6. Робота з портами МК.

Порти мікроконтролера – це пристрої введення / виводу, які призначені для обміну інформацією мікроконтролера із зовнішніми пристроями. Стандартний порт МК сімейства AVR має вісім розрядів даних, тобто є паралельними. Кожному розряду відповідає вивід МК. Залежно від моделі МК може мати в своєму складі кілька портів. Кожному порту введення / виведення присвоєно буквене позначення – A, B, C, D ... Усі паралельні порти в МК рівнозначні. Залежно від того який зовнішній пристрій буде працювати з портом, він може приймати або передавати інформацію.

Крім того, кожен вивід МК може виконувати альтернативні фукції, тому необхідно виконати налаштування портів МК.

Порти класифікуються за типом сигналу:

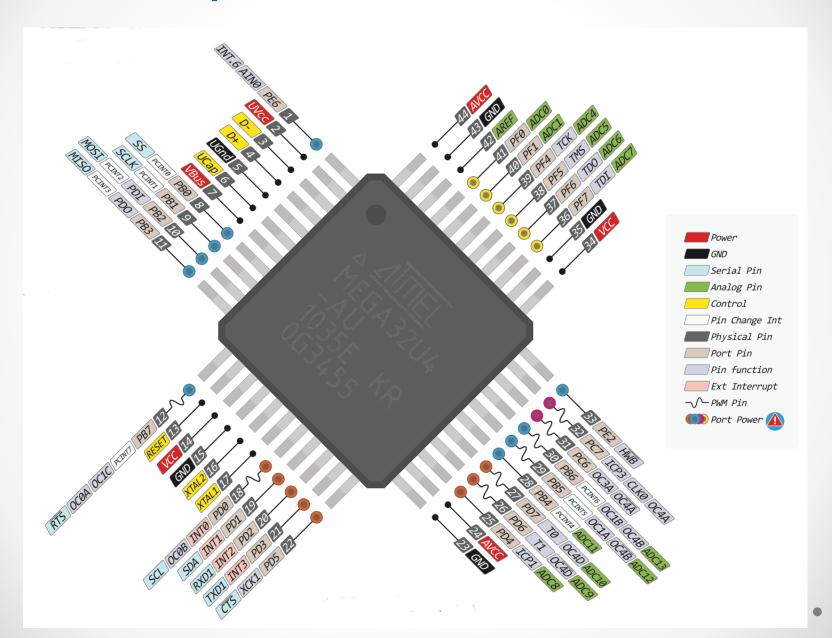
- цифрові порти (працюють з цифровими сигналами 0/1);
- аналогові порти (працюють з аналоговими сигналами, які є безперервними функціями);
- змішані порти (можуть працювати в обох режимах).

Всі порти МК AVR позначаються наступним чином:

- "Р" перша буква, що означає слово "порт";
- "А" (В, С, D, ...) друга буква, що позначає конкретний порт;
- "0" (1, 2, 3, 4, 5, 6, 7) третій символ (цифра), що позначає конкретний вивід (біт) порту.

Наприклад, у порту А п'ятий розряд буде позначений РА5.

Призначення виводів МК



В адресному просторі вводу/виводу для кожного порту відведено по 3 регістри:

- DDRx напрямку роботи (ввод/вивід);
- PORTx вихідних даних;
- PINх вхідних даних.

Наприклад, для порту A - DDRA, PORTA, PINA.

Розряди цих регістрів мають назви:

- DDx7... DDx0 для регістрів DDRx;
- Px7...Px0 для регістрів PORTx;
- PINx7... PINx0 для регістрів PINx.

Регістри PINx дозволяють здійснити доступ до фізичних значень сигналів на виводах порту. Відповідно вони доступні лише для читання. Регістри PORTx та DDRx доступні як для читання, так і для запису. Після рестарту МК в регістри PORTx та DDRx записуються початкові нулеві значення. Це відповідає режиму приймача.

