AN3988
Application note

Clock configuration tool for STM32F40xx/41xx/427x/437x microcontrollers

Introduction

This application note presents the clock system configuration tool (STSW-STM32091) for the STM32F4xx microcontroller family.

The purpose of this tool is to help the user configure the microcontroller clocks, taking into consideration product parameters such as power supply and Flash access mode.

The configuration tool is implemented in the "STM32F4xx_Clock_Configuration_VX.Y.Z.xls" file which is supplied with this application note and can be downloaded from *www.st.com*.

This tool supports the following functionalities for the STM32F4xx:

- Configuration of the system clock, HCLK source and output frequency
- Configuration of the Flash latency (number of wait states depending on the HCLK frequency).
- Setting of the PCLK1, PCLK2, TIMCLK (timer clocks), USBCLK, and I2SCLK frequencies.
- Generation of a ready-to-use *system_stm32f4xx.c* file with all the above settings (STM32F4xx CMSIS Cortex-M4 Device Peripheral Access Layer System Source File).

The STM32F4xx_Clock_Configuration_VX.Y.Z.xls is referred to as “clock tool” throughout this document.

Before using the clock tool, it is essential to read the STM32F4xx microcontroller reference manual (RM0090). This application note is not a substitute for the reference manual.

Note: For VX.Y.Z, please refer to the tool version, example V1.0.0

Contents

List of tables

List of figures

1 Glossary

Table 1. Definition of terms

Term	Description
HCLK	AHB clock
PCLK1	APB1 clock
PCLK2	APB2 clock
TIMCLK	Timer clock
USB OTG FS	USB on-the-go at full-speed
F_{CPU}	Cortex-M4 clock
Ext.Clock	External clock
V_{DD}	Power supply
HSI	High-speed internal clock
HSE	High-speed external clock
MCLK	Master clock
I2S	Integrated interchip sound
Fs	Sampling frequency
I2SCLK	I2S clock
RNG	Random number generator
SDIO	Secure digital input/output interface

2 Getting started

This section describes the requirements and procedures needed to start using the clock tool.

2.1 Software requirements

To use the clock tool with Windows operating system, a recent version of Windows®, such as Windows XP, Vista or Windows 7 must be installed on the PC with at least 256 Mbytes of RAM.

Before starting to use the clock tool, make sure that Microsoft® Office is installed on your machine and then follow these steps:

- Download the latest version of the **clock tool** for the STM32F4xx product from *www.st.com*.
- Enable macros and ActiveX controls:

Excel® 1997-2003 version

1. Click **Tools** in the menu bar
2. Click **Macro**
3. Click **Security**
4. Click **Low (not recommended)**

Note: If ActiveX controls are not enabled, a warning message is displayed asking you to enable ActiveX. In this case, you should click "OK" to enable it.

Excel 2007 version

1. Click the **Microsoft Office** button and then click **Excel options**.
2. Click **Trust Center**, click **Trust center settings**, and then click **Macro settings**.
3. Click **Enable all macros (not recommended, potentially dangerous code can run)**.
4. Click **Trust Center**, click **Trust center settings**, and then click **ActiveX settings**.
5. Click **Enable all controls without restrictions and without prompting (not recommended; potentiality dangerous controls can run).**
6. Click **OK**.

Note: For more information about how to enable macros and ActiveX controls please refer to the Microsoft Office website.

2.2 Hardware requirements

2.2.1 Introduction

The clock tool is designed to configure the system clocks and generate the *system_stm32f4xx.c* file for STM32F4xx microcontrollers.

The *system_stm32f4xx.c* file is provided as a template system clock configuration file which can be easily modified to select the corresponding system clock frequency and to configure the Flash latency.

2.2.2 Clock scheme for STM32F4xx microcontrollers

This section describes the system clock scheme that is dependent on the voltage requirements (V_{DD}) versus the system clock frequency and Flash latency versus the system clock frequency.

