy nemete hodnotu stavu stanice priamo, použite call-back procedúru *SetStationStatus*.

uvTmAval

Aktuálna hodnota absolútnych asových typov bodov TiA a ToA.

uvTmRval

Aktuálna hodnota relatívnych asových typov bodov TiR a ToR.

uvQval

Aktuálna hodnota štvorstavových typov bodov Qi.

```
typedef enum {Q_Trans, Q_Off, Q_On, Q_Err, Q_Oscillate} TQVal;
```

Možné hodnoty sú Q_Trans, Q_Off, Q_On, Q_Err alebo Q_Oscillate. Hodnotu Q_Oscillate nepoužíva, nastavuje ju proces [D2000 KOM](#) v prípade vyhodnocovania oscilácie.

uvTxtval

Pointer na string aktuálnej hodnoty textových typov bodov TxtI a TxtO.

Poznámka: Komunikaný proces si alokuje vlastné kópie hodnôt textových premenných po volaní call-back procedúry [PointNewValue](#).



Súvisiace stránky:

[D2000 KomAPI - štruktúry](#)