Use of Complete Analog Stream on Microcontroller

Targeted competences: Use of Digital to Analog converter, of Analog to Digital Converter and use of timer for sampling rate.

Hardware: STM32F7 Nucleo board

Framework: Keil µvision and CubeMX from STMircoelectronics

The aim of this document is to show how to use the DAC and ADC components in order to realize a complete analog acquisition stream. The Figure 1 presents the stream that we want to obtain. This is a classical way to get an analog input and to provide an analog output. Of course, in this document, we do not present all the ways whose the ADC and DAC components can be used but this is a first approach in order to be able to use this kind of component.



Figure 1: Acquisition Stream

To be able to get an analog input we can use an ADC; this ADC needs to have a sampling frequency in order to transform the continuous value (analog) into a discrete digital one. So, to obtain this sampling frequency we need to add a timer into this scheme. Furthermore, we need to provide the output signal regularly so this frequency will be also used for the output value. It is possible to use one timer for each component (ADC and DAC) but in several cases the output value will be provide at the same speed than the input one will be sampled even if a mathematical treatment will be realized between the two.



Figure 2: Global Scheme

1. Microcontroller configuration

The first step is to configure the microcontroller. In our case, we use the NUCLEO-F767ZI platform based on a STM32F7 architecture. In order to configure this board we will use the CubeMx software. After the right board was chosen on CubeMx, you can obtain this:



Figure 3: CubeMx configuration for ADC and DAC

Here, I have already chosen the ADC and DAC component that I would like to use. I decided that the output of the DAC will be the output of DAC1 so I configure the Pin PA4 as DAC_OUT1 by doing a left click on the Pin PA4. I have also configure the Pin PA3 as the input for the ADC1 by doing a left click on the Pin PA3 (ADC1_IN3).

The next step is to configure one timer to generate the sampling frequency. In this example, I chose to use the Timer Tim2 and I realized the configuration shown below (Figure 4).

You can see that I use the timer in its simplest mode because I would like only that this timer generate an interruption at the frequency F_{e} . So, we have only to choose the clock source for the timer; here we chose the internal clock.



Figure 4: CubeMx configuration for Timer

Now we have to configure the clock for the whole board; to do this click on clock configuration item. With this configuration panel, we can choose the frequency of different μ controller components as CPU frequency, AHB, APB1 buses and so on. I decide to use the maximum frequency of the SYSCLK that is 216MHz. In this case, the frequency of the APB1 and APB2 timer clocks are respectively 108MHz and 216MHz (see Figure 5). These frequencies will be very important in the next step when we will have to configure the sampling frequency.

Now we are going to configure the timer so that generates the sampling frequency. First click and configuration panel then on TIM2 now, we are going to determine the values for the prescaler and the counter period. The prescaler value divides the clock timer frequency in order to obtain a small one; it is useful if you want to have a small sampling frequency. Indeed, as the frequency in our case is 216 MHz and as the maximum count value is 4294967295 (32 bits timer) the minimum of the frequency scaling can be 0.05Hz. The prescaler parameter has more importance when you use a 16 bits timer.

