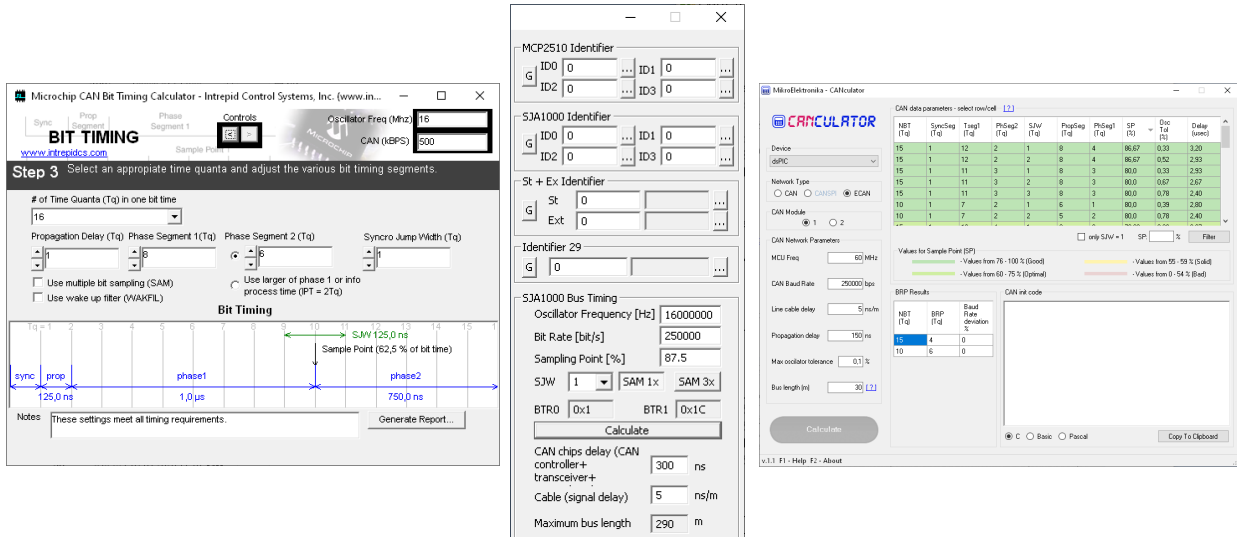


V některých případech je třeba v programu PP2CAN nastavit nestandardní komunikační rychlost, případně přizpůsobit bod vzorkování délce CAN sběrnice a podobně. Program PP2CAN dovoluje pro převodník USB2CAN uživatelsky nastavit parametry časování.

Pro výpočty lze použít různé programy, například Mikroelektronika CANculator, program PP2CAN a podobně. Různé programy podporují rozdílné CAN řadiče a různé parametry.



Také je možné použít online kalkulačku na adrese: <http://www.bittiming.can-wiki.info/>. Ta podporuje asi všechny běžně dostupné externí i interní CAN řadiče.

Použijeme v tomto příkladu tuto online kalkulačku. Jako CAN bus řadič vybereme NXP SJA1000, parametr Clock nastavte na 8Mhz, jelikož krystal v převodníku má hodnotu 16Mhz viz poznámka označená červeně na následujícím obrázku. Zadejte v tomto případě rychlost 250 kbit/s.

NXP SJA1000(Philips) or Intel

Clock Rate **8** in MHz, **from 1 to 300**. Use the value of the clock rate at the first stage of the BaudRatePrescaler BTR, not the clock of the controller or crystal (typically for a 16 MHz clocked NXP SJA1000 use '8').

Sample-Point at: **87.5** in %, **from 50 to 90** (87.5 % is the preferred value used by CANopen and DeviceNet, 75 % is the default value for ARINC 825).

SJW: **1** numerical value **from 1 to ..** (1 is the preferred value used by CANopen and DeviceNet. The value is currently not used in all calculations, please look at the values used below the bit timing table).

The table will be calculated for all CANopen defined bit rates. If you like to have the calculation for one special arbitrary bit rate, enter the the value here in kbit/s **250**

Debug: generates debugging information to the calculation after the table.

[Request Table](#)

A table can be requested by passing parameters on the http request line like <http://www.bittiming.can-wiki.info/?CLK=8&ctype=Philips&calc=1> With possible parameters: CLK=, SJW=, SamplePoint=, ctype=, calc=. Where ctype is the controller type. If calc=1, calc is true, the table is immediately calculated and displayed.

Yellow background rows are settings with an bittime consisting of 16 time quanta tq. At the time this tool was first developed, this seems to be the best value. in these days where we talk about CAN FD, as much as possible time quanta should be used to construct a bit time.

