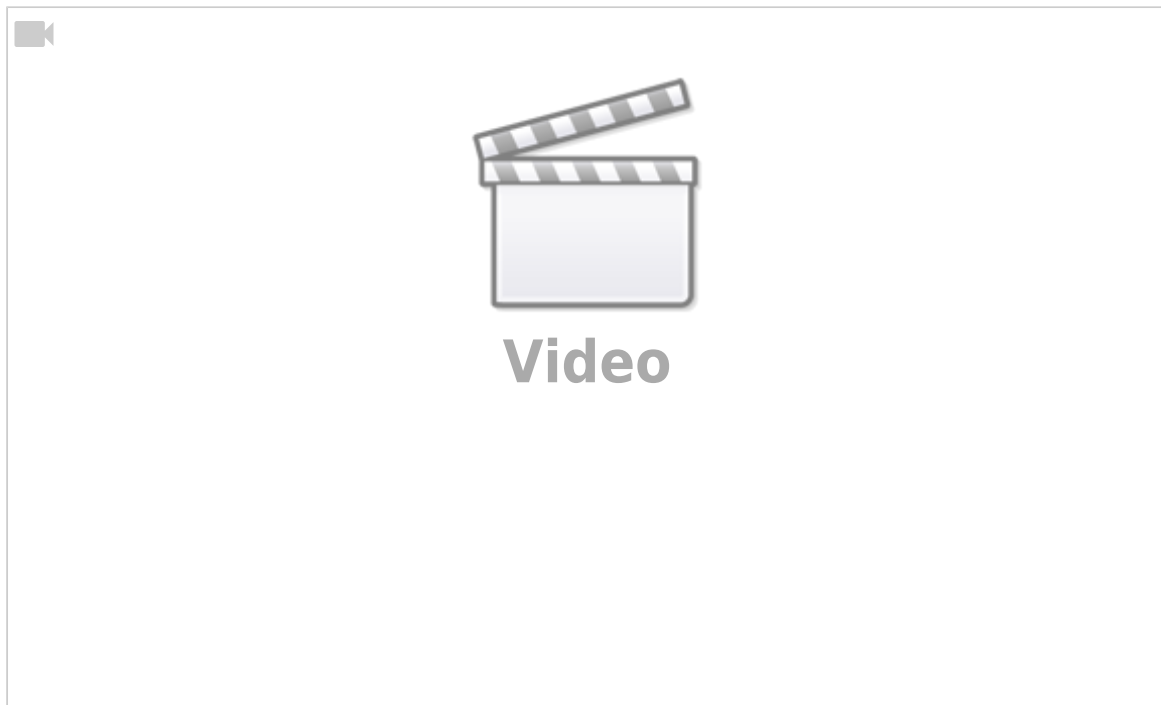
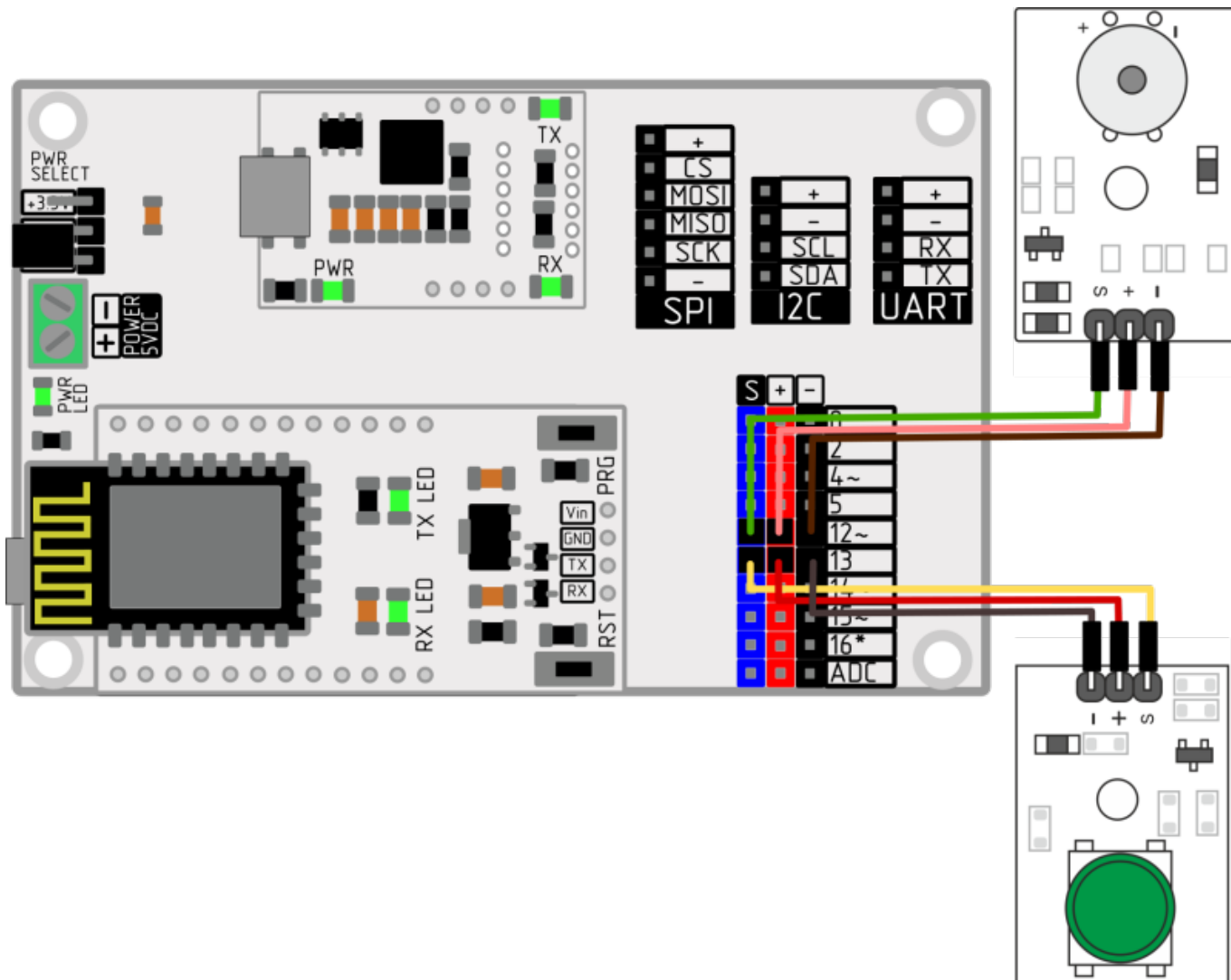