STM32CubeMX STCubeGenerated.ioc*: STM	32F767ZITX			_				
File Project Clock Configuration Window	Clock							9 🖬 🖬 🗗 🛧
Phout Clock Configuration Configuration Po	configura	tion						
HSE	comigura	uon						
Input freque	itom						216	To Ethernet PTP (MHz)
0-1000 KHz	num				-	•	216	memory and DMA (MHz)
32 1042	5				(11 -		216	To Cortex System timer (H)
	32 To IWDG	(KHz)		AP81 Prescaler	PCLK1		216	FCLK Cortex clock (MHz)
HSI RC	System Clock P	Aux		•• /4 •	54 MHz max		54	APB1 peripheral clocks (MF
16 MHz	HSE	SYSCLK (MHz) AHB	Prescaler HCLK (MHz)	AP82 Prescaler	X 2 PCLK2		108	APB1 Timer clocks (PIHz)
PLL Source Mux	PLICIK	216 / 1	216 MHz max	•• /2 •	106 MHz max		108	APB2 peripheral clocks (MF
HSI @ PLLM		220			A 2	USART1 Clock Hus	210	AP82 timer clocks (PIHz)
Input freque	X /2 PLLO	-55	• 95	To LCD-TFT (MHz)		SYSCLK		
4-36 MHz	- /2 + 216	CLK48 Clock Hus	▶ 192	To SPIDFRX (MHz)	USART2 Clock Nuo PCLKI	HSI	108	To USART1 (MHz)
	PLL /2 - 216	PLLQ	▶ 216	To RNG (MHz)	SYSCLE	LSE O		
	PLISAIP	PLLSAIP 12C1	Clock Mux 216	To USB (MHz)	LSE	USART3 Clock Mus	54	To USART2 (PHHz)
	*N /P	V 100				SYSCLK		
	- <u>/2 + /1 +</u> /Q	125 Clock Mare		To I2C1 (MHz)	PCLI2	HSI O	54	To USART3 (MHz)
	PLISAI1 4 /2 4 /2 4 -1	PLLIZSR (9)	•	The state of the state of	SYSCUL	LSE .		warmen dances
	X	125_CON 12C2	Clock Mux 192	101251,2,3 (MHz)	LSE	UART4 Clock Mux	108	To USACTE (FIRE)
	*N /P		•		-0	SYSCLK		
	PLLI2S /Q PLLI2SR	PCUG		To I2C2 (MHz)	PCUG	HSL	54	To UART4 (MHz)
	12 + 192 /R	SYSCLK	•	T- 1202 (MM-)	SYSCU 0	LSE		T- HARTS (MIL-)
12.288		HS1 12C4	Clock Mux 54	To taca (role)	LSE	BART7 Clock Nux-	54	Ju over s (ring)
MH2 MC01 s	ource Mux		•		-0	SYSCLK		
0	LSE	PLISAIO HSI		To I2C4 (HHz)	PCLICI CIGOR PAGE	HSI O	54	To UART7 (HHz)
(MHz) MCOT ++ 26 + /1 +	HSL	PLLIZSQ	•		SYSCLE			
0	PLICIK	PLLSRC DFSE	M1 Audio Clock Mux	To SAI1 (HHz)	LSE	POLKI N.	54	To UARTS (MHz)
MC02 s	ource Hux	SAll	CLK		-	151 (8)		
0	SYSCLK BI(1258	SAT2 Clock Hux PLLSATQ	CLK 192	To DFSDH1 Audio (M	HIELEC Clock Mux	HSI 0	54	To LPTIM1 (NHz)
(MHz) MC0244 215 (/1)	HSE	PLLI2SQ	V	HSI	/ 482 - (0)			and the second second
0	PLICIK	PLLSRC	• 192	10 SA12 (HH2)	LSE	SDMMC1 Clock Ma	n.032787 K	TO CEC (MM2)
		-P			DF5DH1 Clock Ha	- 1.	216	To SDMMC1 (MHz)
(*)			m					

Figure 5: Clock configuration for the board

For this example, I would like a sampling frequency of 2.1Hz so I chose the following values: prescaler value 30000 (in this case, the clock frequency of the timer will be 7.2 KHz instead of 216 MHz), the counter period value 3323 (3323 ticks at 7.2 KHz represents a sampling period of 0.46s).

STM32CubeMX STCubeGenerated.ioc*: STM32F767ZITx			
File Project Window Help			
B 🐸 🖹 🖬 🖏 💺 🧶 🎒 🕂 🕂 🕐	Configuration		💿 🖬 🖸 🗗 🔀
	goiniguration		
Pinout Clock Configuration Configuration Poil or Cons	nanol		
Additional Software	panei		
HiddleWares			
E FAIrs			
E SE CONCE			
Enabled		NVIC Setting	75
B- CHAPHICS		11110 Detting	50
- STemWin			
-Display Interface Di +	TIM2 Configuration		×
e 🧕 LIBJPEG	Parameter Settings all the Constant	MA Settings	
- Enabled	Configure the below parameters a	Contract Secondar Contract Secondary	
B MBEDTLS	Compare the below parameters :		
Enabled	Search : Search (Crt/+F)	• •	H
Peripherals	Counter Settings		
	Prescaler (PSC - 16 bits value)	30000	
	Counter Mode	Up	
	Counter Period (AutoRelast Registe	r - 32 lats value) 3323	-
E DAC	Internal Clock Division (CKD)	No Division	
OUT1 Configuration: Set	auto-reload preload	Disable	
e @ DMA2D Multime	dia. 😑 Trigger Output (TRGO) Parameters		Security
- Activated	Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)	
🖶 🍥 IWDG	Trigger Event Selection TRGO	Reset (UG bit from TIMx_EGR)	
- Activated			
e 💿 RNG			
- Civated			
e sys			
Timebase Source:SysTick			
⊖ • TIM2			
Clock Source :Internal Clo			
E TIME			
Activated	Counter Period (AutoReload Register -	32 bits value)	
	Period must be between 0 and 4 294 967 2	95.	
E Article			
El Australias			
G WWDG	Restore Default	Apply Ok Cancel	
- M Activated			

