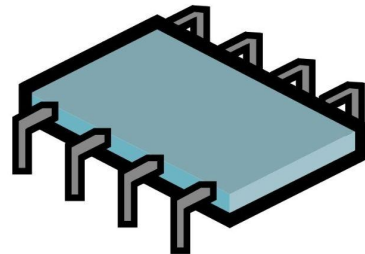


Wskaźnik w argumencie funkcji

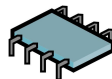
JĘZYK C
dla mikrokontrolerów



Przekazanie przez wskaźnik

— — —

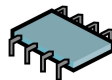
W końcu przechodzimy do prawdziwych zastosowań
wskaźników



Przekazanie przez wskaźnik

— — —

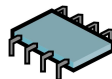
Pora na drugi sposób przekazywania danych do funkcji



Przekazanie przez wskaźnik

— — —

1. Przekazywanie przez wartość
2. **Przekazywanie przez wskaźnik**



Przekazanie przez wskaźnik



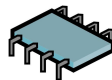
Przekazanie przez wskaźnik

```
void Fun(uint8_t *Ptr)

{

    // fun

}
```



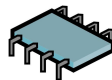
Przekazanie przez wskaźnik

```
void Fun(uint8_t *Ptr)

{

    // fun

}
```



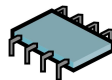
Co teraz?



Przekazanie przez wskaźnik

— — —

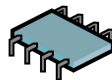
Skoro mamy w środku funkcji adres do oryginalnej zmiennej, to możemy zmienić jej wartość!



Przekazanie przez wskaźnik

— — —

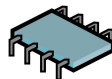
Posługujemy się nazwą z argumentu, a nie oryginalną



Przekazanie przez wskaźnik

— — —

Skoro to jest wskaźnik, to obowiązują go jego zasady



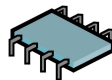
Przekazanie przez wskaźnik

```
void Fun(uint8_t *Ptr)

{

    *Ptr = 5;

}
```



Jak tam wsadzić
zmienną?

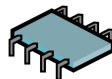


Przekazanie przez wskaźnik

— — —

Funkcja spodziewa się wskaźnika do jakichś danych.

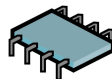
Masz 2 opcje.



Przekazanie przez wskaźnik

— — —

1. Przekazujesz już gotowy wskaźnik pokazujący na odpowiednie dane



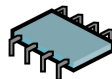
Przekazanie przez wskaźnik

```
void Fun(uint8_t *Ptr)
{
    (*Ptr)++;
}
```

```
uint8_t Var = 5;
uint8_t *Ptr;
```

```
Ptr = &Var;
```

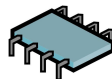
```
Fun(Ptr);
```



Przekazanie przez wskaźnik

— — —

1. Od razu ładujesz adres do tej zmiennej
 - bez pośrednika

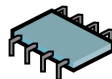


Przekazanie przez wskaźnik

```
void Fun(uint8_t *Ptr)
{
    (*Ptr)++;
}
```

```
uint8_t Var = 5;
```

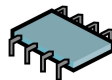
```
Fun(&Var);
```



Przekazanie przez wskaźnik

— — —

Przekazanie bezpośrednio adresu będzie zdecydowanie
częściej metodą na przekazanie danych



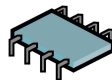
Jaką mamy tutaj
magię?



Przekazanie przez wskaźnik

— — —

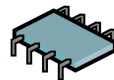
Czy kompilator widząc wskaźnik magicznie działa
inaczej?



Przekazanie przez wskaźnik

— — —

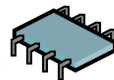
NIE



Przekazanie przez wskaźnik

— — —

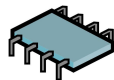
Nadal tworzona jest kopia argumentu



Przekazanie przez wskaźnik

— — —

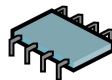
Można więc powiedzieć, że jest to tak samo przekazanie kopii wartości, ale nie zmiennej, a jej adresu



Przekazanie przez wskaźnik

— — —

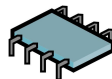
Tyle że ta kopia posiada bezpośredni adres do
oryginału



Przekazanie przez wskaźnik

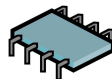
— — —

Więc operowanie na wartości spod tego wskaźnika z
argumentu...



Przekazanie przez wskaźnik

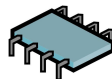
...jest tak naprawdę operowaniem na oryginalnej zmiennej pod adresem, który ten wskaźnik trzyma



Przekazanie przez wskaźnik

— — —

Trenowaliśmy to, prawda? Dwa wskaźniki pokazujące na jedną zmienną. To się tutaj dzieje.



