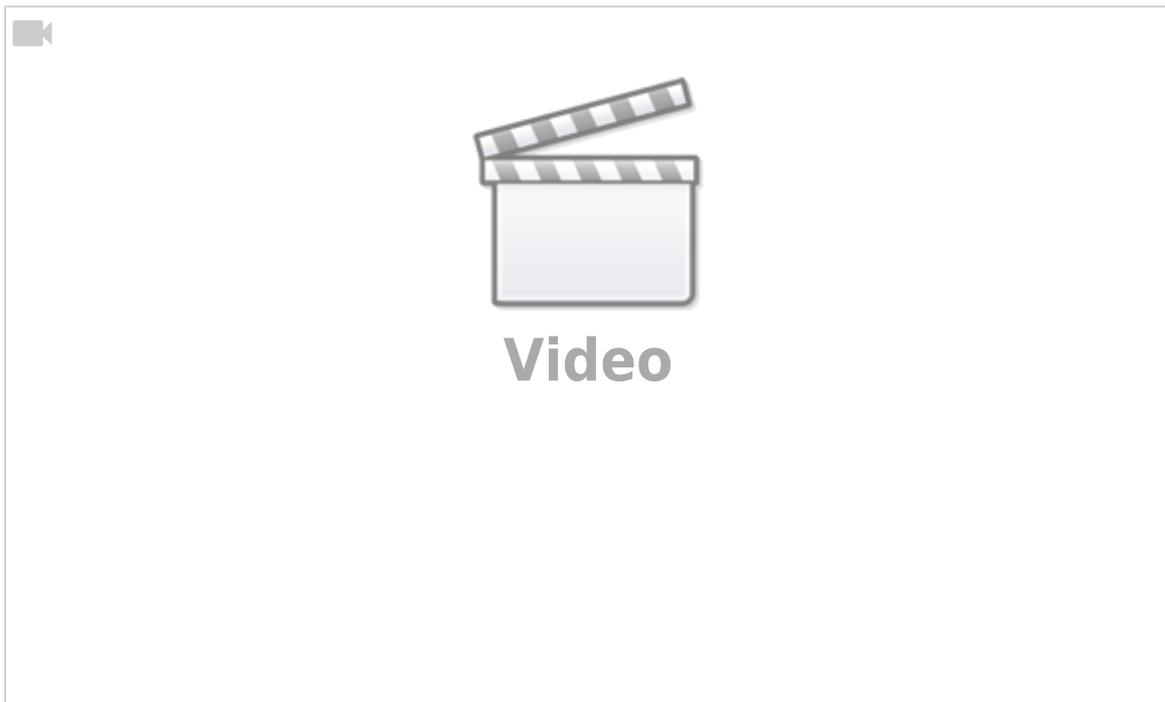
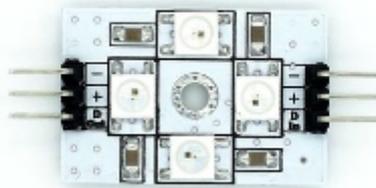