Figure 6: Timer configuration

The last thing to do here is to enable the interruption for the timer. To do this, click on the NVIC Settings panel and enables the TIM2 global interruption by checking Enabled. Then click on Ok.

Now, we are going to configure the ADC component for that click on ADC1 panel; the figure below appears.

- B L K - 6 6 5 +	P			🎯 🖪 🗹 🖸
Clock Configuration Pe	ower Consumption Calculator			
Configuration Configuration Configurati	Healtheedfa	AC1 Configuration Torunits Setting Team Contacts Setting Torunits Setting Torunits Setting Setting	*MACE Settings	

Figure 7: ADC configuration

Here we are going to use the ADC in its simplest mode so we choose the configuration presented in figure 7 (for more details see the ADC document that I wrote).

The last component to configure is the DAC. For that, we click on the DAC configuration panel and we choose the settings presented in the Figure below.

STM32CubeMX STCubeGenerated.ioc*: STM32F7672ITx		
File Project Window Help		
B = B U B ≤ 6 ≤ + - 8 P		🚳 🖬 🖬 🖎
Pnout Clock Configuration Configuration Power Consumption Calculator		
Tributed Source Source Source Source Source Source Source Sour	DEC Configuration Parameter Settings Loss Containes Marco Settings Onlo Onlo Settings Onlo Settings Onlo Onlo Setting	

Figure 8: DAC configuration panel

Now, we configured all the necessary components to realize the application. As, most of them used interruption capability we have to define the priority of these interruptions in order to obtain the desired behavior.

To do this, choose the NVIC configuration panel and change the preemption priority so that the TIMER 2 had the higher priority level (more the priority is high most the number is near 0).