Запис у порт означає запис необхідного стану для кожного виводу порту у відповідний регістр даних порту РОКТх. А читання стану порту виконується читанням або регістра даних порту РОКТх, або регістра виводів порту РІNх. При читанні регістра виводів порту РІNх відбувається зчитування логічних рівнів сигналів, що присутні на виводах порту. А при читанні регістра даних порту відбувається зчитування даних, що знаходяться у регістрі РОКТх.

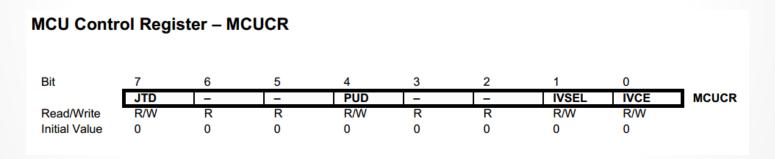
Регістри портів розташовані в сегменті регістрів вводу/виводу, для якого можливі бітові маніпуляції. Це суттєво спрощує роботу з окремими лініями портів.

Розподілення адресів регістрів (частина) приведено у таблиці:

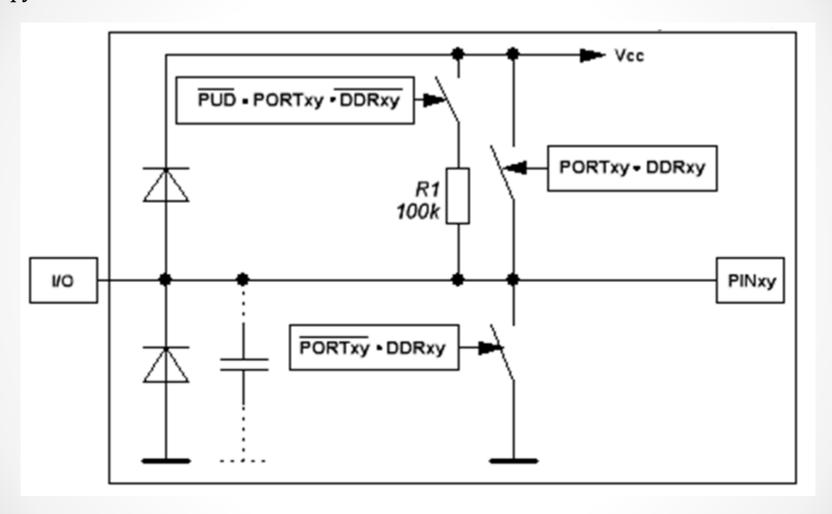
| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Page |
|-------------|----------|--------|--------|--------|--------|--------|--------|--------|--------|------|
| 0x17 (0x37) | Reserved | - | - | - | - | - | - | - | - | |
| 0x16 (0x36) | TIFR1 | - | - | ICF1 | - | OCF1C | OCF1B | OCF1A | TOV1 | |
| 0x15 (0x35) | TIFR0 | - | - | - | - | - | OCF0B | OCF0A | TOV0 | |
| 0x14 (0x34) | Reserved | - | - | - | - | - | - | - | - | |
| 0x13 (0x33) | Reserved | - | - | - | - | - | - | - | - | |
| 0x12 (0x32) | Reserved | - | - | - | - | - | - | - | - | |
| 0x11 (0x31) | PORTF | PORTF7 | PORTF6 | PORTF5 | PORTF4 | - | - | PORTF1 | PORTF0 | |
| 0x10 (0x30) | DDRF | DDF7 | DDF6 | DDF5 | DDF4 | - | - | DDF1 | DDF0 | |
| 0x0F (0x2F) | PINF | PINF7 | PINF6 | PINF5 | PINF4 | - | - | PINF1 | PINF0 | |
| 0x0E (0x2E) | PORTE | - | PORTE6 | - | - | - | PORTE2 | - | - | |
| 0x0D (0x2D) | DDRE | - | DDE6 | - | - | - | DDE2 | - | - | |
| 0x0C (0x2C) | PINE | - | PINE6 | - | - | - | PINE2 | - | - | |
| 0x0B (0x2B) | PORTD | PORTD7 | PORTD6 | PORTD5 | PORTD4 | PORTD3 | PORTD2 | PORTD1 | PORTD0 | |
| 0x0A (0x2A) | DDRD | DDD7 | DDD6 | DDD5 | DDD4 | DDD3 | DDD2 | DDD1 | DDD0 | |
| 0x09 (0x29) | PIND | PIND7 | PIND6 | PIND5 | PIND4 | PIND3 | PIND2 | PIND1 | PIND0 | |
| 0x08 (0x28) | PORTC | PORTC7 | PORTC6 | - | - | - | - | - | - | |
| 0x07 (0x27) | DDRC | DDC7 | DDC6 | - | - | - | - | - | - | |
| 0x06 (0x26) | PINC | PINC7 | PINC6 | - | - | - | - | - | - | |
| 0x05 (0x25) | PORTB | PORTB7 | PORTB6 | PORTB5 | PORTB4 | PORTB3 | PORTB2 | PORTB1 | PORTB0 | |
| 0x04 (0x24) | DDRB | DDB7 | DDB6 | DDB5 | DDB4 | DDB3 | DDB2 | DDB1 | DDB0 | |
| 0x03 (0x23) | PINB | PINB7 | PINB6 | PINB5 | PINB4 | PINB3 | PINB2 | PINB1 | PINB0 | |
| 0x02 (0x22) | Reserved | - | - | - | - | - | - | - | - | |
| 0x01 (0x21) | Reserved | - | - | - | - | - | - | - | - | |
| 0x00 (0x20) | Reserved | - | - | - | - | - | - | - | - | |

