

CREDO School

HANDS ON LAB KATARZYNA SMELCERZ

CREDO School | Kraków, 2018 Instytut Fizyki Jądrowej mgr inż. Katarzyna Smelcerz, Politechnika Krakowska, IFJ

CREDO SCHOOL - HARDWARE

Sesja została przygotowana dla osób poczatkujących w programowaniu sprzętowym. Poświęcona jest mikrokontrolerowi **STM32L152 RB.** Oprogramowanie będzie pisane w środowisku IAR **Embedded Workbench.** Wszystkie zadania będą wykonywane na podstawie materiałów zawartych w bieżącym dokumencie.

OPROGRAMOWANIE I DOKUMENTACJA

W mailu, który był wysłany znajduje się lista oprogramowania koniecznego do zainstalowania.

Dokumentacja:

- Opis zestawu starter-kit, schemat: <u>http://www.st.com/st-web-</u> <u>ui/static/active/en/resource/technical/document/user_manual/DM00027954.pdf</u>
- Kompilator dla rdzenia ARM: <u>https://www.iar.com/iar-embedded-workbench/arm/</u>
- Opis rdzenia ARM STM32L152 <u>http://www.st.com/st-web-</u>
 <u>ui/static/active/en/resource/technical/document/datasheet/CD00277537.pdf</u>
- Dokumentacja do środowiska STM32CubeMX: <u>http://www.st.com/st-web-</u> ui/static/active/en/resource/technical/document/data_brief/DM00103564.pdf

1. PODSTAWOWE WIADOMOŚCI DOTYCZĄCE MIKROKONTROLERA ARM STM32L152 RB

Mikrokontroler ARM STM32L152 RB, znajdujący się w zestawie starter-kit, wykorzystywanym na laboratorium, należy do rodziny L. Znacznik L w nazwie, oznacza rodzinę Low Power, czyli produkty przeznaczone do projektów, w których niskie zużycie energii (np. pochodzącej z baterii) jest priorytetem. Często jest jednak tak, że w zamian za niskie zużycie energii otrzymujemy mniejszą moc obliczeniową, tak też i jest tym razem. Jeśli prześledzimy na stronie ST rodziny rdzeni, pod względem mocy obliczeniowej, łatwo jest zauważyć, że istnieją rdzenie, które mają szybsze zegary i większe pamięci, np. rodzina F.

Procesory niskomocowe mają często dodatkowe wyposażenie, niespotykane w innych rodzinach. Tak też jest w tym wypadku, procesor STM32L152 ma wbudowany sterownik ekranu LCD. Niski pobór mocy oznacza często mniejszą moc obliczeniową, ale wciąż wielokrotnie większą niż możliwości procesorów 8 i 16 bitowych.



Poniżej przedstawiono porównanie rdzeni serii L1.

Rys 1.1 Porównanie rdzeni serii L1, zródło: ST.com

Ogólna charakterystyka rdzenia STM32L152 RB:

- Niskie zużycie mocy, 1.65 V 3.6 V zasilanie
- 0.3 µA Standby mode
- 0.9 µA Standby mode +ko RTC
- 0.57 μA Stop mode
- 1.2 µA Stop mode + RTC
- 9 µA Low-power run mode
- 214 μA/MHz Run mode
- 10 nA ultra-low I/O
- < 8 μs czas wybudzenia
- Rdzeń: ARM®Cortex®-M3 32-bit CPU
- Od 32 kHz do 32 MHz, częstotliwość zegara

2. KRÓTKI OPIS ZESTAWU STARTER-KIT -STM32L-DISCOVERY



1.2 Zestaw ewaluacyjny STM32L-Discovery

Zestaw ewaluacyjny STM32L-Discovery jest wyposażony w rdzeń STM32L152 RB, który został opisany w poprzednim rozdziale. Nagrywanie programu odbywa się za pomocą ST-Link, który jest wbudowany w płytkę.

Można również go podłączyć jako urządzenie zewnętrzne, za pomocą 4 pinów. Zasilanie jest dostarczone przez kabel USB.

Płytka, na której będą wykonywane zadania laboratoryjne jest między innymi wyposażona w:

- Wyświetlacz LCD
- 4 diody
- 2 przyciski
- Czujnik dotykowy + 4 dotykowe klawisze
- Pomiar prądu zasilania

Więcej szczegółów znajduje się w dokumentacji dostarczonej przez producenta.

