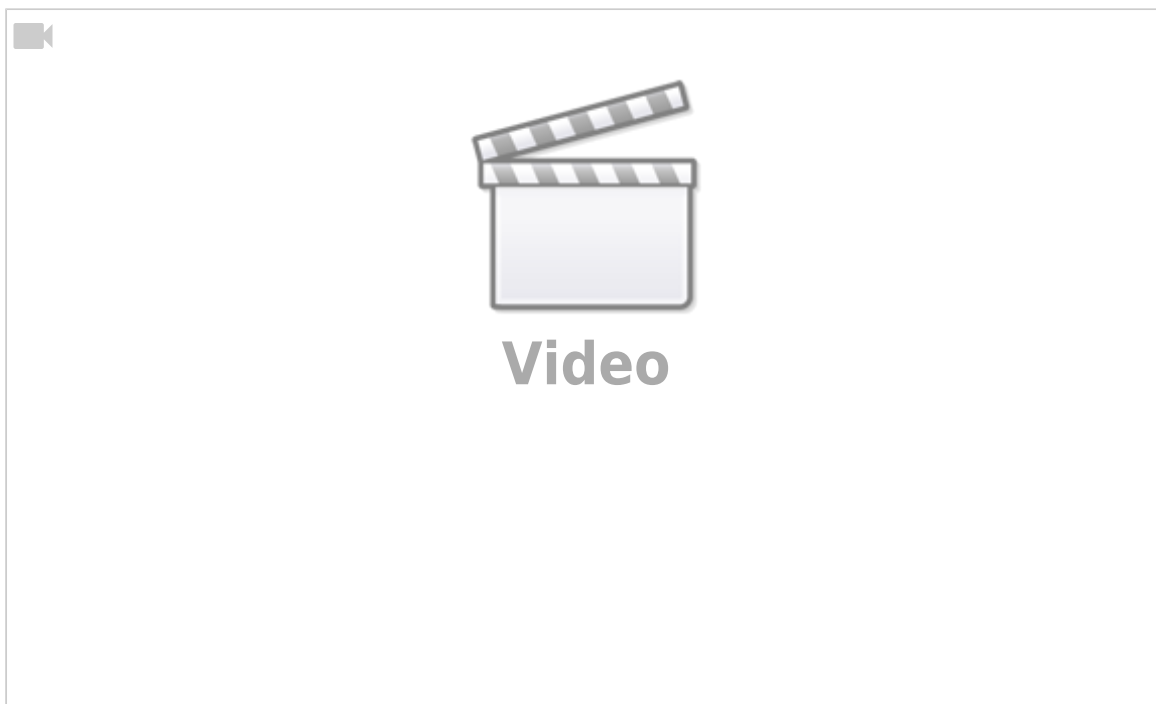


Урок 9. Условные операторы. Оператор if



Усложним наш проект. Подключим к плате расширения *Гиккон Коннект* модуль кнопки. Модуль кнопки работает следующим образом: когда она нажата, она пропускает через себя ток, когда кнопка находится в выключенном состоянии, тока нет. Таким образом кнопку можно использовать как выключатель.



У каждого цифрового вывода есть два состояния: на выводе есть ток или же его нет. Для того чтобы проверить это состояние используется функция `value()` объекта `Pin`. В случае если ток есть, вернётся значение `1 (True)`, в противном случае вернётся `0 (False)`.

Помимо этого цифровые выводы могут работать в двух основным режимах: вход (IN) и выход (OUT). Вход подразумевает что контакт принимает информацию (ток), а выход что контакт отдаёт ток. Именно это значение мы указываем, когда пользуемся объектом `Pin(5, Pin.OUT)`, при его инициализации. Номер вывода, и его режим (IN или OUT). Для светодиода мы использовали режим OUT, но для кнопки нужно использовать режим IN (так как мы будем ждать входящего сигнала от кнопки).

