

Прапорці та розширене керування кнопкою

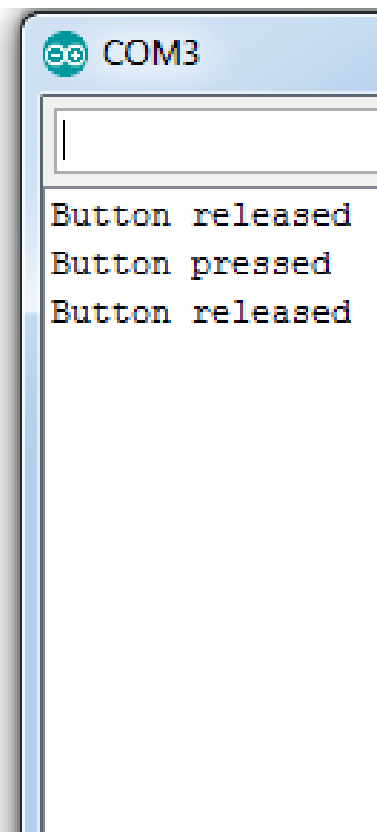
Недолік попереднього скетчу для вмикання світлодіода на 13 цифровому піні Arduino в тому, що він вмикається тільки при утриманні кнопки натиснутою. Якщо кнопку відпустити – світлодіод одразу згасне. Логічно запрограмувати кнопку так, щоб кнопка перемикала стан світлодіода – перше натискання приводило до вмикання світлодіода, наступне – до вимикання.

Цикл **loop()**, як ми пам'ятаємо з попередніх уроків, – нескінчений і виконується 1000 разів за секунду. Нам же потрібно виконати всього одну дію. Для цього введемо поняття логічної змінної – прапорця, яка зберігатиме стан об'єкта. Назвемо її **butt_flag** та присвоїмо їй початкове значення нуль. Кнопку підключимо на 5-й пін, підтягнувши його до землі за допомогою **INPUT_PULLUP**. Змінна **butt** буде приймати поточне значення кнопки. У зв'язку з використанням підтяжки **INPUT_PULLUP**, інвертуємо значення піну 5 на протилежне. Таким чином, якщо кнопка натиснута, **butt** дорівнює одиниці, якщо не натиснута, **butt** дорівнює нулю.

Створимо умову. На початку виконання коду **butt_flag = 0**. Якщо сигнал на піні дорівнює одиниці (кнопка натиснута) **butt = 1**, то перша умова виконується і прапорець змінює значення на 1-цю: **butt_flag = 1**. Коли буде виконуватися наступний цикл **loop** умова вже не виконається. Таким чином, код у фігурних дужках виконається тільки один раз, при натисканні кнопки.

При відпусканні кнопки **butt = 0**, а **butt_flag = 1**, виконається друга умова і значення прапорця знову зміниться на протилежне. Для наочності виведемо в порт значення, які будуть з'являтися при натисканні і відпусканні кнопки. Дивись скетч нижче:

```
boolean butt_flag =0;
boolean butt;
void setup() {
  pinMode(5, INPUT_PULLUP);
  Serial.begin(9600);
}
void loop() {
  butt = !digitalRead(5); //зчитати стан піну
  if (butt == 1 && butt_flag ==0) {
    butt_flag = 1;
    Serial.println("Button pressed");
  }
  if (butt == 0 && butt_flag ==1) {
    butt_flag = 0;
    Serial.println("Button released");
  }
}
```



Пам'ятка з попередніх уроків!

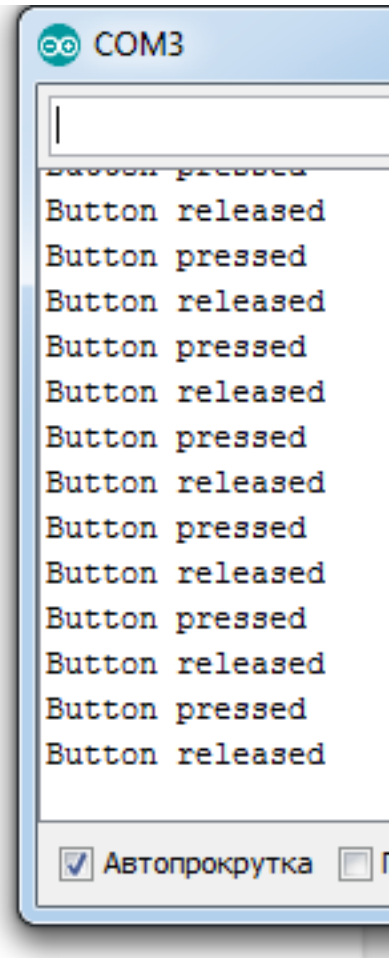
Нехай **A** і **B** – умови.

