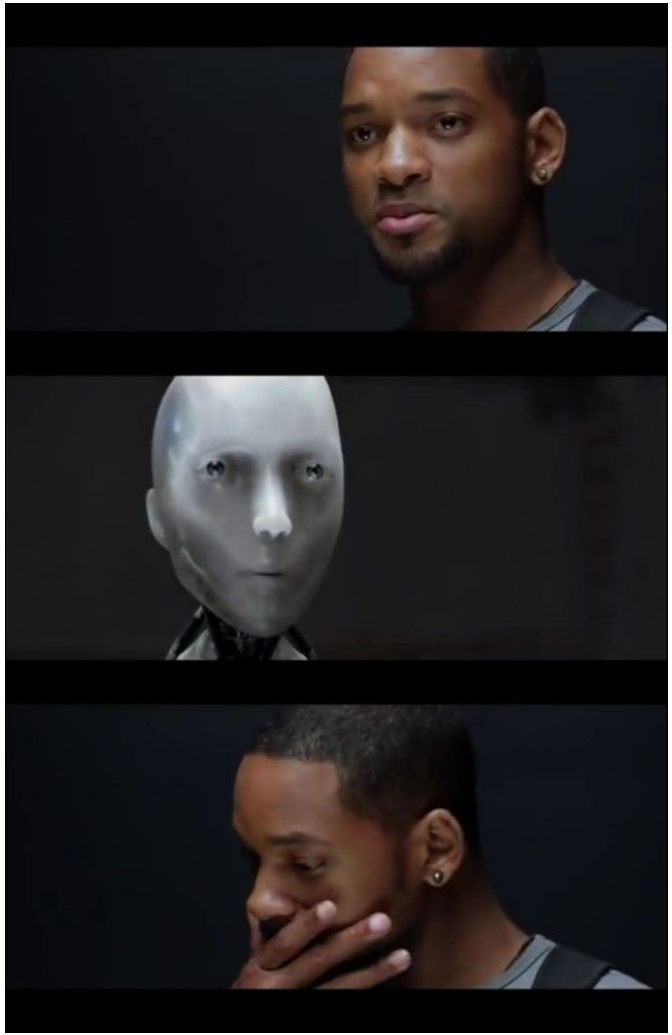
18. Роботика

20 декември 2022

MEME time



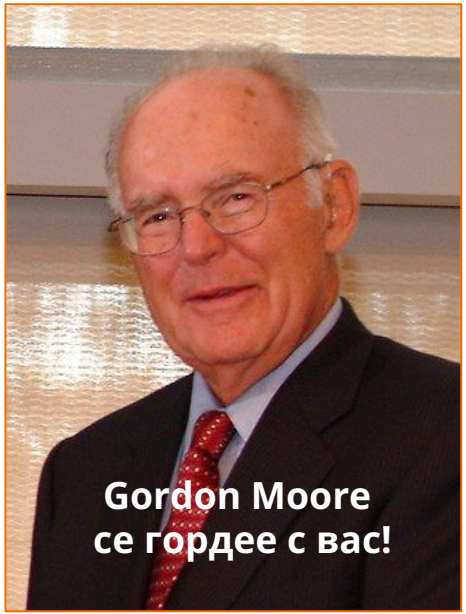
Can a robot create a symphony?
Can a robot turn a canvas into a beautiful masterpiece?

Can a robot submit his homework before the last day of the deadline?

Can you?

Някои неща просто не се променят - 03.11.2022

Но първо - какво е това?



Въпрос

Каква е разликата между конкурентност и паралелизъм?

Конкурентност: Когато две изчисления нямат ясно дефинирана последователност на изпълнение.
Паралелизъм: Две изчисления, които реално се изпълняват едновременно.

Въпрос+

Каква е разликата между `threading.Semaphore` и `threading.Lock`?

Lock позволява само една нишка, а Semaphore позволява няколко.

