AN3252
Application note

Building a wave generator using STM8L-DISCOVERY

Application overview

This application note provides a short description of how to use the STM8L-DISCOVERY as a basic wave generator for various waveforms such as sine, sinc (sin(x)/x), square, ramp and sawtooth signals.

More generally, this document describes STM8L on-chip DAC (digital-to-analog converter) peripheral settings when used with the DMA (direct memory access) controller.

This application does not require any additional hardware. Once the STM8L-DISCOVERY is powered up through a USB cable connected to the host PC, the application starts: a waveform is generated and a message is displayed on the LCD.

Reference documents

- STM8L-DISCOVERY evaluation board user manual (UM0970)
- STM8L-DISCOVERY software user manual *Developing and debugging your STML-DISCOVERY application code* (UM0991)
- Application note *Using the STM8L15x/STM8L162 DMA controller (AN3117)*

The above documents are available at http://www.st.com.

Contents

List of tables

List of figures

1 Application description

1.1 Hardware required

This application uses the STM8L-DISCOVERY on-board LEDs (green LD3 and blue LD4), 6-digit/4-bar LCD glass display and USER push-button.

1.2 STM8L-DISCOVERY hardware settings

The IDD jumper JP1 must be placed in the OFF position.

The solder bridge SB11 must be removed.

Solder bridge SB11 is located at the bottom of the board, close to the LCD device footprint. This operation disconnects PF0/DAC_OUT from the existing IDD measurement circuitry and sets DAC peripheral output for best performance.

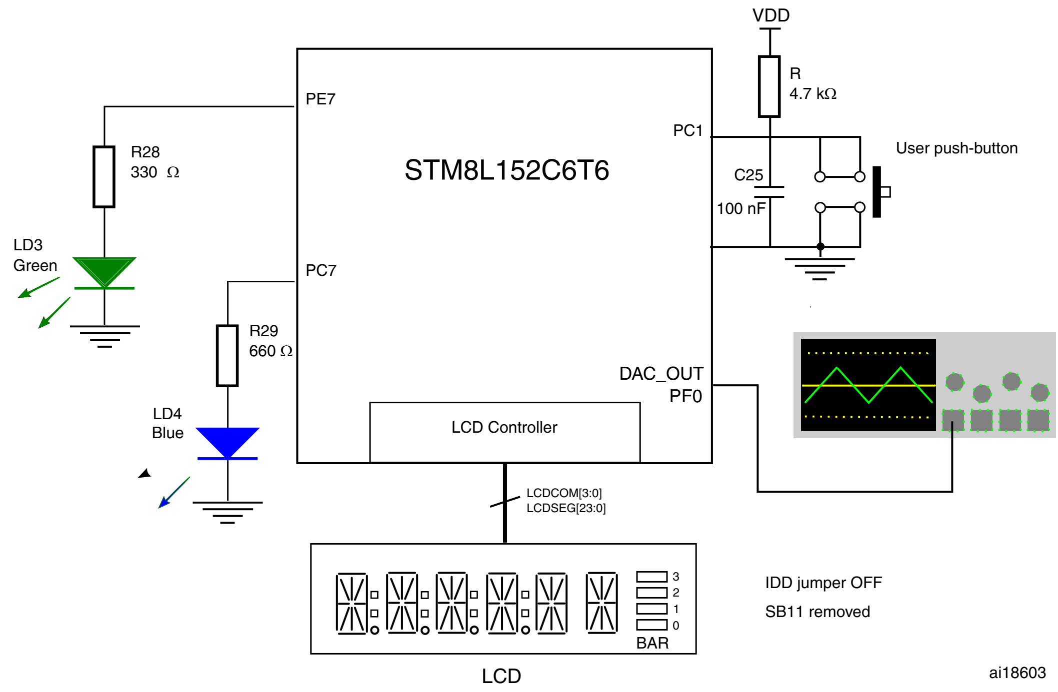
1.3 Application schematics

Figure 1 shows the application electrical schematics.

The generated signal is output on the microcontroller PF0/DAC_OUT pin. It is also available on a pin-head connector for external connection.

As shown, an oscilloscope or a frequency meter can be connected to that pin to monitor the various waveforms and to measure the signal frequency.

Figure 1. Application block diagram



1.4 Application principle

1.4.1 Overview

This example uses the DAC peripheral to generate various waveforms on DAC output.

As the waveforms generated by the wave generator are periodic signals, only one period of each signal needs to be stored in the device Flash memory.

Each signal period is recorded in a table of 65 and 130 samples stored in the microcontroller Flash memory, depending on the required resolution. Each sample corresponds to a digital value that the DAC converts to output voltage by linear conversion.

