

### Раздел 3. «IT-технологии, энергетика, автоматизация и вычислительная техника»

МРНТИ 20.01.15  
УДК 681.57

[DOI: 10.53002/099](https://doi.org/10.53002/099)

Романов Д.Ю.

Карагандинский индустриальный университет, Темиртау, Казахстан  
(E-mail: d.romanov@tttu.edu.kz)

#### Метод преобразования обычной кнопки в кнопку с запоминанием состояния при помощи программирования Arduino

В статье рассматривается программный метод преобразования стандартной кнопки, используемой в проектах на платформе Arduino, в кнопку с функцией фиксации состояния. Предложенный подход позволяет реализовать запоминание предыдущего состояния кнопки без применения дополнительных аппаратных компонентов, что упрощает конструкцию устройства. Рассмотрен принцип работы алгоритма, основанного на изменении логического состояния при каждом нажатии кнопки. Представлены примеры практического применения данного решения в учебных и автоматизированных системах, таких как управление освещением, моделирование пульта управления и разработка прототипов устройств. Метод может быть использован в образовательных целях для демонстрации принципов работы логических состояний и основ программирования микроконтроллеров.

*Ключевые слова:* Arduino, кнопка, кнопка с фиксацией состояния, программирование, автоматизация, управление состоянием, электронные устройства, образование.

#### Введение

Arduino – это платформа для создания различных устройств и даже роботов. Проекты Arduino состоят из самой платы Arduino (например, в данной работе используется Arduino Uno), различных датчиков, и различных исполнительных устройств.

Arduino применяется во многих сферах – в образовании, на небольших производствах, в «умных» домах. Например, в [1] представлена система сигнализации о пожаре/дыме «умного» дома на основе Arduino и детектором дыма.

В Карагандинском индустриальном университете Arduino используется для освоения основ электроники, программирования и автоматизации. Arduino идеально подходит для того, чтобы показать студентам принципы работы автоматизированных систем – есть датчики, есть контроллер, и есть исполнительные устройства. Главное преимущество Arduino в образовательной цели – студент изучает теорию, затем сам создает миниатюрную систему автоматизации на практике, и видит, как теория и программирование превращается в устройство или робота.

Во время создания проектов Arduino возникла проблема – в университете нет кнопок с фиксацией своего состояния; то есть в наличии есть только обычные кнопки, представленные на рисунке 1.



Рисунок 1 – Кнопка (push-button)

### Раздел 3. «IT-технологии, энергетика, автоматизация и вычислительная техника»

Внутренняя структура такой кнопки представлена на рисунке 2.

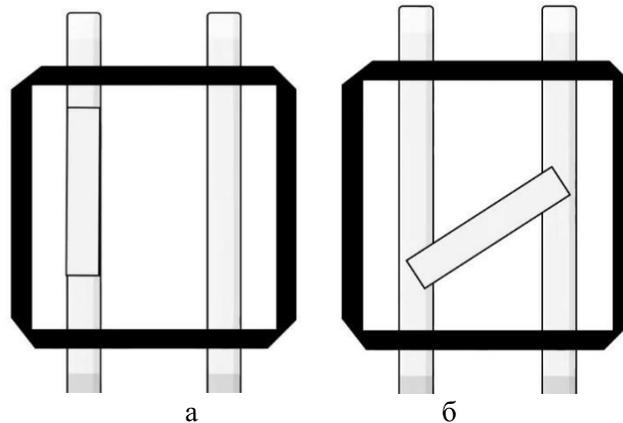


Рисунок 2 – Структура кнопки: а – стандартное состояние контактов кнопки; б – переключение контактов при нажатии на кнопку

В спокойном состоянии (рисунок 2а) параллельные контакты кнопки не замыкаются. При нажатии на кнопку (рисунок 2б), подвижный контакт замыкает две области, и таким образом пропускается электрический ток по нажатии на кнопку.

Подобные кнопки работают в режиме «кнопка нажата – пропускает сигнал; кнопка отпущена – сигнал обрывается». Однако, в некоторых задачах требуется иное поведение кнопки – чтобы при однократном нажатии на кнопку она запоминала, что на нее нажали. То есть, чтобы не приходилось держать кнопку бесконечно долго, а лишь нажать один раз. Это необходимо, например, в проектах с моторами – при нажатии на кнопку, двигатель запускается и вращается, пока не нажмется кнопка стопа; в проектах с переключением состояния – одно нажатие включает первое состояние, повторное нажатие включает второе состояние, и при третьем нажатии снова включается первое состояние устройства или робота.

