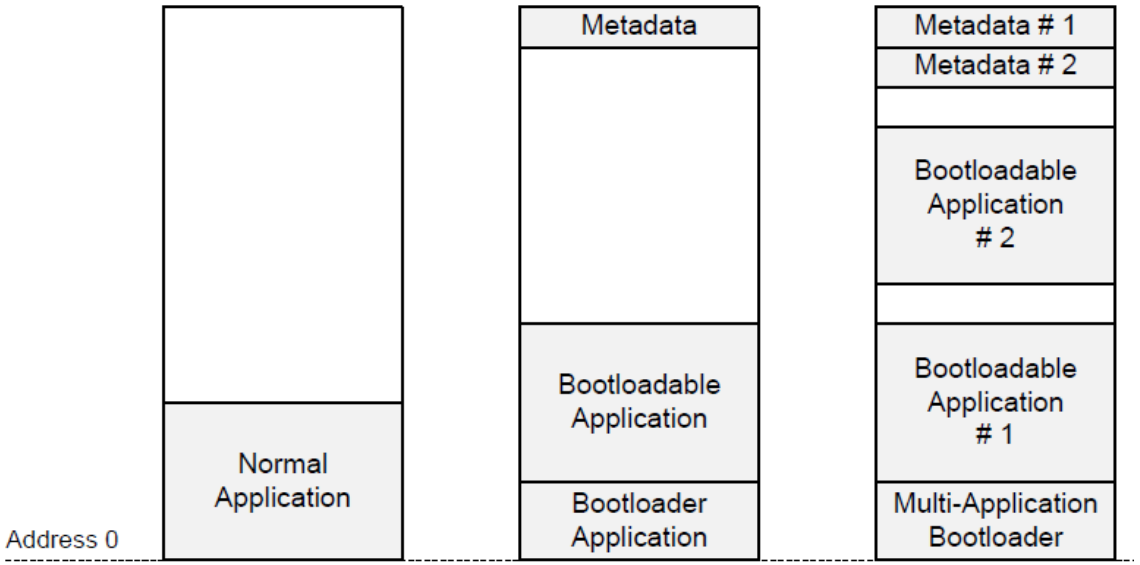
Bootloader

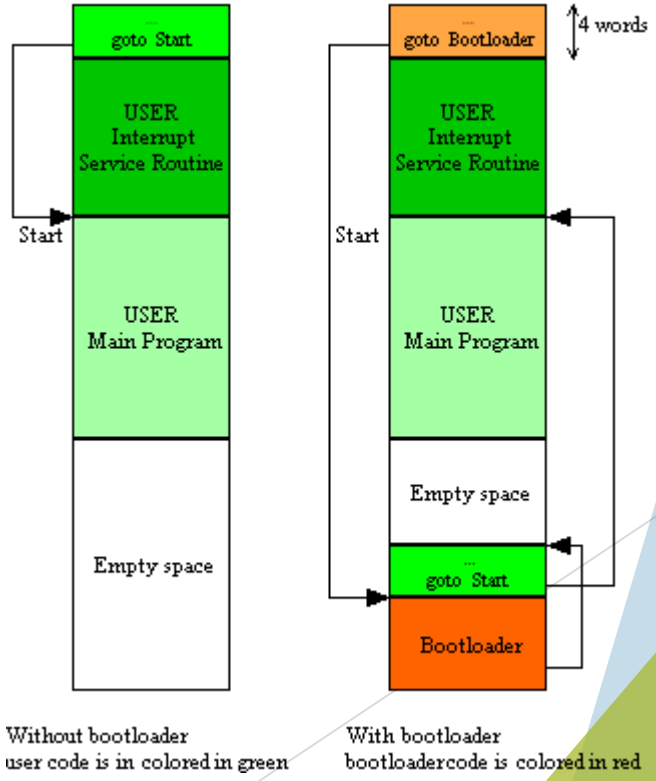
Opis ogólny i możliwości zastosowania w układach PSoC 5LP

inż. Tomasz Żywica

Definicja

- Program uruchamiany przy starcie systemu, pozwalający na załadowanie do pamięci operacyjnej (lub FLASH) właściwego programu, wykonującego funkcję urządzenia (np. system operacyjny).