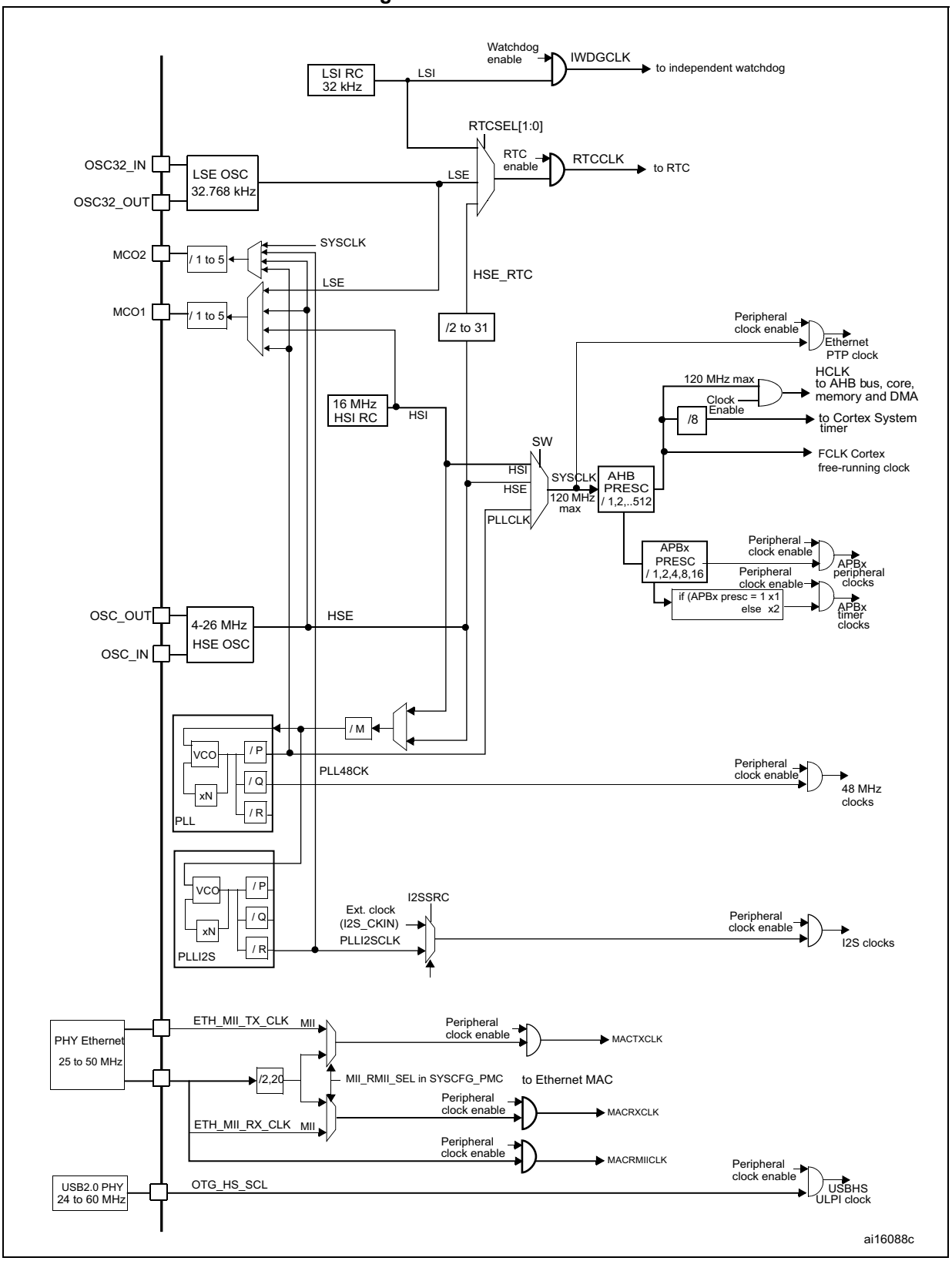
Three different clock sources can be used to drive the system clock (SYSCLK):

1. **HSI** (16 MHz) oscillator clock
2. **HSE** (4 MHz to 26 MHz) oscillator clock
3. **Main phase-locked loop (PLL)** clock with a PLL voltage-controlled oscillator (PLLVCO) input frequency which must be between1 and 2 MHz (2 MHz is recommended to limit the PLL jitter) and with division factors M, N, P, and Q.

All peripheral clocks are derived from the SYSCLK except for:

1. The USB OTG FS clock (48 MHz), the RNG clock (48 MHz), and the SDIO clock (48 MHz) which come from a specific output of PLL (PLL48CLK).
2. The I2S clock. To achieve high-quality audio performance, the I2S clock can be derived either from a specific PLL (PLLI2S) or from an external clock mapped on the I2S_CKIN pin.

Figure 1. Clock scheme



Note: The number of Flash memory wait states (latency) is defined according to the frequency of the CPU (Cortex-M4) and indirectly by the supply voltage of the device (V_{DD}) (see *Table 2: Number of wait states according to CPU clock (HCLK) frequency*).