В общем, кнопка с запоминанием состояния – это важный элемент, необходимый во всех случаях, когда управление системой автоматизации, роботом, или устройством не должно зависеть от длительности нажатия на саму кнопку.

И есть два решения данной проблема – либо покупать отдельно кнопки с фиксацией состояния, либо придумывать логику запоминания состояния обычной кнопки при помощи программирования. В данной работе рассматривается второй вариант решения проблемы (потому что первый вариант не нуждается в публикации статьи).

#### *Методы и материалы*

Для решения данной задачи применялось следующее оборудование: Arduino Uno, макетная плата, резистор 220 Ом, резистор 10 кОм, кнопка, светодиод.

Arduino Uno используется в качестве процессора, в который загружается программный код логики запоминания состояния кнопки, и к которому подключаются все элементы схемы.

Макетная плата в данном случае - необязательный компонент схемы. Здесь она применяется только для того, чтобы не припаивать каждый элемент друг к другу, а просто подключить на плату. Любой желающий может собрать ту же схему без макетной платы, но ему нужно будет припаивать все элементы согласно схеме.

Светодиод применяется для проверки правильности работы логики запоминания кнопки. Чтобы светодиод не сгорел от тока, проходящего через его контакты, к аноду подключается резистор 220 Ом.

### Раздел 3. «IT-технологии, энергетика, автоматизация и вычислительная техника»

К кнопке подключается резистор 10 кОм для предотвращения явления «дребезг контактов» [2].

Написание программного кода выполнялось в среде Arduino IDE – доступной открытой платформе для программирования Arduino, которое можно бесплатно скачать с официального сайта [3].

#### Результаты и обсуждение

Схема подключения кнопки и светодиода к цифровым пинам Arduino представлена на рисунке 3:

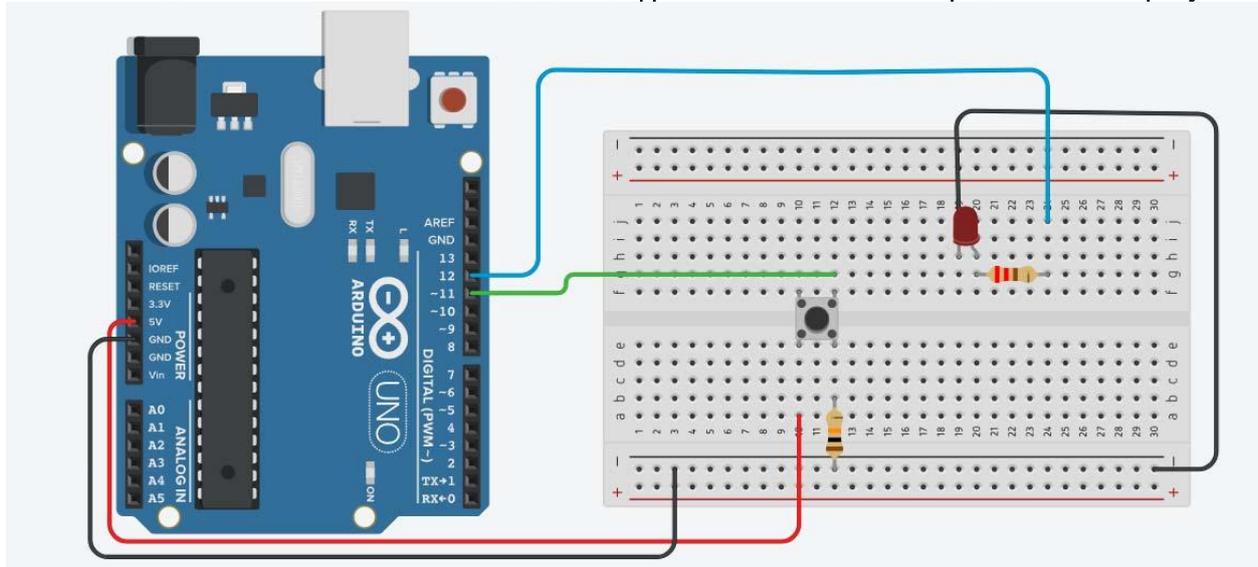


Рисунок 3 – Схема подключения кнопки и светодиода к Arduino

Кнопка подключена к 11 цифровому пину. Светодиод подключен к 12 цифровому пину.

