

Взаимодействие с контроллером через элементы управления.

Мы научились писать программу для визуального общения с человеком посредством светодиодов. Но любое взаимодействие основывается на том, что человек должен как-то оповещать контроллер о своих намерениях.

Кнопки – самый доступный и широко используемый элемент (интерфейс) для взаимодействия цифровых устройств с человеком. Вы все печатали на клавиатуре текст, писали запросы в интернете и т.д. Вы использовали клавиатуру – это набор кнопок. Вот и мы в контроллере ««РУДИРОН»» встречаем 3 кнопки для их использования в наших устройствах.

Кнопки так же, как и светодиоды подключены к выходам (номерам) контроллера и используют внутреннее устройство портов ввода и вывода GPIO микроконтроллера.

В отличие от светодиодов порты должны работать не на выход, а на ввод.

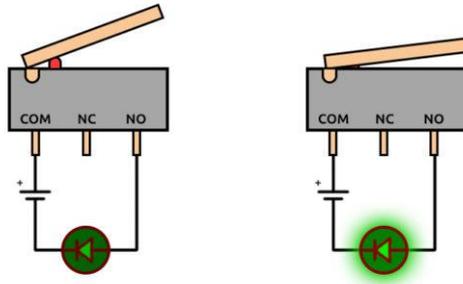
Использование кнопок



Кнопка является простейшим устройством, при помощи которого можно управлять выполнением программы на микроконтроллере, но физически она выполняет очень простую функцию: замыкает и размыкает контакт.

Кнопки бывают нескольких типов:

- С фиксацией – кнопка остаётся нажатой после отпускания, без фиксации – отключается обратно.
- Нормально разомкнутая (Normal Open, NO) – при нажатии замыкает контакты. Нормально замкнутая (Normal Closed, NC) – при нажатии размыкает контакты.
- Тактовые кнопки – замыкают или размыкают контакт. Переключатели – обычно имеют три контакта, общий COM, нормально открытый NO и нормально закрытый NC. При отпущенной кнопке замкнута цепь COM-NC, при нажатой замыкается COM-NO.

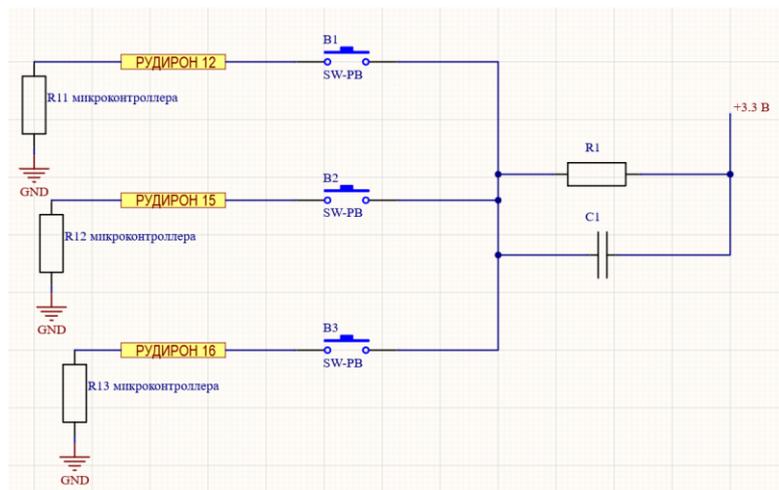


Встроенные кнопки

На плате «РУДИРОН» есть 3 кнопки без фиксации (D1, D2, D3), которые можно использовать для управления выполнением алгоритма программы.