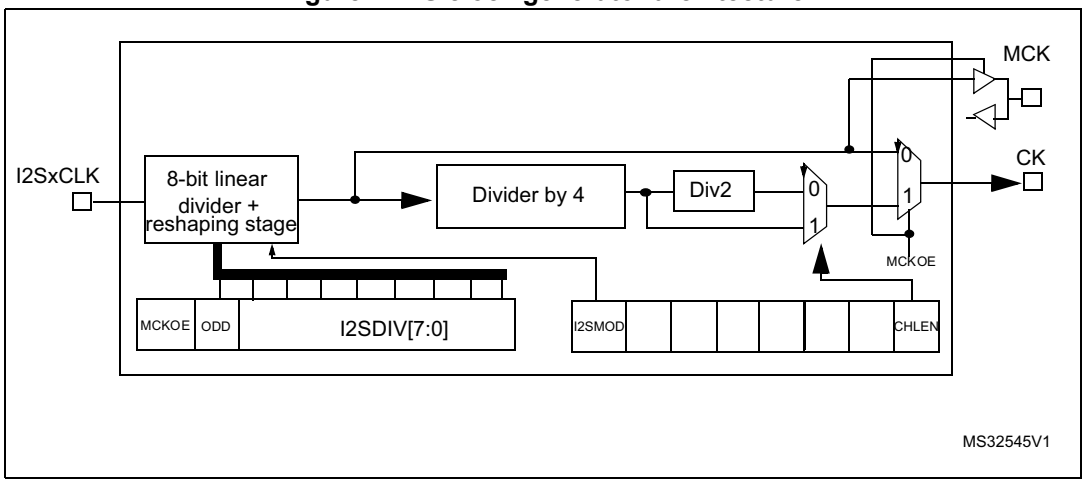
Table 2. Number of wait states according to CPU clock (HCLK) frequency

Wait states (WS) (LATENCY)	HCLK (MHz)			
	Voltage range 2.7 V - 3.6 V	Voltage range 2.4 V - 2.7 V	Voltage range 2.1 V - 2.4 V	Voltage range 1.8 V - 2.1 V Prefetch OFF
0 WS (1 CPU cycle)	0 <HCLK ≤ 30	0 <HCLK ≤ 24	0 <HCLK ≤ 22	0 < HCLK ≤ 20
1 WS (2 CPU cycles)	30 <HCLK ≤ 60	24 < HCLK≤ 48	22 <HCLK ≤ 44	20 <HCLK ≤ 40
2 WS (3 CPU cycles)	60 <HCLK ≤ 90	48 < HCLK≤ 72	44 < HCLK≤ 66	40 < HCLK≤ 60
3 WS (4 CPU cycles)	90 <HCLK ≤ 120	72 < HCLK≤ 96	66 <HCLK ≤ 88	60 < HCLK≤ 80
4 WS (5 CPU cycles)	120 <HCLK ≤ 150	96 < HCLK≤ 120	88 < HCLK≤ 110	80 < HCLK≤ 100
5 WS (6 CPU cycles)	150 <HCLK ≤ 168	120 <HCLK ≤ 144	110 < HCLK≤ 132	100 < HCLK≤ 120
6 WS (7 CPU cycles)		144 <HCLK ≤ 168	132 < HCLK≤ 154	120 < HCLK≤ 140
7 WS (8 CPU cycles)			154 <HCLK ≤ 168	140 < HCLK≤ 160

2.2.3 I2S clock generator

This section describes the I2S clock generator that is dependent on the master clock MCLK (enable or disable), the frame wide, and the I2S peripheral clock (I2SCLK).

Figure 2. I2S clock generator architecture



The audio sampling frequency may be 192 kHz, 96 kHz, 48 kHz, 44.1 kHz, 32 kHz, 22.05 kHz, 16 kHz, 11.025 kHz or 8 kHz. To reach the desired frequency, the linear divider (DIV) needs to be programmed according to the formulas below:

When the master clock is generated (MCKOE in the SPI_I2SPR register is set):

- Fs = I2SxCLK/[(16*2)*((2*I2SDIV)+ODD)*8)] when the channel frame is 16-bit wide
- Fs = I2SxCLK/[(32*2)*((2*I2SDIV)+ODD)*4)] when the channel frame is 32-bit wide

Where ODD is an odd factor for the prescaler.

When the master clock is disabled (MCKOE bit cleared):

- FS = I2SxCLK/[(16*2)*((2*I2SDIV)+ODD))] when the channel frame is 16-bit wide
- FS = I2SxCLK/[(32*2)*((2*I2SDIV)+ODD))] when the channel frame is 32-bit wide

This tool performs the best configuration of the PLLI2S_N and PLLI2S_R with the minimum error on the sampling frequency and according to I2S parameters (frame wide, MCKO, and sampling frequency).

Note: Only the PLLI2S_N and PLLI2S_R are configured in the "system_stm32f4xx.c" file.