Ministeries and a serie of a seri	The set of	and Their Conference Conference Brown Committee Calculation	and the second se					-	
With State Control Image: State And And Control	With Control	In a second state of the second	rue .	_	_			_	
Halfware I Alta I Alta I Status I Status <t< th=""><th>Index Index Index</th><th>Additional Software</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th></t<>	Index	Additional Software							
Not set and set a	HATS Here when and a second and a s	HiddleWares							
Horized Security Configuration Horized Security Configuration Horized Security Configuration Horized Security Hori	How whether How wheth	EATES							
Hullburds	Hold Configuration Hold Configuratio	- El User-defined							
Image: Second	Image: Section of the section of t	C TO CONTRACT							
Horder States Horder	Model M	RETRIOS							
More Comparison for the second provide the first agreement of the second provide th	Michael Construction Parks And	- E Ensbied							
Control benefit C	Control bench in the cont	G GRAPHICS	A MAT Conferentian				-	X D	
Latrice handed Latrice	Latrice for the first acceptance of the first acceptanceceptance of the first acceptance of the first acceptance of the f	- STemWin	W NVXC Configuration				-		
Monte finanze fin	Hong Table Sector Stand Se	-Osplay Interface Di	NVIC of Code generation						
Hord Source State And	Hority Concerning and Preventioner	CO. IN 10100	La desta d						
Section	Image: Section of the section of t	The second	Priority Group 4 bits for pre-emption priority 0 bits for subpriori	ty.		Sort by Premption Pro	onity and Sub Pro	eity	
HUSS HUSS HUSS Hussel	Image: Second	E brauled							
		HIN MEDTLS	Search Search (Citi+F)		٠	Show only enabled in	terrupts		
Provent Tale Provent Tale Provent Tale • All Clinical Clinic	Provide Transition Provide Transition • Additional of the second of the secon	Enabled							
Action of the second seco	Action	Peripherals	Internet Table	En	ables	Dreamotion Drinity	G IN DOWEN		
• OX 64 ************************************	• OX 5d + 0	D-ADC1	Non-markable intern et		120	in the second se	in.		
CdC CdC CdC CdCC CdC CdCC CdC	Oth Channel Do Do Do Image: Stand of Control Contrel Control Control Contrel Control Control Contrel Co	IN3: Set	inter of the driver of		173	0	6		
Image: Second diamond Image: Second diamondiamond Image: Second diamond	Protects flast, rows access fail. D D • • OAC • • OAC 0 0 • • OAC • • OAC 0 0 0 • • OAC • • OAC 0 0 0 0 • • OAC • • OAC 0 0 0 0 0 • • OAC • • OAC 0	e er	Mamoru management facili		121	là.	0	-11	
With and the set of the set	Visit of the second	- III Activated	Pre-fetch fault, memory access fault		13	ő	0		
Model Model Bit Model D D D Model Model D D D D Model Model Model D D <	Million Multimetric D D D • Made • Made		Undefined instruction or illegal state	_	157	0	0	100	
- OT Configuration: Etc Description: The Description of the	Oth Codeparation St 0 0 0 0 Image: Second State	E- S DAC	System service call via SWI instruction		121	0	0		
Image: Second	Image: Second	OUT1 Configuration: Set	Debug monitor		121	0	0		Securit
Image: Second and Second an	Image: Second and the second and secon	DMA2D	Pendable request for system service		121	0	0		
MOG Mode	WOG Cheeded Cheeded Gos Cheeded Gos Cheeded Gos Cheeded Cos	- P Activated	Time base: System tick timer		12	0	0		
Constant in the second interval in	Constant of the second of	CH TWDG	PVD interrupt through EXTI line 16	_	11	0	0		
Constant of the second of	Verond	The Associated	Plash global interrupt		12	0	0		
Constant Constan	The share defined The share defined and the share defined		RCC gobel interrupt	_	-	0	0	-0.0	
Petered P	Botherd B	G- P RNG	AUC1, AUC2 and AUC3 good interrupts	-	1	3	0		
Source Status Transaction Status	Vision Security (1) Transfer Securi	- E Activated	TML debut interrupt	-	1	6	0	-201	
Terret and a source spirit a source spiri	The second second splits The second	10-19 SYS	FRU data internat	-	-14-	ő	0	-13 I III	
VICE	VID2 Code Sector Share Color Table Code Sector Share Color Table Code Sector Share Color Code Sector Share Color Code Sector Secto	Timebase Source:SvsTick			-	14		-21	
WYNG WYNG Actived Active	WYNG WYNG WYNG WYNG WYNG	D. 0 THIS							
STANDE Concol	The Active of Constant State of Constant St	Clock Service (Internal Clo							
Vice Adverter Adverter	Vice Adverted Vice Adverted Adverted Adverted Adverted								
Koned Koned Koned Koned Koned Koned Koned Koned	Advised Advised Advised	- · · · · · · · · · · · · · · · · · · ·							
Tore Phane Hole T	State Today State State Today	Activated							
WYOC Const Addy Addy Const	WING Montel Montel Montel Montel Montel Montel	- Che Pulse Mode							
Conclusion	Concluded	E TIM7							
The share hader	Constant Hole: Constant	- E Activated							
Protection from the second secon		- El One Bular Mode	Constant Process	a new li		I name			
Advied Approx 0. Const	Advied		Dabled Preenpts	er robecy		- second +			
Athulad	Advised	THAT IS A REAL PROPERTY OF THE							
		Activated				Apply Ok	Cancel		
			L						

Figure 9: NVIC configuration panel

After this last step, all the configurations for this application have been realized so we can generate the code for our application. To do this, click on the code generation icon as shown in Figure 10. Then give the name for your project and the path for its location finally choose the toolchain/ide; in this example, I chose to use the MDK-ARM environment.



Figure 10: Project configuration panel

After clicking on ok, CubeMx generates the application code and create the MDK-ARM project as shown in the figure below.



Figure 11: MDK-ARM code generation

At the code level, few things are to do indeed, we only want that the sample provided from the ADC to be transferred to the DAC at each sampling period. Therefore, the first thing to do is to start the timer in interruption mode. In the main.c file it is the only thing to write.

The second step is to write the code for the interrupt functions. When the timer interrupt occurs, we have to start the ADC conversion and in function of the etat value (0 or 1) either we transfer the ADC value to the DAC output or we transfer a '0' value to the DAC output. This is just an example; the behavior can be another one.

The code to realize this behavior is shown in figure 12 and 13.

