



# Knihovna TCOM052.DLL

## popis knihovny



### 1. Úvod

Tento dokument obsahuje popis rozhraní komunikační knihovny TCOM052.DLL určené pro komunikaci s moduly MicroUnit série pomocí protokolu AIBus-2 přes standardní seriový COM port. Knihovna je určena pro operační systémy Windows XP a vyšší.

### 2. Deklarace datových struktur

Jako parametr se v některých funkcích používají pole s následující deklarací:

```
uint32_t Buff[1024];      // pole 1024 prvků (uint32)
uint32_t ErrText[100];    // pole 100 prvků (uint32)
```

### 3. Funkce

Všechny funkce vrací 0 (uint32\_t) v případě úspěšného provedení. Nenulová hodnota signalizuje chybu, pro její identifikaci lze použít funkci TCOM052\_Error.

#### 3.1. TCOM052\_Version

##### Popis

Funkce vrací verzi knihovny DLL (nutno vydělit 100, např. vrácená hodnota 100 znamená verzi 1.00).

##### Definice

```
__stdcall uint32_t TCOM052_Version(uint32_t *Ver);
```

##### Parametry

Ver                   ukazatel na uint32\_t, verze knihovny

##### Návratová hodnota

Chybový kód provedení funkce (viz. Přehled chybových kódů)

##### Příklad

```
uint32_t result;
uint32_t VerzeDLL;

result = TCOM052_Version(&VerzeDLL);

// vyhodnocení chyby volání funkce TCOM052_Version
if (result!=0)
{
    // chyba volání funkce
}

// výpis verze
printf("Verze knihovny Tcom052.dll: %d.%02d\n", (VerzeDLL/100), (VerzeDLL%100));
```

### 3.2. TCOM052\_Init

#### Popis

Inicializuje knihovnu a otevře zadaný standardní COM. Nastaví default komunikační parametry (9600Bd, timeout 10ms, 2 opakování při chybě komunikace). Vrací handle otevřeného zařízení.

#### Definice

```
__stdcall uint32_t TCOM052_Init(uint32_t *PHNDL, uint32_t COM);
```

#### Parametry

PHNDL        ukazatel na uint32\_t, handle otevřeného zařízení  
COM         uint32\_t, číslo COM (např. 1 = COM1)

#### Návratová hodnota

Chybový kód provedení funkce (viz. Přehled chybových kódů)

#### Příklad

```
uint32_t result;  
uint32_t HNDL;  
uint32_t PortCOM;  
  
// otevření COM1  
PortCOM = 1;  
result = TCOM052_Init(&HNDL, PortCOM);  
  
// vyhodnocení chyby volání funkce TCOM052_Init  
if (result!=0)  
{  
    // chyba volání funkce  
}
```

### 3.3. TCOM052\_SetConfig

#### Popis

Nastaví komunikační rychlost, timeout a opakování pro komunikaci s moduly MicroUnit na lince RS-485. Povolené rychlosti odpovídají modulům MicroUnit, tj. 600Bd, 1200Bd, 2400Bd, 4800Bd, 9600Bd, 19200Bd, 38400Bd, 57600Bd a 115200Bd. Timeout, tj. max. doba čekání na zahájení odpovědi modulu, je nastavitelný od 1 do 65535ms. Opakování určuje počet zopakovaných volání na modul MicroUnit v případě chybné komunikace, tzn. např. pro Opakování=2 je na modul MicroUnit voláno v případě chyby komunikace celkem 3x.

Pokud vyhoví default hodnoty konfigurace (viz. TCOM052\_Init), není nutno tuto funkci volat.

#### Definice

```
__stdcall uint32_t TCOM052_SetConfig(uint32_t HNDL, uint32_t Baud, uint32_t Timeout, uint32_t Opakovani);
```

#### Parametry

HNDL	uint32_t, handle otevřeného zařízení.
Baud	uint32_t, komunikační rychlost koncového modulu MicroUnit
Timeout	uint32_t, doba čekání na odpověď (1-65535ms)
Opakovani	uint32_t, počet opakování volání při chybě komunikace (0-15)

#### Návratová hodnota

Chybový kód provedení funkce (viz. Přehled chybových kódů)

#### Příklad

```
uint32_t result;
uint32_t HNDL;
uint32_t BaudMU;
uint32_t TimeoutMU;
uint32_t OpakovaniMU;

BaudMU = 9600;
TimeoutMU = 100;
OpakovaniMU = 2;
result = TCOM052_SetConfig(HNDL, BaudMU, TimeoutMU, OpakovaniMU);

// vyhodnocení chyby volání funkce TCOM052_SetConfig
if (result!=0)
{
    // chyba volání funkce
}
```

### 3.4. TCOM052\_GetConfig

#### Popis

Funkce vrací aktuální komunikační parametry knihovny.

