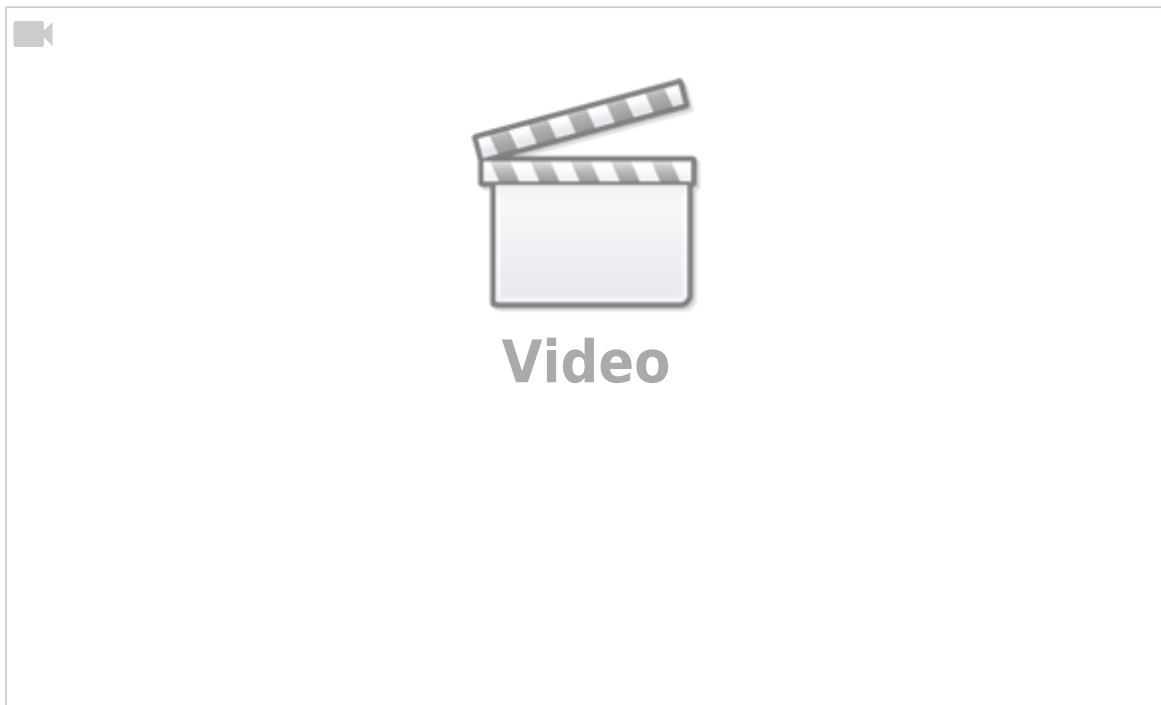
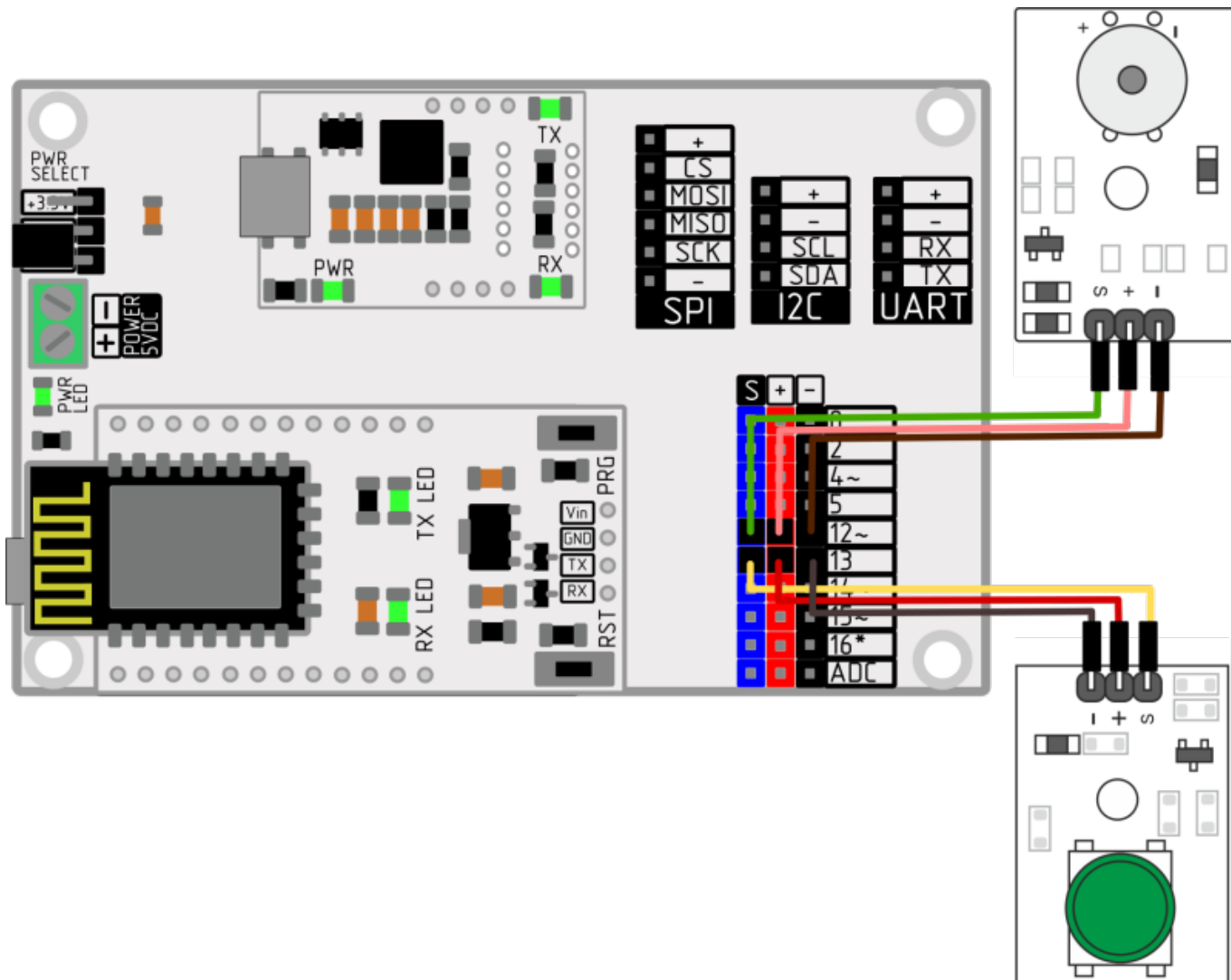