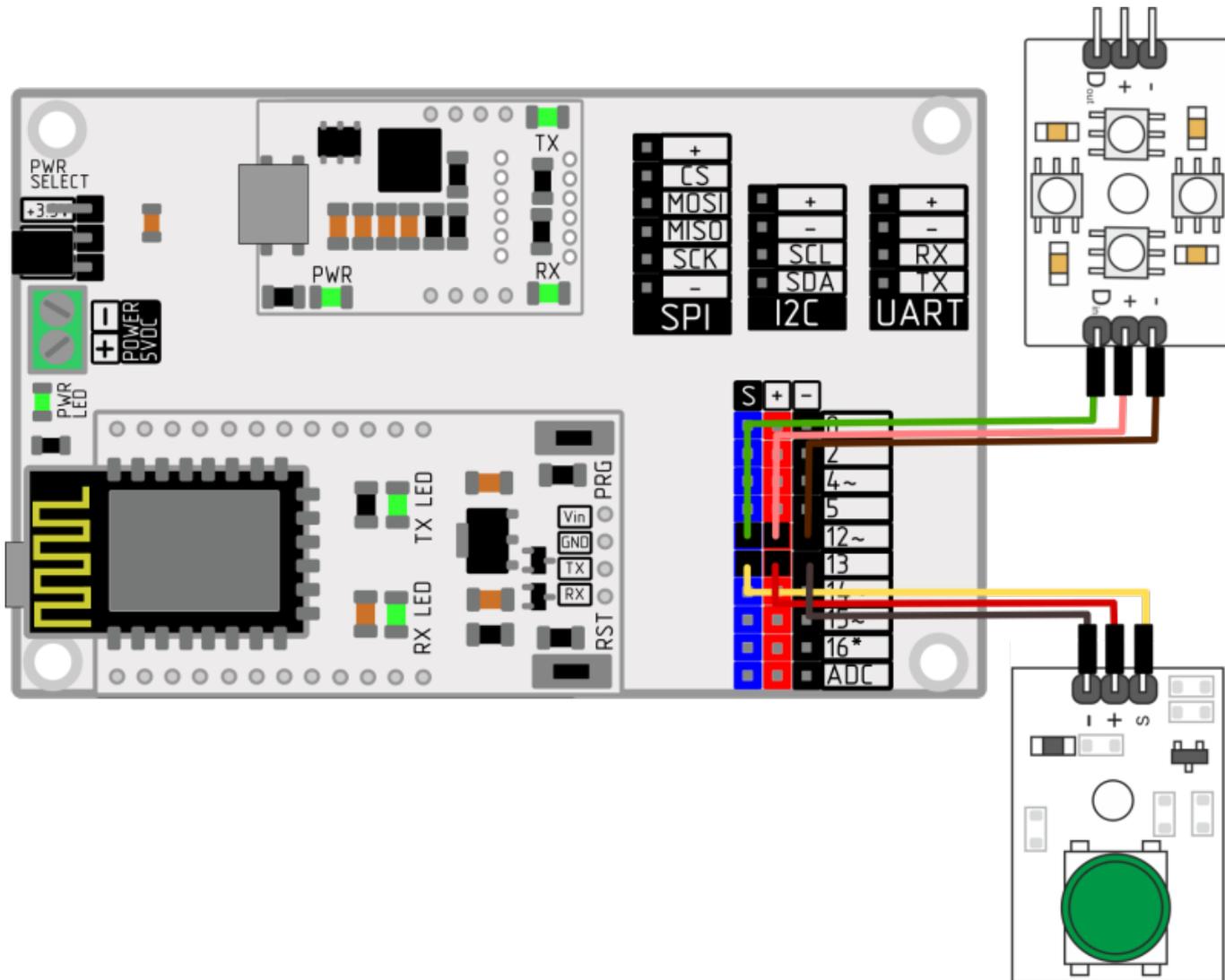
Урок 15. Кортежи



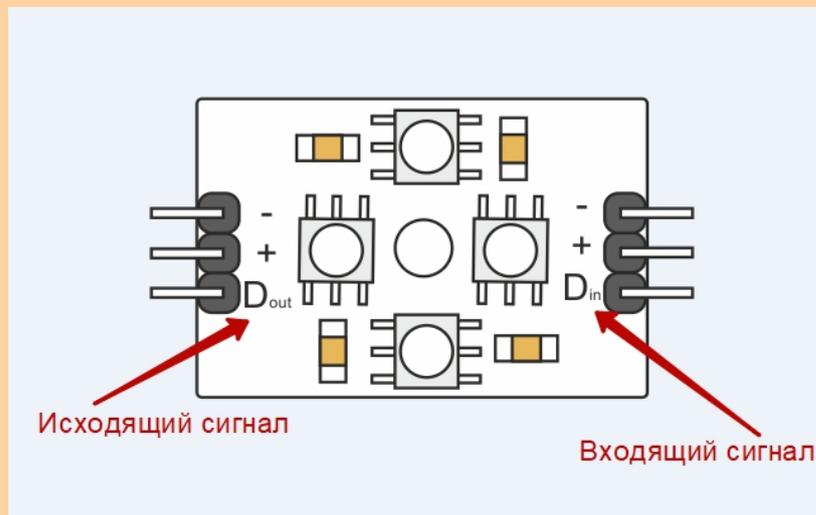
Перейдём к ещё одному интересному модулю в наборе - это модуль адресных светодиодов.