These samples are sent cyclically from the Flash memory to the DAC using a DMA transaction and synchronized by a hardware trigger event provided by TIM4 timer. The DMA is set to Circular mode so that the operation is repeated indefinitely and that the produced signal is transmitted continuously. Refer to the application note *Using the STM8L15x DMA controller (AN3117)* for more details.

The wave generator is intended to approximately cover the audio range, i.e. 20 Hz to 20 kHz. This example uses various resolution tables (65 and 130 samples) and different TIM4 prescaler values to illustrate DAC output frequencies from 30 Hz to 22.5 kHz.

1.4.2 Waveform generation and frequency ranges

The DAC output value is computed from all the digital samples that make up a waveform period using the following formula:

DAC output = V_{REF} x (DOR / 4096), where V_{REF} = 3 V and DOR is the 12-bit digital sample value.

For example, to generate one period of a sine wave, the table should contain x samples calculated using the sin(x) function and calibrated by applying an offset to shift the signal generation into the positive area. For a ramp signal, the table is calculated by using the 'f(x)=x' function, where x varies from 0 to 4095 (12 bits) and by selecting only n samples per period.

The wave generator can deliver 5 different waveforms in 2 resolution groups, as shown in *Table 1* below.

Depending on the chosen output frequency, select the resolution which best meets your needs.

Table 1. Waveform types

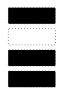
Wave_type #	Waveform type
0	Ramp
1	Square
2	SineC (sin(x)/x)
3	Sine
4	Sawtooth

Converting a 130-sample table takes twice as long as converting a 65-sample table. This means that waveforms sampled with 65 values have twice the frequency of waveforms sampled with 130 values (see ranges 1 and 3 in *Table 2*). Furthermore, TIM4 prescaler values can be modified to create 2 (or more) additional ranges (ranges 2 and 4).

A dedicated 4-bar display shows the current frequency range and resolution.

The corresponding bars display on the right side of the LCD glass (see *Figure 1* for more details).

Table 2. Frequency ranges and bar display

Range #	Start frequency	Stop frequency	Sample size	Bars	Waveform type
1	30 Hz	700 Hz	130		All
2	480 Hz	11 kHz	130		All
3	60 Hz	1.4 kHz	65		All
4	960 Hz	22.5 kHz	65		All

This table shows 4 frequency ranges with various start/stop limits. There are 2 main groups:

- Ranges 1 and 2 use a 130-sample sized table and provide the best resolution signals.
- Ranges 3 and 4 use a 65-sample sized table and provide higher frequency values.

BAR0 is located at the lowest position of the 4-bar display.

BAR3 is used to highlight the table resolution (BAR3 OFF = 130 s, BAR3 ON = 65 s), whereas BAR0 and BAR1 are used to display LOW and HIGH ranges for a table with the same number of samples. BAR2 remains unused.

