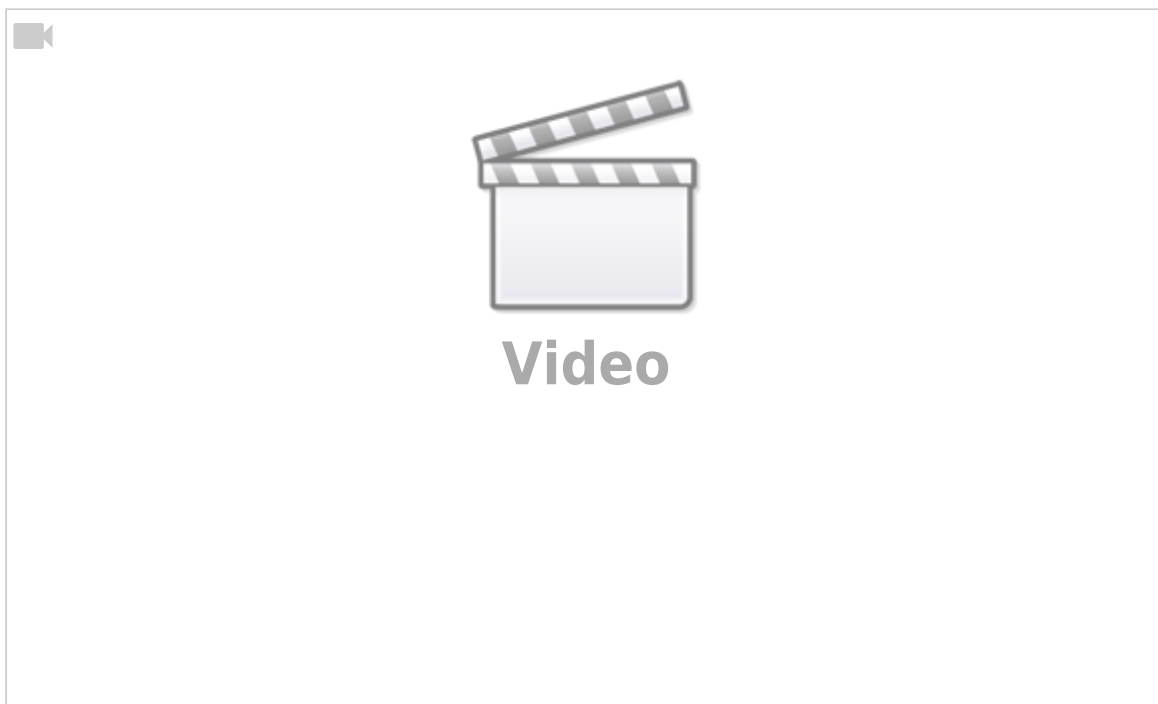
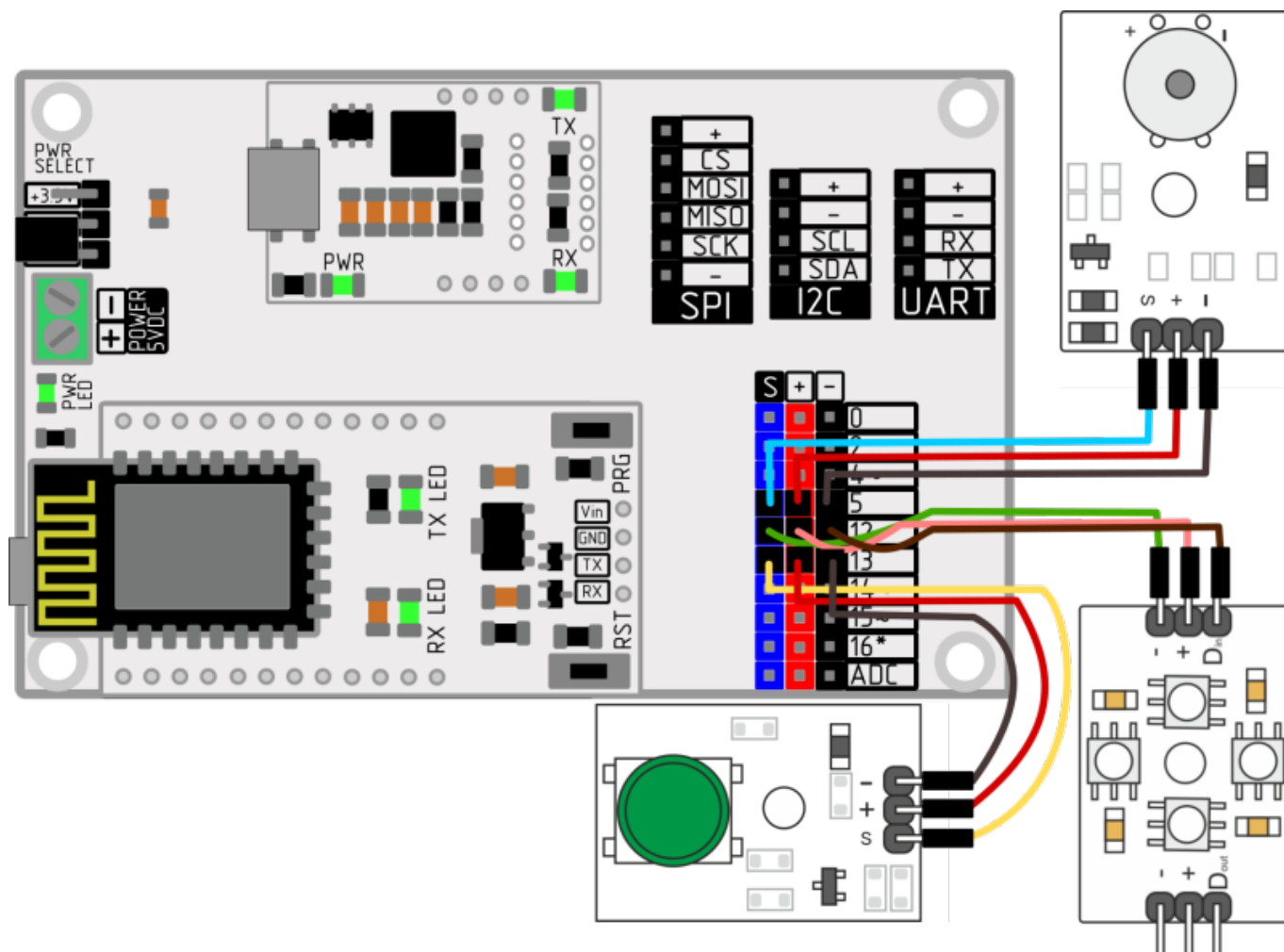


