

Laboratorium 1

I. Sposoby zapisu liczb w języku C:

unsigned char x = 15; // liczba w kodzie dziesiętnym
unsigned char x = 0b00001111; // liczba w kodzie binarnym
unsigned char x = 0x0f; // liczba w kodzie heksadecymalnym

II. Wprowadzenie do programowania ATmega16 (rodzina: AVR, producent: ATMEL).

1. Definicje

Pin – można utożsamić z jednym z wyprowadzeń mikrokontrolera - małe, metalowe, wystaje z obudowy; potocznie nazywany nóżką;

Port – grupa pinów, zwykle 8; nadane im są oznaczenia literowe (A, B, C, D); w ramach portu piny są rozróżniane na podstawie numerów (0-7).

Piny w mikrokontrolerze mogą pracować w wielu trybach, jednym z nich jest tryb ogólnego przeznaczenia: wejście/wyjście (I/O). Ponadto piny mogą pełnić funkcje specjalne, ale takie zostaną przedstawione później.

Do obsługi pinów w danym porcie służą trzy rejestry 8-bitowe:

DDRx – rejestr ten ustala kierunek pracy poszczególnych pinów w ramach portu, każdy bit odpowiada określonemu pinowi; 1 – pin pracuje jako wyjście, 0 – pin pracuje jako wejście.

PORTx – jeżeli pin pracuje jako wyjście, wpisanie „0” do odpowiadającego mu bitu w rejestrze PORTx spowoduje ustawienie stanu niskiego na odpowiednim pinie, natomiast wpis „1” powoduje ustawienie stanu wysokiego. Odczyt tego rejestru powoduje odczytanie poprzednio ustawionych, a nie stanu pinów.

PINx – jeżeli pin pracuje jako wejście, odczyt odpowiadającego mu bitu w rejestrze powoduje sprawdzenie stanu na fizycznym wyprowadzeniu mikrokontrolera.

x – należy zastąpić literą reprezentującą dany port

2. Przykład dla portu D

Ustawienie kierunku portów:

DDRD = 0x27;

Nr bitu	7	6	5	4	3	2	1	0
DDRD	0	0	1	0	0	1	1	1
Tryb	Wejście	Wejście	Wyjście	Wejście	Wejście	Wyjście	Wyjście	Wyjście

Ustawienie stanu pinów na wyjściu:

PORTD = 0xe5;

Nr bitu	7	6	5	4	3	2	1	0
PORTD	1	1	1	0	0	1	0	1
Stan	Wysoki	Wysoki	Wysoki	Niski	Niski	Wysoki	Niski	Wysoki

Odczyt stanu pinów:
 $x = \text{PIND};$

Nr bitu	7	6	5	4	3	2	1	0
Stan	Wysoki	Niski	Niski	Wysoki	Niski	Wysoki	Niski	Wysoki
PIND	1	0	0	1	0	1	0	1

III. Operacje bitowe

Operacje na poszczególnych bitach bajtów; nie należy ich mylić z operacjami logicznymi, stosowanymi w funkcjach warunkowych oraz pętlach.

Operacja binarna	Operacja logiczna
&	&&
~	!
^	brak odpowiednika

1. Operacje na dwóch bajtach

Iloczyn binarny AND
 $\text{wynik} = x \& y;$

Bajt x	1	1	1	0	0	1	0	1
Bajt y	0	1	1	0	1	1	0	0
Wynik &	0	1	1	0	0	1	0	0

Suma binarna OR
 $\text{wynik} = x | y;$

Bajt x	1	1	1	0	0	1	0	1
Bajt y	0	1	1	0	1	1	0	0
Wynik	1	1	1	0	1	1	0	1

Alternatywa wykluczająca XOR
 $\text{wynik} = x \wedge y;$

Bajt x	1	1	1	0	0	1	0	1
Bajt y	0	1	1	0	1	1	0	0
Wynik ^	1	0	0	0	1	0	0	1

2. Operacje na jednym bajcie

Negacja NOT

wynik = $\sim x$;

Bajt x	1	1	1	0	0	1	0	1
Wynik $\sim$	0	0	0	1	1	0	1	0

Częsty błąd:

wynik = !x;

Bajt x	1	1	1	0	0	1	0	1
Wynik !	0	0	0	0	0	0	0	0

Bajt x	0	0	0	0	0	0	0	0
Wynik !	0	0	0	0	0	0	0	1

Przesunięcia bitowe:

- w lewo o „x” pozycji
zmienna $\ll x$
- w prawo o „x” pozycji
zmienna $\gg x$

Bity, które zostaną wypchnięte z bajtu, giną bezpowrotnie, natomiast puste miejsca są uzupełniane „0”.

wynik = (zmienna $\ll 3$);

zmienna	1	1	1	0	0	1	0	1
wynik	0	0	1	0	1	0	0	0

Suma binarna OR

wynik = (zmienna $\gg 3$);

zmienna	1	1	1	0	0	1	0	1
wynik	0	0	0	1	1	1	0	0