Ниже представлен программный код обычной кнопки без фиксации для запоминания своего состояния.

```
int Last_State = 0;
bool Svet_State = LOW;

void setup()
{
  pinMode(12, OUTPUT);
  pinMode(11, INPUT);
}

void loop()
{
  int Current_State = digitalRead (11);
  if (Current_State != Last_State) {
    Last_State = Current_State;
    if (Current_State == LOW) {
      if (Svet_State == HIGH) {
        Svet_State = LOW;
      }
    }
    else {
      Svet_State = HIGH;
    }
  }
}
```

### Раздел 3. «IT-технологии, энергетика, автоматизация и вычислительная техника»

```

    }
    digitalWrite (12, Svet_State);
  }
}
}

```

Код начинается с объявления двух переменных - Last\_State и Svet\_State. Переменная Last\_State нужна для хранения памяти о последнем состоянии кнопки. В коде объявляется, что кнопка изначально не нажата (поэтому = 0). Переменная Svet\_State нужна для хранения памяти о состоянии светодиода: либо горит (HIGH), либо не горит (LOW) (поэтому назначен тип переменной bool). Так как изначально светодиод не горит, нужно объявить «= LOW».

В функции void setup объявлены два цифровых пина – 11 для кнопки (INPUT), и 12 для светодиода (OUTPUT).

В функции void loop представлена логика запоминания состояния кнопки. Первым делом инициализируется переменная Current\_State (необходима для запоминания текущего состояния кнопки), которая считывает состояние кнопки – «нажата ли кнопка?». Данная переменная объявлена в функции loop для того, чтобы после очередного цикла программы проверялось состояние кнопки, и, если состояние кнопки изменилось, то переменная перезаписывалась в реальном времени.

Далее выполняется проверка – «текущее состояние кнопки не равняется ли последним состоянием кнопки?». Это необходимо для того, чтобы проверить, отпустили ли нажатие кнопки. Если нет, то код дальше не выполняется. Если да, то выполняется код - переменная Last\_State получает значение Current\_State, то есть Arduino запомнило, что кнопка уже была нажата. Далее проверяется – «текущее состояние кнопки LOW?», - то есть, кнопка отпущена. Если кнопку отпустили, то выполняется код дальше – если светодиод уже горит (Svet\_State == HIGH), то отключить светодиод (Svet\_State = LOW). Если светодиод не горел, то он фиксированно загорается (else {Svet\_State = HIGH;}). В этом месте можно вставить любой код, который должен выполняться при нажатии кнопки; в данном случае для демонстрации правильности работы логики используется обычный светодиод. На его месте может быть любое другое исполнительное устройство.

Демонстрация работы данной схемы и данной программы представлены на рисунке 4. На рисунке 4б видно, что после однократного нажатия кнопки светодиод горит, то есть кнопка запомнила, что ее нажали.

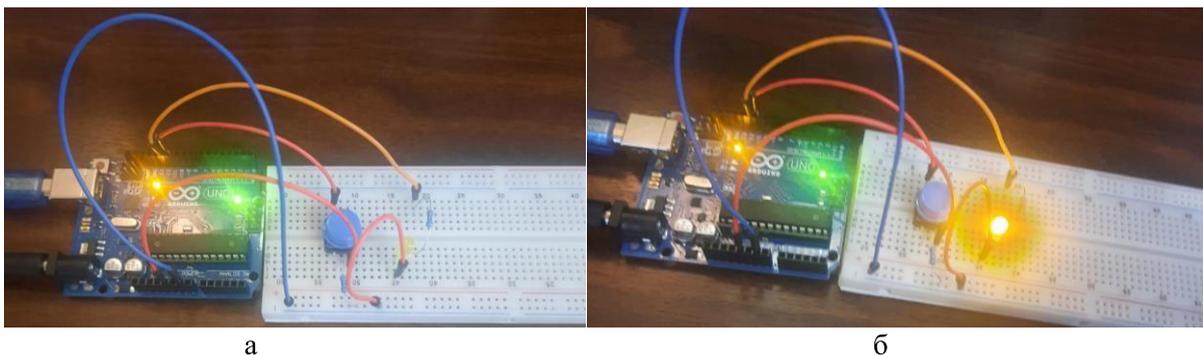


Рисунок 4 – Демонстрация работы: а – до нажатия кнопки; б – после нажатия кнопки

#### Выводы

1. Предложен метод превращения обычной кнопки без фиксации состояния в кнопку с запоминанием своего состояния при помощи программного кода.
2. Продемонстрирован программный код запоминания состояния кнопки с подробными пояснениями.