Przekazanie przez wskaźnik

— — —

`uint8_t Value`

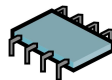
A: 0x1111

0x25

`uint8_t *Ptr`

A: 0x2222

0x1111



Przekazanie przez wskaźnik

— — —

uint8_t Value

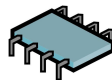
A: 0x1111

0x25

uint8_t *Ptr

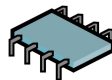
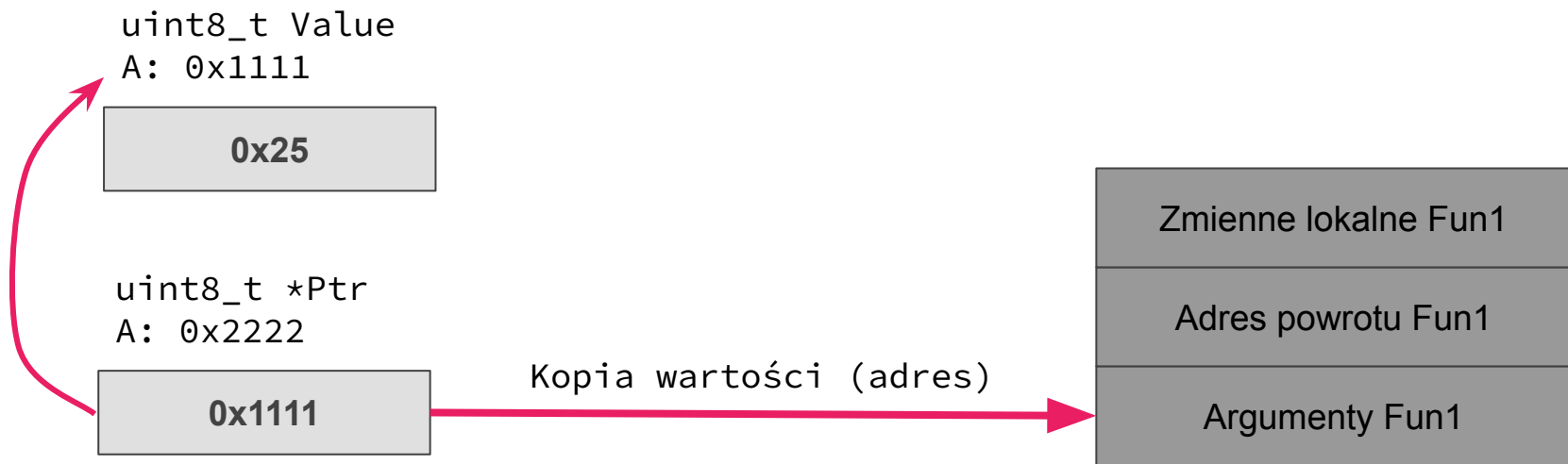
A: 0x2222

0x1111



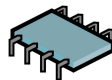
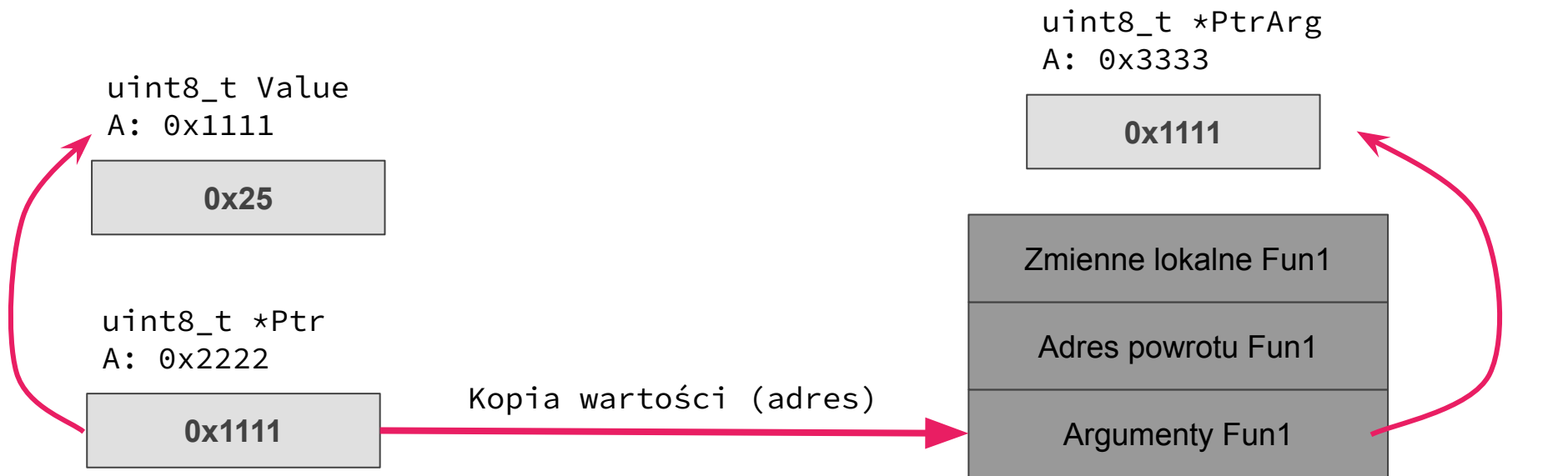
Przekazanie przez wskaźnik

