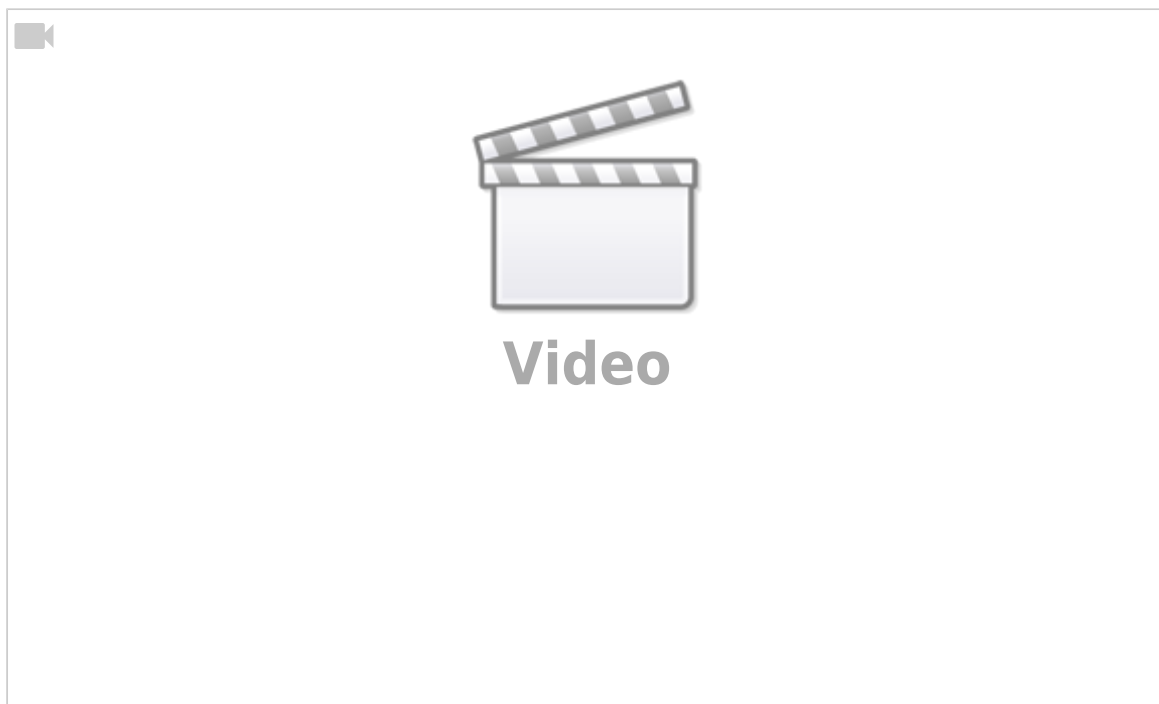
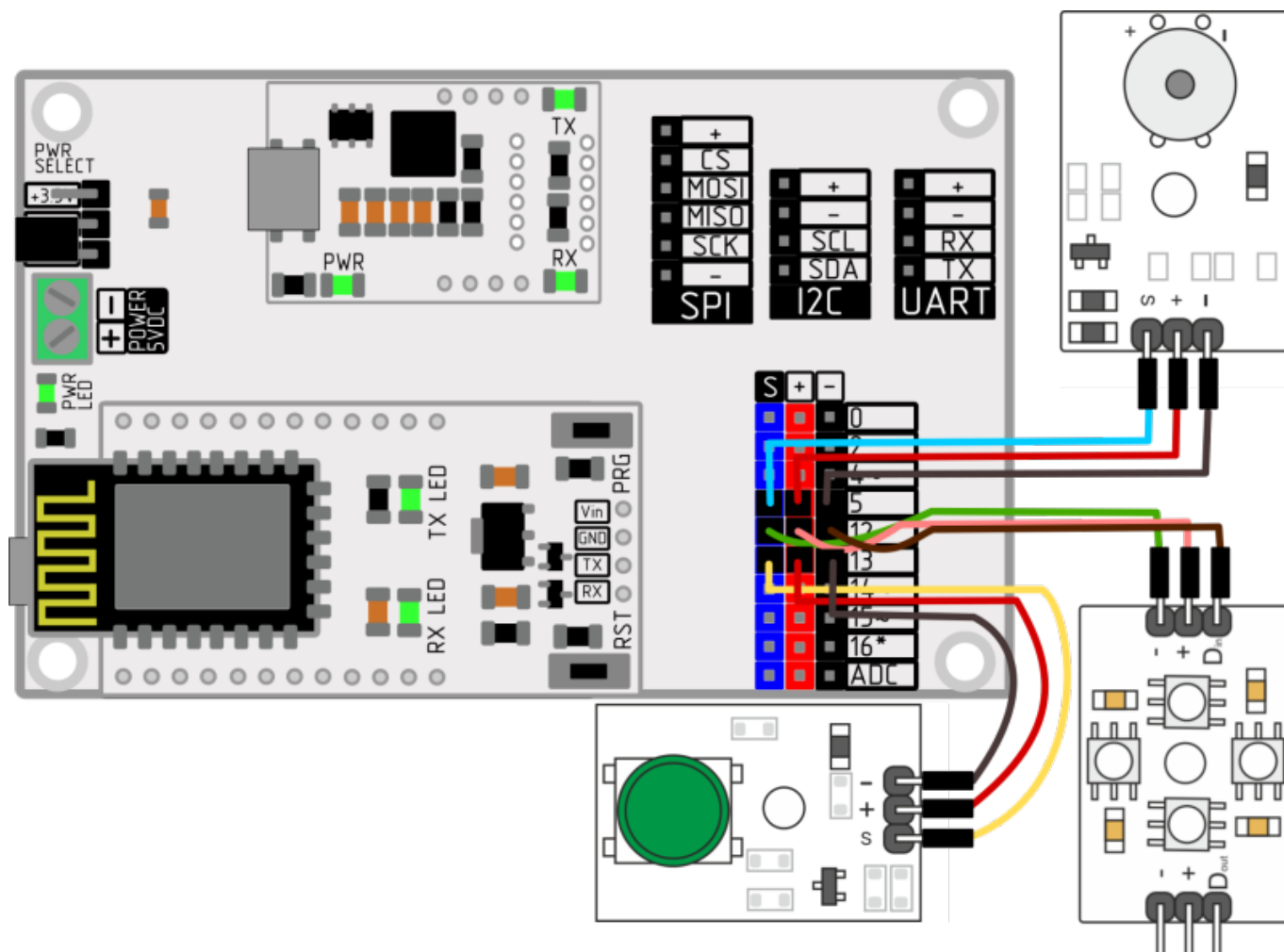