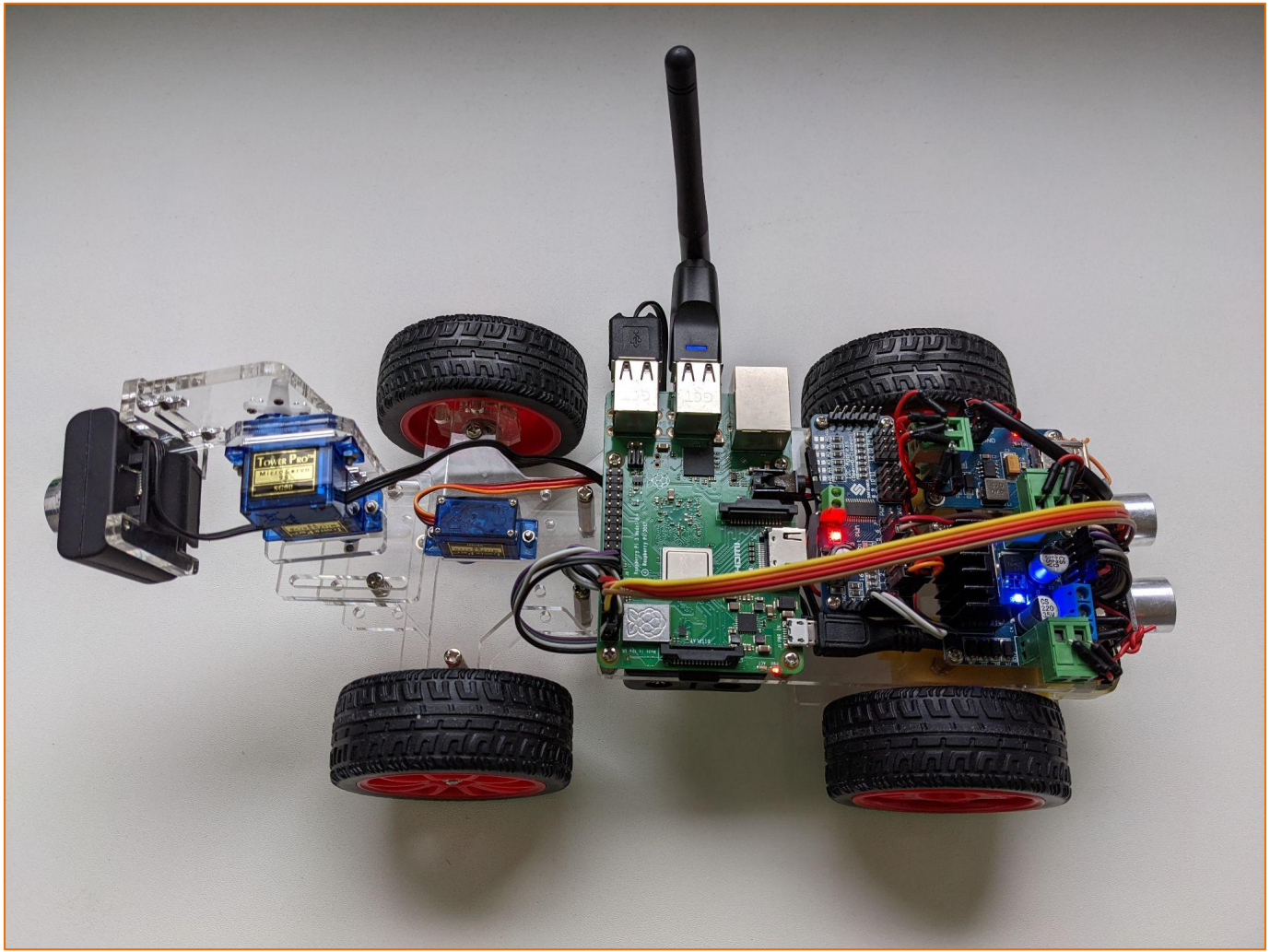
Въпрос++

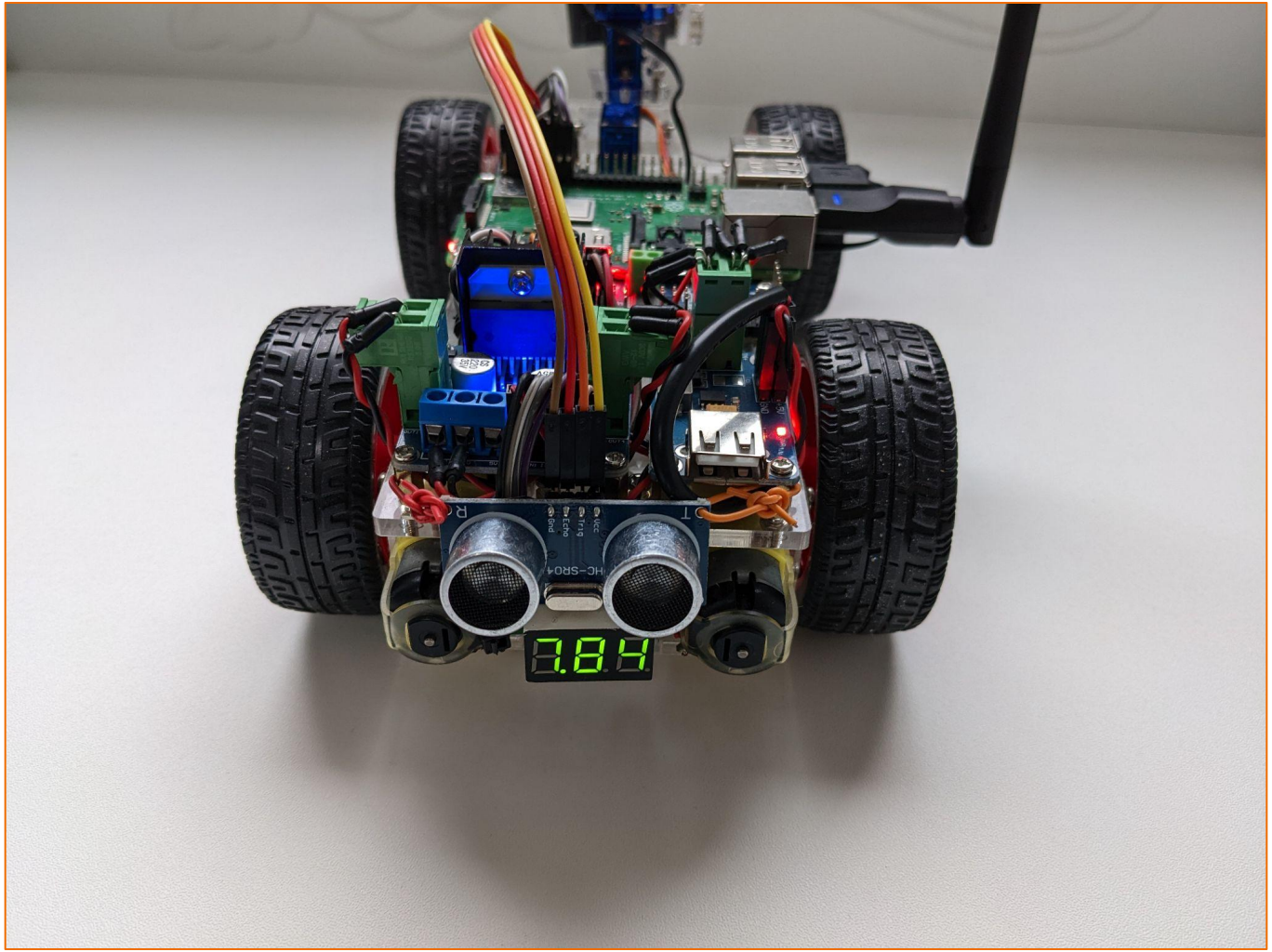
За какво се използва `os.fork`?

За разделяне на програмата в два отделни процеса.

Какво ще правим днес?

Комплектът преди да запретнете ръкави


TOWER PRO


HC-SR04
Vcc
Trig
Echo
Gnd
7.84

Интерфейсът изглежда така