Налаштування регістрів портів здійснюється відповідно до рисунку:

| DDxn | PORTxn | PUD (in MCUCR) | I/O | Pull-up | Comment |
|------|--------|-------------------|--------|---------|--|
| 0 | 0 | Х | Input | No | Tri-state (Hi-Z) |
| 0 | 1 | 0 | Input | Yes | Pxn will source current if ext. pulled low |
| 0 | 1 | 1 | Input | No | Tri-state (Hi-Z) |
| 1 | 0 | X | Output | No | Output Low (Sink) |
| 1 | 1 | X | Output | No | Output High (Source) |

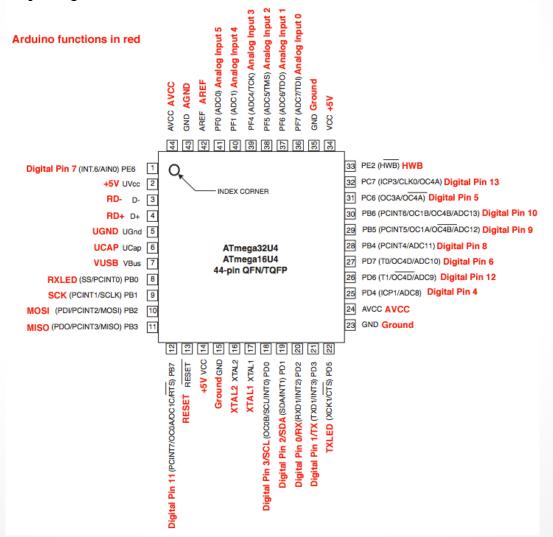


Функціональна схем одного виводу порта без урахування альтернативних функцій:



Особливість управління цифровими виводами портів з використанням функцій Arduino в тому, що конфігурування і управління здійснюється бітам порту, а не всім портом.

