



## STR71X CLOCK CONFIGURATIONS EASY SETUP

---

### INTRODUCTION

This application note describes how to use the “*STR71x\_Calculation.xls*” file which is intended as a tool to assist in the configuration of the STR71x clocks and baud rates.

The STR71x microcontroller contains a variety of peripherals running at different frequencies. Some peripherals have a prescaler to adjust their working frequency, but some others don't and need a specific frequency to run correctly.

The file “*STR71x\_Calculation.xls*” calculates all the working frequencies and shows the related register values for each peripheral.

**Note:** This application note doesn't substitute the fact that it is mandatory to read the reference manual before the use of this tool.

## 1 PRCCU

This peripheral allows the configuration of the clock Prescaler values to adjust the needed frequencies for the CPU, the APB1 and the APB2.

This peripheral contains two calculation sheets:

- CCU: Clock Control Unit calculation
- CDU: Clock Distribution Unit calculation

### 1.1 CCU

In this calculation sheet, it is necessary to enter the input clock frequency and the PLL configuration parameters (the Prescaler ratio, the PLL multiplier ratio, the PLL divider factor, the input clock source and the running mode).

**Figure 1. CCU calculation sheet**

### STR 71x : Clock Control Unit Calculation

Clock-IN:  MHz  
 Divided by:  Ratio

Hint: Set Bit 6 of PRCCU PLL1CR to 1

PLL Multiply:	12	16	20	24		
Direct:	8	96	128	160	192	MHz
Div. 1/16:	0.5					MHz
Divide 1/N						
1	not spec.	not spec.	not spec.	not spec.		MHz
2		48	64	not spec.	not spec.	MHz
3		32	42.667	53.333	64	MHz
4		24	32	40	48	MHz
5		19.2	25.6	32	38.4	MHz
6		16	21.333	26.667	32	MHz
7		13.714	18.286	22.857	27.429	MHz

**Actual calculated RCLK Frequency**

Registers "PRCCU\_PLL1CR"  (lower 16 bit of 32)

"PRCCU\_CFR"  (lower 16 bit of 32)

Register:	"PRCCU_CFR"	Bit 3	Bit 0
Clk/16	Not allowed	0	0
Direct	0.5 MHz	1	0
PLL	40 MHz	1	1

"RCLK" is: 40 MHz

(C) COPYRIGHT 2004 STMicroelectronics  
MCD Application Team

**Setting the operating Frequency:**

1) Fill in Input Clock Frequency here  MHz  
*(externally supplied Clock / Xtal)*

2) Select Prescaler ratio  Prescaler  
*(Input Clock divider / 1:1 or 1:2)*  
Reg: "PRCCU\_CFR" / Bit 15

3) Select PLL Multiplier Ratio   
*programm PLL Multiplication Ratio by*  
Reg: "PRCCU\_PLL1CR" / Bit 5 & Bit 4  binary

4) Select 1/N Divider   
*programm 1/N divider by*  
Reg: "PRCCU\_PLL1CR" / Bits 2, 1, 0  binary

5) Select Direct, 1/16 or PLL Clock Source  
*programm Clock Multiplexer by*  
Reg: "PRCCU\_CFR" / Bit 0  Bit 0  
  
and Bit 3  Bit 3

6) PLL free running mode  
Reg: "PRCCU\_PLL1CR" / Bit 7  Bit 7

This sheet calculates all the frequencies which can be generated by the PLL.

As an example, as shown in Figure 1, the input frequency is set to 16MHz, the Prescaler ratio (1/2) is activated, the multiplier ratio is set to 20, the divider factor is set to 4 and the PLL is locked. This configuration allows the RCLK frequency to be set to 40MHz.

2/19

## 1.2 CDU

This calculation sheet is used to configure the working clock frequency for the CPU, the APB1 and the APB2.

**Figure 2. CDU calculation sheet**

**RCLK = 40 MHz** (Clock Control Unit)  
defined in "Clock Control Unit", to modify click here

**MCLK = 40 MHz** (CPU & Memory Clock)  
CPU and Memory Interface runs with this frequency

---

**PCLK1 = 20 MHz** (Peripheral Clock 1)  
Peripheral Clock 1 is supplied for:  
I<sup>2</sup>C-Bus, UARTs, USB, BSPI, HDLC - Interfaces

---

**PCLK2 = 10 MHz** (Peripheral Clock 2)  
Peripheral Clock 2 is supplied for:  
Wake-Up, Interrupt, I/O-Ports, ADC, Timer, RTC, Watchdog

Registers: "PRCCU\_MDIVR"  (16 bit Reg)  
"PRCCU\_PDIVR"  (16 bit Reg)