Он отличается от обычного модуля светодиода тем, что имеет четыре светодиода, каждый светодиод может гореть любым цветом и к каждому из светодиодов можно обратиться адресно. Именно поэтому светодиоды называются адресными. Подключим модуль адресных светодиодов к плате Гиккон Коннект.



Обратите внимание что на модуле имеется два ряда выводов. Они нужны потому что платы можно последовательно подключать друг к другу. Входящий сигнал подключается к выводу **Din**, выходящий к выводу **Dout**.



Для работы с платой и зажигания светодиодов на ней нам потребуется модуль `neopixel`, а точнее его часть `NeoPixel`. Также будут нужны уже знакомые нам `machine` и `time`. Цвет светодиода задаётся тремя параметрами - это три составляющих цвета свечения: красный, зелёный и синий. Яркость свечения устанавливается от 0 до 255. Где **0 - цвет погашен, 255-максимально яркий**.

Указать цвет свечения нужно одновременно указав три параметра свечения для светодиода. Например, красный будет задаваться как 255 - яркость красного свечения, 0 - зелёного и 0 - синего. Одновременно три параметра мы могли бы указать в списке: `[255, 0, 0]`. Но для данного модуля требуется указание цвета свечения в **кортеже**.

Кортеж - это тоже коллекция значений, которая очень похожа на списки. Основное отличие кортежа в том, что его нельзя изменить. Это делается для того, чтобы случайно не поменять переменную. Например, определим переменную `red` и присвоим ей значение составляющих красного цвета. Кортежи, в отличие от списков, определяются круглыми скобками:

```
red = (255, 0, 0) # красный цвет
```

Теперь мы точно знаем, что цвет останется красным, и этот кортеж случайно не изменится нигде в программе. Теперь определим цвет, когда ничего не горит (светодиод погашен):

```
dark = (0, 0, 0) # не горит ничего
```

Создадим список светодиодов, к которым можно будет обращаться по номеру и указывать каким цветом должен гореть каждый. Для этого нам потребуется импортированная функция `NeoPixel`, которой нужно передать вывод к которому подключена плата со светодиодами и количество светодиодов на плате или платах.

```
# создание списка светодиодов  
leds = NeoPixel(addr_leds, 4)
```

Далее напишем функцию в которой пока зажжём первый светодиод красным цветом и погасим его. Для этого укажем номеру светодиода (*нумерация начинается с 0*) каким цветом гореть. Затем вызовем функцию `write()` объекта `leds` и именно в этот момент светодиод загорится. Пусть он горит 1 секунду. А для того чтобы погасить светодиод, укажем что он должен гореть "никаким" цветом:

```
def light_leds():  
    leds[0] = red  
    leds.write()  
    sleep(1)  
    leds[0] = dark  
    leds.write()
```

Ну а теперь вызовем эту функцию по нажатию на кнопку.

Целиком скрипт будет выглядеть следующим образом:

```
# импорт модулей
```

```
from neopixel import NeoPixel
from machine import Pin
from time import sleep

# выводы к которым подключены платы:
but = Pin(13, Pin.IN) # кнопка
addr_leds = Pin(12) # адресные светодиоды

# переменные для определения цветов
dark = (0, 0, 0) # не горит ничего
red = (255, 0, 0) # красный цвет

# создание списка светодиодов
leds = NeoPixel(addr_leds, 4)

# функция зажигания светодиода
def light_leds():
    leds[0] = red
    leds.write()
    sleep(1)
    leds[0] = dark
    leds.write()

# основной цикл программы
while True:
    if but.value():
        light_leds()
```

Запомнить:

- Кортежи - это коллекции значений которые нельзя изменять
- Они нужны, чтобы точно быть уверенным, что коллекцию не меняли
- С кортежами можно работать так же как и со списками, за исключением изменения значений
- Кортежи обозначаются круглыми скобками

[Предыдущий урок](#)

[Следующий урок](#)

From:
<https://know.gikkon.ru/> -

Permanent link:
https://know.gikkon.ru/main/gikkon_start/p1_I15

Last update: 2023/10/06 14:17