SJA1000 equivalent (can4linux for many board available)

Bit Rate	accuracy	Pre-scaler	Number of time quanta	Seg 1 (Prop_Seg+Phase_Seg1)	Seg 2	Sample Point at	Register BTR0	Register BTR1
250	0.0000	2	16	13	2	87.5	0x01	0x1c
250	0.0000	4	8	6	1	87.5	0x03	0x05



Type: Philips, Clock: 8MHz, max brp: 64, SP: 87.5%, min tq: 8, max tq: 25, FD factor: undefined, SJW: 1

V tabulce jsou vypočtené hodnoty, použijeme první řádek:

Bit BTR0	7	6	5	4	3	2	1	0
Vypočteno	0x01							
Binárně	0	0	0	0	0	0	0	1
Položka	SJW			BRP				
Část	0x00 = 0d			0x01 = 1d				

Vypočtena byla hodnota 0x01, po převodu do binární podoby a rozdělení na části dle mapy registru získáme hodnoty:

BRP je Baud Rate Prescaler. Dělička začíná hodnotou 1, nelze dělit kmitočty 0. Tedy nastavení bitů na 0 odpovídá děličce 1

$$BRP = 32 \times BRP.5 + 16 \times BRP.4 + 8 \times BRP.3 + 4 \times BRP.2 + 2 \times BRP.1 + BRP.0 + 1$$

Ze vzorce je taktéž zřejmé, že minimální hodnota je 0.

Tedy pro hodnotu bitů BRP v registru 1 je hodnota délky BRP 2.

SJW je Synchronization Jump Width, minimální délka je 1 časové kvantum. Hodnota bitů SJW 0 tedy odpovídá 1 kvantu.

$$SJW = 2 \times SJW.1 + SJW.0 + 1$$

Tedy pro hodnotu bitů SJW v registru 0 je hodnota délky SJW 1.

Bit BTR1	7	6	5	4	3	2	1	0
Vypočteno	0x1C							
Binárně	0	0	0	1	1	1	0	0
Položka	SAM	TSEG2			TSEG1			
Část	0	0x01 = 1d			0x0C = 12d			

Vypočtena byla hodnota 0x1C, po převodu do binární podoby a rozdělení na části dle mapy registru získáme hodnoty:

$$TSEG1 = 8 \times TSEG1.3 + 4 \times TSEG1.2 + 2 \times TSEG1.1 + TSEG1.0 + 1$$

Tedy pro hodnotu bitů TSEG1 = 12 v registru, je hodnota délky TSEG1 v kvantech je rovna 13. Opět platí že 0 v registru odpovídá nejmenší možné délce – tedy jednomu kvantu.

$$TSEG2 = 4 \times TSEG2.2 + 2 \times TSEG2.1 + TSEG2.0 + 1$$

Tedy pro hodnotu bitů TSEG2 = 1 v registru, je hodnota délky TSEG2 2.

V konfiguračním souboru X2CAN.cfg programu PP2CAN lze najít sekci USR_TIMING. Upravme ji podle vypočtených hodnot. Parametry jsou skutečné hodnoty děličky/kvant. Parametr PRSEG není pro převodník USB2CAN použit.

```
[USR_TIMING]
BRP=2
SJW=1
PRSEG=0
PHSEG1=13
PHSEG2=2
SAM=No
```

V programu PP2CAN je obsažen tento kód:

```
t1 = (UserTiming.SJW-1) << 6;
t1 += ((UserTiming.BRP-1)& 0x3F);
```

```

if(UserTiming.SAM) t2 = 0x80;
else t2 = 0x00;
t2 += (((UserTiming.PHSEG2-1)&0x07)<<4);
t2 += ((UserTiming.PHSEG1-1)&0x0F);

```

Tedy je prováděn zpět přepoččet na hodnoty v registrech z délky kvant.

Z toho je zřejmé, že při tomto nastavení se vypočtou zpětně hodnoty

```

t1 = 0x01
t2 = 0x1C

```

Zkusme nějakou nestandardní rychlost. V ojedinělých případech lze narazit na rychlost CANu označenou 667k. Reálně se jedná o rychlost 666,6666.....kbit.

Stránka www.bittiming.can-wiki.info nám vypočte hodnoty: BTR0 = 0x00, BTR1 = 0x09

Bit BTR0	7	6	5	4	3	2	1	0
Vypočteno	0x00							
Binárně	0	0	0	0	0	0	0	0
Položka	SJW				BRP			
Část	0x00 = 0d				0x00 = 0d			

SJW je tedy rovno 1, BRP je také rovno 1.

Bit BTR1	7	6	5	4	3	2	1	0
Vypočteno	0x09							
Binárně	0	0	0	0	1	0	0	1
Položka	SAM	TSEG2			TSEG1			
Část	0	0x00 = 0d			0x09 = 9d			

TSEG2 je tedy rovno 1, TSEG1 je rovno 10 (dekadicky).

Příslušná sekce je pak nastavena takto:

```

[USR_TIMING]
BRP=1
SJW=1
PRSEG=0
PHSEG1=10
PHSEG2=1
SAM=No

```

Program PP2CAN nedovoluje nastavit automaticky po startu nestandardní hodnoty rychlosti. Proto odpojte převodník USB2CAN od CAN sběrnice, otevřete program PP2CAN, nastavte rychlost CANu na volbu USER_TM. Převodník se nastaví a je možné jej nastavit na měřenou CAN sběrnici.



Při použití vlastního programu s využitím X2CAN API pro převodník USB2CAN se používá pro nastavení funkce `UsrCANSpeedPrepare` a při volání `USB2CAN_Open` se jako hodnota rychlosti použije hodnota `SPEED_USR`.