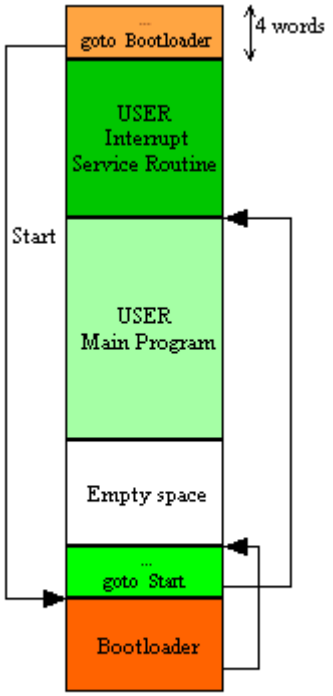




Metadane zawierają np. wielkość aplikacji, jej identyfikator, wersję bootloadera, adres końca bootloadera

Zasada działania

- Krok 1:
  - Zaraz po resecie wykonywany jest skok do miejsca w pamięci w którym zawarty jest dalszy ciąg kodu bootloadera;
- Krok 2:
  - Następuje oczekiwanie na dane (kod programu) który zostaje przysłany przez wybrany interfejs (USB, RS232…);
- Krok 3:
  - Przychodzące dane zostają wpisane w miejsce, w którym powinien znajdować się kod programu;
- Krok 4:
  - Gdy cały program został załadowany do pamięci, następuje skok do miejsca w którym się ten program zaczyna, powodując jego uruchomienie.



With bootloader
bootloadercode is colored in red

Część układów ma dedykowany obszar pamięci w którym należy umieścić bootloader

Powody stosowania bootloadera

- Wyeliminowanie potrzeby posiadania programatora
  - Oczekiwanie na dane (program) z USB/COM wysyłany przez program z poziomu PC
  - No ale bootloader trzeba jakoś umieścić w pamięci programu… ☺
- Możliwość odczytywania kodu programu z pamięci EEPROM
  - Komunikacja bootloadera z pamięcią za pośrednictwem protokołu I^2C/SPI
  - Łatwa wymiana kości (w podstawce) w przypadku uszkodzenia lub zmiany kodu programu
- Jest przydatny, ale niekoniecznie musi być użyty
  - W przypadku gdy po resecie nie podamy nowego kodu źródłowego, bootloader po zadanym czasie oczekiwania wykona skok do miejsca w pamięci, gdzie zaczyna się program użytkownika (np. załadowany poprzednio)

Jak zabezpieczyć się przed nadpisaniem fragmentów pamięci?

- Dobra praktyka – umieszczanie głównej części kodu bootloadera na końcu przestrzeni pamięci programu a na początku jedynie skoku do tej części;
  - Zdarza się, że początkowe adresy w pamięci programu są zarezerwowane i nie mogą zostać zmapowane, lub jest to rozwiązanie czasochłonne.
- Określenie konkretnego miejsca w przestrzeni adresowej od którego będzie się zaczynał program użytkownika;
  - Miejsce nie zarezerwowane przez inne składniki systemu (wektor przerwań, resetu).
- Określenie maksymalnej wielkości programu;
  - Żeby przypadkiem nie nadpisać bootloadera…

Wiele układów posiada wyznaczone miejsce w pamięci, w którym powinien znaleźć się kod bootloadera.

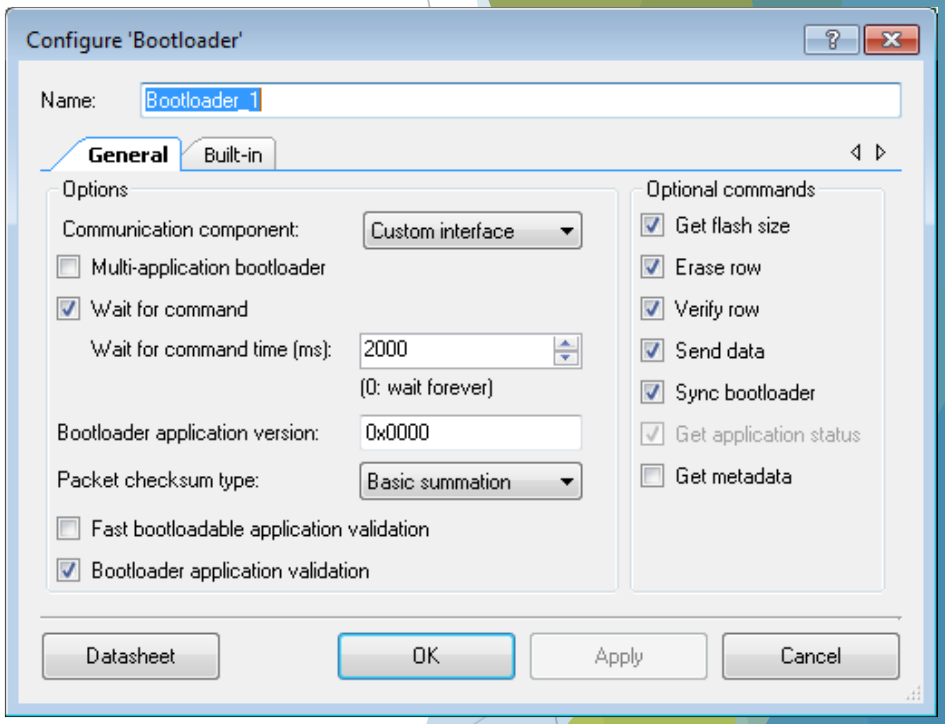
Implementacja bootloadera w układach Cypress PSoC 5LP