This tool does not configure the I2S register.

The sampling frequency error is computed as an indicator according to the I2S parameters which are not configured in the output file "system_stm32f4xx.c".

3 Tutorials

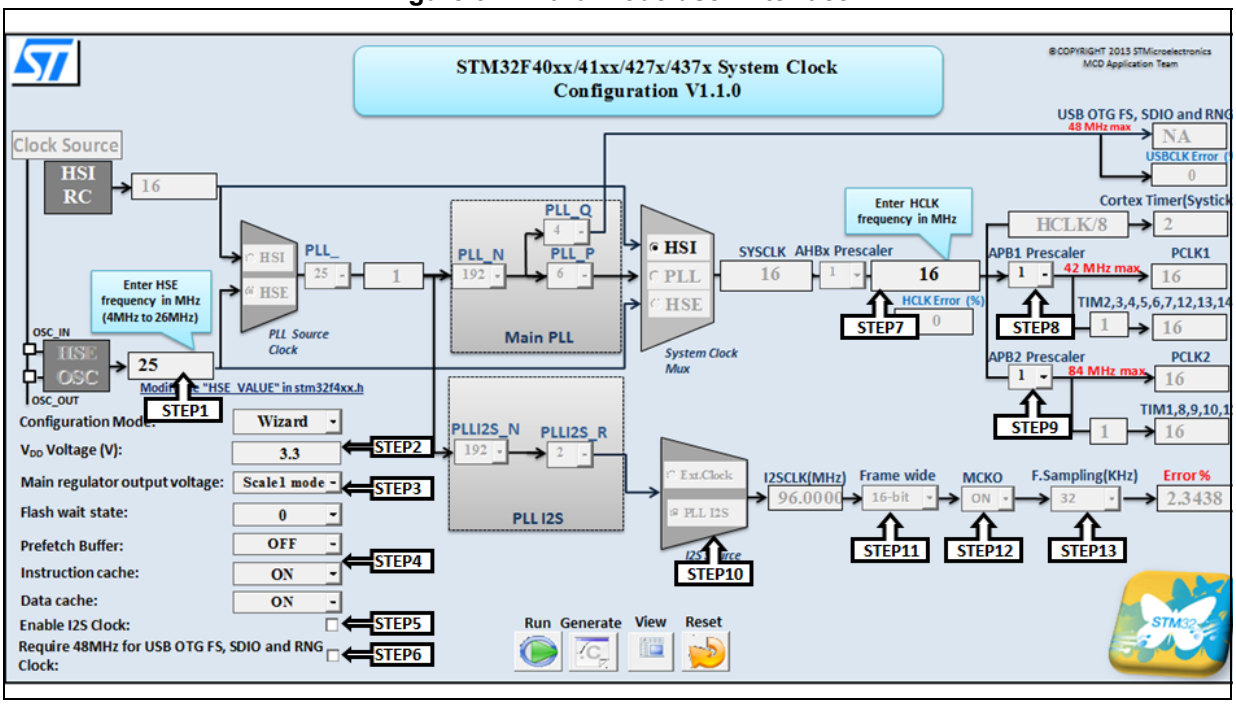
This section describes step by step how to use the clock tool to configure all system clocks and generate the system_stm32f4xx.c file.

Two modes are available: **Wizard** and **Expert**. The selection is made in the **Configuration mode** list box.

3.1 Wizard mode

This mode (default mode) guides you through a series of steps to obtain the desired clock system configuration quickly and easily.

Figure 3. Wizard mode user interface



*Note: The '**View**' button permits viewing of the xls file in full screen to be activated or deactivated.*

*The '**Reset**' button permits the system clock to the default configuration to be set.*

3.1.1 Step-by-step procedure

1. If the HSE is used in your application, set its frequency between a minimum of 4 MHz and a maximum of 26 MHz if a crystal oscillator is used for STM32F4xx

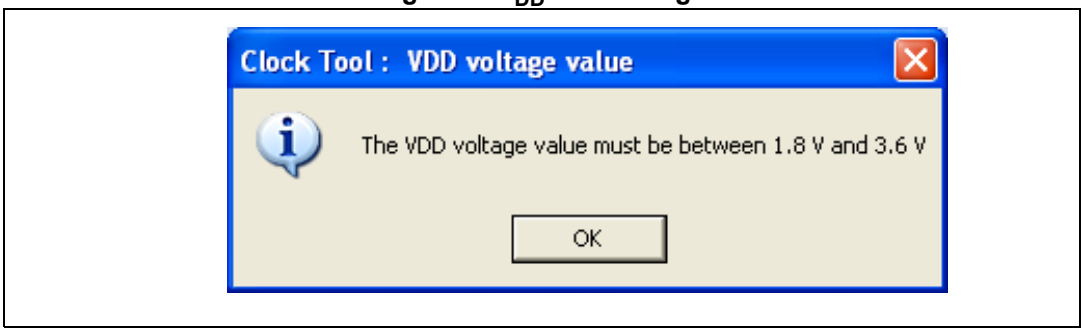
Note: The definition, HSE_VALUE, in the stm32f4xx.h file must be modified each time the user changes the HSE oscillator value.