**(C) COPYRIGHT 2004 STMicroelectronics**  
MCD Application Team

Set "Prescaler" for CPU & Memory   
Reg: "PRCCU\_MDIVR", Bit 1&0  binary

**Setting the Serial Interface Clock (APB1 Bridge)**

Set "Prescaler" for APB1 Peripherals   
Reg: "PRCCU\_PDIVR", Bit 1&0  binary

**Setting the System Peripherals Clock (APB2 Bridge)**

Set "Prescaler" for APB2 Peripherals   
Reg: "PRCCU\_PDIVR", Bit 9&8  binary

Just select the required Prescaler and obtain as a result the MCLK, the PCLK1 and the PCLK2 working frequency values.

As an example, as shown in [Figure 2](#), the Prescaler for CPU & memory is set to 1, the Prescaler for the APB1 peripherals is set to 2 and the Prescaler for the APB2 peripherals is set to 4. This configuration allows the MCLK to be set to 40MHz, the PCLK1 to 20 MHz and the PCLK2 to 10MHz.

## 2 UART

In this calculation sheet we have two cases:

- Enter the Prescaler and obtain as a result the correspondent baud rate,
- Or, just enter the required baud rate and obtain as a result the corresponding Prescaler.

Figure 3. UART calculation sheet

Clk In  MHz  
 Prescaler  Ratio    PCLK1  Ratio

---

PCLK1 =  MHz

"UART\_BR" Register is set to:

Register: "UART\_BR"  (16 Bit Register)

BAUD Rate calculates to:  $PCLK1 / (16 \times \langle UART\_BR \rangle) =$

**BAUD Rate = 9.615 kBAUD**

(C) COPYRIGHT 2004 STMicroelectronics  
MCD Application Team

**Calculate Baud Rates of STR 71x UART's**

**1) Fill in Reload-Value to calculate Baud-Rate**  
 Reg: "UART\_BR" Register  
 max. Value = 65534     decimal  
*(see left table)*

**2) Fill in Baud-Rate to calculate Reload-Value**  
 max. Value = 635 *(see right table)*     kBaud

Home
Back
Continue >

Reload-Value calculated for a certain BAUD-Rate is rounded by Spreadsheet

**Calculate the UART's Baud-Rate**

Value: provides BAUD rate

PLL Multiply:	12	16	20	24	
Direct:	1.923				kBaud
Div. 1/16:	0.12				kBaud
<b>Divide 1/N</b>					
1	not spec.	not spec.	not spec.	not spec.	kBaud
2	11.538	15.385	not spec.	not spec.	kBaud
3	7.692	10.256	12.82	15.385	kBaud
4	5.769	7.692	9.615	11.538	kBaud
5	4.615	6.154	7.692	9.231	kBaud
6	3.846	5.128	6.41	7.692	kBaud
7	3.297	4.396	5.494	6.594	kBaud
0 (Zero) means not available					
<b>min. Deviation:</b>	<b>no match</b>	<b>no match</b>	<b>0.156006</b>	<b>no match</b>	in %
	%	%	%	%	

**Calculate the Register Value for Baud Rate**

kBaud: is achieved with Ratio

Deviation:		12	16	20	24	
%						
no match	26					decimal
no match	2					decimal
<b>Divide 1/N</b>						
1		0	0	0	0	decimal
2		156	208	0	0	decimal
3		104	139	174	208	decimal
4		78	104	130	156	decimal
5		63	83	104	125	decimal
6		52	69	87	104	decimal
7		45	60	74	89	decimal
0 (Zero) means not available						

In this sheet there are two tables, the first one gives the baud rate for all the PLL frequency for a given Prescaler and the deviation for the actual PLL configuration. The second table calculates the corresponding prescaler for a given baud rate.

Figure 3 shows how to configure the actual baud rate to 9600 BAUD by setting the Prescaler to 130.

**Note:** For more details on the allowed value and standard baud rates, refer to the UART section in the reference manual.

4/19

### 3 I2C

The I2C has two functioning modes:

- The Standard I2C mode,
- The Fast I2C mode.

