

necha millisekund davomida bir holatdan boshqasiga bir necha marotaba o'tadi, so'ng HIGH signal qiymati o'rnatiladi. Mikrokontroller bu holatlarning barchasini qayd qilib oladi, chunki titrashda hosil bo'ladigan signal bosishda hosil bo'ladigan signaldan farq qilmaydi. Tutashning ta'sirini dastur orqali bartaraf etish mumkin. Uning algoritmi quyidagicha:

1. Tugmaning oldingi holatini va hozirgi holati saqlanadi (LOW initsializatsiyalashda).
2. Tugmaning hozirdagi holatini o'qiladi.
3. Agarda tugmaning hozirdagi holati oldindagi holatidan farq qilsa, 5 ms kutiladi, chunki tugma holatini o'zgartirgan bo'lishi mumkin.
4. 5 ms dan so'ng tugma holatini o'qiladi va uni hozirdagi holati sifatida ishlatiladi.
5. Agarda tugmaning oldingi holati LOW bo'lgan bo'lsa, hozirdagi holati HIGH bo'lsa, u holda yorug'lik diodining holatini o'zgartiriladi.
6. Tugmaning oldingi holatini hozirgi holati uchun o'rnatiladi.
7. 2 qadamga qaytish.

Sketchga titrashni bartaraf etish dasturostisini qo'shiladi.

2.3 listingda ko'rsatilgan kod olinadi.

### 2.3 listing

```
const int LED=10; // 10 ulanish nuqta yorug'lik diodini ulash uchun
const int BUTTON=2; // 2 ulanish nuqta tugmani ulash uchun
int tekButton = LOW; // Tugmani hozirgi holatini saqlash uchun o'zgartuvchi
int prevButton = LOW; // Tugmani oldingi holatini saqlash uchun o'zgartuvchi
// tugmalar
boolean ledOn = false; // Yorug'lik diodining hozirgi holati (yoqiq/o'chiq)
void setup()
{
  // Yorug'lik diodining ulanish nuqtasini chiqish kabi o'zgartirish
  pinMode(LED, OUTPUT);
  // Tugmaning ulanish nuqtasini kirish kabi o'zgartirish
  pinMode(BUTTON, INPUT);
}
// Titrashni yo'q qilish funksiyasi. Tugmaning oldingi holatini
// argument sifatida qabul qiladi va haqiqiysini beradi,
boolean debounce(boolean last)
{
  boolean current = digitalRead(BUTTON); // Tugma holatini o'qish,
  if (last != current) // agarda o'zgarsa...
```