If the frequency entered is out of range, an error message is displayed, as shown in *Figure 4*. A valid frequency must be entered.

Figure 4. HSE value out of range

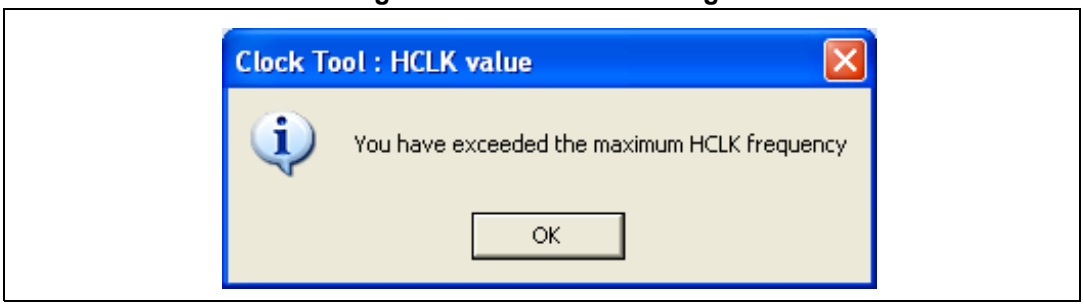
2. Enter the V_{DD} power supply voltage range which is between 1.8 V and 3.6 V (refer to *Figure 3: Wizard mode user interface*). If the V_{DD} voltage is out of range, an error message is displayed as shown in *Figure 5*.

Figure 5. V_{DD} out of range



3. Configure the main regulator output voltage:
   - Select Scale1 mode from the list box to obtain a maximum system clock frequency (f_{HCLK}) of 168 MHz.
   - Select Scale2 mode, to obtain a maximum system clock frequency (f_{HCLK}) 144 MHz.
4. Configure the **Prefetch buffer**, **Instruction Cash** and the **Data cash** (select ON or OFF from the list box).
5. Specify if the PLLI2S is needed. If it is needed, enable it and follow step 9, 10, 11 and 12. Otherwise, go to step 5.
6. Specify if a 48 MHz clock is needed for USB OTG FS, RNG or SDIO operations. If it is needed, this adds a constraint to the parameter setting in PLL configuration. If it is not needed, no USB constraint has been added.
7. Set the desired HCLK frequency. The maximum HCLK frequency depends both on the main regulator voltage output Scale1/Scale2 mode (see step 4.) and on the V_{DD} voltage (see *Table 2*). If the value entered is higher than the maximum HCLK frequency, an error message is displayed as shown in *Figure 6*.