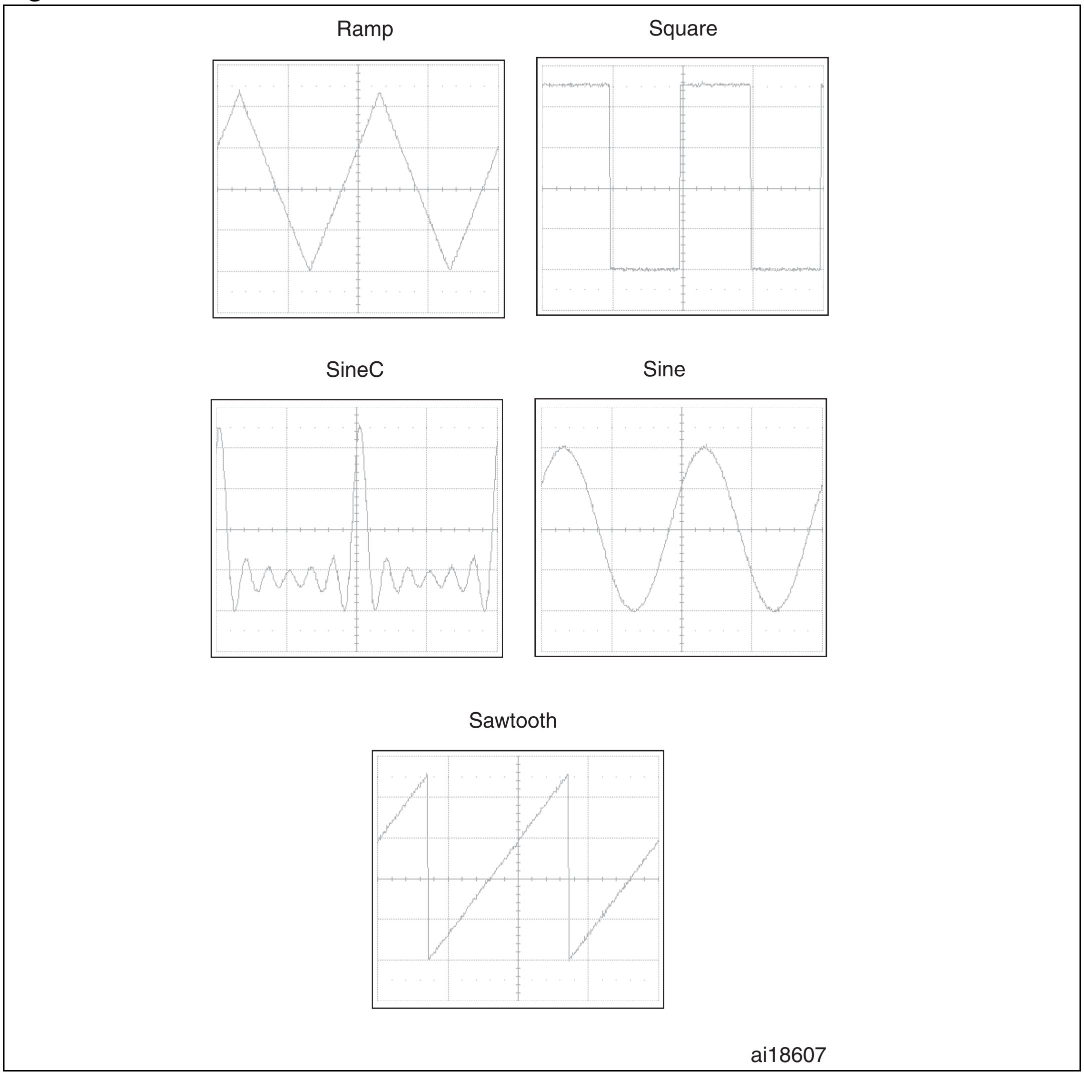
Ranges are automatically switched when the stop limit is reached and the USER push-button is pressed continuously.

Example1: To obtain a signal with a frequency adjusted around 200 Hz, choose Range 1 to obtain the best resolution. The sample size per period is then 130 and the bar display shows BAR0 ON, and BAR1 and BAR3 OFF.

Example2: To obtain a signal with a frequency adjusted around 15 kHz, choose Range#4 for higher frequencies. The sample size per period is then 65 and the bar display shows BAR0, BAR1 and BAR3 ON.

Figure 2 shows the various waveforms the application provides and demonstrates the capability of the DAC. The waveforms are shown in their actual order of appearance, from left to right: Ramp, Square, SineC, Sine, Sawtooth.

Figure 2. Waveforms



1.4.3 Known limitations

The output frequency of a waveform is determined both by the frequency of the TIM4 trigger output and by the update rate of the DAC peripheral. For instance, an output signal with a frequency of 10 kHz and built with a 100-sample resolution requires an update rate of 1 Msps (1 Msps = 1 Mega sample per second).

The following formula is used to calculate the theoretical update rate of the DAC peripheral:

In our example, f_{MAX} = 22.5 kHz, so $t_{MAX} = 1/f_{MAX}$ = 44.4 µs.

If a sample table contains 65 samples per signal period, then the update rate for maximum frequency is approximately:

$65/(44.4 \times 10^{-6}) = 1.46$ Msps.

The resulting value is higher than the update rate allowed to comply with the specifications described in DAC characteristics limited to 1 Msps (refer to datasheet *STM8L152x*).

Therefore, the above method can introduce linearity errors in the signal reproduction for frequencies higher than 15.5 kHz. Hence, designers must take care to find a solution taking into account the expected output frequency and the amount of required samples, and to evaluate the link with the DAC update rate in order to run the DAC in the optimal operating way. So, for your own application using the DAC for signal generation, take care to correlate the required samples per period with the DAC update rate to run the DAC in the optimal operating way.

1.5 Using the wave generator

The application starts generating waveforms automatically as soon as it is powered up via the USB cable (or an external power supply). The first waveform is a ramp signal (130 samples) with a default frequency of 100 Hz.

The USER push-button is the only interface available to interact with the application.

