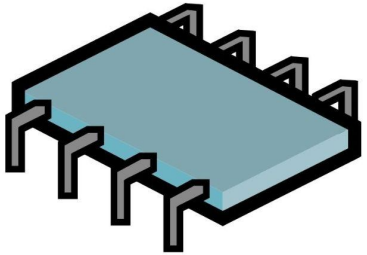
Wskaźnik i struktury

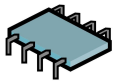
JĘZYK C
dla mikrokontrolerów



msalamon.pl

Wskaźnik i struktury

Możemy mieć wskaźnik na typ strukturalny, który utworzyliśmy



Wskaźnik i struktury

```
typedef struct {
    char Name[32];
    uint8_t Age;
    uint16_t Height;
    uint8_t Weight;
} Human_t;
```

Wskaźnik i struktury

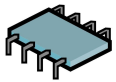
```
typedef struct {
    char Name[32];
    uint8_t Age;
    uint16_t Height;
    uint8_t Weight;
} Human_t;
```

```
Human_t Mati;
Human_t *HumanPtr;
```

Wskaźnik i struktury

```
typedef struct {
    char Name[32];
    uint8_t Age;
    uint16_t Height;
    uint8_t Weight;
} Human_t;
```

```
Human_t Mati;
Human_t *HumanPtr;

HumanPtr = &Mati;
```

Odwołanie do pól

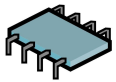
Wskaźnik i struktury - odwołanie do pól

Jeśli chcemy odwołać się do pola struktury we wskazywanych danych to będzie to nieco inaczej

Wskaźnik i struktury - odwołanie do pól

Zamiast operatora `.` musimy zrobić to przez

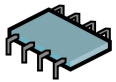
operator `->`



Wskaźnik i struktury

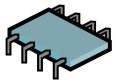
```
typedef struct {
    char Name[32];
    uint8_t Age;
    uint16_t Height;
    uint8_t Weight;
} Human_t;
```

```
Human_t Mati;
Human_t *HumanPtr;

HumanPtr = &Mati;

Mati.Age = 30;
HumanPtr->Age = 30;
```

Wskaźnik i struktury - odwołanie do pól

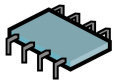
Najlepsze jest to, że jeśli spróbujemy użyć kropki to kompilator sam nam powie, że robimy to źle



Wskaźnik i struktury - odwołanie do pól

Wystawi error

i spyta się, czy chodziło nam o ->



Wskaźnik i struktury - odwołanie do pól

Więc nie popełnimy jakiegoś dziwnego błędu w odwołaniach między zmienną struct,

a wskaźnikiem na nią

Wskaźnik na strukturę

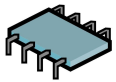
…do funkcji

Wskaźnik i struktury - do funkcji

Pamiętasz, że struktura była w argumencie kopiowana?

Wskaźnik i struktury - do funkcji

Jeśli mieliśmy sporą strukturę, to kopia była ogromna



Wskaźnik i struktury - do funkcji

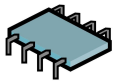
Przekazując wskaźnik (adres do) zawsze tworzony jest

na stosie wskaźnik (2, 4, 8 bajtów)

Wskaźnik i struktury - do funkcji

Więc oszczędzamy cenne miejsce w RAM

i cenny czas obsługi funkcji



Wskaźnik i struktury - do funkcji

Dodatkowo program będzie szybszy bez kopiowania danych

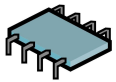
tam i z powrotem

Jedna z zasad embedded

Jedna z zasad embedded

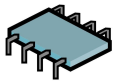
Program pisany w C z wykorzystaniem wskaźników jest

dlatego szybki i efektywny, ponieważ...