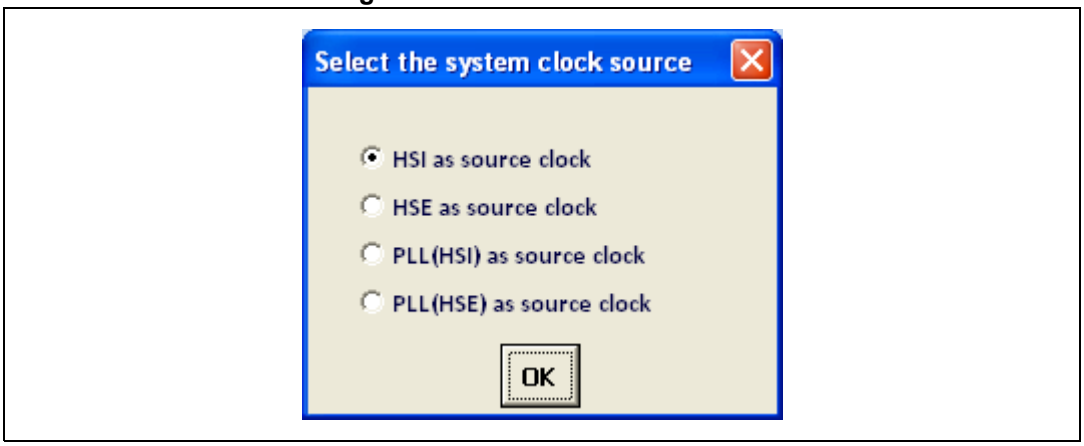
Figure 6. HCLK error message



8. Select the PCLK1 and PCLK2 prescaler settings from the list box to obtain the desired PCLK1 and PCLK2 frequencies. The TIMCLK frequencies are configured automatically depending on the PCLK1 and PCLK2 prescaler settings.
9. See step 8.
10. Select the I2S clock source from the I2S source. Ignore steps 10, 11, and 12 if the external clock is selected as the clock source for the I2S peripheral.
11. If the PLLI2S is selected as the I2S clock source, select the frame wide (16 or 32 bits).
12. Specify if the master clock is enabled or disabled (Select ON/OFF from the list box).
13. Select the Fs from the list box. The Fs value can be 192 kHz, 96 kHz, 48 kHz, 44.1 kHz, 32 kHz, 22.05 kHz, 16 kHz, 11.025 kHz, and 8 kHz.
14. Click the **RUN** button.

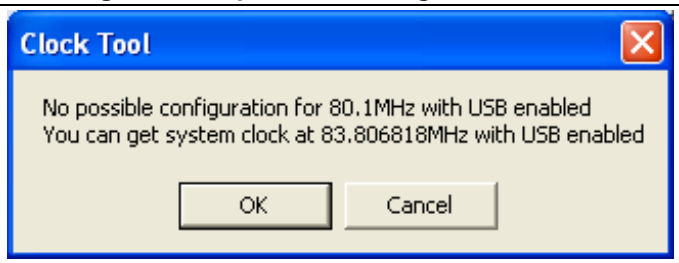
If more than one clock source is possible, a message box displays the clock sources that can be selected. (see *Figure 7*). Choose HSE, HSI or PLL (which are sourced by the HSI or HSE).

Figure 7. Select the clock source



1. When the USB and/or the I2S are enabled (checkbox selected in the clock tool) and the selected HCLK frequency is not possible, a message box displays the nearest HCLK clock frequency to use.

Figure 8. No possible configuration error



Configure the Flash Latency: after running the application, the number of wait states is configured automatically with the best value (lowest possible value) which can be modified by another number higher then the selected one.

Clicking the **RUN** button displays a progress bar.

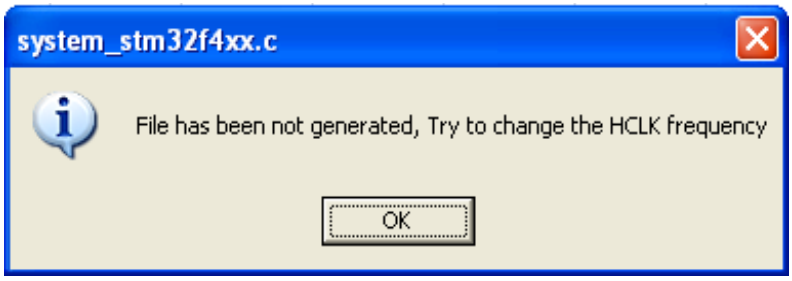
15. Finally, click the **Generate** button to automatically generate the *system_stm32f4xx.c* file.

The *system_stm32f4xx.c* is generated in the same location as the clock tool. Display the file to verify the value of the system clock, *SystemCoreClock,* and the values of HCLK, PCLK1, PCLK2, Flash access mode, and other parameters which are defined in the *SetSysClock* function.

The *system_stm32f4xx.c* file must be added to the working project to be built.

If the file is not generated, an error message is displayed as shown *Figure 9*.