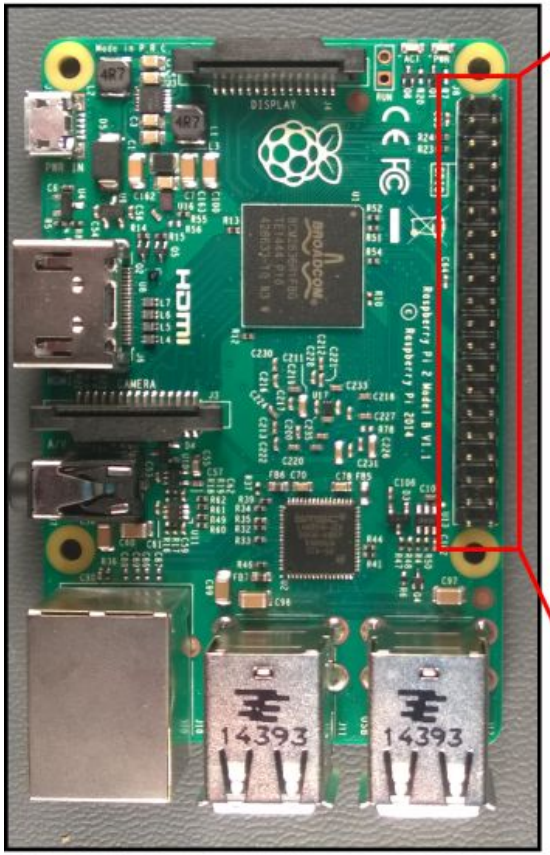
Да разгледаме компонентите

- Raspberry Pi 3 Model B+
- Sunfounders DC-DC converter module
- HC-SR04 Ultrasonic Distance Sensor
- L298N Motor Driver
- PCA9685 16 Channel Servo I2C Module
- 3x SG90 180° Servo
- 2x DC Gear Motor
- USB Video Camera
- 7 segment voltmeter
- USB Wifi adapter