Jedna z zasad embedded

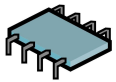
Unikamy kopiowania danych z miejsca na miejsce



Jedna z zasad embedded

Dlatego przekazując strukturę do funkcji na 99%

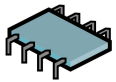
będzie to przy użyciu wskaźnika



Jedna z zasad embedded

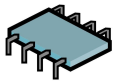
Podajemy bezpośredni adres w pamięci i

bezpośrednio na nim działamy



Jedna z zasad embedded

Jednocześnie takie pisanie jest mniej bezpieczne i jest łatwiej popełnić błąd przekazania złego wskaźnika



Tablica struktur

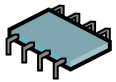
...i wskaźnik

Tablica struktur

Wróćmy jeszcze do tablicowania struktur

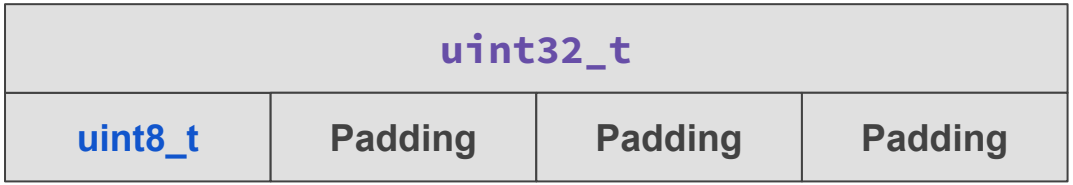
Tablica struktur

Jak układają się kolejne elementy?



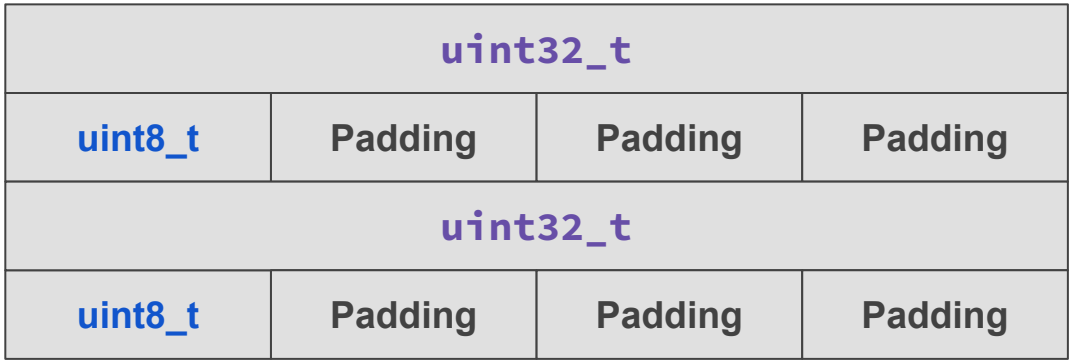
Tablica struktur

```
typedef struct {
    uint32_t AgeInDays;
    uint8_t Weight;
} Human_t;
```



Tablica struktur

```
typedef struct {
    uint32_t AgeInDays;
    uint8_t Weight;
} Human_t;

Human_t Tab[2];
```



Tablica struktur

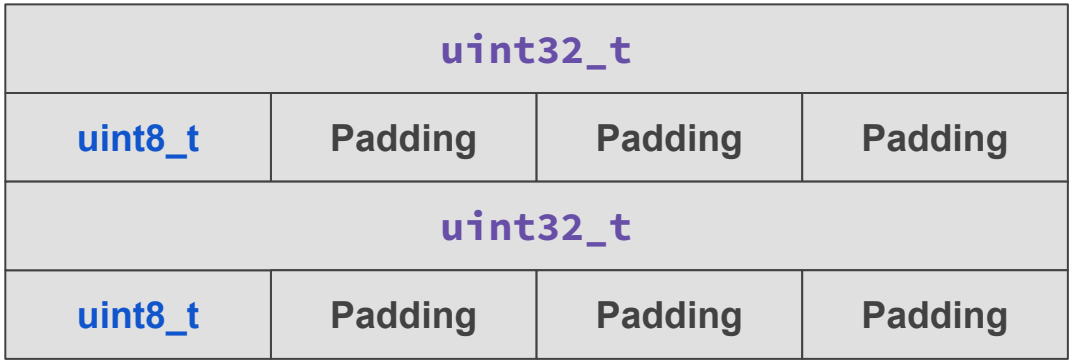
Wskaźnik będzie “skakał” o całość

łącznie z Paddingiem!