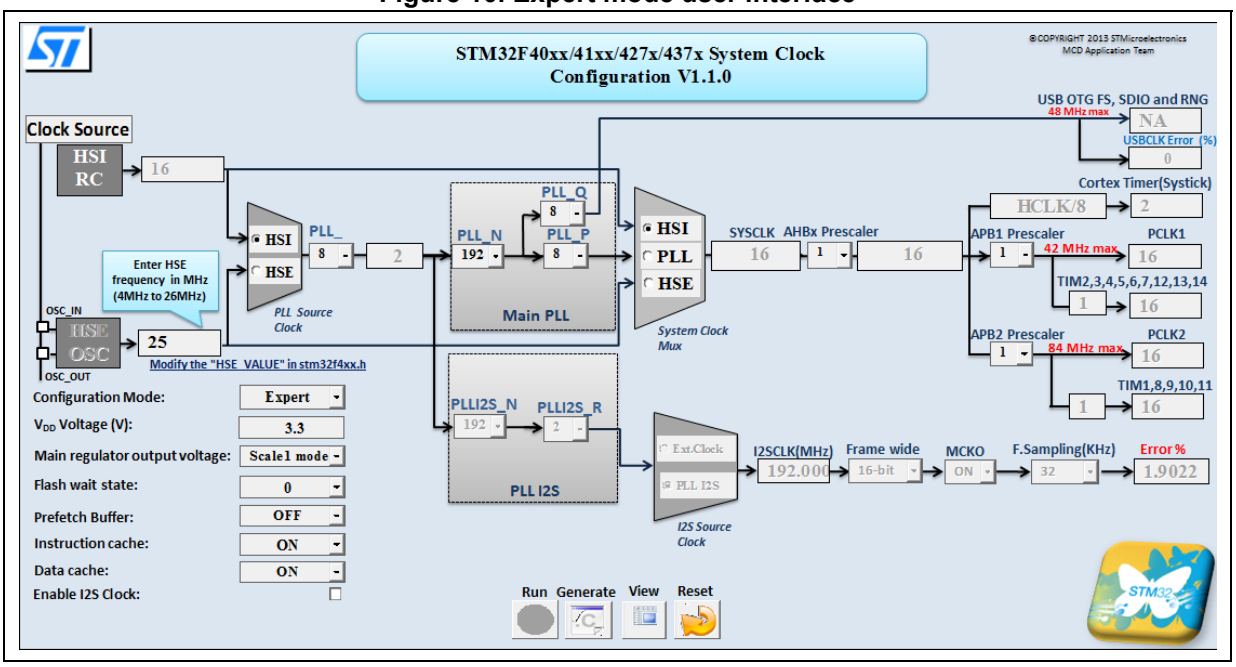
Figure 9. File generation error



3.2 Expert mode

This mode provides more flexibility regarding the configuration setup but, it is up to the user to ensure that configuration is correct.

Figure 10. Expert mode user interface

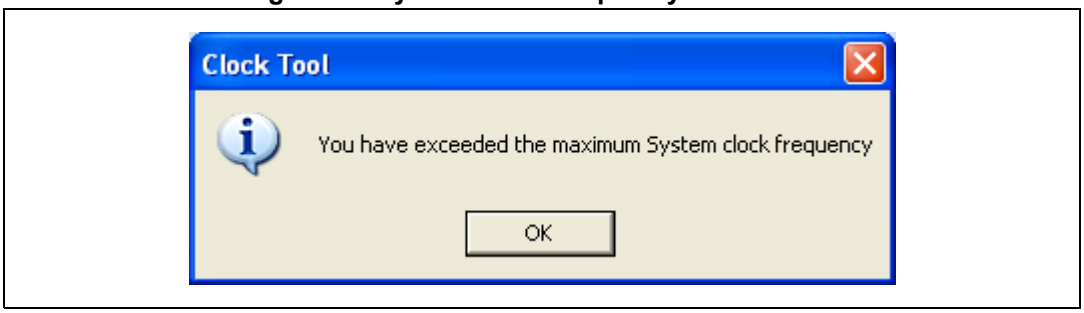


The '**View**' button permits viewing of the xls file in full screen to be activated or deactivated.

The '**Reset**' button permits the system clock to the default configuration to be set.

1. Configure the SYSCLK frequency as follows:
   a) If the HSE is used in your application, set its frequency between:
   – a minimum of 4 MHz
   and
   – a maximum of 26 MHz if a crystal oscillator is used for STM32F4xx.
   Note: The definition, HSE_VALUE, in the stm32f4xx.h file must be modified each time the user changes the HSE oscillator value.
   If the frequency entered is out of range, an error message is displayed, as shown in *Figure 4*. A valid frequency must be entered.
   b) Enter the V_{DD} power supply voltage range which is between 1.8 V and 3.6 V (refer to *Figure 10: Expert mode user interface*).
   c) Configure the main regulator output voltage:
   – Select Scale1 mode from the list box to obtain a maximum system clock frequency (f_{HCLK}) of 168 MHz.
   – Select Scale2 mode, to obtain a maximum system clock frequency (f_{HCLK}) 144 MHz.
   d) Configure the SYSCLK source (PLL, HSE or HSI). If the clock source selection is invalid (the HCLK frequency is too high) an error message is displayed as shown in *Figure 11*.

