SC0069
Sample Code

AT32_TMR and DMA Update PWM Duty Cycle Dynamically

Introduction

This document takes AT32F403A as an example to introduces how to use AT32 TMR and DMA to update PWM duty cycle dynamically.

This sample code use TMR4 update event to trigger DMA1 to transmit data to TMR4 channel 2 data register (TMRx_C2DT), so that to dynamically output PWM waveforms with different duty cycles in the TMR4 channel 2 (PB7).

Note: The corresponding code in this application note is developed on the basis of V2.x.x BSP provided by Artery. For other versions of BSP, please pay attention to the differences in usage.

Applicable products:

Part number	ARTERY AT32 family

List of peripherals:

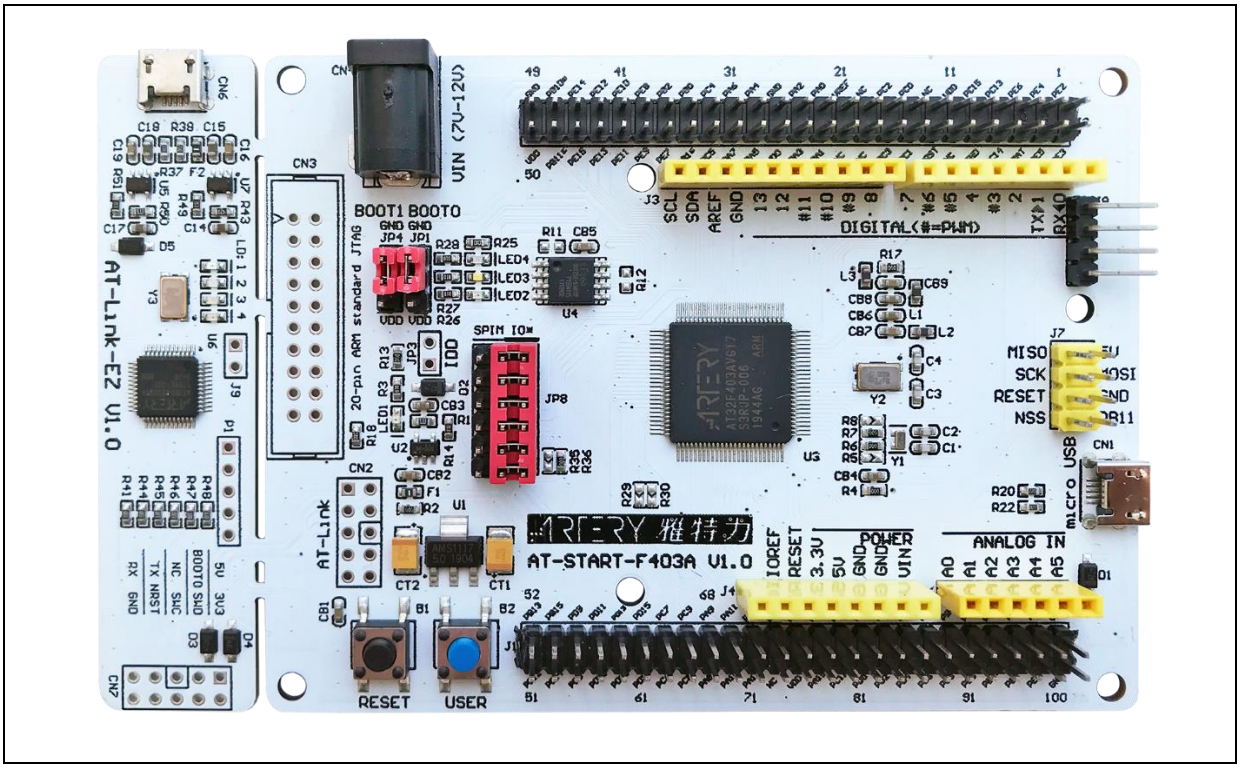
Main peripherals	TIMER
	DMA

1 Application method

1.1 Hardware requirements

1) AT-START-F403A V1.x evaluation board;
2) Logic analyzer or oscilloscope

Figure 1. AT-START-F403A V1.0 evaluation board



1.2 Software requirements

1) This demo uses AT32F403A, and the BSP version is AT32F403A_407_Firmware_Library_V2.0.6.
2) TMR configuration:
   - Configure the TMR4 channel 2(PB7) to generate PWM waveforms.
   - Configure DMA1 and set channel 7 (TMR4 update event) for request mapping; transmit different duty cycles in the memory to the TMR4 channel 2 data register (TMRx_C2DT) to update the duty cycle dynamically.
3) Modify the “src_buffer[]” array in *main.c* to set the required duty cycle; the array length is calculated by software automatically.

Note: All projects are built around keil 5. If users want to use them in other compiling environments, please refer to AT32xxx_Firmware_Library_V2.x.x\project\at_start_xxx\templates (such as IAR6/7, keil 4/5) for a simple change.

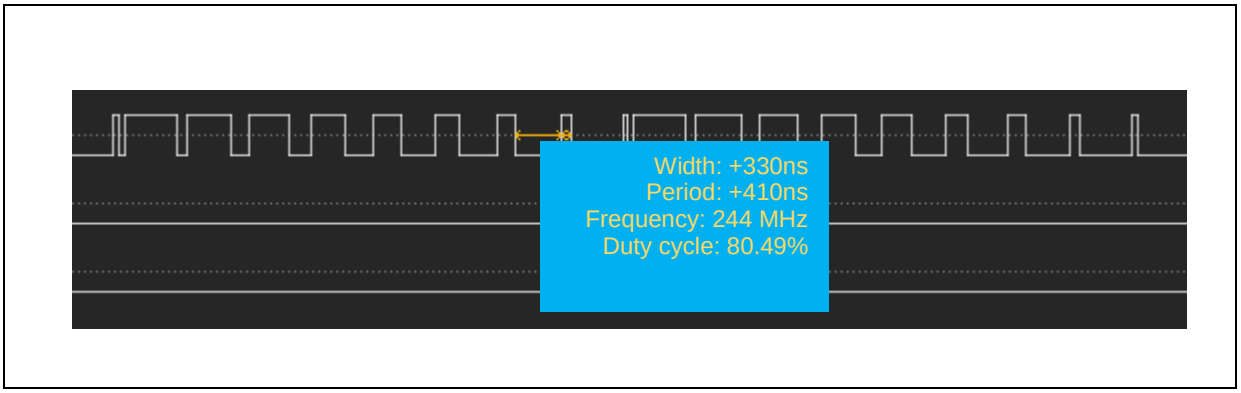
1.3 Example of application

1) Hardware connection: Connect PB7(TMR4_CH2) to logic analyzer or oscilloscope;
2) Open \SourceCode\SC0069_SourceCode_V2.0.0\utilities\SC0069_Demo\mdk_v5\Duty.uvprojx

then compile and download to the evaluation board;

3) Capture the PB7 waveform and verify the effect. In normal operation, TMR4_CH2 (PB7) outputs the PWM waveform with corresponding duty cycle according to the variables in “uint16_t src_buffer[] = {10,20,30,40,50,60,70,80,90,10,20,30,40,50,60,70,80,90};”. The PWM waveform in this Demo is shown in Figure 2.

Figure 2. LA captures waveform



2 Revision history

Table 1. Document revision history

Date	Version	Revision note
2022.01.25	2.0.0	Initial release.