**Figure 4. I2C calculation sheet**

	<b>16</b>	MHz	<b>Input Clock Frequency (PLL)</b>			
	<b>2</b>	Ratio	<b>Prescaler Ratio (Clk In)</b>			

<b>PLL Multiply:</b>	<b>12</b>	<b>16</b>	<b>20</b>	<b>24</b>	
Direct:	8	96	128	160	192 MHz
Div. 1/16:	0.5				MHz
Divide 1/N					
1	not spec.	not spec.	not spec.	not spec.	MHz
2		48	64	not spec.	MHz
3		32	42.667	53.333	64 MHz
4		24	32	40	48 MHz
5		19.2	25.6	32	38.4 MHz
6		16	21.333	26.667	32 MHz
7		13.714	18.286	22.857	27.429 MHz

**Calculating the I<sup>2</sup>C Bus Speed** (in Master Mode)

<b>PLL Multiply:</b>	<b>12</b>	<b>16</b>	<b>20</b>	<b>24</b>	
Direct:	10.02506				KHz
Div. 1/16:	0.626566				KHz
Divide 1/N					
1	not spec.	not spec.	not spec.	not spec.	KHz
2		60.15	80.201	not spec.	KHz
3		40.1	53.467	66.834	80.201 KHz
4		30.075	40.1	50.125	60.15 KHz
5		24.06	32.08	40.1	48.12 KHz
6		20.05	26.734	33.417	40.1 KHz
7		17.186	22.914	28.643	34.372 KHz

(C) COPYRIGHT 2004 STMicroelectronics  
MCD Application Team

**Calculate Clock Rate of STR 71x's - I<sup>2</sup>C**

**1) Fill in Value for Clock Divider**

Reg. *lsb's* I2C\_CCR / Bits 0..6  decimal

*msb's* I2C\_ECCR/Bits 0..4

**entered value is within limits**

value must be lower than 4096 and higher than 1 !!!!

**2) Select Fast / Slow Mode in CCR**

Reg. I2C\_CCR/Bit 7

**Actual I<sup>2</sup>C Bus Settings:**

PCLK1 for I<sup>2</sup>C Bus is  MHz

Register "OAR2" (8 Bit)  binary

Register "CCR" (8 Bit)  binary

Register "ECCR" (8 Bit)  binary

I<sup>2</sup>C Bus works at: 50.125 kHz

Standard Mode is specified for 0... 100 kHz, fast mode at 100..400kHz

In this calculation sheet, it is necessary to select the functioning mode (Standard or Fast) by pushing the "Mode" button, and then to input the required Prescaler. We obtain as a result the working frequency of the I2C bus.

Figure 4 illustrates how to configure the I2C mode to Standard mode and the working frequency of the Bus to 50KHz.

**Note:** For more information about the calculation formulas, refer to the I2C section in the reference manual.

4 BSPI

In this calculation sheet, it is necessary to enter the corresponding divider factor, to configure the clock polarity, the clock phase and the BSPI mode. We obtain as a result the working frequency of the BSPI bus.

Figure 5. BSPI calculation sheet

	16	MHz	Input Clock Frequency (PLL)		
	2	Ratio	Prescaler Ratio (Clk In)		

PLL Multiply:	12	16	20	24	
Direct:	8	96	128	160	192 MHz
Div. 1/16:	0.5				MHz
Divide 1/N					
1	not spec.	not spec.	not spec.	not spec.	MHz
2		48	64	not spec.	MHz
3		32	42.667	53.333	64 MHz
4		24	32	40	48 MHz
5		19.2	25.6	32	38.4 MHz
6		16	21.333	26.667	32 MHz
7		13.714	18.286	22.857	27.429 MHz

**Calculating the BSPI Speed**

PLL Multiply:	12	16	20	24	
Direct:	210.5263				KHz
Div. 1/16:	13.15789				KHz
Divide 1/N					
1	not spec.	not spec.	not spec.	not spec.	KHz
2	1263.158	1684.211	not spec.	not spec.	KHz
3	842.105	1122.807	1403.509	1684.211	KHz
4	631.579	842.105	1052.632	1263.158	KHz
5	505.263	673.684	842.105	1010.526	KHz
6	421.053	561.404	701.754	842.105	KHz
7	360.902	481.203	601.504	721.805	KHz

(C) COPYRIGHT 2004 STMicroelectronics  
MCD Application Team

**Calculate Clock Rate of STR 71x's - BSPI**

**1) Fill in Value for Clock Divider**

Reg: **BSPI\_CLK / Bits 0..7**

max. Value = 255       decimal

INFO: value in range

**2) Set Clock Phase & Polarity**

Reg: **BSPI\_CSR1 / Bit 8, Bit 9**

Clock Polarity    Bit 8        **Polarity**    active low

Clock Phase      Bit 9        **Phase**      sec. edge

**3) Configure the BSPI mode**    **BSPI\_CSR1 / Bit1**

BSPI mode        Bit 1        **Mode**        Master

**Actual BSPI Settings:**

Register **"BSPI\_CLK"**    xxxx|xxxx|0001|0011

Register **"BSPI\_CSR1"**    xxxx|xx11|xxxx|xx1x

BSPI works at:    1052.632 kHz

This example, as shown in Figure 5, illustrates how to configure the Clock Divider to 20, to set the BSPI working frequency to 1MHz, to set the polarity to "active low", to set the phase to "second edge" and to configure the BSPI as master.