Tablica struktur

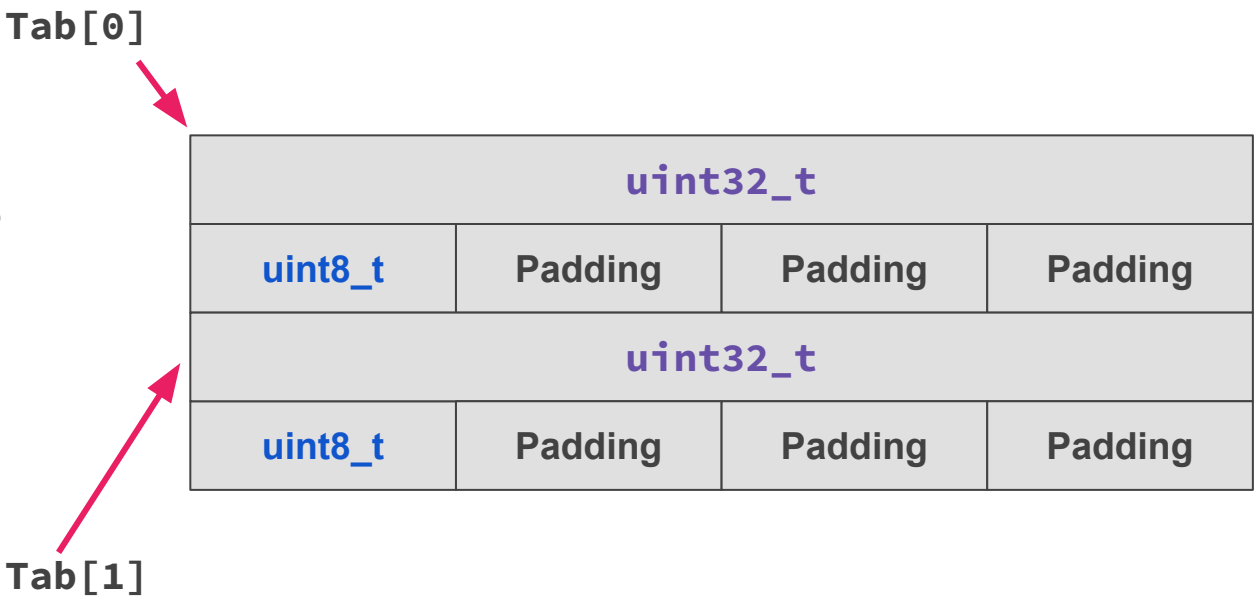
```
typedef struct {
    uint32_t AgeInDays;
    uint8_t Weight;
} Human_t;

Human_t Tab[2];
```

Tab[0]



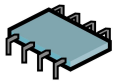
Tablica struktur

```
typedef struct {
    uint32_t AgeInDays;
    uint8_t Weight;
} Human_t;

Human_t Tab[2];
```



