

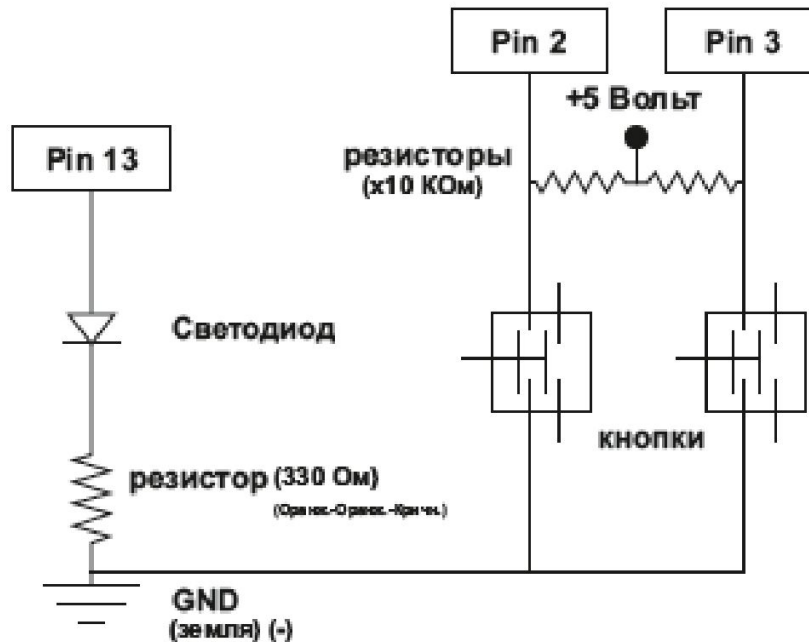
Практическое занятие 2

Ввод информации через цифровые
порты

Схемотехнические особенности ввода!

- На портах **Arduino**, настроенных на ввод, должен быть **однозначный** сигнал либо низкого (**LOW**), либо высокого (**HIGH**) уровня.
- Для этого порт должен быть соединен либо с землей (**GND**), либо с выводом (**+ 5 V**) через **«подтягивающий»** резистор величиной (10-20 кОм).
- **Неподключенный** порт ввода будет иметь **неопределенный** уровень!
- * В Arduino есть возможность использования встроенных подтягивающих резисторов.

- Пример использования цифрового ввода. Принципиальная схема.



Работающие на
ввод цифровые
выводы **Arduino**

«Подтягивающие»
резисторы **10 кОм**

Кнопки управления

Назначение подтягивающих резисторов: 1) обеспечить однозначный уровень на цифровых выводах; 2) предотвратить короткое замыкание при нажатии кнопки.

Если кнопка **не нажата**, то на входном выводе будет **высокий HIGH** уровень (5 В), т.к. напряжение на вывод будет поступать от источника питания через подтягивающий резистор.

Если кнопка **нажата**, то на входном выводе будет **низкий** уровень (0 В), т.к. вывод напрямую соединяется с общим проводом (землей)

Реализация программного кода

- 1) Объявить переменную в которую будет производиться запись данных с порта **перед** областью `Void Setup`.
- Для цифровых данных используется переменная типа `boolean`.
- Синтаксис: `boolean NameVar;`
`boolean` – тип переменной, `NameVar` – имя переменной.

- 2) конфигурация выбранного порта на вывод в области однократного выполнения кода `Void Setup`:

```
pinMode(pin, INPUT);
```

- 3) записать данные с порта в переменную в основной области `Void loop`:

```
NameVar = digitalRead(pin);
```

- * значение этой переменной можно вывести в другой цифровой порт с помощью функции `digitalWrite`

- * При написании программ для реализации условных переходов (**Если, то**) можно пользоваться логическими операциями:

==	ЭКВИВАЛЕНТНОСТЬ	A == B - true (истина), если A и B равны
!=	НЕ РАВНО	A != B - true (истина), если A и B не равны
&&	логическое "И"	A && B - истина, если оба и A и B - истина
 	логическое "ИЛИ"	A B - истина, если один из A или B истина
!	ОТРИЦАНИЕ	A ! B - превращает ложь в истину, и наоборот

- Это позволит осуществить выполнение кода при выполнении нескольких условий.

- # Пример использования условного перехода и логических операций.

- `if (NameVar1== LOW & NameVar2==HIGH) // условие`
- `{`
- `digitalWrite(13, HIGH); // действие при выполнении условия`
- `} else {`
- `**** ; // действие при невыполнении условия`
- `}`

Задание 1

- Подключить кнопку к одному из цифровых выводов с использованием подтягивающего резистора 10 кОм.
- Написать программу включения светодиода, подключенного к выводу 13, если кнопка нажата.

Задание 2

- Подключить вторую кнопку к другому цифровому порту.
- Написать программу включения светодиода, подключенного к выводу 13, только если обе кнопки нажаты.

Задание 3

- Собрать модель светофора с включением красного света для машин по требованию пешехода.
- Алгоритм работы: нормальный режим работы светофора – зеленый (для машин). При **однократном** нажатии на кнопку зеленый начинает мигать, затем загорается желтый и затем красный. Красный горит какое-то время и затем включается опять зеленый.