3. KONFIGURACJA I PODSTAWOWE FUNKCJE ŚRODOWISKA IAR EMBEDDED WORKBENCH

Kompilator IAR wymaga konfiguracji do pracy pod konkretnym rdzeniem. Wszystkie ustawienia są przedstawione na poniższych print screenach. Nagranie programu odbywa przyciskiem "Download and

Debug" 🌛 , uruchomienie "go" 🚧 , pozostałe funkcje debuggera "step over", "step into" itd również są

dostępne 🔁 🕒 🗁 🕹 👉 😕 😕 . Podgląd zmiennych odbywa się w trybie debuggowania w oknach

Watch i Locals – należy je wywołać w zakładce "View", na górnej belce kompilatora.

Live Watch		×
Expression	Value	
etato machino	0	

. г						
ategory:						
General Options						
C/C++ Compiler						
Assembler	Target	Output	Library Cor	figuration	Libran Ontione	MISRAC-200 4
Output Converter	rargor	Output	Libidiy Cui	ingulation	ubialy options	MI3NA-5.200
Custom Build						
Lioker	Proc	essor va	riant			
Debugger		-	Cortoy M2		_]	
Simulator	0.	Die	COREX-MD			
Angel	0	levice	ST STM32	152x8	1	Ē.
CMSIS DAP		20100			0	
GDB Server	1 E					
IAR ROM-monitor	- E-2	Ya		TTU		
I-jet/JTAGjet	End	an mode		EPU		
J-Link/J-Trace	01	ittle		Non	e	*
TI Stellaris	OE	Big		03		
Macraigor	1	BE32				
PE micro	6	BER				
RDI		g or <u>o</u>				
ST-LINK						
Third-Party Driver						
TT XDS100/200	L					

Category:	-				
Seneral Options					
Assembler			1	i	
Output Converter	Target	Output	Library Configuration	Library Options	MISRA-C:200
Custom Build	Outr	ut file			
Build Actions		Evecutab			
Linker		Library	101 1		
Simulator		Donary			
Angel	Outp	ut directo	ories		
CMSIS DAP	Ex	ecutables	s/libraries		
GDB Server		obulo\ Eve			
IAR ROM-monitor	0	ebug \Exe	8:		
I-jet/JTAGjet	Ob	ject files:			
J-Link/J-Trace	D	ebug\Obj			
TI Stellaris Macraigor	Lis	t files:			
PE micro	D	ebug\List			
RDI					
ST-LINK					
Inird-Party Driver					

Category: General Options C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator Angel CMSIS DAP	Output Output Image: Contrast additional output Output format: Dutput file
GDB Server IAR ROM-monitor I-jet/JTAGjet J-Link/J-Trace TI Stellaris Macraigor PE micro RDI ST-LINK Third-Party Driver TI XDS 100/200	M 1560.bin

ategory:	-					Factory Setting
eneral Options						Ļ
C/C++ Compiler						
Assembler	1	-				
Output Converter	Setup	Download	Images	Extra Options	Plugins	
Custom Build	5 			125221125015		
Build Actions	Driver			Run to		
Linker	ST-L	INK	+	main		
Debugger				1. <u>1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1</u>		
Simulator	Setu	ip macros				
Angel		Use macro f	ile(s)			
CMSIS DAP						
GDB Server						
TAR ROM-monitor						+++*
1-jet/JIAGjet						
J-Link/J-Trace	Dev	i <u>c</u> e descriptio	on file			
Macraigor		<u>O</u> verride def	ault			
PE micro	ST	OOLKIT_DI	R\$\CONF	IG\debugger\S	T\STM32L152	x8.ddf
RDI						
ST-LINK						
Third-Party Driver						
TT YDS 100/200						

ategory:				(Factory Settings
eneral Options					
C/C++ Compiler					
Assembler	STJUNK				
Dutput Converter					
Lustom Build	Reset				
inker	Connect dur	ing reset		-	
Debugger					
Simulator	Interface	Clock setup			
Angel	· · · · · · · · · · · · · · · · · · ·				
CMSIS DAP	O JTAG	CPU clock:	72.0	MHz	
GDB Server	- 0141D				
IAR ROM-monitor	<u> </u>	SVVU CIOCK:			
I-jet/JTAGjet			2000	kHz	
J-Link/J-Trace					-k
11 Stellaris Macraigor					
PE micro					
RDI					
ST-LINK					
Third-Party Driver	1				
TI XDS100/200					

4. PODSTAWOWE WIADOMOŚCI DOTYCZĄCE ŚRODOWISKA STM32CUBEMX

Środowisko Cube, jest środowiskiem graficznym, służącym do generowania kodu w języku C. Wygenerowany kod, jest kompatybilny z takimi kompilatorami, jak Keil, IAR oraz GCC. Środowisko to, bardzo ułatwia konfigurację peryferii, portów IO, czy też zegarów.