Схема подключения кнопок B1, B2, B3 (BUTTON1, BUTTON2, BUTTON3):



По схеме видно, что на вход контроллера поступит логический 1 (напряжение близкое к 3.3 вольт), когда мы нажмем кнопку (вход контроллера соединяется с питанием 3.3 вольта через резистор R1). Если кнопка не нажата, то на входах присутствует логический ноль (за счет

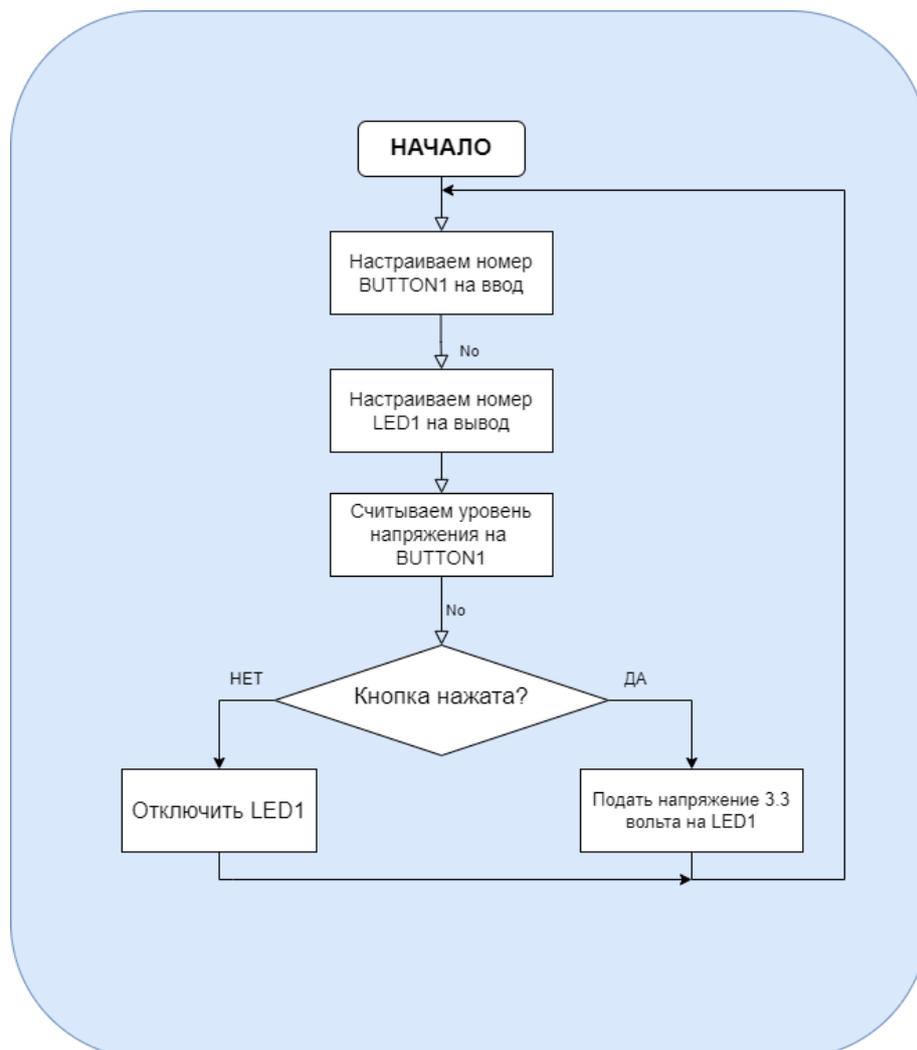
внутреннего резистора микроконтроллера, подключенного к земле так называемый подтягивающий резистор).

Задача – написать программу по нажатию и удержанию кнопки В1 (BUTTON1) контроллер подает напряжение 3.3 вольт на светодиод L1 (светится). Когда кнопка отпускается, то светодиод L1 должен погаснуть. Такой простейший электронный переключатель свечения светодиода L1.

Для получения информации о уровне напряжения на входе контроллера необходимо использовать команду (функцию) `digitalRead(«Номер контроллера»)`.

Команда считывает с номера контроллера уровень напряжения и если он около 3 вольт, то это 1 (true) и кнопка не нажата. Если кнопку нажмут, то номер будет соединен с землей схемы и это будет считано как (false).