Raspberry Pi

- Напълно функционален едноплатков компютър
- CPU, GPU, RAM, Networking, USB, HDMI, Audio
- Но най-важното - GPIO:
  - General Purpose Input/Ouput
- Моята версия е 3B+:
  - 1.2 GHz 64-bit quad core ARM processor
  - 1GB RAM
- Последният модел е 4B:
  - 1.5 GHz 64-bit quad core ARM processor
  - 8GB RAM
- Има специално Линукс дистро - Raspberry Pi OS
  - Debian базирано

Pinout



Alternate Function							Alternate Function
	3.3V PWR	1		2	5V PWR		
I2C1 SDA	GPIO 2	3		4	5V PWR		
I2C1 SCL	GPIO 3	5		6	GND		
	GPIO 4	7		8	UART0 TX		
	GND	9		10	UART0 RX		
	GPIO 17	11		12	GPIO 18		
	GPIO 27	13		14	GND		
	GPIO 22	15		16	GPIO 23		
	3.3V PWR	17		18	GPIO 24		
SPI0 MOSI	GPIO 10	19		20	GND		
SPI0 MISO	GPIO 9	21		22	GPIO 25		
SPI0 SCLK	GPIO 11	23		24	GPIO 8		SPI0 CS0
	GND	25		26	GPIO 7		SPI0 CS1
	Reserved	27		28	Reserved		
	GPIO 5	29		30	GND		
	GPIO 6	31		32	GPIO 12		
	GPIO 13	33		34	GND		
SPI1 MISO	GPIO 19	35		36	GPIO 16		SPI1 CS0
	GPIO 26	37		38	GPIO 20		SPI1 MOSI
	GND	39		40	GPIO 21		SPI1 SCLK

Захранването

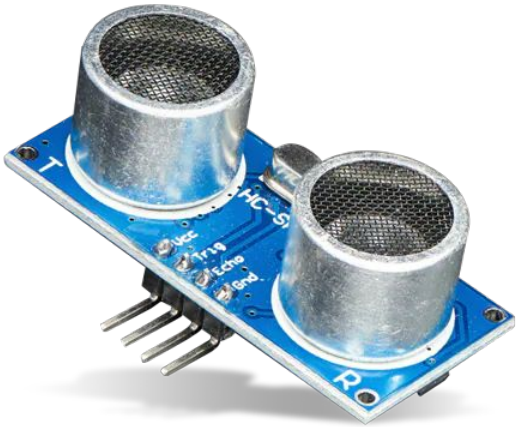
ИЗХОД->
5V/3A

ВХОД->
3V-12V

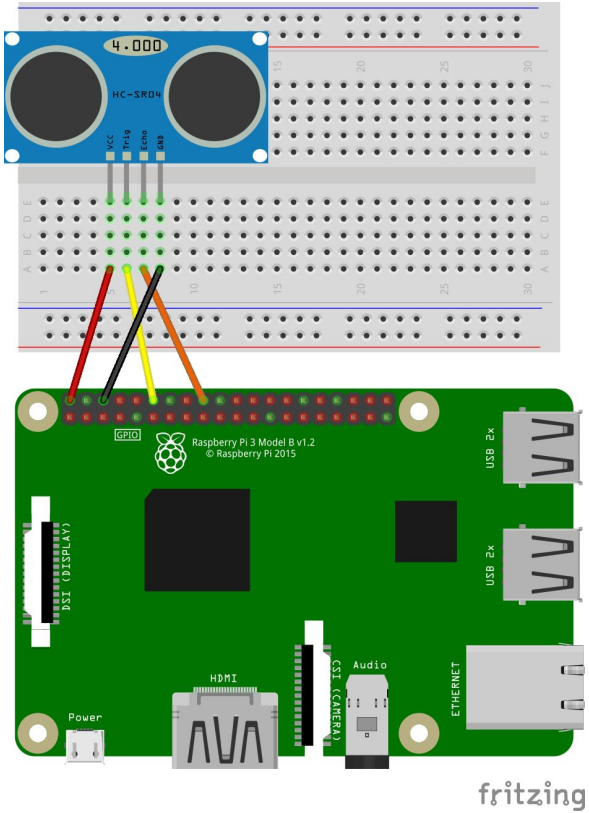
RPi.GPIO

- pip install RPi.GPIO
- `import RPi.GPIO as GPIO`
- `GPIO.setmode(GPIO.BOARD/GPIO.BCM)`
- `GPIO.getmode()`
- `GPIO.setwarnings(False)`
- `GPIO.setup(channel, GPIO.IN/OUT)`
- `GPIO.setup(channel, GPIO.OUT, initial=GPIO.LOW/HIGH)`
- `GPIO.input(channel)`
- `GPIO.output(channel, GPIO.LOW/HIGH)`
- `GPIO.cleanup()`
- `GPIO.RPI_INFO`
- `GPIO.VERSION`