Взаємозв'язок нумерації виводів МК з виводами Arduino:



```
Константи:
LOW – догічний "0";
HIGH – догічна "1";
INPUT – призначити цифровий вивід Arduino на введення;
INPUT_PULLUP – призначити цифровий вивід Arduino на введення з підтягуючим резистором;
OUTPUT – призначити цифровий вивід Arduino на виведення;
```

Функції:

```
pinMode (pin, mode) – конфігурує режим роботи вказаного виводу; digitalWrite (pin, value) – відправляє на цифровий вивід значення; digitalRead (pin) – зчитує значення з зазначеного цифрового виводу; де pin – номер виводу, mode – може приймати значення INPUT, OUTPUT або INPUT_PULLUP, value – значення на цифровому виводі HIGH або LOW.
```

Переваги:

- 1. Універсальність;
- 2. Інтуїтивність.

Недоліки:

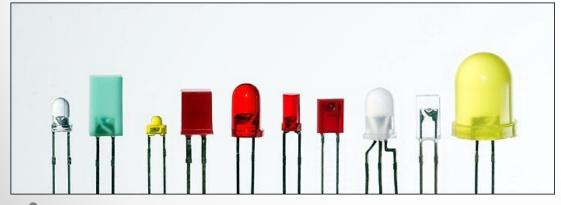
- 1. Низька швидкодія;
- 2. Додаткові витрати ресурсів пам'яті.

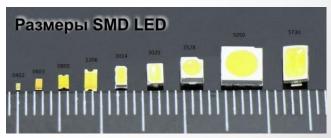
Підключення пристроїв введення/виводу інформації

Світлодіод

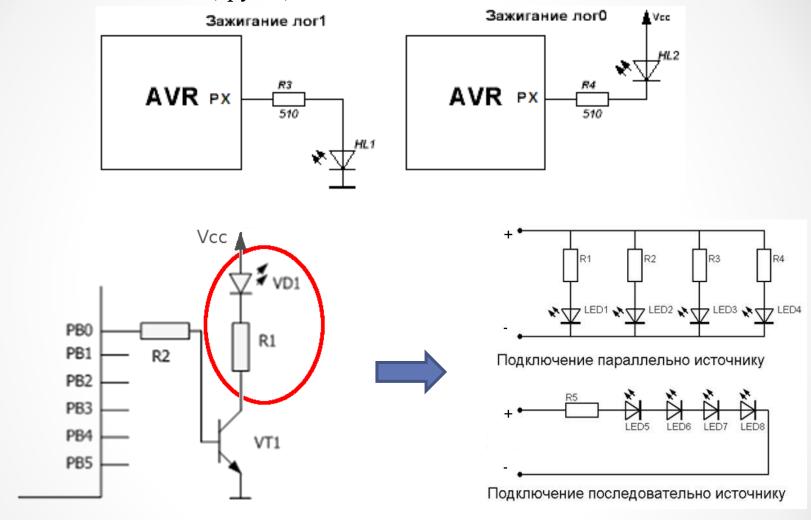
Світлодіод – це пристрій, який являє собою напівпровідниковий прилад, здатний випромінювати світло при пропусканні через нього електричного струму в прямому напрямку (від анода до катода).