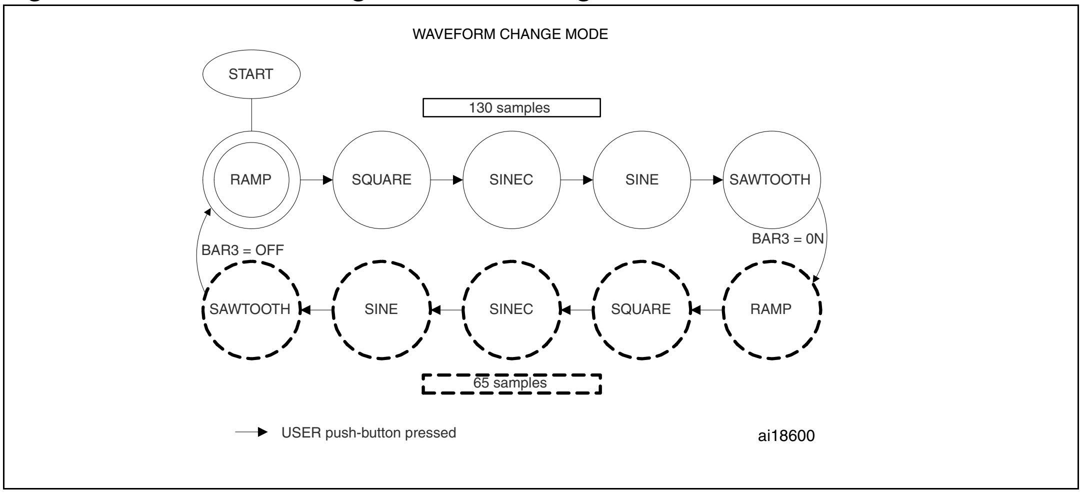
The USER push-button has 2 modes:

- Waveform change
- Frequency change

Waveform change mode

Press the push-button briefly to select waveform types from the 10 tables available. The first five waveforms are made of 130 samples per period (BAR3 OFF), while the five following ones contain 65 samples per period (BAR3 ON). Each time a new waveform is selected, a message is displayed on the LCD to provide information about the current waveform type. LED3 (green) is briefly switched ON when a waveform change occurs.

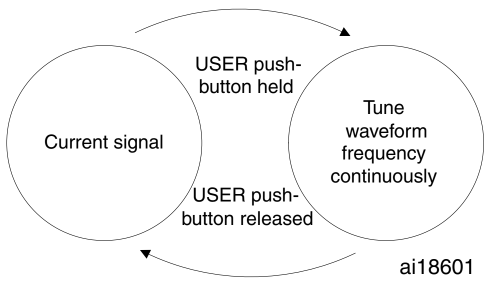
Figure 3. Waveform change mode state diagram



Frequency change mode

Hold down the push-button for more than 250 ms to adjust the signal frequency incrementally over the full frequency range. In this mode, the message “TUNING” is displayed on the LCD and LED4 (blue) switches ON. Keep the push-button pressed to increase the signal frequency, which can be easily observed using an oscilloscope. This allows you to adjust the frequency accurately or to cover a wider range of frequencies. The frequency can be changed during any of the 10 states shown in *Figure 3*.

Figure 4. Frequency change mode state diagram



2 Software description

2.1 STM8L peripherals used by the application

This application example uses the following STM8L peripherals configured as described below:

Direct Memory Access controller

The DMA controller is used to provide high-speed transfer of data from the memory to the DAC peripheral without using CPU resources. The DMA is configured as follows:

- DMA Channel 3 selected
- Transfer from memory to peripheral (destination address range from 0x4000 to 0x5FFF)
- Circular mode (auto-reload mode)
- Memory Increment mode
- Channel priority level very high (DMA takes priority over the CPU)
- Transfer size set to 16-bit mode
- Amount of data to transfer, either 65 or 130, as provided by *sample_size* value (depending on resolution/frequency selected)

DAC

The DAC performs digital-to-analog conversions from data available in memory and transferred via the DMA controller. The DAC output is directly connected to pin PF0 (STM8L152C6T6 48-pin package). The DMA must be configured as follows:

- DAC output buffer disabled (minimizes offset errors)
- DAC conversion sequence is performed when a trigger occurs (TIM4_TRGO)
- DAC DMA enabled

TIM4

TIM4 is used to generate a DMA request on a counter update event (counter overflow). TIM4 must be configured as follows:

- TIM4 prescaler used either with prescaler value DIV 1 or DIV 16 (depending on the frequency range selected)
- TIM4->ARR (auto-reload value) determines intrinsically the frequency value of the DAC output signal

GPIOs

Port C and Port E are interfaced with USER push-button and LEDs.