HC-SR04 Ultrasonic Distance Sensor

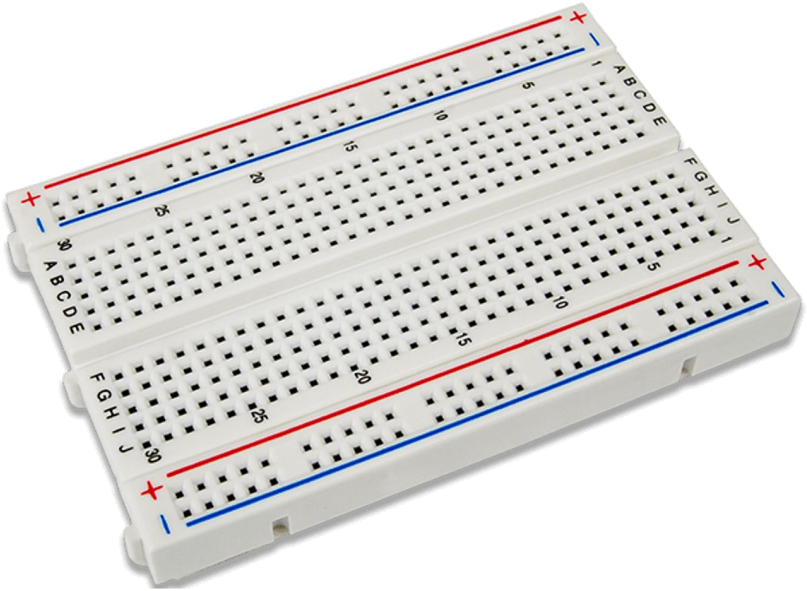


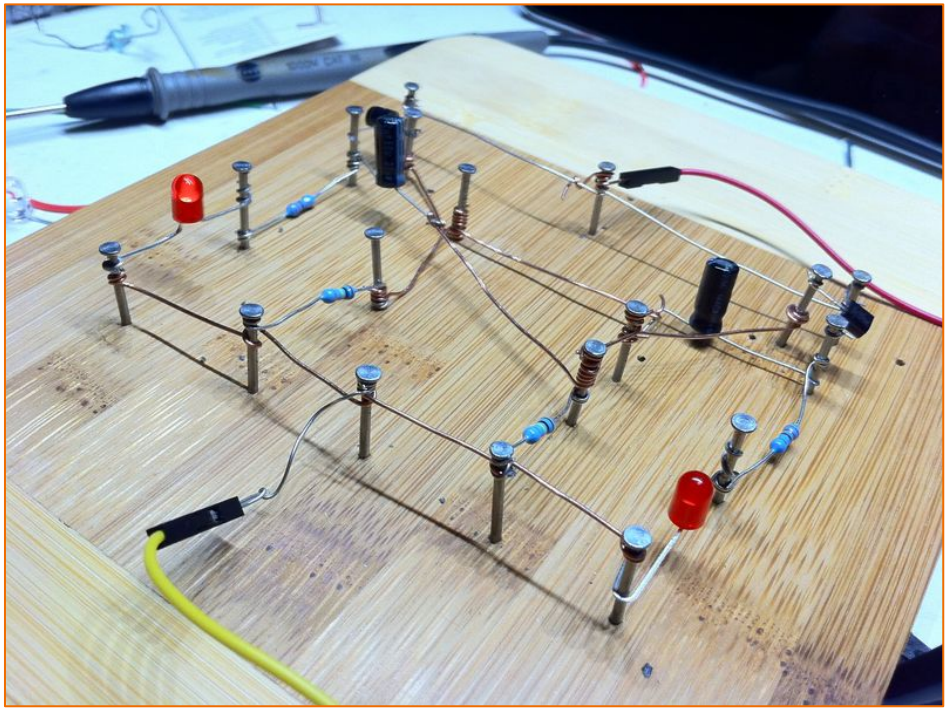