Складена умова **A || B** виконується, якщо виконується хоча б одна із умов: або умова **A**, або умова **B**. Складена умова **A && B**

виконується лише тоді, коли виконуються обидві умови: і умова **A**, і умова **B**.

Напишемо скетч, що буде вмикати і вимикати кнопкою на 5-му піні світлодіод, що підключений до 13 піну:

```
boolean butt_flag = 0;
boolean butt;
boolean led_flag = 0;
void setup() {
  pinMode(5, INPUT_PULLUP);
  Serial.begin(9600);
  pinMode(13, OUTPUT);
}
void loop() {
  butt = !digitalRead(5); //зчитати стан піну
  if (butt == 1 && butt_flag == 0) {
    butt_flag = 1;
    Serial.println("Button pressed");
    led_flag = !led_flag;
    digitalWrite(13, led_flag);
  }
  if (butt == 0 && butt_flag == 1) {
    butt_flag = 0;
    Serial.println("Button released");
  }
}
```



Натискаючи кнопку, ми побачимо небажане явище, яке називають «брязкіт контактів⁴». Воно з'являється завдяки тому, що під час натискання кнопки може кілька разів замкнутися та розімкнутися контакт. Для виключення «брязкіта контактів» доопрацюємо програму затримкою після натискання кнопки. Можна використати **delay(50)**, але використання цієї функції небажане, тому підемо іншим шляхом, використавши функцію **millis**, як це ми робили в попередніх уроках.

⁴ Небажане замикання й розмикання контактів в момент комутації

Скетч що виключає «брязкіт контактів»:

```
boolean butt_flag = 0;
boolean butt;
boolean led_flag = 0;
unsigned long last_press;
void setup() {
  pinMode(5, INPUT_PULLUP);
  Serial.begin(9600);
  pinMode(13, OUTPUT);
}
void loop() {
  butt = !digitalRead(5); //зчитати стан піну
  if (butt == 1 && butt_flag == 0 && millis() - last_press > 50) {
    butt_flag = 1;
    Serial.println("Button pressed");
    led_flag = !led_flag;
    digitalWrite(13, led_flag);
    last_press = millis();
  }
  if (butt == 0 && butt_flag == 1) {
    butt_flag = 0;
    Serial.println("Button released");
  }
}
```

А як же відпрацювати не просто натискання, а наприклад, утримання кнопки або подвійного натискання? Для цього існує бібліотека **OneButton**, яка дозволяє відпрацювати всі перераховані варіанти натискання кнопки.

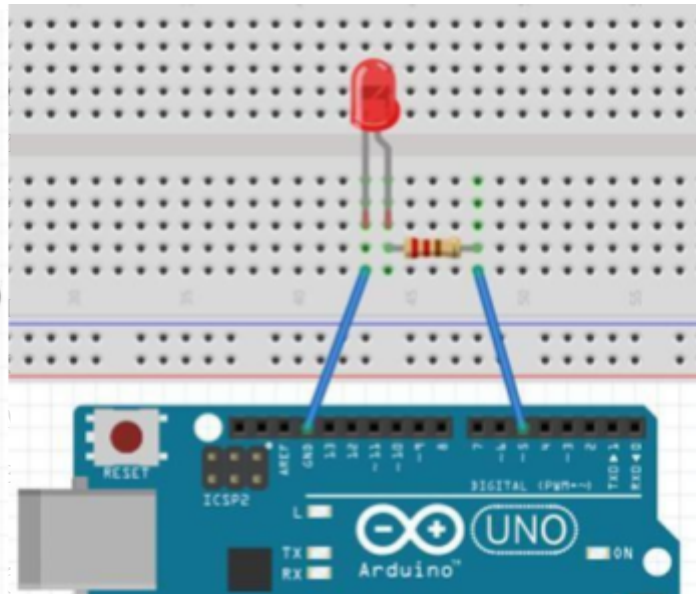
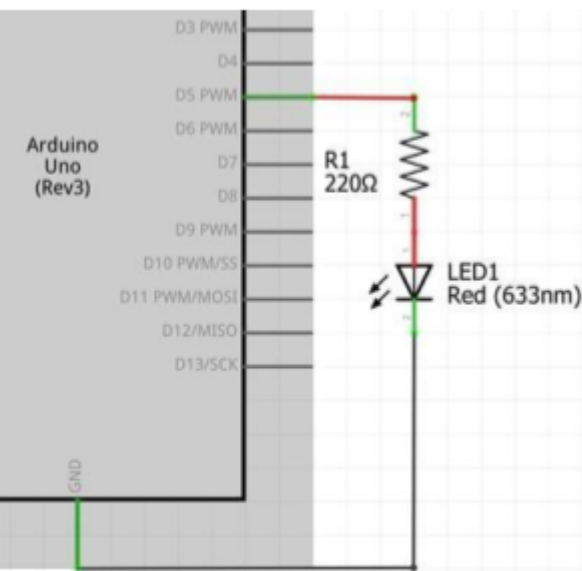
Скачати можна за посиланням:

<https://github.com/mathertel/OneButton>

Як працювати з бібліотеками ми роздивимося на 19 уроці.