Ręczna konfiguracja jest mozolna i łatwo w niej popełnić błędy.

5. PORTY WEJŚCIA/WYJŚCIA

Porty w omawianym na laboratorium mikrokontrolerze zazwyczaj występują w postaci 16 bitowej. Tak jak we wszystkich architekturach, tak i tu, służą do komunikacji ze światem zewnętrznym. Port może być skonfigurowany w różnych trybach, takich jak te przedstawione poniżej.

```
typedef enum {
GPIO_Mode_IN = 0x00, /*!< GPIO Input Mode */
GPIO_Mode_OUT = 0x01, /*!< GPIO Output Mode */
GPIO_Mode_AF = 0x02, /*!< GPIO Alternate function Mode */
GPIO_Mode_AN = 0x03 /*!< GPIO Analog Mode */
} GPIOMode_TypeDef;</pre>
```

W mikrokontrolerach ARM port konfiguruje się jako strukturę. Opis tej struktury jest zawarty w bibliotece stm32l1xx_gpio.h, dołączonej do projektu.

typedef struct		
{		
uint32_t GPIO_Pin;	/*!< S	pecifies the GPIO pins to be configured.
	This pa	rameter can be any value of @ref GPIO_pins_define */
GPIOMode_TypeDef GPIO_M	Mode;	/*!< Specifies the operating mode for the
		selected pins. This parameter can be a value
		of @ref GPIOMode_TypeDef */
GPIOSpeed_TypeDef GPIO_	Speed;	/*!< Specifies the speed for the selected pins.
		This parameter can be a value of @ref
		GPIOSpeed_TypeDef */
GPIOOType_TypeDef GPIO_	_OType;	/*!< Specifies the operating output type for the
		selected pins. This parameter can be a
		value of @ref GPIOOType_TypeDef */
GPIOPuPd_TypeDef GPIO_H	PuPd;	/*!< Specifies the operating Pull-up/Pull down
		for the selected pins. This parameter can
		be a value of @ref GPIOPuPd_TypeDef */
<pre>} GPIO InitTypeDef;</pre>		

UWAGA! Jak ustawić konkretny Pin dla danego portu w stan wysoki lub niski, jest opisane w pliku stm32l1xx_hal_gpio.c, dołączanym automatycznie przez Cube na etapie generowania kodu w języku C.

6. KONFIGURACJA PORTÓW W ŚRODOWISKU CUBE

Jak widać na Listingach zawartych w poprzednim rozdziale, konfiguracja portów jest dosyć żmudnym i czasochłonnym procesem. Środowisko Cube dostarcza programiście narzędzia, które pomaga znacznie zaoszczędzić czas. Poniżej przedstawiono przykładową konfigurację portu oraz konfigurację Cube dla naszego rdzenia.

Kliknij "new project", wybierz odpowiedni rdzeń, zgodnie z nr układu na płytce:

ICU Selector	Board Sele	ctor										
	Dual u Sele	CLON										-
MCU Hitters		Line				Dadrage (
All		All			3771	All		1 -		-		-
<u>^"</u>	*	-			•	~	~		MON	e riiters •		_
Peripheral Se	lection			MCUs List: 817 Ite	ms							
Periphera	ls	Nb	Max	MCU	Lines		Package	Flash	Ram	Eeprom	IO	Ę
ADC 12-b	t	0	40	CTM22L1E17CT	CTMO		LOEDIAA	250	22	0102	115	1
ADC 16-bi	t	0	3	STM32L1512CTX	511432	1151/152	LQFP144	200	32	0192	115	-
CAN		0	2	STM32L1512D1X	511432	1151/152	LQFP144	384	48	12288	115	-
COMP		0	7	STM32L1512ETx	STM32	1151/152	LQFP144	512	80	16384	115	-
DAC 12-bi	t	0	3	STM32L152C6Tx	STM32	1151/152	LQFP48	32	10	4096	37	
DCMI			-	STM32L152C6TxA	STM32	L151/152	LQFP48	32	16	4096	37	
DFSDM		4	1	STM32L152C6Ux	STM32	1151/152	UFQFPN48	32	10	4096	37	
DSIHOST	-	님	-	STM32L152C6UxA	STM32	L151/152	UFQFPN48	32	16	4096	37	
Ethernet		님	-	STM32L152C8Tx	STM32	L151/152	LQFP48	64	10	4096	37	
EMDT2C		⊢	-	STM32L152C8TxA	STM32	L 151/152	LQFP48	64	32	4096	37	
ESMC		井		STM32L152C8Ux	STM32	L151/152	UFQFPN48	64	10	4096	37	
	0	븜		STM32L152C8UxA	STM32	L151/152	UFQFPN48	64	32	4096	37	1
HRTTM		H		STM32L152CBTx	STM32	L151/152	LQFP48	128	16	4096	37	1
0 I2C		0	4	STM32L152CBTxA	STM32	1151/152	LOFP48	128	16	4096	37	
12S		0	5	STM32L152CBUx	STM32	151/152	UFOFPN48	128	16	4096	37	
IRTIM				STM32L152CBUXA	STM32	151/152	LIEOEPN48	128	16	4096	37	
LPTIM		0	2	STM32L152CCTx	STM32	1 151/152	LOEP48	256	32	8192	37	
LPUART				STM32L152CCUV	STM32	1 151/152	LIEOEDNI48	256	32	8102	37	-
OPAMP		0	4	STM32L152CCOX	CTM22	1 151/152	UEBCA 122	250	22	0192	112	
QUADSPI				STM32L152QCHX	511432	1151/152	UFDGA132	200	32	0192	112	
RTC				STM32L152QDHX	51M32	1151/152	UFBGA132	384	48	12288	112	-
SAI		0	2	STM32L152QEHx	STM32	1151/152	UFBGA132	512	80	16384	112	
SDIO			-	STM32L152R6Hx	STM32	1151/152	TFBGA64	32	10	4096	50	
SDMMC		4	-	STM32L152R6HxA	STM32	L151/152	TFBGA64	32	16	4096	50	
SPDIFRX		<u> </u>		STM32L152R6Tx	STM32	L151/152	LQFP64	32	10	4096	51	
SWPMT				STM32L152R6TxA	STM32	L151/152	LQFP64	32	16	4096	51	
Segment	CD	+		STM32L152R8Hx	STM32	L151/152	TFBGA64	64	10	4096	50	
TFTLCD		+		STM32L152R8HxA	STM32	L151/152	TFBGA64	64	32	4096	50	
Timer 16-ł	pit	0	14	STM32L152R8Tx	STM32	L151/152	LQFP64	64	10	4096	51	1
Timer 32-ł	pit	0	2	STM32L152R8TxA	STM32	L151/152	LQFP64	64	32	4096	51	1
Touch Ser	nsing			STM32L152RBHx	STM32	L151/152	TFBGA64	128	16	4096	50	1
UART		0	4	STM32L152RBHxA	STM32	L151/152	TFBGA64	128	32	4096	50	1
USART		0	8	STM32L152RBTx	STM32	1151/152	LQFP64	128	16	4096	51	ſ
USB Devic	e			STM32L152RBTxA	STM32	151/152	LOFP64	128	32	4096	51	1
USB OTG	FS			STM32L152RCTx	STM32	1 151/152	LOEP64	256	32	8192	51	1
USB OTG	HS		-	CTMODE & COD CT.	CTMO	H 4 F 4 /4 F 9	LOEDCA	250	22	0102		- 1

1.1 Wybranie rdzenia

	STM32CubeMX Untitled*: STM32L152RBTx	×
File Project Pinout Window Help		
[🔄 🖦 🖶 🦺 🎂 🧾 🗌 Keep Current Signals Placement 🥑 🧒	j — 🥝 🔶 Find 🛛 🗸 🔍 🔍 🗹 Show user Label 🛛 ৈ 🧈 🦆	
Pinout Clock Configuration Configuration Power Consumption Calculator		
Configuration → → MiddleWares → ● → ● → ● → ● → ● → ● → ● → ● → ● → ● → ● → ● → ● → ● → ● → ● → ○ → ● → ● → ○ → ○ → ○ → ○ → ○ → ○ → ○ → ○ → ○ → ○ → ○ → ○ → ○ → ○ → ○ → ○ → ○ <th>VCC VCC VCC VCC PCL PCL PCL PCL PCL PCL PCL PCL PCL PCL PCL PCL</th> <th></th>	VCC VCC VCC VCC PCL PCL PCL PCL PCL PCL PCL PCL PCL PCL PCL PCL	