- VCC
- GND
- TRIG
- ECHO



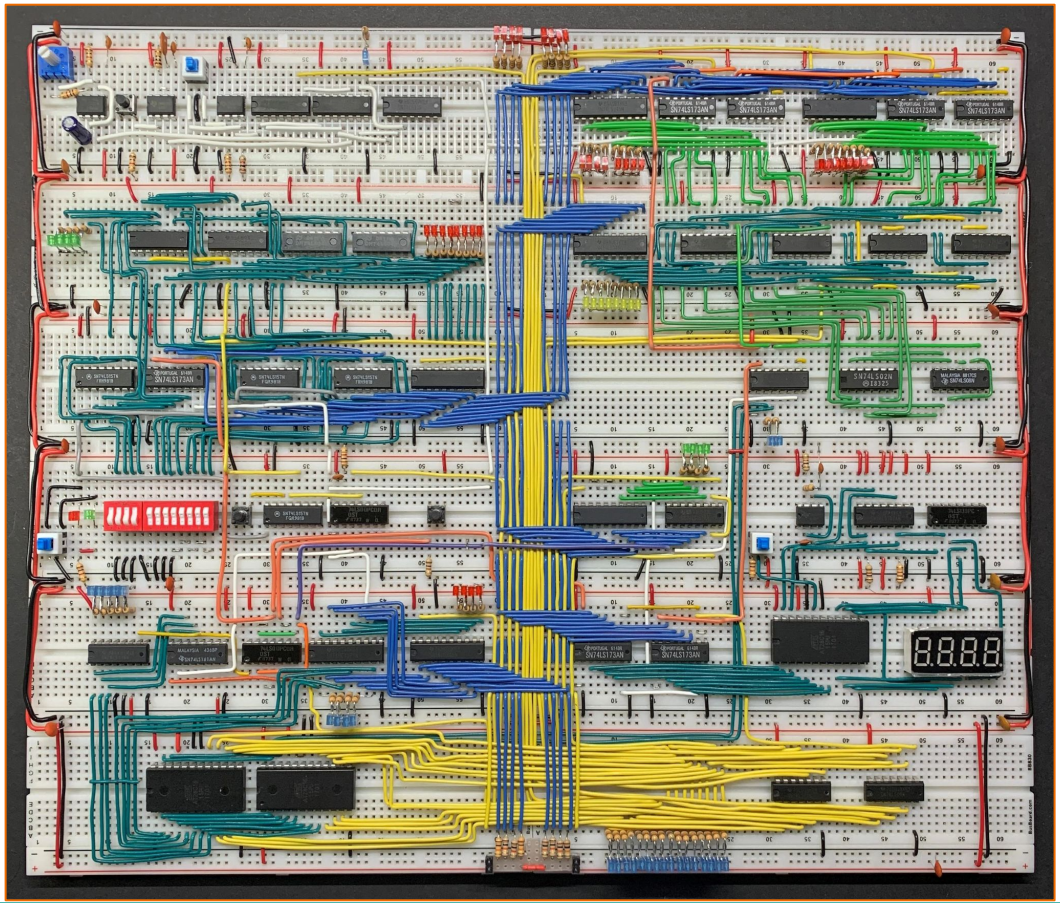
Breadboard?