Сначала нарисуем алгоритм программы:



Используя алгоритм пишем программу:

```
#include "Arduino.h"
void setup()
{
    pinMode(BUTTON_BUILTIN_1, INPUT_PULLDOWN);
    pinMode(LED_BUILTIN_1, OUTPUT);
}
bool btnState;    // создаем переменную для хранения true или false
void loop()
{
    digitalWrite(LED_BUILTIN_1, false);
    btnState = digitalRead(BUTTON_BUILTIN_1);
    if (btnState)
    {
        // обработчик нажатия
        digitalWrite(LED_BUILTIN_1, true);
    }
    else
    {
        // обработчик отпущения
        digitalWrite(LED_BUILTIN_1, false);
    }
}
```

Рассмотрим нашу программу.

Первое изменение в блоке `setup` мы видим новый вариант установки режима работы вывода контроллера `pinMode(BUTTON_BUILTIN_1, INPUT_PULLDOWN)`. Здесь самое интересное это `INPUT_PULLDOWN`. Это как раз подключение внутреннего подтягивающего резистора вывода к земле R12. Такая установка позволит при отсутствии нажатой кнопки В1 однозначно иметь уровень 0 напряжения на входе.

Хорошим тоном в программировании является хранение состояния кнопки (нажата или не нажата) в отдельной переменной. Такой стиль позволит в дальнейшем организовать с помощью кнопок разветвленное меню управления устройством. Поэтому создаем переменную `bool btnState`. Причем она создана между блоками `setup` и `loop`. Это будет глобальная переменная, к которой мы сможем обращаться из любой части программы. Переменная имеет свой тип `bool` – может хранить 1 или 0 (`true` или `false`).

Для считывания уровня напряжения на входе контроллера на `BUTTON1`(кнопка В1) используем команду:

`btnState = digitalRead(BUTTON_BUILTIN_1)` с занесением в переменную `btnState` значения `true` если кнопка нажата (напряжение на входе 3.3 вольта за счет резистора R1 на схеме выше) или `false` если кнопку не нажали (напряжение близкое к нулю за счет подключения входа контроллера к земле внутренним подтягивающим резистором R12).

Для определения какое значение было записано в переменную `btnState` используем оператор ветвления `if`:

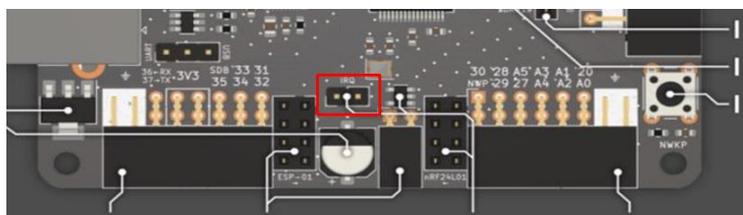
```
if (btnState)
{
    // обработчик нажатия
    digitalWrite(LED_BUILTIN_1, true);
}
```

Если переменная `btnState` имеет значение (кнопка B1 нажата 3.3 вольта на входе контроллера) `true`, то выполняется команда `digitalWrite(LED_BUILTIN_1, true)` и светодиод загорается.

Для обработки состояния когда кнопка не нажата воспользуемся возможностями оператора `if`, который позволяет исполнить блок кода если условие не совпало с `true` :

```
else
{
    // обработчик отпущения
    digitalWrite(LED_BUILTIN_1, false);
}
```

Важно: Работа кнопки B2 может иногда работать наоборот. Программа будет считать, что кнопка нажата, хотя вы ее не трогали. Это происходит из-за установки переключки IRQ необходимой для работы радио модуля.



Вы можете снять переключку. И программа корректно заработает. Или не снимая ее изменить условие проверки факта нажатия на обратное добавив «!»:

```
if (!btnState)
```

- Задание:**
1. Напишите программу при нажатии на кнопку В2 светится светодиод L2.
 2. Напишите программу при нажатии кнопки В1 светится светодиод L1, при нажатии В2 светится светодиод L2. При нажатии на кнопку В3 светятся оба светодиода L1 и L2.