Як підключити світлодіод до Arduino

Виходи Arduino служать тільки для керування іншими приладами. До плати Arduino можна безпосередньо підключати деякі елементи з споживаним струмом до 40 міліампер. Це можуть бути 2 світлодіоди, якась пищавка, датчик. Світлодіоди підключають через резистор, який обмежує струм.



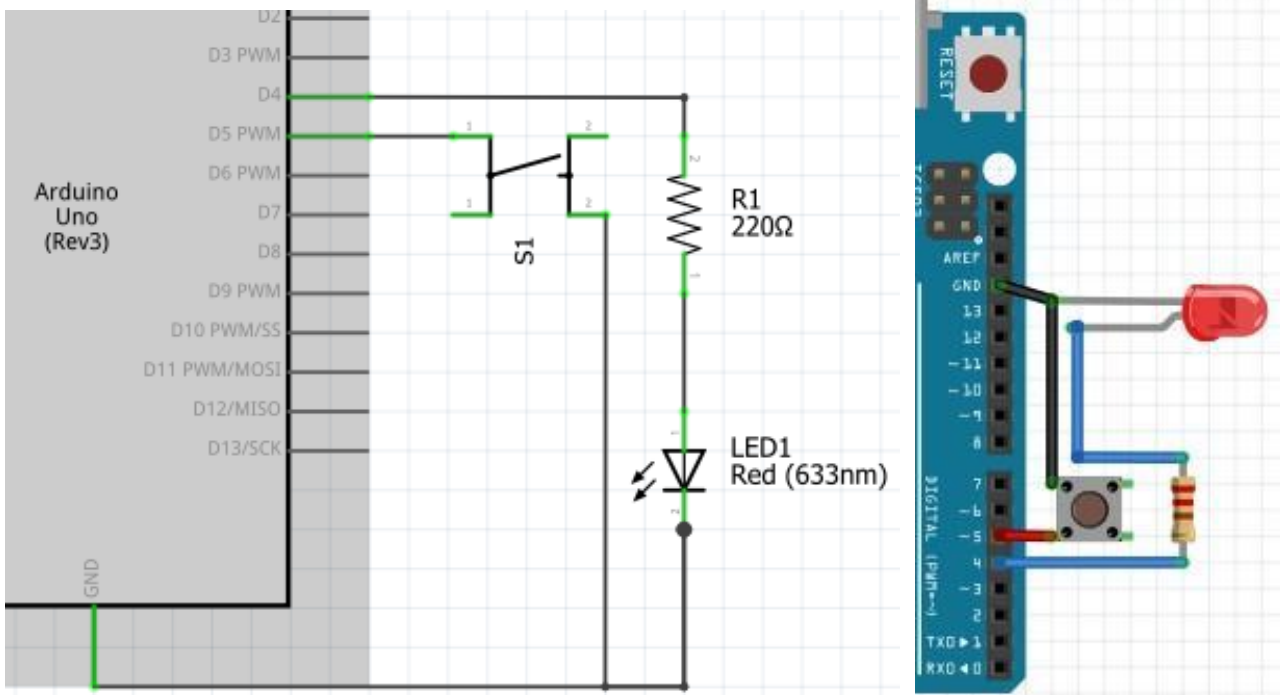
деякими
живлення
мій платі
д-якому
итися на
пруги на
ні:
руги на
знайдемо

$$R = \frac{V_{дж} - V_R}{I_{сп}} = \frac{5 - 3}{0,02} = 100 \text{ Ом}$$

Це буде R для максимального значення струму, що витримає світлодіод. Для номінального значення можемо збільшити значення R до 220 Ом. Для того, щоб зібрати макетну схему, потрібно ще знати, що світлодіод має полярність – анод та катод. Звичайно, мінусовий вихід

(катод) має меншу довжину. Це обов'язково треба приймати до уваги, збираючи схему.