- PC1 set as input floating pin with interrupt (USER push-button)
- PC7 and PE7 set as output push-pull pin (LED4 and LED3 respectively)
- An action on PC1 generates an interrupt with interrupt sensitivity set to falling edge only

LCD controller

The LCD controller is initialized by the `LCD_Glass_Init()` function, which sets the LCD contrast to a mid-range value (LCD_Contrast_3V0). In some cases, this initial value must be adjusted to adapt the contrast to the LCD display panel used or to the application requirements.

Clock

The clock source selected is HSI (16 MHz high speed internal oscillator). The SYSCLK prescaler is set with prescaler /1 so that CPU frequency is 16 MHz.

2.2 Configuring the STM8L standard firmware library

The *stm8*l15x_conf.h file of the STM8L standard firmware library is used to configure the library by enabling the peripheral functions used by the application.

The following statements must be present:

#include stm8l15x.h

#include stm8l15x_clk.h enables clock configuration used in LCD functions.

#include stm8l15x_lcd.h enables LCD controller.

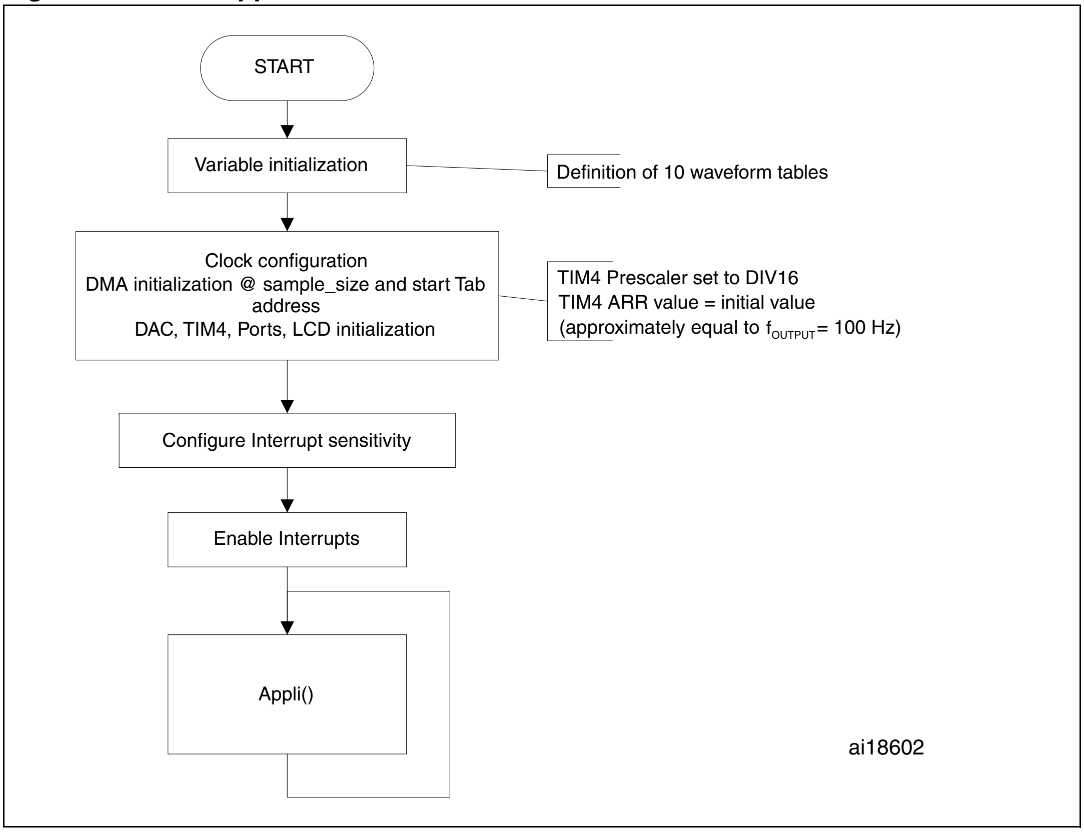
This example uses LCD driver from the library while the main part of the code using DMA, DAC and TIM4 has been optimized in terms of code size and execution time.

2.3 Application software flowcharts

2.3.1 Main application flowchart

The main application initializes all peripherals (such as DMA, DAC, TIM4, Ports and LCD) using the above-defined settings (see *Section 2.1*). After this initialization step, interrupts are enabled and the `Appli()` loop refreshes the display according to the selected menu option.

Figure 5. Main application flowchart



2.3.2 Appli() loop routine description

The `Appli()` loop first displays the application welcome message “STM8L DISCOVERY WAVE GENERATOR”.