— — —



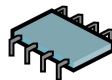
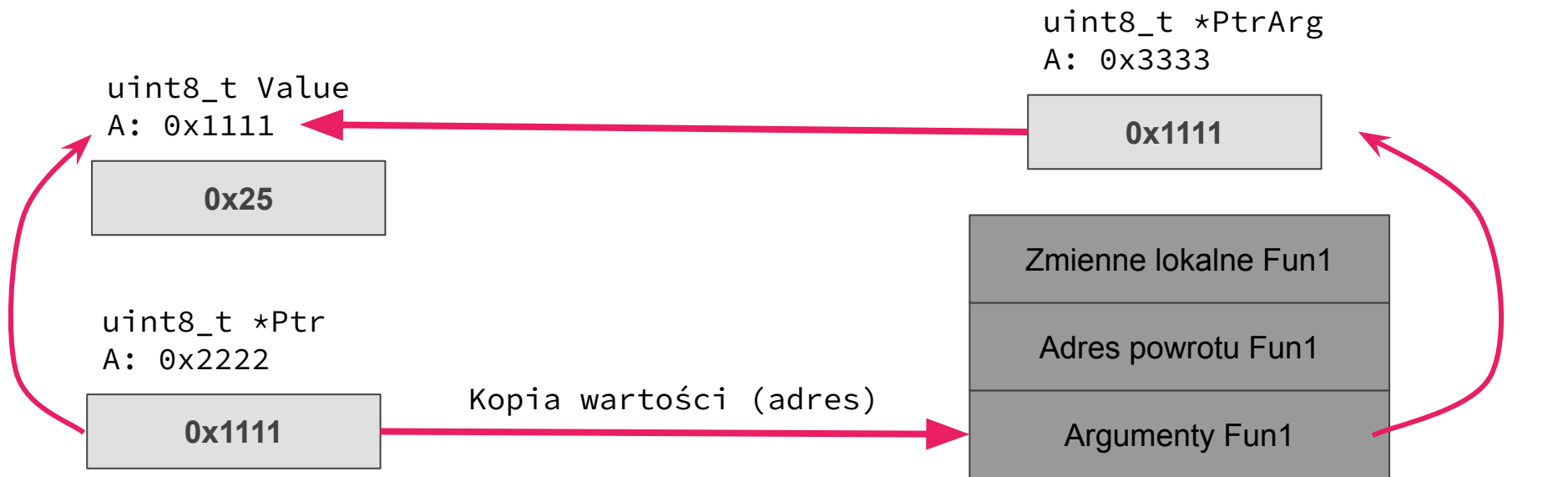
Przekazanie przez wskaźnik

— — —



Przekazanie przez wskaźnik

— — —



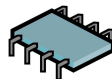
Przekazanie dalej



Przekazanie dalej

— — —

Wskaźnik (adres do danych) można przekazywać głębiej
do kolejnych funkcji



Przekazanie dalej

```
void Fun(uint8_t *Ptr)
```

```
{
```

```
    (*Ptr)++;
```

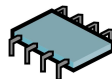
```
}
```

```
void Fun2(uint8_t *Ptr)
```

```
{
```

```
    Fun(Ptr);
```

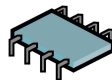
```
}
```



Przekazanie dalej

— — —

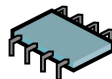
Przy przekazaniu wartości spod wskaźnika pamiętamy o gwiazdce!



Przekazanie dalej

```
void Fun(uint8_t Val)
{
    printf("%d", Val);
}
```

```
void Fun2(uint8_t *Ptr)
{
    Fun(*Ptr);
}
```



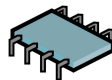
Wiele
“returnów”



Wiele “returnów”

— — —

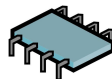
Teraz dzięki przekazywaniu przez wskaźnik możemy dowolnie modyfikować dane podawane w argumentach



Wiele “returnów”

— — —

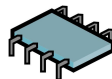
Możemy “zwracać” wartości obliczeń przez argumenty



Wiele “returnów”

— — —

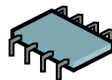
Dowolną ilość :)



Wiele “returnów”

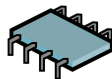
— — —

Jednocześnie zwracać status powodzenia funkcji przez
return



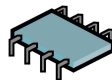
Wiele “returnów”

```
Status_t Fun(uint8_t *DataOut, uint8_t *DataOut2)
{
    (*DataOut)++;
    (*DataOut2)--;
    return STATUS_OK;
}
```



Podsumowanie

- Przekazanie przez wskaźnik umożliwia modyfikację argumentów
- Nadal tworzone są kopie, ale adresu do oryginałów
- Możemy zwracać z funkcji wiele wyników



Dzięki!

JĘZYK C
dla mikrokontrolerów