CAUsers/JOHANM/Desktop/essai_dac/Stream_ADC	DACIMDK-ARM\Stre	am_ADC_DAC.uvprojx - µVision	
File Edit View Project Flash Debug Periph	erals Tools SVCS	Window Help	
D 12 2 2 4 × 2 15 9 ~ 1 + -	P 2 2 3	🐨 🕸 //: //: 🙆 tmp_dac_value 🕞 🍌 🌮 🍭 - 💿 🔗 🍂 🛄 - 🔍	
😌 🖽 🥵 - 📖 🙀 Stream ADC DAC		* 7 8	
Project 3	E) main.c*) stm32f7xx_it.c*	* X
Charge Stream, ADC, DAC Portugal, ADC, DAC Portugal, ADC, DAC Portugal, and ADC, DAC	15 16 17 18 19 20 21 22 23 24 25 26	Modfor other materials provided with the distribution. Multither heams of Thitteneiterrowice nor the mass of the contributors wy be used to notore or promote products derived from this software without specific prior within the permission This SortMake 19 Horiverbe D'I The Corpting to the Cortisorods "Ad 14" This SortMake 19 Horiverbe D'I The CortFord Notes Adv CortFistorods "Ad 14" The SortMake 19 Horiverbe D'I The CortFistorods Table 10" The CortFistor D'I The CortFistor Notes Adv CortFistorods "Ad 14" The CortFistor D'I The CortFistor D'I The CortFistorods and the CortFistorods "Ad 14" The CortFistor D'I The CortFistor D'I The CortFistorods Table 10" The CortFistor D'I The CortF	
8 🥃 Drivers/CMSS	27 28 29 31 31 32 34 35 36 36 36 36 36 40 40 42 41 41 42 43 41 42 43 41 42 43 52 45 52 52 52 52 52 52 52 53 52 54 52 55 52 54 54 54 54 54 54 54 54 54 54 54 54 54	<pre>Control and on any timeout or insality, matters is contract, first Lisbity of THIS contract, first if Anvisto or THE FOSSISILTY of SUC DANKS. ''' '' Include 'HESIF's has'' '' Include 'HESIF's has'' '' Include 'HESIF's has'' ''' Danks 'HESIF's has'' ''''''''''''''''''''''''''''''''''</pre>	Here I define the variable to store the ADC value conversion and the state variable etat. Be careful: write your own code between the two beacons "USER CODE BEGIN" and "USER CODE END" otherwise your code will be deleted if you regenerate a configuration from CubeMx
Deget Oran Orantees Retenuites	33 + - 55 + - 56 58	<pre>Brid Full function handles System tick timer. //id SysTick_Hendler(void) // UESR COME Satts SysTick_INGD, 0 */ // // UESR COME Satts S</pre>	, .
		ST-Unk Debugger	

Figure 12: Interruption variable statement

C\Users\JOHANN\Desktop\essai_dac\Stream_ADC_D	DACVMDK-ARM/Stream_ADC_DAC.uvprojx - µVision	
File Edit View Project Flash Debug Peripher	als Taols SVCS Window Help	
N 😅 🖬 🖉 🔺 🗠 🙇 🔊 (~) 🖛 🛁	🕐 🔃 🔃 🔃 🕼 /// 🕼 🐲 tmp_dac_value 💿 🔜 🥐 🚳 🔹 🔗 🔕 💽 🔦	
😕 🕮 🥶 • 🗔 🗱 Stream_ADC_DAC	- ※ ▲ 吉 ◆ 今 翰	
Project a	3	
Bergert Stream, ABC, DAC Borgert Mana, ABC, DAC Borgert Mana, DAC, DAC Borgert Mana, Stream,	1 * Bitset This Function bandles ADC1, ADC2 and ADC1 global interrupts.	ADC conversion value
	<pre>11 (Free_contents) (State () (Stat</pre>	In function of the etat value the DAC output provides the ADC value or '0'. The state of etat changes at each sampling period
⊒Payet ⊜trast 0 Function 0 ₄ respirer. Built Collect	<pre>ising</pre>	
4	Static Designer , -	

Figure 13: interruption code function

At the end we have to compile the code in order to obtain the executable file for the STM32F7. When the compilation is done without error, we can transfer the code into the board. A demo video can be found here: <u>https://youtu.be/ u9r6jeGoiE</u>