Breadboard

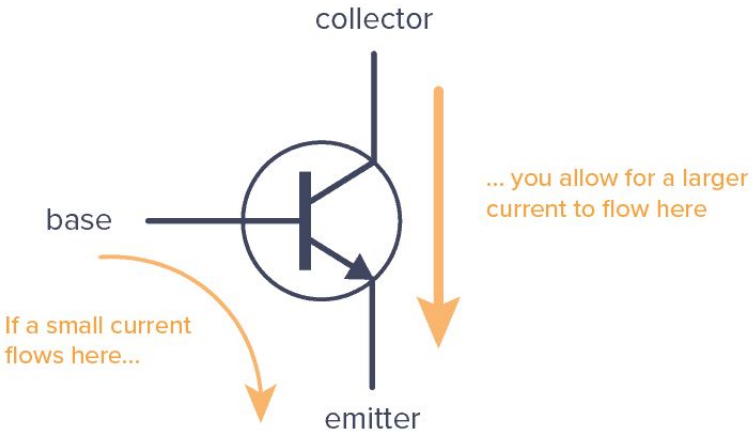


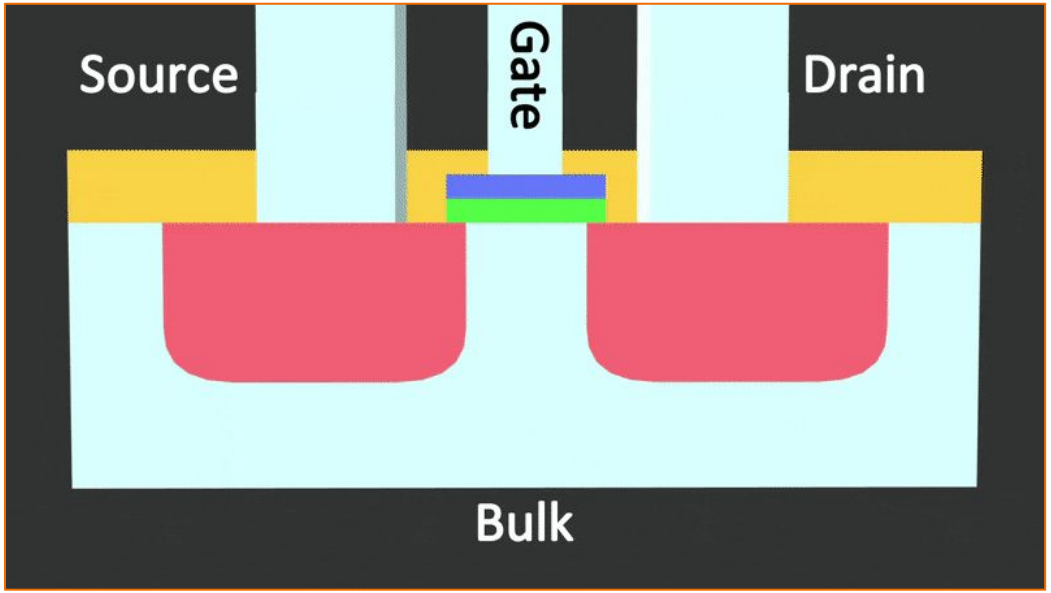


Breadboard 8-bit computer by Ben Eater



Основа - транзистор





Преди транзисторите имаше ламБи

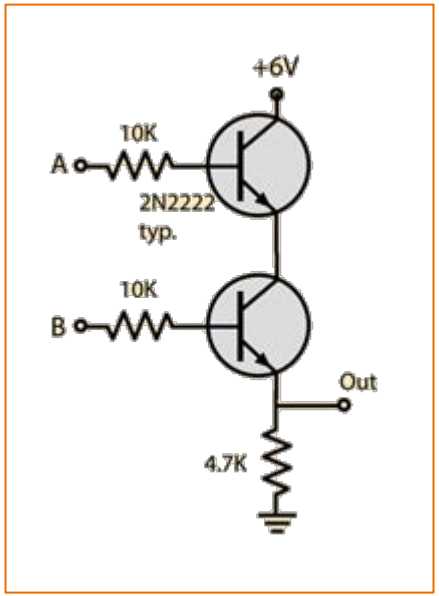
