

Akademia ETI

26 kwietnia 2017

Michał Czyż

SKN CHIP



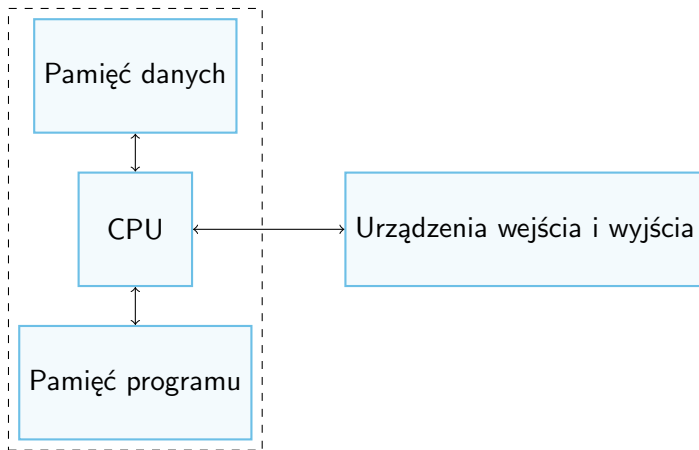
WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI I INFORMATYKI



Plan wykładu

- 1 Budowa mikroprocesora
 - Architektura harwardzka
 - Zadania CPU
- 2 Budowa mikrokontrolera
 - Mikrokontroler
 - Schemat blokowy
 - Peryferia
- 3 Logika a napięcie
 - Poziomy napięć
 - LED
- 4 Programowanie mikrokontrolerów
 - Język C
 - Główny proces
 - Główna pętla
 - Biblioteka HAL

Budowa mikroprocesora



Rysunek: Architektura Harwardzka

Zadania CPU

CPU:

- Dekoder instrukcji i logika kontroli przepływu.
- ALU.
- Rejestry adresu i danych.
- Szyna do pamięci głównej.

Przykładowe polecenie ADD R1,R2.

Układ synchroniczny

Każdy układ mikroprocesorowy jest układem **synchronicznym**. Oznacza to, że każda operacja odbywa się z kolejnym taktem zegara. Częstotliwość, czyli liczba cykli zegara na sekundę, jest najczęściej podawanym parametrem. Często wynosi ona: 2-4GHz.

Wniosek

Czas CPU jest współdzielony między procesami.

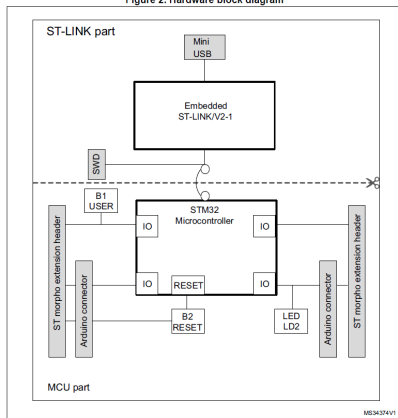
Budowa mikrokontrolera

Mikrokontroler to pojedynczy chip, w którym znajdują się:

- o mikroprocesor,
- o dodatkowa pamięć (opcjonalnie),
- o peryferia (na płytkach uruchomieniowych także zewnętrzne).

Schemat blokowy STM32L4Nucleo

Figure 2. Hardware block diagram



Peryferia

Niektóre peryferia:

- GPIO : General Purpose Input Output,
- TIM : Timer,
- ADC : Analog Digital Converter,
- I2C, USART, ..., COM Interface,
- programator.

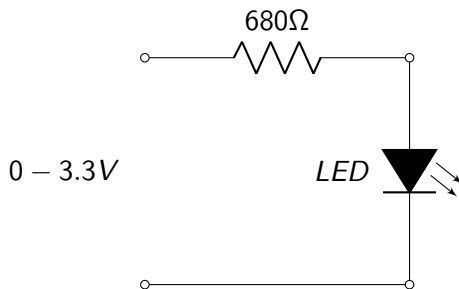
Logika a napięcie

Powszechnie stosowane poziomy napięcia: 0, 3.3, 5, 9, 12V.

W zależności od potrzeb różnym napięciom przypisuje się poziom logiczny : "0" lub "1".

Rodzina STM32L4xx wykorzystuje 0V i 3.3V lub 5V.

LED



Język C

```
int liczbaCalkowita = 5;
struct mojaStruktura {
int poleStruktury1;
//... tu wiecej pol
};
double dodaj(zmienna1 ,zmienna2);
```

Główny proces

W każdym programie istnieje funkcja main: punkt startowy.

```
int main(void) {  
    //TU WPISZ SWOJ KOD  
    return 0;  
}
```

Główna pętla

W programowaniu mikrokontrolerów najczęściej celem jest ciągle wykonywanie pewnych powtarzanych zadań. Do tego wykorzystuje się nieskończoną pętlę `while`:

```
int main(void) {  
    //INICJALIZACJA MIKROKONTROLERA  
    while(true) {  
        //ZADANIA MIKROKONTROLERA  
    }  
    return 0;  
}
```

Komendy STM HAL

```
HAL_GPIO_WritePin(GPIO_PORT, GPIO_PIN_NO,  
    BOOL);
```

```
HAL_GPIO_ReadPin(GPIO_PORT, GPIO_PIN_NO);
```

```
HAL_GPIO_TogglePin(GPIO_PORT, GPIO_PIN_NO);
```