Урок 17. Прерывания



Оставим схему сборки с прошлого урока, однако, нам понадобится только модуль кнопки.



В прошлом уроке мы смастерили сирену для машины экстренных служб (например, Скорой помощи). Но сирена наша звучит только несколько циклов. Если запустить сирену на постоянной основе (то есть сделать “вечный” цикл), то непонятно, как её выключить? Ведь пока у нас исполняется код в “вечном” цикле, остальные команды срабатывать вовремя не будут. В том числе обработка нажатия кнопки.

На помощь приходят **прерывания**. Примером прерывания в жизни служат отвлечения от работы. Например, Вы занимаетесь неким делом, скажем, смотрите кино. И тут Вам позвонил курьер, и сообщил, что он привёз вкусную еду. Вам приходится поставить кино на паузу, встретить курьера, забрать заказ, распаковать его, и уже смотреть кино далее, с того места, где Вы его ставили на паузу.

В контроллере происходит примерно так же. Какой бы код не выполнялся, мы можем “дёрнуть” ножку контроллера, он прервёт все свои дела, выполнит определенные инструкции и вернётся к выполнению кода, на том месте, где остановился.

На контроллере Гиккон для прерывания подойдёт любой вывод контроллера кроме вывода 16.

Для того чтобы вызвать прерывание, нужно сначала описать функцию, которая будет вызвана, когда это прерывание произойдёт. В эту функцию параметром передаётся номер вывода, с которого поступило прерывание. Опишем такую функцию:

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

    print("Вызвано прерывание на выводе №", pin)
```

Наша функция выводит сообщение в окно терминала, в котором указан номер вывода с которого произошло прерывание.

Обратите внимание, что есть рекомендации, по поводу написания кода прерываний. Не стоит выполнять сложные действия и перегружать функции прерываний. Также не рекомендуется выводить сообщения с помощью инструкций `print()`. В данном примере `print()` используется только для наглядности. Ниже приведены основные рекомендации при написании кода прерываний:

- Старайтесь, чтобы код был как можно короче и проще.
- Избегайте выделения памяти: никаких добавлений к спискам или вставок в словари, никаких операций с плавающей запятой.
- Код должен делать только то, что должно быть сделано сразу после события, вызвавшего его: операции, которые можно отложить, должны делегироваться в основной цикл программы.
- Следует избегать циклических конструкций или сводить к минимуму.
- Обычно следует избегать ввода-вывода на устройства, отличные от прерывающего устройства: ввод-вывод, доступ к диску, операторы `print()` и доступ к UART

Далее, тому выводу от которого это прерывание ожидается, нужно это прерывание подключить с помощью инструкции функции `irq`, где в параметрах нужно указать тип прерывания, который мы отлавливаем и функцию, которую нужно запустить, в случае возникновения этого прерывания. Два основных типа прерываний:

- `IRQ_FALLING` (когда напряжение меняется с высокого на низкое - с HIGH на LOW)

- `IRQ_RISING` - наоборот.

Подключим прерывание для нашего вывода кнопки:

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

Ну а теперь, нужно описать основной цикл программы. Но так как у нас пример только для демонстрации прерываний, в цикле мы ничего делать не будем. Для того чтобы описать пустой цикл, мы будем использовать ключевое слово `pass` - которое как раз и означает - "ничего не делать".

```
while True:
    pass
```

Целиком код программы будет выглядеть следующим образом:

```
# импорт модулей
from machine import Pin

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

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

    print("Вызвано прерывание на выводе №", pin)

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

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

Загрузите данный скрипт на контроллер и попробуйте нажимать кнопку. В окне терминала будет видно сообщение о нажатой кнопке.

Запомнить:

- Прерывания нужны для срочного прекращения запущенного кода.
- Функции для выполнения прерываний должны быть максимально простыми.
- Прерывания на контроллере Гиккон можно настроить на любой из выводов, кроме 16.

[Предыдущий урок](#)

[Следующий урок](#)

From:

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

Permanent link:

https://know.gikkon.ru/main/gikkon_start/p1_l17

Last update: **2023/10/27 13:18**