Урок 19. Ключевое слово global



Оставим схему сборки из урока 16, однако, нам понадобится только модуль кнопки.



В предыдущем уроке мы поговорили о видимости переменных. Ещё раньше мы работали с прерываниями. И здесь нужно знать ещё одну особенность прерываний. Помимо того что код прерываний должен быть максимально простым, нужно учитывать тот момент что у прерываний нет доступа к памяти устройства. То есть внутри функции прерывания мы не можем использовать переменные, которые были объявлены вне функции-обработчика прерывания.

Но как быть, если нужно знать состояние одной из переменных? Здесь придёт на помощь ключевое слово `global`, которое указывает на то, что переменная берётся из зоны глобальной видимости (общей видимости всего скрипта).

Вернёмся к [уроку 17](#) по работе с прерываниями. Там мы описали функцию, которая выполняется при прерывании. Это был вывод сообщения о том, с какого вывода поступило прерывание. Доработаем данную функцию. Будем считать количество наступивших прерываний. Для этого введём переменную-счётчик количества прерываний и затем будем при каждом прерывании добавлять к этой переменной единицу.

В области объявления переменных объявим новую переменную и назовём её `irq_q`, присвоим ей значение 0.

```
irq_q = 0 # счётчик количества прерываний
```

А теперь в функции-обработчике прерывания, прибавим к переменной единицу и выведем её значение.

```
# функция, которая будет вызвана при возникновении прерывания
def callback(pin):

    irq_q = irq_q + 1 # увеличение количества прерываний
    print("Количество прерываний: ", irq_q)
```

Если попробовать запустить исправленный скрипт в таком виде, то получим ошибку. Как было сказано выше, у обработчика прерывания нет доступа к переменным, которые были объявлены вне функции. Для того чтобы обеспечить доступ к этой переменной, нужно добавить следующую строку:

```
global irq_q # "достать" переменную
```

Весь код будет выглядеть так:

```
# импорт модулей
from machine import Pin

# выводы к которым подключены платы:
but = Pin(13, Pin.IN) # кнопка
irq_q = 0 # счётчик количества прерываний

# функция, которая будет вызвана при возникновении прерывания
def callback(pin):

    global irq_q # "достать" переменную
```

```
irq_q = irq_q + 1 # увеличение количества прерываний
print("Количество прерываний: ", irq_q)

# подключение прерывания на вывод кнопки
but.irq(trigger=Pin.IRQ_RISING, handler=callback)

# основной цикл программы
while True:
    pass # ничего не делать
```

Если теперь запустить скрипт, то при возникновении прерывания в терминале будет выводиться количество прерываний, которое будет соответствовать нажатию кнопки.

Запомнить:

- Функция-обработчик прерываний “не видит” переменные, которые были объявлены вне функции
- Для того чтобы прочитать или изменить переменную вне функции-обработчика, нужно предварительно “достать”
- Ключевое слово global “достаёт” переменную из памяти

[Предыдущий урок](#)

[Следующий урок](#)

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

Permanent link:
https://know.gikkon.ru/main/gikkon_start/p1_l19

Last update: **2023/10/06 14:19**