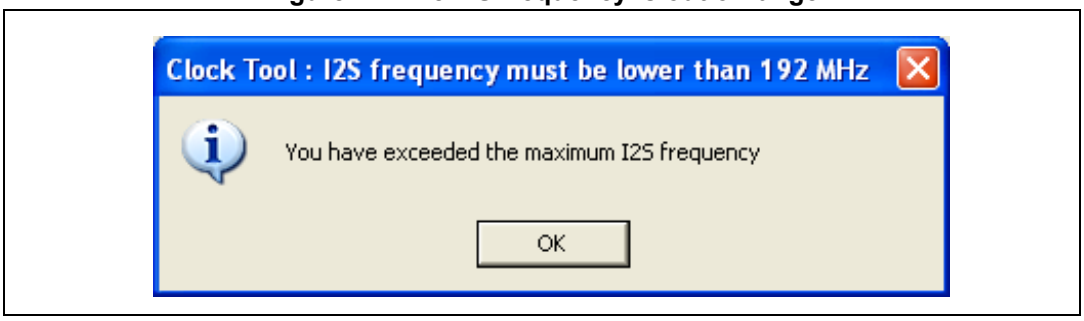
Figure 11. System clock frequency is exceeded



e) If PLL is selected as the SYSCLK source, it is necessary to select the source clock for the PLL (HSE or HSI).

f) If the PLL is selected as the SYSCLK source, configure the main PLL(M) division factor to achieve a PLLVCO frequency between 0.95 MHz and 2.1 MHz (for more details, please refer to “PLL characteristics” in STM32F4xx datasheet). If the selected division factor is invalid, an error message is displayed. If the I2S frequency is higher than 192 MHz an error message is displayed as shown in *Figure 12*.

Figure 12. The I2S frequency is out of range



g) Set the HCLK prescaler using the **AHBPrescaler** list box to obtain the desired HCLK frequency.

h) Select the PCLK1 prescaler settings from the list box to obtain the desired PCLK1 frequency. The TIMCLK frequencies are configured automatically depending on the PCLK1 prescaler settings.

i) Select the PCLK2 prescaler settings from the list box to obtain the desired PCLK2 frequency. The TIMCLK frequencies are configured automatically depending on the PCLK2 prescaler settings.

j) Configure the Flash Latency: after setting the HCLK prescaler, the number of Flash wait states is configured automatically with the best value (*lowest* possible value) *which can be modified by an*other number higher then the selected one.

k) Generate the clock configuration files by clicking on the **Generate** button.

2. Configure the I2S clock frequency as follows:
   a) If the PLLI2S is needed, enable it and fellow the steps below.
   b) If the external clock source is selected as the I2S clock source, the following steps can be ignored.
   c) If the PLLI2S is selected as the clock source for the I2S peripheral, configure the PLLI2S(N) multiplication factor.
   d) If the PLLI2S is selected as the clock source for the I2S peripheral, configure the PLLI2S(R) division factor.
   e) If the PLLI2S is selected as the I2S clock source, select the frame wide (16 or 32 bits) and specify if the master clock is enabled or disabled.
   f) Select the Fs from the list box. The Fs value can be 192 kHz, 96 kHz, 48 kHz, 44.1 kHz, 32 kHz, 22.05 kHz, 16 kHz, 11.025 kHz and 8 kHz.
3. Configure the USB OTG FS, RNG or SDIO clock

   Configure the PLL(Q) division factor for USB OTG FS, SDIO and the RNG.
4. Optional configuration

   Configure the **Prefetch buffer**, **Instruction Cash** and the **Data cash**.
5. Generate the *system_stm32f4xx.c* file

   Click the **Generate** button to automatically generate the *system_stm32f4xx.c* file.

   The *system_stm32f4xx.c* file is generated in the same location as the clock tool. It can be displayed to verify the value of the SYSCLK “SystemCoreClock” and the values of HCLK, PCLK1, PCLK2, Flash access mode, and other parameters which are defined in the “SetSysClock” function.

   The system_stm32f4xx.c file must be added to the working project to be built.

4 Known limitations

This sections describes the known limitations of the clock configuration tool.

This tool does not support configurations that use the HSE external clock source (HSE bypass).

5 Conclusion

This application note provides a description of how to use the clock tool with respect to STM32F4xx microcontroller devices. Using either one of the two configuration modes, this tool generates a source code file *system_stm32f4xx.c* to configure the clock system of the STM32F4xx.

Two modes are available:

- Wizard mode is the first mode and provides a quick and easy way to configure the system clocks.
- Expert mode is the second mode. It offers more flexibility in setting up the system clock configuration while still respecting all the product constraints.

6 Revision history

Table 3. Document revision history

Date	Revision	Changes
10-Oct-2011	1	Initial release
05-Aug-2013	2	Updated: – Document's title:replaced "STM32F40x/41x" by "STM32F40xx/41xx/427x/437x". – *Section : Introduction*. – *Table 2: Number of wait states according to CPU clock (HCLK) frequency* – *Section 3.1: Wizard mode*. – *Section 3.2: Expert mode*.