## Урок 13. Цикл for



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



Можно заметить что в предыдущих уроках мы писали код не очень универсальным. Например, в прошлом уроке, проигрывая нотный стан, мы вручную писали как проиграть каждую ноту. Но если добавить ещё несколько нот, на октаву повыше. Или несколько нот проиграть не один раз, то придётся снова переписывать несколько функций. Для того чтобы исправить это неудобство, мы воспользуемся ещё одним видом циклов - это циклы **for**.

Цикл `for` используется для выполнения действий ограниченное число раз. Синтаксис будет таким. После ключевого слова `for` (с английского *for* - для) указывается переменная, затем ключевое слово `in` (с английского *in* - в) и указывается коллекция, откуда будет браться эта переменная.

Перепишем функцию `show_all_sounds()` из предыдущего урока и сделаем её более универсальной. Для этого сначала введём переменную типа список значений и укажем ноты, которые мы хотим воспроизвести.

```
# СПИСОК НОТ
song = ["do", "re", "mi", "fa", "sol", "la", "si"]
```

Затем заменим функцию `show_all_sounds()`, где опишем цикл для прохождения по каждой ноте из списка.

```
# функция для воспроизведения всех звуков
def show_all_sounds():
```

```
for note in song:
    sound(notes[note])
```

Теперь наша функция стала намного универсальнее. И для того чтобы воспроизвести новую песню, достаточно поменять переменную `song`.

Полностью скрипт будет выглядеть следующим образом:

```
# 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

# список нот
song = ["do", "re", "mi", "fa", "sol", "la", "si"]

# 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():
    for note in song:
        sound(notes[note])

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

Запомнить:

- Цикл `for` используется для выполнения действий ограниченное число раз
- Сначала указывается переменная, в которую при каждом проходе цикла будет помещаться значение, а затем коллекция значений, из которой будут браться значения

[Предыдущий урок](#)

[Следующий урок](#)

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

Permanent link:  
[https://know.gikkon.ru/main/gikkon\\_start/p1\\_I13](https://know.gikkon.ru/main/gikkon_start/p1_I13)

Last update: **2023/10/06 14:16**