Урок 12. Словари



В этом уроке мы будем использовать ту же схему подключения что и в прошлом уроке:



Есть ещё один очень полезный контейнер (или коллекция значений) - это **словарь**. Он, судя по названию, очень похож на словарь. Как и в настоящих словарях, в этой коллекции сначала идёт слово, а потом его определение. Нам будет удобно воспользоваться словарём, чтобы описать ноты.

Люди с музыкальным слухом (если у Вас он тоже музыкальный, и Вы много экспериментировали с частотой звучания зуммера) определили что музыкальным нотам соответствует определённая частота, например, ноте do - 262 Гц, ноте re - 294 Гц и так далее. Такое соответствие частот нотам как раз и удобно хранить в словаре.

Создадим словарь для хранения соответствия частот нотам. Словарь обозначается фигурными скобками (например, пустой словарь выглядит так: `notes = {}`). В фигурных скобках сначала указывается ключ (в нашем случае название ноты, например, "do"), через двоеточие указывается значение, которое соответствует ключу (в нашем случае ноте do соответствует значение 262). Через запятую можно указать сколько угодно пар ключ - значение. Для двух нот определяться словарь будет так:

```
notes = {"do": 262, "re": 294}
```

Подобно спискам, к каждому элементу можно обратиться, только в словарях обращаются по ключу. Ключ в таком случае указывается в квадратных скобках. Например, считать в переменную частоту ноты do можно так:

```
do_freq = notes["do"]
```

Добавить значение в уже существующий словарь можно просто установив новое значение для нового ключа:

```
notes["mi"] = 330
```

Проще представить словарь с помощью таблицы из двух колонок, где первая колонка - это ключ, а вторая - значение, соответствующее ключу.

Ключ	Значение
«do»	262
"re"	294
"mi"	330

А теперь немного модифицируем прошлый скрипт, и проиграем все ноты по очереди при нажатии на кнопку.

```
# импорт модулей
from machine import Pin, PWM
from time import sleep

# создадим ШИМ вывод buz на 12 выводе
buz = PWM(Pin(12, Pin.OUT))

# словарь нот
notes = {}
notes["do"] = 262
notes["re"] = 294
notes["mi"] = 330
notes["fa"] = 349
notes["sol"] = 392
notes["la"] = 440
notes["si"] = 494

# кнопка подключена к выводу 13
but = Pin(13, Pin.IN)

# функция для воспроизведения звука определённой частоты
def sound(new_freq):
    buz.freq(new_freq) # установка частоты из параметра
    buz.duty(512) # установить заполнение в 512
    sleep(1) # задержка в 1 секунду
    buz.duty(0) # установить заполнение в 0
    sleep(0.5) # задержка в половину секунды

# функция для воспроизведения всех звуков
def show_all_sounds():
    sound(notes["do"]) # вызов функции sound со значением ноты
    sound(notes["re"]) # вызов функции sound со значением ноты
```

```
sound(notes["mi"]) # вызов функции sound со значением ноты
sound(notes["fa"]) # вызов функции sound со значением ноты
sound(notes["sol"]) # вызов функции sound со значением ноты
sound(notes["la"]) # вызов функции sound со значением ноты
sound(notes["si"]) # вызов функции sound со значением ноты
```

основной цикл программы

```
while True:
    if but.value():
        show_all_sounds() # если нажали на кнопку, воспроизведём все звуки
```

Запомнить:

- Словарь - это контейнер для хранения пар ключ - значение
- Словарь обозначается фигурными скобками (notes = {})
- В словарь можно добавлять значения, удалять их или заменять
- Из словаря можно считать значение по ключу (notes[«do»])

[Предыдущий урок](#)

[Следующий урок](#)

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

Permanent link:
https://know.gikkon.ru/main/gikkon_start/p1_l12

Last update: **2023/10/06 14:15**