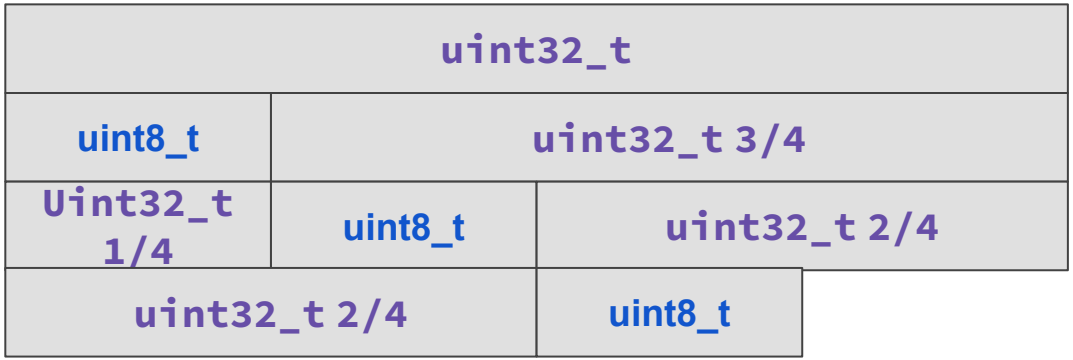
Tablica struktur

A jak spakujemy np. do jednego bajtu?



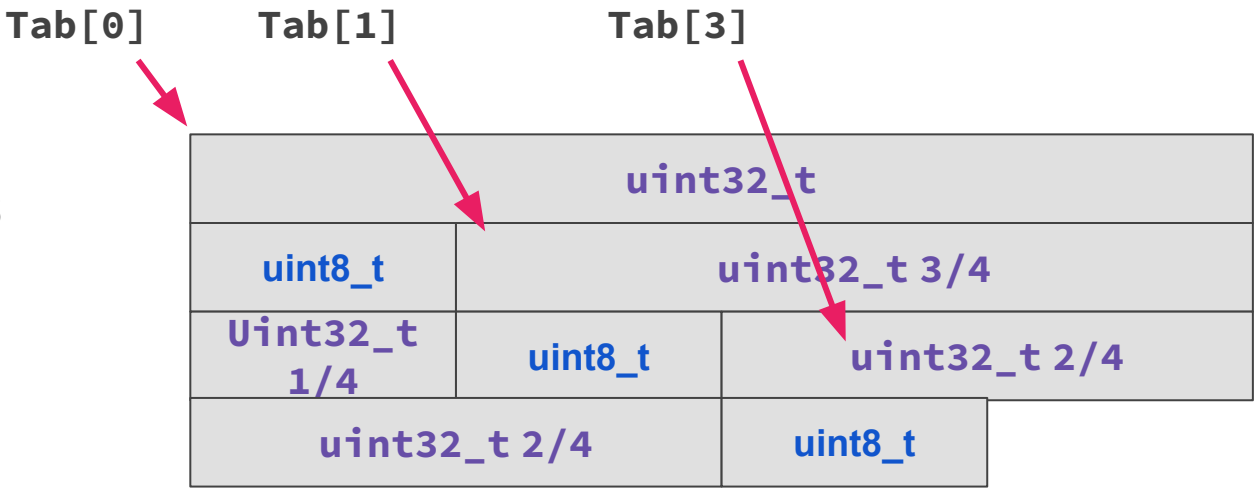
Tablica struktur

```
#pragma pack(1)
typedef struct {
    uint32_t AgeInDays;
    uint8_t Weight;
} Human_t;

Human_t Tab[3];
```



Tablica struktur

```
#pragma pack(1)
typedef struct {
    uint32_t AgeInDays;
    uint8_t Weight;
} Human_t;

Human_t Tab[3];
```

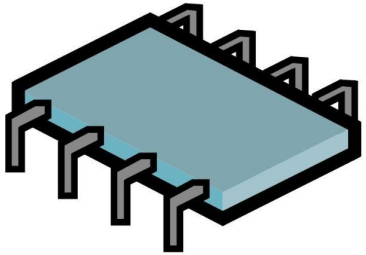


Podsumowanie

- Do pól struktury przez wskaźnik odwołujemy się przez operator ->
- Kompilator ostrzega przed błędnym operatorem odwołania do pola
- Przekazywanie struktur do funkcji będzie głównie przez wskaźnik
- Tablicowanie struktur i “skok” do kolejnego elementu uwzględnia padding

Dzięki!

JĘZYK C
dla mikrokontrolerów



msalamon.pl