## 5 WDG

In this calculation sheet, it is necessary to enter the corresponding Prescaler, to set the watchdog Preload value, and finally to select the functioning mode (Free Running Timer mode or Watchdog mode).

We obtain as a result the Free running Timer frequency value if this mode is selected or the Timeout value if the Watchdog mode is selected.

**Figure 6. WDG calculation sheet**

Reg: <b>WDG_CR / Bit 0, Bit 1, Bit 2</b> (16 Bit)		(C) COPYRIGHT 2004 STMicroelectronics MCD Application Team	
<input type="text" value="xxxx xxxx xxxx x001"/>		<b>Calculate time interval for WatchDoG timer</b>	
Bit 0 if set: Watchdog is enabled, else free running Timer <input type="button" value="WE"/>		<b>1) Fill in Value for Watchdog Prescaler</b>	
Bit 1 if set: Counter loads "Preset Value", else stopped <input type="button" value="SC"/>		Reg: <b>WDG_PR / Bits 0..7</b> max. Value = 255 <input type="text" value="19"/> decimal	
Reg: <b>WDG_KR</b> (16 Bit)		<b>INFO:</b> <input type="text" value="Input Frequency for WDG_CNT = 0.5"/> MHz	
Bit 0.. Bit 15 writing two consecutive values to this register reloads the counter with the content of WDG_VR		<b>2) Set Watchdog Count Preload Register</b>	
Reg: <b>WDG_SR</b> (16 Bit)		Reg: <b>WDG_VR / Bit 0...15</b> max. Value = 65535 <input type="text" value="9999"/>	
Bit 0 Bit is set by HW at "End of count", reset by software		<b>INFO:</b> <input type="text" value="valid setting of 'reload register'"/>	
Reg: <b>WDG_MR</b> (16 Bit)		<input type="button" value="Home"/> <input type="button" value="Back"/> <input type="button" value="Continue &gt;"/>	
Bit 0 if set: "End of count" interrupt is enabled, else disabled		<b>Actual Watchdog Settings:</b>	
Free running Timer : <input type="button" value="Disabled"/>		Current "PCLK2" Frequency is: <input type="text" value="10"/> MHz	
		To change PCLK2 frequency go to "Clock Distribution Unit" <input type="button" value="CDU"/>	
		Register "WDG_PR" <input type="text" value="xxxx xxxx 0001 0011"/>	
		Register "WDG_VR" <input type="text" value="0010 0111 0000 1111"/>	
Watchdog Timeout: <input type="button" value="20 ms"/>			

As an example, as shown in Figure 6, this calculation sheet illustrates how to configure the WDG Prescaler to 19, the WDG Count Preload value to 9999 and to select the Free Running Timer mode.

6 TIM

In this calculation sheet, it is necessary to select the clock source (internal or external), to select the external clock edge if external clock was selected, and to configure the needed prescaler. We obtain as a result the counter input frequency and the TIM period.

This sheet shows how to configure the different registers to:

- Enable/Disable the Timer Counter,
- Setting the operation mode of the Timer,
- Setting the Input Capture edge,
- Setting the Output Compare function,
- Setting the Output Compare signal Level,
- Enable the Output Compare signals.

Figure 7. TIM calculation sheet

Reg: **TIMn\_CR1** 16 Bit

**Enable the Timer Counter:**

Bit 15  **Bit 15** if set: the Timer Counter is enabled  **ON**

**Setting the Operation Mode of the Timer:**

Bit 4  **Bit 4** if set: PWM Output is active  **none**

Bit 5  **Bit 5** if set: One Pulse Mode is active  **none**

Bit 14  **Bit 14** if set: PWM Input is active  **ON**

**Setting the Input Capture Edge:**

Bit 2 (IEDGA)  **Bit 2**  **pos.** Bit 3 (IEDGB)  **Bit 3**  **neg.**

**Setting the Output Compare Function:**

Bit 6 (OCAE)  **Bit 6**  **OFF** Bit 7 (OCBE)  **Bit 7**  **OFF**

**Setting The Output Compare Signal Level:**

Bit 8 (OLVL A)  **Bit 8**  **OFF** Bit 9 (OLVL B)  **Bit 9**  **OFF**

**Enable the Output Compare Signals:**

Bit 10 (FOLVA)  **Bit 10**  **OFF** Bit 11 (FOLVB)  **Bit 11**  **OFF**

INFO:

(C) COPYRIGHT 2004 STMicroelectronics  
MCD Application Team

**Calculate time interval for Timer (TIM)**

**1) Select Clock (internal / external)**

Reg. **TIMn\_CR1 / Bit 1 & Bit 0**   **Bit 1**  **Bit 0**

INFO:

external Clock on Port 1.1 is:  kHz

**2) Fill in Value for TIM Prescaler**

Reg. **TIMn\_CR2 / Bits 0..7**

max. Value = 255

INFO:  kHz

**Actual Timer Settings:**

"CNTR" Input Frequency is:  kHz

To change PCLK2 frequency go to "Clock Distribution Unit"

Register **"TIMn\_CR2"**

Register **"TIMn\_CR1"**

---

TIM period: 327.675 ms

TIM Frequency: 3.05180437933928 Hz

As an example, as shown in Figure 7, this calculation sheet shows how to configure the TIM as follows:

- Select the Pulse Width Modulation Input mode,
- Configure ICAP A to a positive edge (Rising edge),
- Configure ICAP B to negative edge (Falling edge),
- Select the clock as internal and configure the Prescaler to 49,
- Enable the Timer Counter.

### 7 USB

The USB controller on STR71x works with an external clock generator supplying 48MHz as defined in the specification.

**Note:** However, due to USB data rate and packet memory interface requirements, the APB1 clock frequency must be greater than 8MHz to avoid data overrun/underrun problems.

## 8 CAN

In this calculation sheet, it will be needed to enter the two baud rates Prescaler (Baud Rate Prescaler and Ext. Baud Rate Prescaler) and to configure the bit time parameters (TSeg1, TSeg2 and SJW).

We obtain as a result:

- The total Prescaler Divider,
- The Time Quanta,
- The total Bit-Time for CAN,
- The total CAN-Bus Bitrate,
- The tolerance calculation.

**Figure 8. CAN calculation sheet**

The CAN Module uses PCLK1:  MHz

Total Prescaler Divider:  value

The "Time Quanta" is:   $\mu$ s

---

*Total Bit-Time is calculated as: (Sync\_Seg + TSeg1 + TSeg2 + 2) \* TimeQuanta*

Total Bit-Time for CAN:   $\mu$ s

Total CAN-Bus Bitrate:  kBits/s

---

*Tolerance calculation:  $\min(PB1, PB2) / 2 * (13 * (\text{bit time} - PB2))$*

Tolerance of CAN\_CLK:  %

---

**CAN Section Settings:**

---

Register "CAN\_BTR"  bin

Register "CAN\_BRPR"  bin

*(C) COPYRIGHT 2004 STMicroelectronics  
MCD Application Team*

**Setting the Bit Timing Register:**

Setting CAN\_BTR (Baud Rate Prescaler):  value

Setting CAN\_BRPR (Ext. Baud Rate Pre.):  value

**Values in range**

---

**Bit time is set by:**

Sync\_Seg (fixed)

TSeg1 (pre sampling): Reg: BTR  value

TSeg2 (post sampling): Reg: BTR  value

SJW (sampling shift): Reg: BTR  value

**Note:** For further details about the formulas, refer to the CAN section in the reference manual.

As an example, the table 1 lists the different standard bitrates of the CAN with their timing parameters with PCLK1 = 20MHz.

**Table 1. Standard bitrate configurations (PCLK1 = 20 MHz)**

Bitrate	NTQ	TSEG1	TSEG2	SJW	BRP
100Kbit/s	20	13	4	3	10
125Kbit/s	16	10	3	3	10
250Kbit/s	8	3	2	2	10
500Kbit/s	8	3	2	1	5
1Mbit/s	5	1	1	1	4

With:

NTQ:  $TSEG1 + TSEG2 + 3$

TSEG1: Time Segment before the sampling point minus Sync\_seg

TSEG2: Time Segment after the sampling point

SJW: Synchronization Jump Width

BRP: Baud Rate Prescaler

### 9 RTC

In this calculation sheet, it will be needed to enter the Prescaler, the Alarm value and the counter value. We obtain as a result:

- The period,
- The frequency of the RTC counter,
- The Alarm time out.