#### Definice

```
__stdcall uint32_t TCOM052_GetConfig(uint32_t HNDL, uint32_t *Baud, uint32_t *Timeout,  
uint32_t *Opakovani);
```

#### Parametry

HNDL	uint32_t, handle otevřeného zařízení.
Baud	ukazatel na uint32_t, komunikační rychlost koncového modulu MicroUnit
Timeout	ukazatel na uint32_t, doba čekání na odpověď (1-65535ms)
Opakovani	ukazatel na uint32_t, počet opakování volání při chybě komunikace (0-15)

#### Návratová hodnota

Chybový kód provedení funkce (viz. Přehled chybových kódů)

#### Příklad

```
uint32_t result;  
uint32_t HNDL;  
uint32_t BaudMU;  
uint32_t TimeoutMU;  
uint32_t OpakovaniMU;  
  
result = TCOM052_GetConfig(HNDL, &BaudMU, &TimeoutMU, &OpakovaniMU);  
  
// vyhodnocení chyby volání funkce TCOM052_GetConfig  
if (result!=0)  
{  
    // chyba volání funkce  
}  
  
// výpis komunikačních parametrů  
printf("Baud: %lu Timeout: %lu Opakovani: %lu\n", BaudMU, TimeoutMU, OpakovaniMU);
```

### 3.5. TCOM052\_Send

#### Popis

Funkce vyšle dotaz (BuffT) na modul MicroUnit, vrátí odpověď modulu (BuffR), Status registr modulu (StatusR) a Status příjmu (StatusP).

BuffT a BuffR jsou pole obsahující vyslanou resp. přijatou zprávu. Obsah BuffT odpovídá vysílané zprávě protokolu AIBus-2, bez CRC. Přesný význam je uveden v popisu protokolu.

BuffT[0] - adresa volaného modulu MicroUnit (0-255)

BuffT[1] - požadovaná funkce AIBus-2 (0,1,2,3,100,101,102,103,200)

BuffT[2] - požadovaná periferie modulu MicroUnit (0-255)

BuffT[3] - přímý povel (0)

BuffT[4] - D1 datový byte dotazu (0-255)

BuffT[5] - D2 datový byte dotazu (0-255)

BuffT[6] - D3 datový byte dotazu (0-255)

BuffT[7] - D4 datový byte dotazu (0-255)

Pole BuffR obsahuje pouze přijaté datové byty odpovědi.

BuffR[0] - D1 datový byte odpovědi (0-255)

BuffR[1] - D2 datový byte odpovědi (0-255)

BuffR[2] - D3 datový byte odpovědi (0-255)

BuffR[3] - D4 datový byte odpovědi (0-255)

StatusR obsahuje přijatý Status registr modulu. Ve StatusP je uložen kód úspěšnosti komunikace s modulem MicroUnit. Slovní popis kódu lze získat pomocí funkce TCOM052\_Error.

#### Definice

```
__stdcall uint32_t TCOM052_Send(uint32_t HNDL, uint32_t *BuffT, uint32_t *BuffR, uint32_t *StatusR, uint32_t *StatusP);
```

#### Parametry

HNDL            uint32\_t, handle otevřeného zařízení.

BuffT            ukazatel na pole uint32\_t, vysílaná AIBus-2 zpráva

BuffR            ukazatel na pole uint32\_t, přijatá data odpovědi

StatusR          ukazatel na uint32\_t, přijatý Status registr modulu

StatusP          ukazatel na uint32\_t, chybový kód komunikace

#### Návratová hodnota

Chybový kód provedení funkce (viz. Přehled chybových kódů)

**Příklad**

```
uint32_t result;
uint32_t HNDL;
uint32_t BuffT[1024];
uint32_t BuffR[1024];
uint32_t StatusR;
uint32_t StatusP;
uint8_t  DIN_0_7;
uint8_t  DIN_8_15;
uint8_t  DIN_16_23;
uint8_t  DIN_24_31;

// dotaz AIBus-2 protokolu (čtení digitálních vstupů DIN0 až DIN31)
BuffT[0] = 1;           // adresa modulu MicroUnit
BuffT[1] = 0;           // funkce modulu MicroUnit (0=čtení, 1=zápis)
BuffT[2] = 64;          // periferie modulu MicroUnit (64=DIN)
BuffT[3] = 0;           // pomocný povel
BuffT[4] = 0;           // data D0
BuffT[5] = 0;           // data D1
BuffT[6] = 0;           // data D2
BuffT[7] = 0;           // data D3 (zapisovaná data jsou při čtení bez významu)
result = TCOM052_Send(HNDL, BuffT, BuffR, &StatusR, &StatusP);

// vyhodnocení chyby volání funkce TCOM052_Send
if (result!=0)
{
    // chyba volání funkce
}

// vyhodnocení chyby komunikace s modulem MicroUnit - StatusP
if (StatusP!=0)
{
    // chyba komunikace s modulem
}

// komunikace v pořádku - přijata platná odpověď
DIN_0_7  = (uint8_t)BuffR[0];      // D1 (první byte dat odpovědi) - LSB
DIN_8_15 = (uint8_t)BuffR[1];      // D2 (druhý byte dat odpovědi)
DIN_16_23 = (uint8_t)BuffR[2];      // D3 (třetí byte dat odpovědi)
DIN_24_31 = (uint8_t)BuffR[3];      // D4 (čtvrtý byte dat odpovědi) - MSB
```