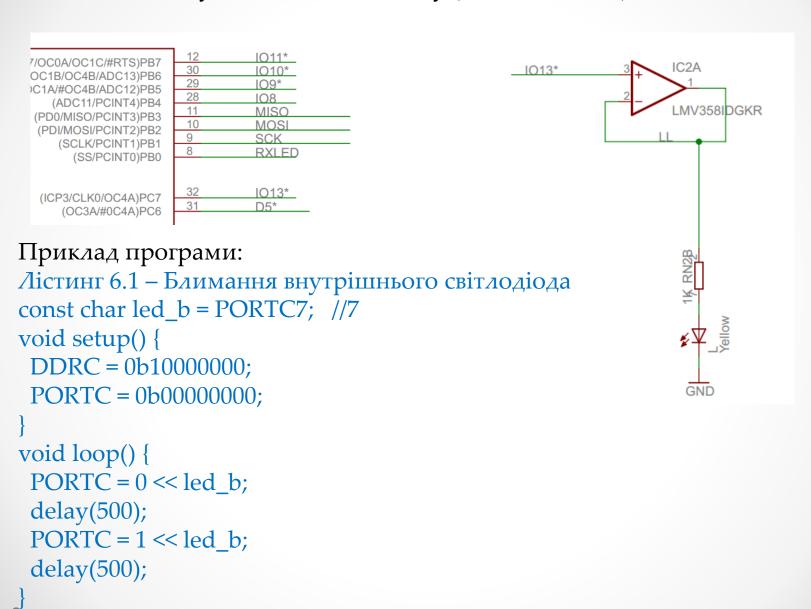




Підключення одного (групи) світлодіодів до МК:



Підключення вбудованого світлодіоду (частина схеми):



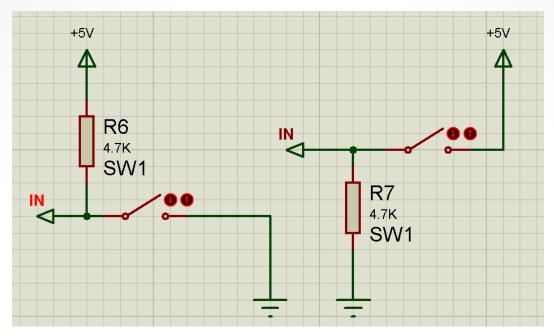
Перемикачі:

Перемикач – це електромеханічний пристрій для з'єднання і роз'єднання електричних ланцюгів. Існують різні перемикачі, але типовий механічний перемикач складається з клем, які можна замикати між собою механічно.

| Кнопочный переключатель | Тумблерный переключатель | Клавишный переключатель | Микро- переключатель | DIL- переключатель |
|--|-----------------------------|----------------------------|-------------------------|-----------------------|
| | 3 | | | |
| \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\ | | | | 9 9 |

Кнопочний перемикач (кнопка) – механічний пристрій для передачі сигналу (введення інформації), елемент інтерфейсу людина-машина: елементарний фізичний механізм передачі електричного сигналу різних пристроїв шляхом замикання або розмикання двох або більше контактів. По суті своїй є датчиком зовнішнього фізичного впливу (зусилля натискання), що передає далі факт фізичного впливу до МК.

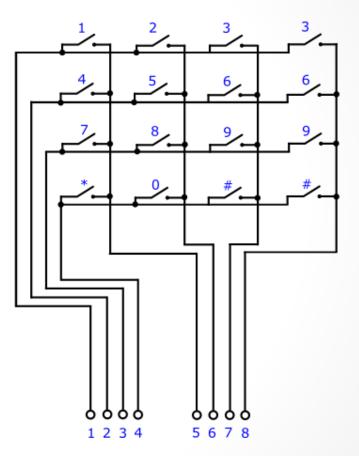
Підключення однієї кнопки до МК:



```
Приклад програми:
Лістинг 6.2 – Робота с кнопкою
char button_b = A1;
char led_b = 13;
bool buttonState = 0;
void setup() {
 pinMode(led_b, OUTPUT);
 pinMode(button_b, INPUT_PULLUP);
void loop() {
 buttonState = digitalRead(button_b);
 if (buttonState == true) {
  digitalWrite(led_b, HIGH);
 } else {
  digitalWrite(led_b, LOW);
```

Якщо необхідно підключити велику кількість кнопок (клавіатуру), то даний варіант не використовується, тому що необхідна велика кількість виводів мікроконтролера. Замість цього використовують матричне включення кнопок:





Принцип роботи такий клавіатури досить простий. МК по черзі подає логічний «0» на кожен з виводів 4х рядків, а з виводів стовпців навпаки – зчитує значення. Якщо натиснути будь-яку з кнопок, то вона замкне вивід рядка і вивід стовпчика, з якими пов'язана, в результаті чого на відповідному виводі стовпчика з'явиться логічний «0».