1.2 Konfiguracja pinów - ustawienie pinu jako wyjście

0		STM32CubeMX Untitled	*: STM32L152RBTx		
ile Project Window Help					
🖸 📂 🖪 🖬 🖳 🧄 🚄 🔶 🗕	🕐 🧈 🦆				
Pinout Clock Configuration Configuration Pow	wer Consumption Calculator				
Configuration HiddleWares HiddleWares G FATFS G FATFS G FAERTOS G FAERTOS G FAERTOS F Fachaled Peripherals					
e o CRC			Middlewares		
Activated					
O IWDG Activated					
- Activated	Multimedia	Control	Analog	Connectivity	System
One Pulse Mode One Pulse Mode Activated One Pulse Mode One Pulse Mode					

1.3 Dalsza konfiguracja pinów

			Pin Configuration			
PIO						
Search Signals						
Search (Crtl+F	=)				Show only	y Modified Pins
Pin Name	Signal on Pin	GPIO mode	GPIO Pull-up/Pull-down	Maximum outpu	User Label	Modified
7	n/a	Output Push Pull	No pull-up and no pull-do	. High		~
PB7 Configuratio	on :					
PB7 Configuratio GPIO mode	on :		Output Push Pull			~
287 Configuratio GPIO mode GPIO Pull-up/Pu	on : Ill-down		Output Push Pull No pull-up and no	pull-down		~
287 Configuratio GPIO mode GPIO Pull-up/Pu Maximum outpu	on : Ill-down It speed		Output Push Pull No pull-up and no High	pull-down		~
287 Configuratio GPIO mode GPIO Pull-up/Pu Maximum outpu User Label	on : ill-down it speed		Output Push Pull No pull-up and no High	pull-down		~

1.4 Tryb, prędkość i pozostałe opcje pinu

Generowanie kodu w języku C:

	Project Settings		
vject Code Generator			
Project Settings Project Name			
IO_LED			
Project Location			
E:\systemy_wbudowane_pk\			Browse
Toolchain Folder Location			
E:\systemy_wbudowane_pk\IO_LED\			
Toolchain / IDE			
EWARM		~	
Mcu and Firmware Package Mcu Reference STM32L152RBTx			
Firmware Package Name and Version			
STM32Cube FW_L1 V1.4.0			

1.5 Zrzut po wygenerowaniu kodu w języku C - nazwa projektu dla IAR

Po naciśnięciu przycisku OK, zostaje uruchomione środowisko IAR. Znajdują się w nim wygenerowane pliki w języku C, zgodne z konfiguracją ustawioną w środowisku CUBE.



1.6 Zrzut z IAR - kod programu

7.TIMERY

Rdzeń ARM STM32L152 RB jest wyposażony w 10 timerów. Sześć 16-bitowych, do 4 kanałów IC/OC/PWM oraz dwa 16-bitowe podstawowe timery i 2 do obsługi watchdoga.

Poniżej znajduje się proste zestawienie wszystkich Timerów wraz z ich najważniejszymi funkcjami.

Timer	x-bitowy	Typ licznika	Prescaler	Obsługa DMA	Capture/compare Ilość kanałów
TIM2, TIM3, TIM4	16-bit	Up, down, up/down	Int 1-65536	Tak	4
TIM9	16-bit	Up	int 1-65536	Nie	2
TIM10, TIM11	16-bit	Up	int 1-65536	Nie	1
TIM6, TIM7	16-bit	Up	int 1-65536	Tak	0

8.NVIC (NESTED VECTOR INTERRUPT CONTROLLER)

Rdzenie ARM wyposażone są w NVIC (Nested Vector Interrupt Controller). Jest to rozwiązanie sprzętowe pozwalające na obsługę przerwania o wyższym priorytecie, nawet jeśli w danej chwili jest wykonywane inne przerwanie, bez interwencji CPU. Więcej szczegółów znajduje się w dokumentacji.

9. KONFIGURACJA TIMERÓW W ŚRODOWISKU CUBE

Poniższe print screeny przedstawiają konfigurację Timera 3.