**Pozn:**

*BuffT a BuffR jsou z důvodu kompatibility pole uint32\_t, i když je využit pouze jeden byte.*



**Pozn:**

*Při čtení analogového vstupu resp. periferie s real datovým formátem AIBus-2 (IEEE488.2) je nutno přijatá data převést do formátu IEEE754 pomocí následující funkce:*

```
// Funkce pro převod přijaté zprávy na hodnotu real. Vstupem je přijatá
// datová část odpovědi modulu MicroUnit, výstupem je spočítaná real hodnota.

double PrevodAIBusReal(uint32_t D1, uint32_t D2, uint32_t D3, uint32_t D4)
{
    uint8_t i, Exp;
    double R;

    R=(D3&0x7F)*1.0;
    R=(R*256.0+D2)*256.0+D1;
    if(D3>127) R=-R;
    Exp=D4&0x7F;
    if (Exp)
    {
        for(i=0;i<Exp;i++) R=R/10.0;
    }
    return R;
}
```

### **3.6. TCOM052\_Done**

#### **Popis**

Funkce uzavře otevřené zařízení a uvolní knihovnu z paměti.

#### **Definice**

```
__stdcall uint32_t TCOM052_Done(uint32_t *PHNDL);
```

#### **Parametry**

PHNDL        ukazatel na uint32\_t, handle otevřeného zařízení.

#### **Návratová hodnota**

Chybový kód provedení funkce (viz. Přehled chybových kódů)

#### **Příklad**

```
uint32_t result;  
uint32_t HNDL;  
  
result = TCOM052_Done(&HNDL);  
  
// vyhodnocení chyby volání funkce TCOM052_Done  
if (result!=0)  
{  
    // chyba volání funkce  
}
```

### 3.7. TCOM052\_Error

#### Popis

Funkce převede návratový resp. chybový kód na text.

#### Definice

```
__stdcall uint32_t TCOM052_Error(uint32_t ErrCode, uint32_t *ErrText);
```

#### Parametry

ErrCode        uint32\_t, návratový resp. chybový kód

ErrText        ukazatel na pole uint32\_t, ASCII znaky textového popisu chyby

#### Návratová hodnota

Chybový kód provedení funkce (viz. Přehled chybových kódů)

#### Příklad

```
uint32_t StatusP;
uint32_t ErrText[100];

// vyhodnocení chyby komunikace s modulem MicroUnit - StatusP
if (StatusP!=0)
{
    TCOM052_Error(StatusP, ErrText);
    printf("Status komunikace: %lu - %s\n", StatusP, PrevodTextError(ErrText));
}

// Funkce pro převod pole uint32_t na string. Vstupem je text ve formě pole,
// výstupem je text ve formě stringu.

uint8_t* PrevodTextError(uint32_t *B)
{
    static uint8_t s[128];
    uint32_t i;

    for(i=0; (B[i]!=0) && (i<100); i++)
    {
        s[i] = (uint8_t)B[i];
    }

    return s;
}
```

#### 4. Přehled chybových kódů

<i>Kód chyby</i>	<i>Význam</i>
0	V pořádku
1	Chybná komunikační rychlost
2	Neplatný timeout nebo počet opakování
3	Chyba při vysílání zprávy
4	Timeout - nepřišla žádná odpověď
5	Timeout - nepřišla správná odpověď
6	Odpověď přerušena
7	Chyba parity odpovědi
8	Chybná hlavička odpovědi
9	Chyba CRC odpovědi
10	Nelze vytvořit datovou strukturu knihovny
11	Neplatný handle zařízení
12	Neplatné číslo zařízení
13	Nelze otevřít zařízení
14	Zařízení již inicializováno
15	Chyba knihovny (volání API)
16	Chyba knihovny (volání API)
17	Chyba knihovny (volání API)
18	Chyba knihovny (volání API)
19	Chyba knihovny (volání API)
20	Chyba knihovny (volání API)