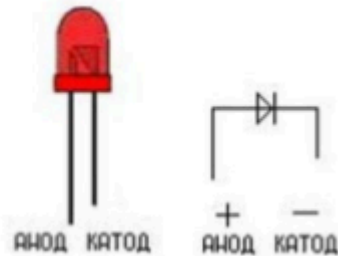
Вивчивши теорію, спробуємо на основі теорії попереднього уроку створити схему, де зовнішній світлодіод буде вмикатися та вимикатися КНОПКОЮ:



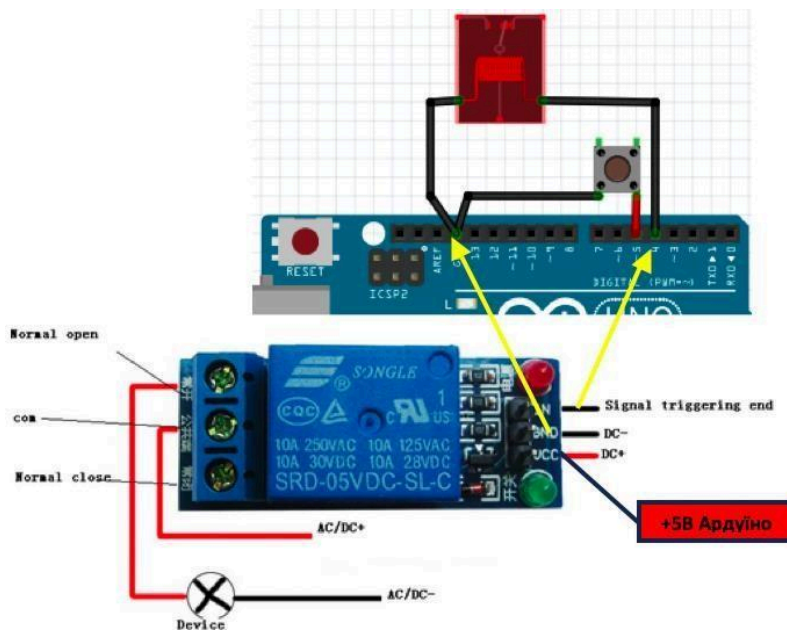
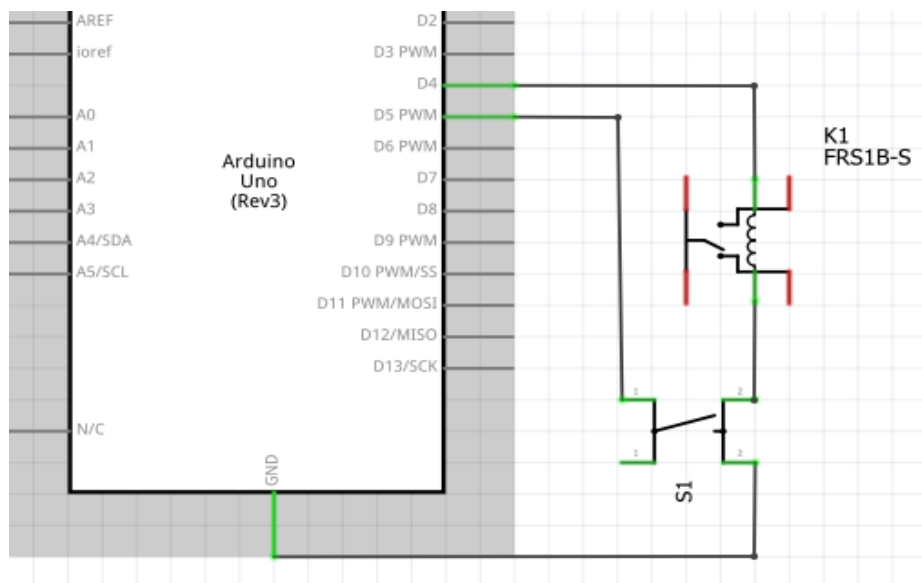
```
boolean butt_flag = 0;
boolean butt;
boolean led_flag = 0;
unsigned long last_press;
void setup() {
  pinMode(5, INPUT_PULLUP);
  Serial.begin(9600);
  pinMode(4, OUTPUT);
}
```

```
void loop() {
  butt = !digitalRead(5); // считать текущее положение кнопки

  if (butt == 1 && butt_flag == 0 && millis() - last_press > 100) {
    butt_flag = 1;
    Serial.println("Button pressed");
    led_flag = !led_flag;
    digitalWrite(4, led_flag);
    last_press = millis();
  }
  if (butt == 0 && butt_flag == 1) {
    butt_flag = 0;
    Serial.println("Button released");
  }
}
```



А тепер спробуємо замість світлодіода підключити РЕЛЕ, яке буде перемикати контакти і вмикати споживача з великим навантаженням, наприклад, лампу або бойлер у системі «Розумний дім»:



На схемі вище відсутнє підключення входу реле DC+. Його треба підключити до +5В на платі Arduino. До силових клем можна підключати велике навантаження, величина якого обмежена параметрами реле. Для нашого варіанту – це 10А та 250В, або 2 кВт. Але краще використовувати навантаження вдвічі менше за максимальне – тоді реле служитиме довго.

Скетч підходить від попереднього завдання, просто замість світлодіода з резистором на ті ж піни потрібно причепити реле. Можна звичайно замінити деякі назви – замість led прописати relay. Але це не обов'язково.