Zrzut 1.1 Ustawienie TIM3 z zegarem wewnętrznym w odpowiednim trybie na kanale 1 (sygnał nie jest wyprowadzony na pin, ale może być wykorzystany, np. do przerwania)

			iws coniguration
		Parameter Settings 🚽 User Constants 🤘	NVIC Settings 🛷 DMA Settings
		Configure the below parameters :	
		Counter Settings	
		Prescaler (PSC - 16 bits value)	64
		Counter Mode	Up
		Counter Period (AutoReload Register - 1	6 bits va 2000
		Internal Clock Division (CKD)	No Division
		Trigger Output (TRGO) Parameters	
		Master/Slave Mode	Disable (no sync between this TIM (Master) and its Sla
lultimedia	Control	Trigger Event Selection	Reset (UG bit from TIMx_EGR)
		🕞 Output Compare No Output Channel 1	
/	TIM3	Mode	Frozen (used for Timing base)
		Pulse (16 bits value)	0
	6. C	CH Polarity	High
		_	

Zrzut 1.2 Ustawienie odpowiednich parametrów TIM3

	TIM3 Configu	ration		
🖋 Parameter Settings 🛛 🎻 User Constants	🦪 DMA Settings			
Interrupt Table	Enabled	Preemption Priority	Sub Priority	
TIM3 global interrupt		0	0	

Zrzut 1.3 Włączenie przerwania od TIM3

Aby uruchomić Timer 3 oraz PWM należy dopisać funkcje, w wygenerowanym kodzie w języku C, w kompilatorze IAR przedstawione poniżej:

HAL_TIM_Base_Start_IT(&htim3);	//Start	TIM3	
HAL_TIM_PWM_Start_IT(&htim4, TIM_CHANNEL_2);	//Start	PWM,	kanał2

ZADANIA PRAKTYCZNE

Należy przeanalizować pracę mikrokontrolera poprzez analizę kodu dostarczonego przez prowadzącego. Należy uruchomić kompilator IAR dla ARM, następnie nagrać program na płytkę ewaluacyjną za pomocą kabla USB. Zadaniem jest zaobserwowanie przebiegu programu, ustawienie pracy krokowej w kompilatorze, oraz uruchomienie podglądu zmiennej n w oknie "Locals". Po zatrzymaniu programu
 , a następnie jego wznowieniu
 można zaobserwować zmianę wartości.

1.1 Proszę zmienić górną wartość zmiennej n, jaki ma to wpływ na świecenie diody?

- 2. W programie STM32CubeMX proszę skonfigurować pin obsługujący diodę LED4. Należy sprawdzić na schemacie płytki ewaluacyjnej, do którego portu podpięta jest dioda i następnie odpowiednio skonfigurować ten port, tak aby dioda świeciła się światłem ciągłym. Wygenerowany kod przez Cube, należy otworzyć w środowisku IAR, przekompilować i nagrać na płytkę.
 - 2.1 W programie STM32CubeMX proszę skonfigurować odpowiednio porty obsługujące dwie diody LED oraz przycisk USER. Należy sprawdzić na schemacie płytki ewaluacyjnej, do którego portu są one podpięte, następnie odpowiednio skonfigurować te porty. Wygenerowany kod należy przez Cube, należy otworzyć w środowisku IAR, przekompilować i nagrać na płytkę. W środowisku IAR należy dopisać kod, tak aby po naciśnięciu przycisku, pojawiała się jakaś zmiana na diodach (według uznania). Przycisk jest konfigurowany podobnie jak dioda, z tą różnicą, że jest wejściem, nie wyjściem.
- Proszę skonfigurować Timer 3 do wygenerowania przerwania o dowolnym okresie. Proszę nagrać program na płytkę i w podglądzie zmiennych w IAR dla rejestrów (Register) sprawdzić działanie Timera. Proszę wskazać rejestr będący licznikiem.
 - 3.1 Proszę skonfigurować Timer 3 do wygenerowania przerwania o okresie 3000 cykli zegarowych. Proszę w przerwaniu inkrementować dowolną zmienną "czas" typu short. Proszę nagrać program na płytkę i w podglądzie zmiennych w IAR dla rejestrów (Register) sprawdzić działanie Timera poprzez sprawdzenie, czy jego licznik jest inkrementowany. Następnie proszę sprawdzić, czy program wchodzi w przerwanie, albo przez ustawienie breakpointa, albo przez podgląd zmiennej "czas", w podglądzie zmiennych.