When the USER push-button is pressed, the display highlights the current operation (new waveform or tuning frequency operation).

When a new waveform is selected, its name is displayed 10 times, one after the other, then the welcome message is back.

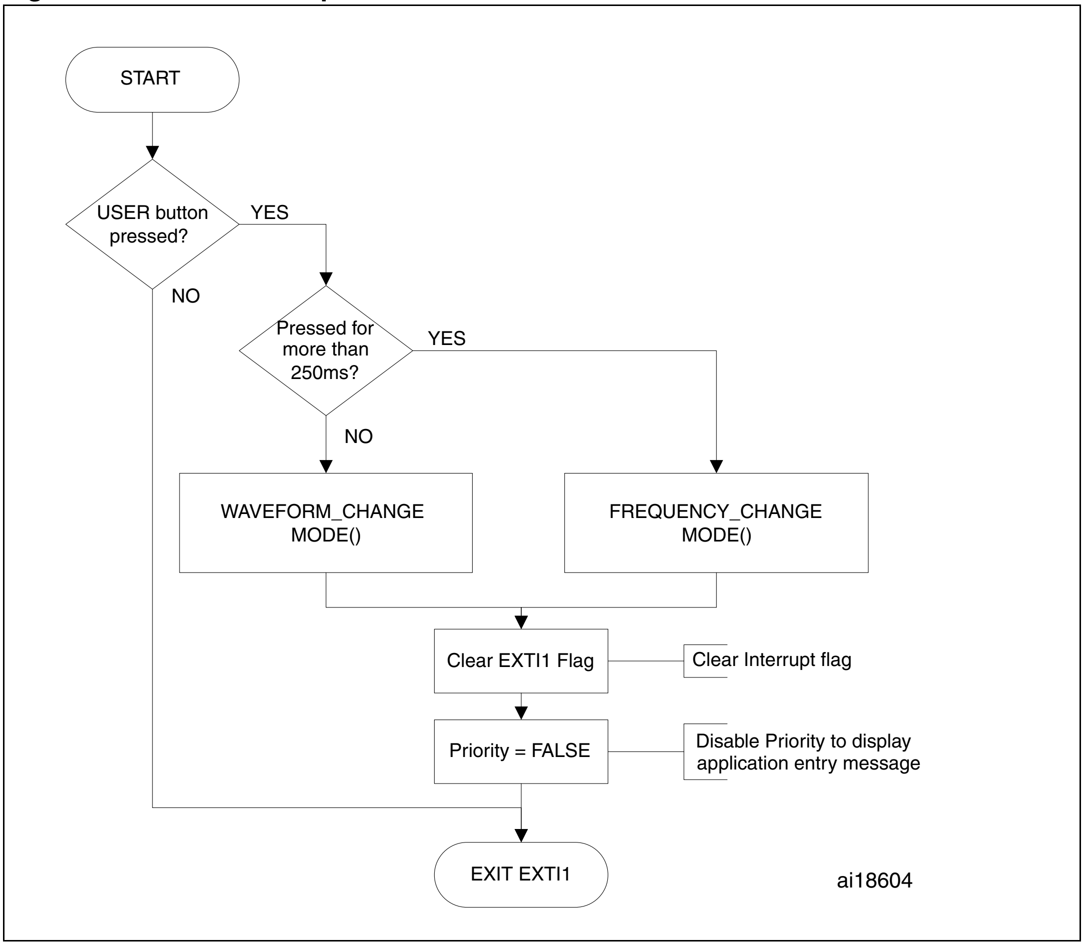
2.3.3 Interrupt subroutine EXTI1 flowchart

Pressing the USER push-button connected to the STM8L generates an interrupt triggered by a falling edge on pin PC1.

If the USER push-button is pressed briefly or is kept pressed for more than 250 ms, the interrupt sends the request to the corresponding routine - either `Waveform_Change Mode()` or `Frequency_Change Mode()`.

In any case, the EXTI1 interrupt flag is cleared and the priority is changed to FALSE so that dedicated messages related to user actions can be displayed.