### **Раздел 3. «IT-технологии, энергетика, автоматизация и вычислительная техника»**

3. Была представлена схема подключения кнопки и светодиода к Arduino.
4. Была собрана и проверена схема запоминания состояние кнопки с демонстрацией правильности работы при помощи светодиода.
5. Полученные результаты могут и будут использоваться на занятиях робототехники в Карагандинском индустриальном университете при построении более продвинутых проектах Arduino.

#### *Список литературы*

1. Kolesnikov A., Atygayev T., Kasimos I. Arduino-based wireless security monitoring system // Bulletin of the M.Kozybayev NKU – 2020 - №1 (46) – с. 190-194.
2. Иванюк А.А., Можейко А.А. Цифровая схема анализа дребезга контактов коммутаторов // Системный анализ и прикладная информатика – 2023 - №3 – с. 24-32.
3. Arduino IDE [Электронный ресурс] – режим доступа: <https://www.arduino.cc/en/software/?ref=zx85.me> (Дата обращения: 26.10.2025).

Романов Д.Ю.

#### **Arduino платформасында бағдарламалау арқылы күйді есте сақтайтын батырманы жасау әдісі**

Мақалада Arduino платформасында қолданылатын стандартты батырманы күйді есте сақтау функциясы бар батырға түрлендірудің бағдарламалық әдісі қарастырылған. Ұсынылған тәсіл батырманың алдыңғы күйін есте сақтауды қосымша аппараттық компоненттерді қолданбай жүзеге асыруға мүмкіндік береді, бұл құрылғының құрылымын жеңілдетеді. Әр батырма басылған сайын логикалық күйдің өзгеруіне негізделген алгоритмнің жұмыс принципі сипатталған. Ұсынылған шешімнің оқу және автоматтандырылған жүйелердегі практикалық қолдану мысалдары келтірілген, мысалы: жарықты басқару, басқару пультін модельдеу және құрылғылардың прототиптерін жасау. Бұл әдіс логикалық күйлердің жұмыс істеу қағидаларын және микроконтроллерлерді бағдарламалаудың негіздерін көрсету мақсатында білім беру саласында қолданылуы мүмкін.

Түйінді сөздер: Arduino, батырма, күйді есте сақтайтын батырма, бағдарламалау, автоматтандыру, күйді басқару, электрондық құрылғылар, білім беру.

Romanov D.Yu.

#### **A Method for Converting a Regular Button into a State-Retaining Button Using Arduino Programming**

The article discusses a software-based method for converting a standard button used in Arduino projects into a button with a state-retaining function. The proposed approach allows implementing state memory of the button without additional hardware components, which simplifies the device design. The principle of the algorithm's operation, based on changing the logical state with each button press, is described. Examples of the practical application of this solution are presented in educational and automation systems, such as lighting control, control panel modeling, and device prototyping. The method can be used for educational purposes to demonstrate the principles of logical states and the basics of microcontroller programming.

*Keywords:* Arduino, button, state-retaining button, programming, automation, state control, electronic devices, education.

**Раздел 3. «IT-технологии, энергетика, автоматизация и вычислительная техника»**

References

1. Kolesnikov A., Atygayev T., Kasimos I. Arduino-based wireless security monitoring system // Bulletin of M. Kozybayev NKU – 2020 – No. 1 (46) – pp. 190–194.
2. Ivanyuk A.A., Mozheyko A.A. Cifrovaya sxema analiza drebezga kontaktov kommutatorov // Sistemny`j analiz i prikladnaya informatika – 2023 – No. 3 – pp. 24–32.
3. Arduino IDE [Electronic resource] – Access mode: <https://www.arduino.cc/en/software/?ref=zx85.me> (Accessed: October 26, 2025).