- Komponent Bootloader
  - Pozwala na zapisanie pamięci FLASH programem dostarczonym za pośrednictwem interfejsu;
  - Należy do projektu dołączyć komponent zapewniający obsługę odpowiedniego protokołu (USB, UART, SPI, I^2C).
- Komponent Bootloadable
  - Tworzy kod który będzie przesłany do bootloadera (można wskazać adres od którego będzie rozpoczynał się program);
  - Zbiera informacje na temat właściwości bootloadera (w jaki sposób kod ma być wysłany, gdzie w pamięci umieszczony jest bootloader itp.).

Architecture	Supported Interfaces				
	Custom Interface	USB	UART	I^2C	SPI
PSoC 3 / PSoC 5LP	✓	✓	✓	✓	✓
PSoC 4	✓			✓	

Parametry komponentu Bootloader

- Komponent komunikacyjny (musi być na schemacie)
  - Bootloader używa go do odbierania poleceń i wysyłania odpowiedzi;
- Wybór trybu bootloadera
  - Bootloader pozwala na max. 2 różne komponenty bootloadable
- Oczekiwanie na komendę
  - Gdy włączone – oczekuje na komendę od hosta przez określony czas. Jeżeli nie – od razu przechodzi do wykonywania programu bootloadable zawartego w pamięci
- Typ sumy kontrolnej przychodzących danych
- Szybka weryfikacja aplikacji
  - Gdy wyłączone – liczy sumę kontrolną za każdym uruchomieniem
- Elementy opcjonalne



Parametry komponentu Bootloadable

- Wersja aplikacji, jej identyfikator;
- Adres w pamięci do którego ma być skopiowana aplikacja (opcjnalne);
  - Przydatne gdy mamy do czynienia z 2 różnymi aplikacjami typu Bootloadable – każde będzie miało swoje stałe miejsce w pamięci;
- Ścieżki do plików HEX oraz ELF (projektowych) wygenerowanych za pośrednictwem PSoC Creator'a;
  - Komponent zbiera potrzebne informacje na temat stworzonego wcześniej bootloadera (miejsce w pamięci, użyty interfejs komunikacyjny)

- **Nasza docelowa aplikacja musi zawierać komponent bootloadable, aby można było ją zaimplementować za pośrednictwem interfejsu bootloadera (stworzonego jako oddzielny projekt odpowiedniego typu)!**

Application Programming Interface – zestaw funkcji do obsługi bootloadera

- Bootloader_Start();
  - Sprawdzenie poprawności kodu bootloadera i aplikacji bootloadable;
  - Ustawienie uruchomienia aplikacji bootloadable przy następnym uruchomieniu;
  - Reset urządzenia;
- Bootloader_GetMetadata (uint8 pole, uint8 ID);
  - Zwraca wartość np. długości programu, jego identyfikatora lub ID bootloadera;
- Bootloader_ValidateBootloadable (uint8 nr_aplikacji);
  - Sprawdza poprawność policzonej sumy kontrolnej aplikacji bootloadable z tą zawartą w sekcji metadanych – dodatkowa weryfikacja czy dane zostały przesłane pomyślnie;
- Bootloader_Exit(Bootloader_EXIT_TO_BTLDB);
  - Pozwala na uruchomienie programu bootloadable przy starcie systemu i go resetuje;
- Bootloadable_Load();
  - Uaktualnia dane zawarte w sekcji Metadata.

Oprogramowanie do testu poprawności wykonania bootloadera

- Firma Cypress dostarcza prosty program pozwalający na sprawdzenie czy bootloader został poprawnie zaimplementowany w urządzeniu;
- Podajemy ścieżkę do pliku HEX naszego programu bootloadable i wysyłamy go do urządzenia poprzez konkretny interfejs komunikacyjny;
- Ponadto, udostępnione są także kody źródłowe tego programu pozwalające na łatwiejszą jego modyfikację i stworzenie własnego programu hosta.