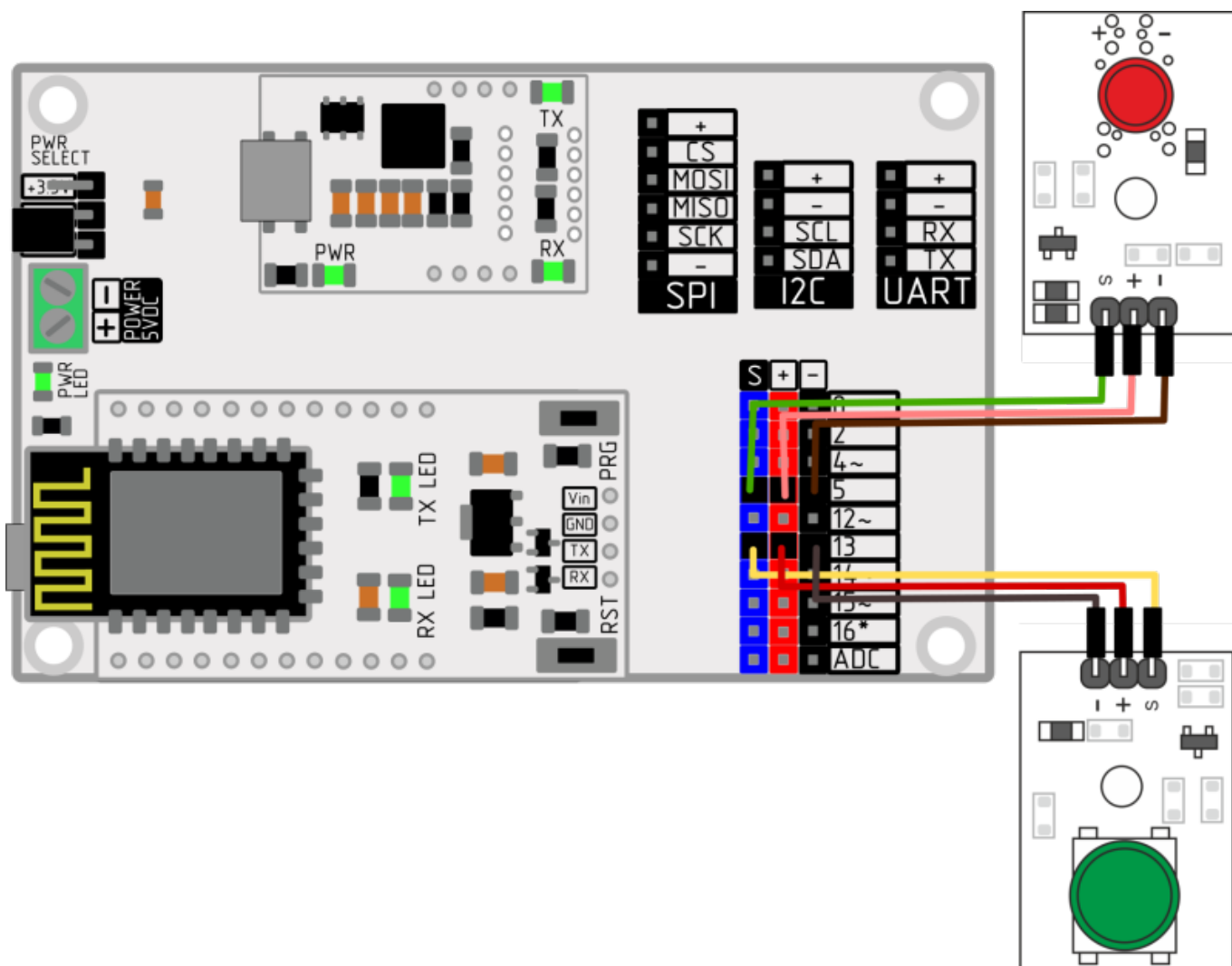
Светодиод будем зажигать только во время нажатия кнопки. Программно проверить какое-либо условие в python можно с помощью условных конструкций. Познакомимся с конструкцией

if. Она очень проста:

```
if выражение1:  
    действия если сработало выражение1  
elif выражение2:  
    действия если сработало выражение2  
else:  
    действие если ничего из предыдущих выражений не сработало
```

После ключевого слова if укажите выражение, которое будет вычисляться. Если это выражение истинно, то будут выполняться действия следующие с отступами после части if. Если выражение ложно, то вычисляется выражение следующее за ключевым словом elif (смесь от английского else if). Здесь так же выполнится код, если выражение 2 истинно. Далее может идти сколько угодно много выражений elif. В конце можно указать после ключевого слова else (иначе) какие действия нужно выполнить, если не сработало ни одно из условий. Части elif и else не являются обязательными.

Подключим к плате Гиккон Коннект модуль кнопки к выводу 13.



Итак, скорректируем наш код. Добавим переменную but (от английского button - кнопка), укажем что она будет подключена к 13 выводу, в режиме входа: but = Pin(13, Pin.IN).

Затем в цикле будем проверять нажата ли сейчас кнопка. Для этого получим значение с вывода кнопки: `but.value()`. Оно будет истинным если кнопка нажата, и ложным если не нажата. Если выражение истинно, включим светодиод `led.on()`, в противном случае (`else`) светодиод погасим. Полностью изменённый код будет выглядеть следующим образом:

```
# импорт модулей
from machine import Pin

# переменные
led = Pin(5, Pin.OUT)
but = Pin(13, Pin.IN)

# основной цикл программы
while True:
    if but.value():
        led.on()
    else:
        led.off()
```

Запомнить:

- У цифровых выводов есть два состояния, которые можно проверить с помощью функции `value()`
- Вывод может работать в одном из режимов: `Pin.IN` (вход) или `Pin.OUT` (выход)
- Выполнять действия только при определённых условиях можно при помощи конструкции `if`
- Части `elif` и `else` не являются обязательными в условных выражениях

[Предыдущий урок](#)

[Следующий урок](#)

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

Permanent link:
https://know.gikkon.ru/main/gikkon_start/p1_l9

Last update: **2023/10/06 14:13**