Figure 9. RTC calculation sheet

**ALARM & COUNTER Settings:**

Setting RTC\_ALRH (Alarm Register High)  value

Setting RTC\_ALRL (Alarm Register Low)  value

INFO:

Setting RTC\_CNTH (Counter High)  value

Setting RTC\_CNTL (Counter Low)  value

INFO:

**Actual RTC Settings:**

Total Prescaler Divider:  value

Total Alarm Value:  value

Total Counter Value:  value

RTC period:

RTC Frequency:

Alarm Time Out:

**RTC Section Settings:**

Register "RTC\_PRLH"  bin

Register "RTC\_PRL"  bin

Register "RTC\_ALRH"  bin

Register "RTC\_ALRL"  bin

Register "RTC\_CNTH"  bin

Register "RTC\_CNTL"  bin

(C) COPYRIGHT 2004 STMicroelectronics  
MCD Application Team

Calculate the Oversampling Clock of STR 71x - ADC

**Fill in Value for Clock Divider**

Setting RTC\_PRLH (Prescaler Divider High)  value

Setting RTC\_PRL (Prescaler Divider Low)  value

INFO:

As an example, as shown in [Figure 9](#):

- To configure the RTC period to 1s, just set the Prescaler to 32768;
- To configure the Alarm time out to 1500s just set the Alarm value to 2000 and the Counter value to 500.

### 10 ADC

In this calculation sheet, it is necessary to enter the clock divider. As a result, we will recuperate the ADC sampling frequency.

Figure 10. ADC calculation sheet

	<b>16</b>	MHz	<b>Input Clock Frequency (PLL)</b>			
	<b>2</b>	Ratio	<b>Prescaler Ratio (Clk In)</b>			

PLL Multiply:	12	16	20	24	
Direct:	8	96	128	160	192 MHz
Div. 1/16:	0.5				MHz
Divide 1/N					
1	not spec.	not spec.	not spec.	not spec.	MHz
2		48	64	not spec.	MHz
3		32	42.667	53.333	64 MHz
4		24	32	40	48 MHz
5		19.2	25.6	32	38.4 MHz
6		16	21.333	26.667	32 MHz
7		13.714	18.286	22.857	27.429 MHz

**Calculating the ADC Sampling frequency**

PLL Multiply:	12	16	20	24	
Direct:	98				Hz
Div. 1/16:	6				Hz
Divide 1/N					
1	not spec.	not spec.	not spec.	not spec.	Hz
2		586	781	not spec.	Hz
3		391	521	651	781 Hz
4		293	391	488	586 Hz
5		234	313	391	469 Hz
6		195	260	326	391 Hz
7		167	223	279	335 Hz

(C) COPYRIGHT 2004 STMicroelectronics  
MCD Application Team

**Calculate the sampling frequency of STR71x - ADC**

Fill in Value for Clock Divider

Req: **ADC\_CPR / Bits 0..11**

max. Value = 4095  decimal

INFO: value in range

Home
Back
Continue >

**Actual ADC Settings:**

Current "PCLK2" Frequency is:  MHz

Register "ADC\_CPR"

Sampling frequency: 488 Hz

As an example, as shown in Figure 10, setting the Prescaler to 5, configure the sampling frequency to 488 Hz.

## 11 HDLC

This calculation sheet allows the configuration and calculation of the transmit and the receive clocks.

For the two cases it is necessary to select the clock source (Internal or External), to set the value of the clock source if external clock source was selected, to configure the clock divider value and to select the transmission and the receive data code.

We obtain as a result:

- The transmit and the receive clocks,
- The transmit and the receive registers configuration.

**Figure 11. HDLC calculation sheet**

**HDLC Section** Internal Clock:  MHz

---

Input Clock for Transmit Divider :  MHz

**Transmit Clock :**  kHz

Register: "HDLC\_BRR"  16 bit

Register: "HDLC\_TCTL"  16 bit

---

Input Clock for Receive Divider:  MHz

**Receive Clock:**  kHz

Register: "HDLC\_PRSR"  16 bit

Register: "HDLC\_RCTL"  16 bit

---

(C) COPYRIGHT 2004 STMicroelectronics  
MCD Application Team

**Transmit Clock**

Select Clock Source :

kHz

Provide Transmit Clock Divider :  value

Select Transmission Data Code:

**Receive Clock**

Select Clock Source :

kHz

Provide Receive Clock Divider :  value

Select Receive Data Coding:

**Note:** For more details, refer to the HDLC section of the reference manual.

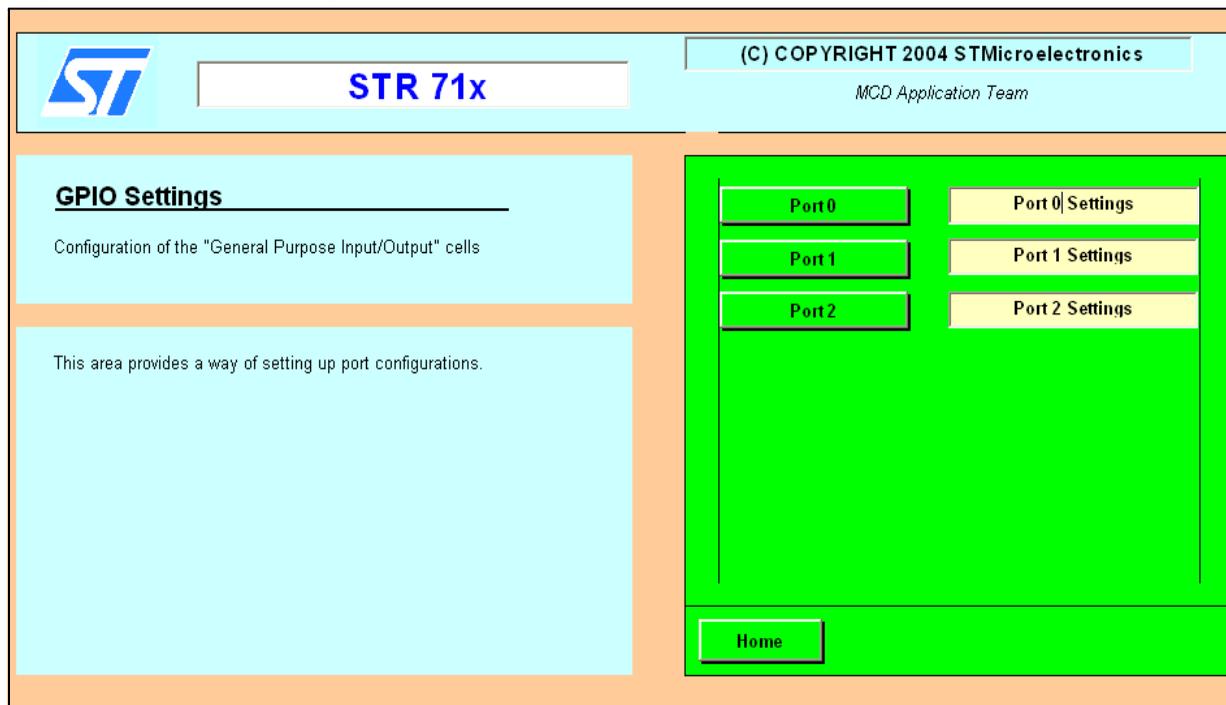
As an example, as shown in [Figure 11](#), the HDLC is configured as follows:

- The transmit clock is set to 1MHz with an internal clock source, a clock divider value set to 9 and with Manchester transmission data code,
- The receive clock is set to 500KHz with an internal clock source, a clock divider value set to 39 and with NRZI transmission data code.

## 12 GPIO

In this sheet, there are three buttons (Port 0, Port 1 and Port 2) as shown in [Figure 12](#).

**Figure 12. GPIO Settings sheet**



## 12.1 PORT 0

This sheet allows the configuration of all the port 0 pins independently. We obtain as a result the values to be loaded into the PC0, PC1 and PC2 registers.

Figure 13 shows how to configure the port 0 as follows:

- Bits 0, 1, 2, 3: BSPIO,
- Bits 4, 5, 6, 7, 12: GPIO,
- Bits 8, 9, 10, 11: UART1,
- Bits 13, 14: UART2,
- Bits 15: Wake-Up.

**Figure 13. Port 0 configuration sheet**

**Port 0: Settings of I/O Pins**

WakeUp	UART-2	UART-2	GPIO	UART-1	UART-1	UART-0	UART-0
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
III-TTL	AFct-OD	III-CMOs	AFct-PP	AFct-PP	III-CMOs	AFct-OD	AFct-OD

GPIO	GPIO	GPIO	GPIO	BSPIO-0	BSPIO-0	BSPIO-0	BSPIO-0
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
III-CMOs	III-CMOs	AFct-OD	III-TTL	AFct-OD	AFct-PP	AFct-PP	III-TTL

**Port 0: Settings of Configuration Registers**

Port 0 Configuration Register 0: 1001100000010111      0x9817h

Port 0 Configuration Register 1: 0111111111101110      0x7FEEh

Port 0 Configuration Register 2: 0101101100101110      0x5B2Eh

**Use BSPI 0 / FC 1 / UART 3:** BSPI-0      Master

**Use BSPI 1:** GPIO

**Use UART 0:** UART-0      Half-Dplx

**Use UART 1 / Smart Card:** UART-1

**Use UART 2 / EFT 2:** UART-2

**Use Wake-Up:** Wake Up

---

**Select I/O Pin Function:**

**Select Pin:** Bit 10

**Select Function:** IN-CMOS      Store

**Alternate Function I/O Configuration is set properly !**

Home
Back
Port 1
Port 2

## 12.2 PORT 1

This sheet allows the configuration of all the port 1 pins independently. We obtain as a result the values to be loaded in to the PC0, PC1 and PC2 registers.

