

Bootloader

Autor prezentacji: Kamil Romanowski

WETI, Gdańsk, 2020

1



### Literatura

- Dokument AN73854 "PSoC Introduction to Bootloaders"
- Nota aplikacyjna komponentu "Bootloader and Bootloadable 1.50"
- Przykład aplikacyjny "USBFS Bootloader and Bootloadable Code Example 3.0"



## Plan prezentacji

- Czym jest Bootloader
- Zastosowanie
- Cykl życia Bootloader'a
- Bootloader w PSoC Creator
- Zabezpieczenie
- Bootloader Host
- Demo



# Czym jest Bootloader

Program uruchamiany (najczęściej) po restarcie systemu, zarządza procesem aktualizowania pamięci flash MCU nowym kodem aplikacji i/lub danych.





#### Zastosowanie

Wykorzystanie istniejącego portu łączącego produkt ze światem zewnętrznym, np. I2C, USB, UART lub inny.

Brak dodatkowych złącz czy brak programatora to mniejsze cena produktu, kosztem jest zużycie dostępnej pamięci MCU.





**Bootloader** 

# Cykl życia Bootloader'a

- Sprawdzenie poprawności aplikacji przed jej uruchomieniem
- Komunikacja z hostem
- Aktualizacja pamięci flash
- Przekazanie kontroli do aplikacji





### **Bootloader w PSoC Creator**

Wykorzystanie Bootloader'a w PSoC Creator wykorzystuje następujące komponenty:

- Bootloader dla głównego projektu z komponentem komunikacyjnym
- Bootloadable dla aplikacji wgrywanej przez Bootloader





# Typy projektów

Components Present in Project	Resultant Project Type		
None	Normal.		
	This project type is not used for bootloading and is not a bootloadable application.		
Bootloader and communication	Bootloader		
	Dual-application bootloader		
Bootloader	Launcher		
Bootloadable	Bootloadable		
Bootloader, communication, and Bootloadable	Combination		



#### **Komponent Bootloader**

Configure 'Bootloader'	? ×
Name: Bootloader	
General Built-in	4 ۵
Options Optional command   Communication component: USBFS   Dual-application bootloader Custom interface   USBFS Optional command   Golden image support I2C	ls 🔨
✓ Auto application switching ✓ Get row checks   ✓ Copier ✓ Verify application   ✓ Wait for command ✓ Send data   ✓ Wait for command time (ms): 10000   (0: wait forever) ✓ Get application	sum on checksum er
Bootloader application version: 0x0000 Get metadata	
Packet checksum type: Basic summation   □ Fast bootloadable application validation   ☑ Bootloader application validation   □ Security key: 0x 11 22 33 44 55 66	~
Datasheet OK Ap	oply Cancel



#### **Komponent Bootloader - API**

#### Classic Bootloader use case specific functions

Function	Description
Bootloader Start	This function is called to execute the following algorithm.
Bootloader GetMetadata	Returns the value of the specified field of the metadata section.
Bootloader ValidateBootloadable	Verifies validation of the specified application.
Bootloader Exit	Schedules the specified application and performs software reset to launch it.
Bootloader Calc8BitSum	Computes the 8 bit sum for the specified data.
Bootloader InitCallback	Initializes the callback functionality.
Bootloadable Load	Updates the metadata area for the Bootloader to be started on device reset and resets the device.

#### Launcher-Combination use case specific functions

Function	Description
Bootloader Initialize	Called for in-application bootloading, to initialize bootloading.
Bootloader HostLink	Called for in-application bootloading, to process bootloader command from the host.
Bootloader GetRunningAppStatus	Returns the application number of the currently running application.
Bootloader GetActiveAppStatus	Returns the application number of the currently active application.
Bootloadable GetActiveApplication	Gets the application which will be loaded after a next reset event.
Bootloadable SetActiveApplication	Sets the application which will be loaded after a next reset event.



#### **Komponent Bootloadable**

Configure 'Bootloadable'		? ×	
Name: Bootloadable			
General Dependencies Built-in	n	4 b	
Application version:	Dx0000		
Application ID:	0×0000	Configure 'Bootloadable'	? ×
Application custom ID:	D×0000000	Name: Bootloadable	
Manual application image placement		General Dependencies Built-in	4 0
Placement address:	0×0000000		
Checksum exclude section size (bytes):	D	Bootloadable projects require a reference to the associated Bootloader project's HEX and ELF files. The HEX files extension is *.hex. The ELF files extension depends on IDE and can be *.elf, *.out, *.axf, or other.	\$
		Bootloader HEX file: \\USEES_Bootloader01.ouder\CostexM2\ABM_GCC_541\Debug\USEES_Bootloader01.bev	
		Bootloader ELF file:	se
		\USBFS_Bootloader01.cydsn\CortexM3\ARM_GCC_541\Debug\USBFS_Bootloader01.elf	
		Brow	se
Datasheet	ОК	Ар	
			Cancel
		Datasrieet OK Apply (	Jancel



## Zabezpieczenie pamięci

Obszar pamięci flash używany przez Bootloader powinien być chroniony, aby uniknąć jego uszkodzenia przez aplikacje.

Poziomy ochrony pamięci flash można zmienić tylko podczas operacji całkowitego kasowania tej pamięci.

	PSoC 3 and PSoC 5LP		PSoC 4	
Protection Setting	Allowed	Not Allowed	Allowed	Not allowed
Unprotected	External read and write, Internal read and write	_	External read and write, Internal read and write	_
Factory Upgrade	External write, Internal read and write	External read	NA	NA
Field Upgrade	Internal read and write	External read and write	NA	NA
Full Protection	Internal read	External read and write, Internal write	Internal read	External write, Internal write (see Note below)



## **Bootloader Host**

PSoC Creator posiada wbudowane narzędzie działające w roli hosta, służące do wgrywania nowych aplikacji.

Współpraca z interfejsami USB, UART, SPI, I2C.

🛓 Bootloader Host		_		×
File Actions Help				
🖆 🗼 BB 📎 🚳				
File: FS_Bootloadable01.cydsn\CortexM3\ARM_GCC_541\De	bug\USBFS_B	ootloadable	01.cyaco	
Ports: Filters	Active applic	ation: No	change	$\sim$
Port komunikacyjny (COM1) - UART Urządzenie szeregowe USB (COM5) - UART	Ox 00	key 00 00	00 00	00
	Baud:	9600		$\sim$
	Data Bits:	8		$\sim$
	Stop Bits:	1		$\sim$
	Parity:	None		$\sim$
Log:				
09:41:44 PM - Selected device: Urządzenie szeregowe USB (CO Generic Serial Port	M5) - UART			
 Ready				



#### Demo

- 1. Add new project example "USBFS Bootloader"
- 2. Ustawienie pinów projektu "USBFS Bootloader"
  - USBFS:Dm P15.7
  - USBFS:Dp P15.6
  - LED4 P6.3
- 3. Kompilacja i wgranie projektu na płytkę
- 4. Odłączenie przewodu USB z gniazda programatora i podłączenie do gniazda komunikacyjnego
- 5. Add new project example "USBFS Bootloadable"
- 6. Ustawienie pinów projektu "USBFS Bootloadable"
  - LED3 P6.2
- 7. Ustawienie ścieżek do pliku \*.hex oraz \*.elf projektu "USBFS Bootloader" w komponencie Bootloadable
- 8. Kompilacja projektu
- 9. Wgranie pliku aplikacji \*.cyacd narzędziem "Bootloader Host"