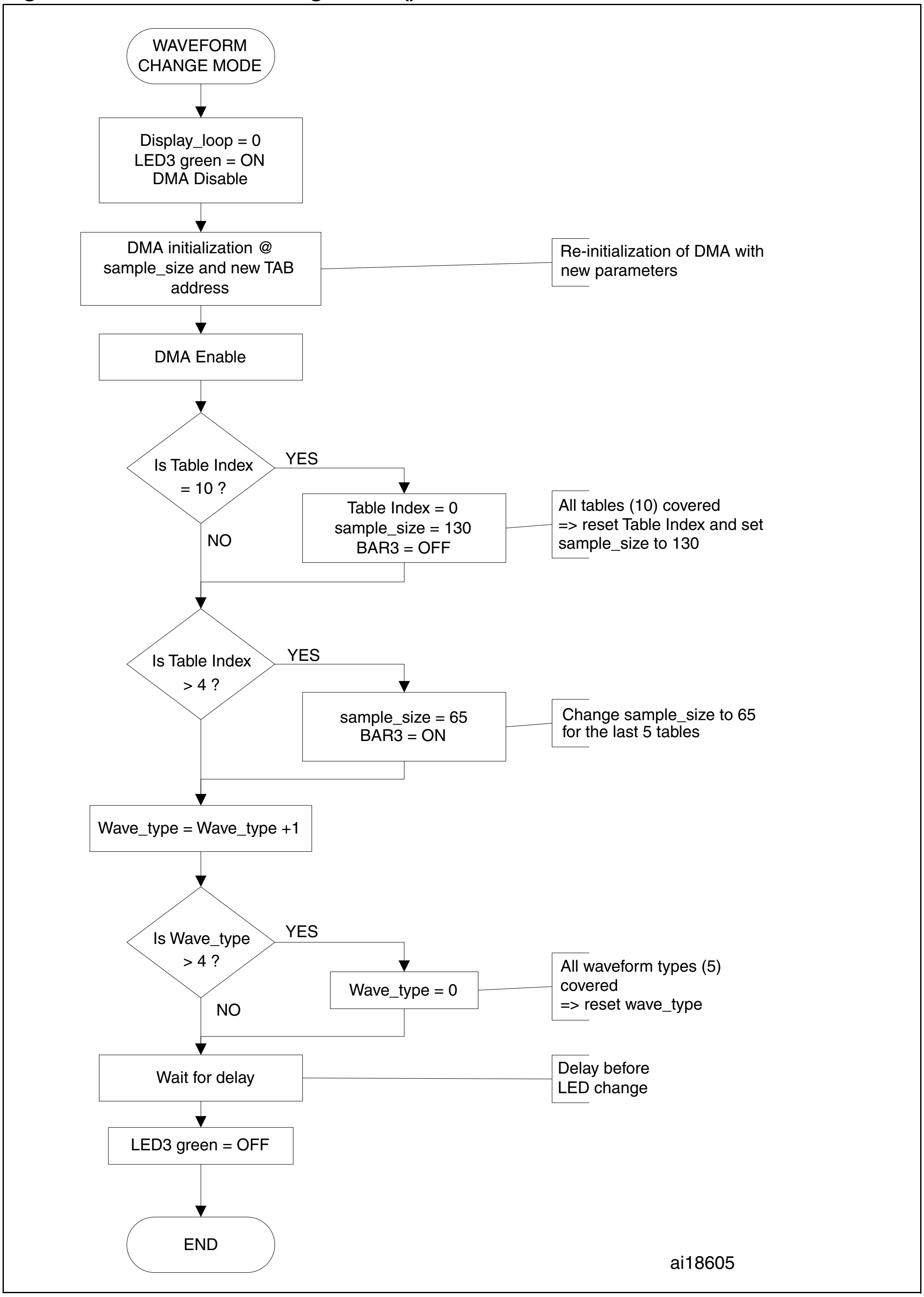
Figure 6. EXTI1 interrupt subroutine



2.3.4 Waveform_Change Mode() flowchart

When waveform change mode is detected (briefly press the USER push-button), the DMA is momentarily disabled until new parameters such as sample size and the table address (TAB) are confirmed or modified. Then the DMA is enabled again with a new set of parameters and a new signal is available on the DAC output. Once the first five tables containing 130 samples have been processed, the *sample_size* value is set to 65 for the last five tables.