The [Figure 14](#) shows how to configure the port 1 as follows:

- Bits 0, 1, 2, 3: Timer 3,
- Bits 4, 5, 6, 7, 8, 9, 15: GPIO,
- Bits 10: USB,
- Bits 11, 12: CAN,
- Bits 13, 14: I2C.

**Figure 14. Port 1 configuration sheet**

### Port 1: Settings of I/O Pins

GPIO Bit 15	PC-0 Bit 14	PC-0 Bit 13	CAN Bit 12	CAN Bit 11	USB Bit 10	GPIO Bit 9	GPIO Bit 8
AFct-PP	AFct-PP	AFct-PP	AFct-PP	III-CMOs	III-CMOs	OUT-PP	AFct-OD

GPIO Bit 7	GPIO Bit 6	GPIO Bit 5	GPIO Bit 4	Timer-3 Bit 3	Timer-3 Bit 2	Timer-3 Bit 1	Timer-3 Bit 0
AFct-PP	AFct-PP	AFct-PP	III-TTL	III-TTL	AFct-OD	III-CMOs	AFct-PP

### Port 1: Settings of Configuration Registers

Port 1 Configuration Register 0:  
 0xF2F9h

Port 1 Configuration Register 1:  
 0xFDE7h

Port 1 Configuration Register 2:  
 0xF3E5h

Use Timer 3:

Use HDLC / Timer 1:

Use CAN:

Use USB:

---

Select I/O Pin Function:

Select Pin:

Select Function:

Alternate Function I/O Configuration is set properly !

## 12.3 PORT 2

This sheet allows the configuration of all the port 2 pins independently. We obtain as a result the values to be loaded in to the PC0, PC1 and PC2 registers.

The [Figure 15](#) shows how to configure the port 2 as follows:

- Bits 0, 1: External memory interface: Select memory bank,
- Bits 2, 3, 12, 13, 14, 15: GPIO,
- Bits 4, 5, 6, 7: External memory interface: Address bus,
- Bits 8, 9, 10, 11: External interrupt.

**Figure 15. Port 2 Configuration sheet**

**Port 2: Settings of I/O Pins**

GPIO Bit 15	GPIO Bit 14	GPIO Bit 13	GPIO Bit 12	INT 5 Bit 11	INT 4 Bit 10	INT 3 Bit 9	INT 2 Bit 8
III-TTL	III-OUT	AFct-PP	AFct-PP	III-TTL	III-CMOs	III-TTL	III-TTL

---

Adr. 23 Bit 7	Adr. 22 Bit 6	Adr. 21 Bit 5	Adr. 20 Bit 4	GPIO Bit 3	GPIO Bit 2	CSN 1 Bit 1	CSN 0 Bit 0
AFct-PP	AFct-PP	AFct-OD	AFct-OD	AFct-OD	AFct-PP	AFct-OD	AFct-OD

**Port 2: Settings of Configuration Registers**

Port 2 Configuration Register 0: 1111|1011|1100|0100 → 0xFBC4h

Port 2 Configuration Register 1: 0111|0100|1111|1111 → 0x74FFh

Port 2 Configuration Register 2: 0011|0000|1111|1111 → 0x30FFh

**Use ext. Bank Select:** use Bank Select

**Use extended Addresses:** use Adr. Lines

**Use ext. Interrupts:** use Interrupts

INT 2 , enabled	<input checked="" type="checkbox"/> select	set Interrupts ▾
INT 3 , enabled	<input checked="" type="checkbox"/> select	
INT 4 , enabled	<input checked="" type="checkbox"/> select	
INT 5 , enabled	<input checked="" type="checkbox"/> select	

Store

**Select I/O Pin Function:**

Select Pin: Bit 1

Select Function: AFct-OD Store

Alternate Function I/O Configuration is set properly !

Home
Back
Port 0
Port 1

“THE PRESENT NOTE WHICH IS FOR GUIDANCE ONLY AIMS AT PROVIDING CUSTOMERS WITH INFORMATION REGARDING THEIR PRODUCTS IN ORDER FOR THEM TO SAVE TIME. AS A RESULT, STMICROELECTRONICS SHALL NOT BE HELD LIABLE FOR ANY DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT TO ANY CLAIMS ARISING FROM THE CONTENT OF SUCH A NOTE AND/OR THE USE MADE BY CUSTOMERS OF THE INFORMATION CONTAINED HEREIN IN CONNECTION WITH THEIR PRODUCTS.”

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics.

All other names are the property of their respective owners

© 2005 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia – Belgium - Brazil - Canada - China – Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

[www.st.com](http://www.st.com)