Figure 7. Waveform_Change Mode() flowchart

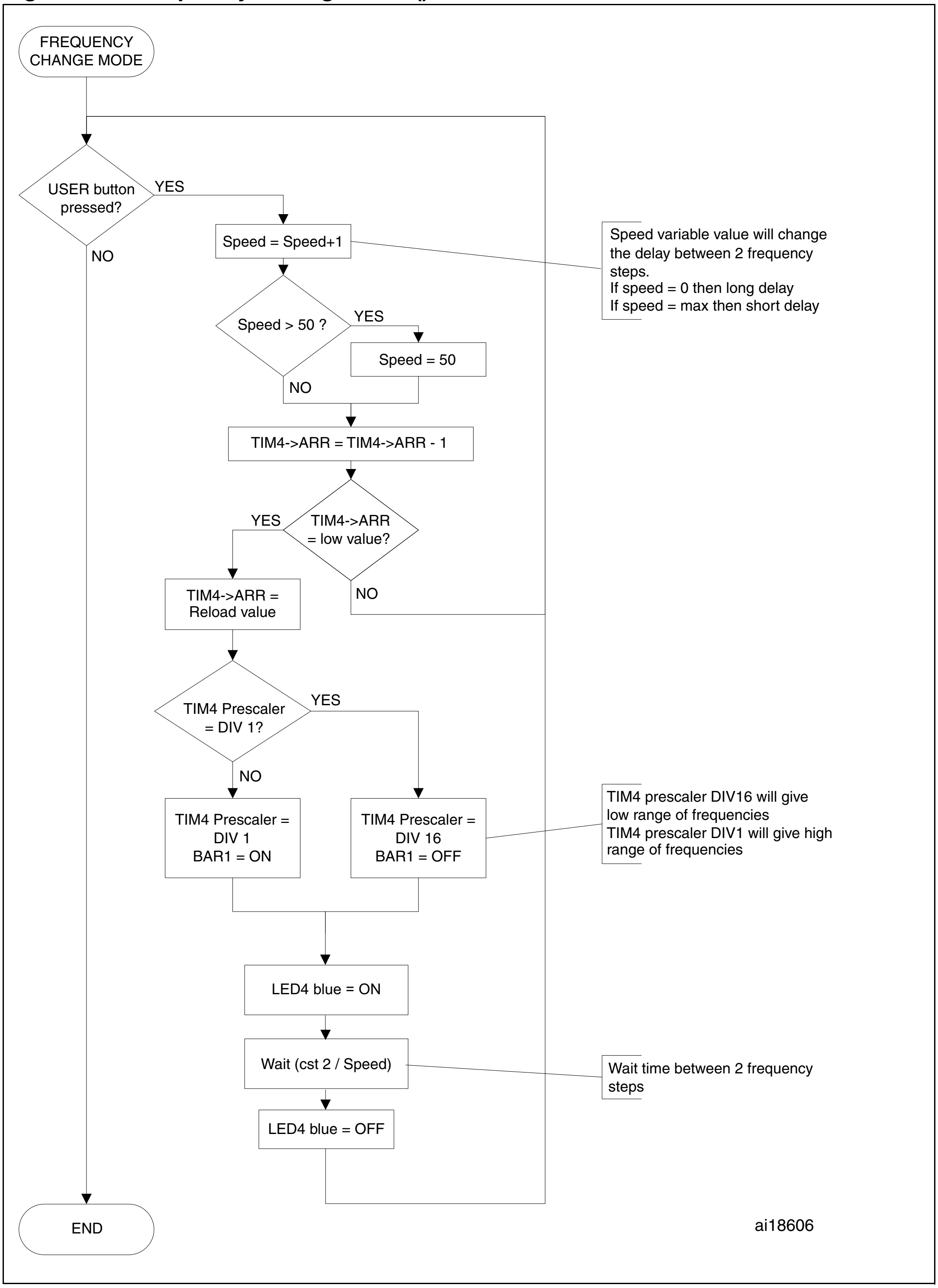


2.3.5 Frequency_Change Mode() flowchart

When the frequency change mode is detected (press the USER push-button for more than 250 ms), the output waveform frequency on the DAC output is gradually increased. If the end of a frequency range is reached, then the next range is accessed automatically, as shown by the BAR0 and BAR1 displays. Once the maximum frequency is reached, the application returns to a lower range (see *Table 2*).

The longer the key is pressed, the quicker the frequency changes. This can be used to make fine adjustments to the frequency or to cover a wide range of frequencies.

Figure 8. Frequency_Change Mode() flowchart



3 Conclusion

Digital-to-analog converters are used in many application fields such as audio converters, signal conditioning, instrumentation, test equipment, process control, communication and industrial systems.

This basic example shows how to use the STM8L152C6 DAC to generate different waveforms based on tables stored in memory. Furthermore, it describes the settings needed to use both the DMA controller and TIM4 together with the DAC peripheral in order to release CPU resources so they can be used for other purposes.

This application note can be a starting point for users to develop their own applications involving STM8L152C6 12-bit DAC combined with high performance core and ultra-low-power modes. It can be adapted to cover various needs and a wide range of applications.

4 Revision history

Table 3. Document revision history

Date	Revision	Changes
17-Sep-2010	1	Initial release