Логиката - Gates

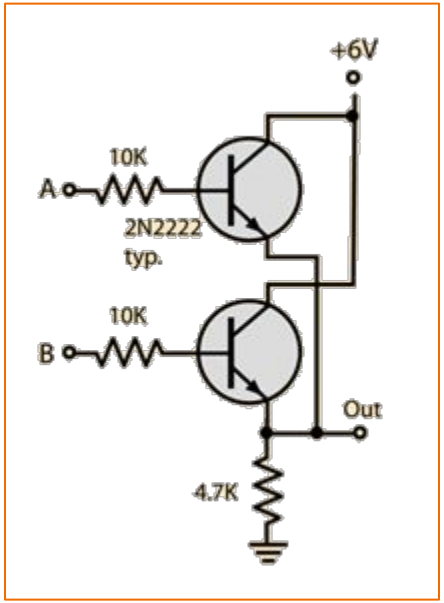
AND

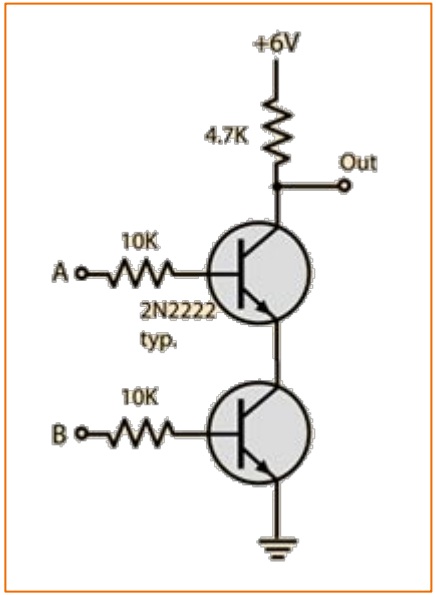
OR

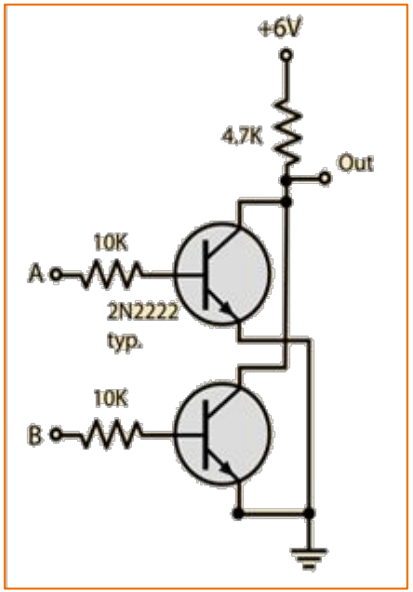
NAND

NOR

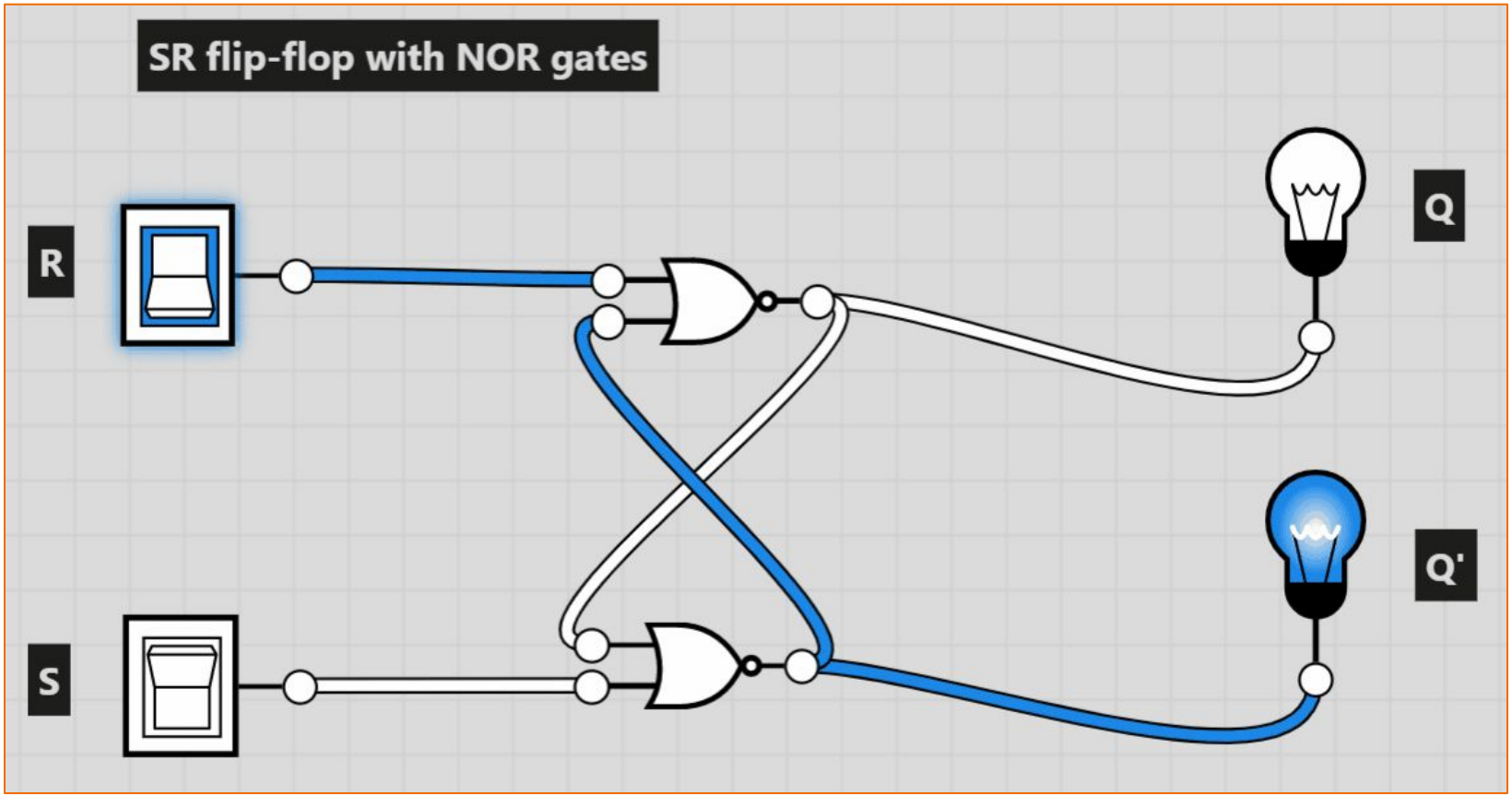




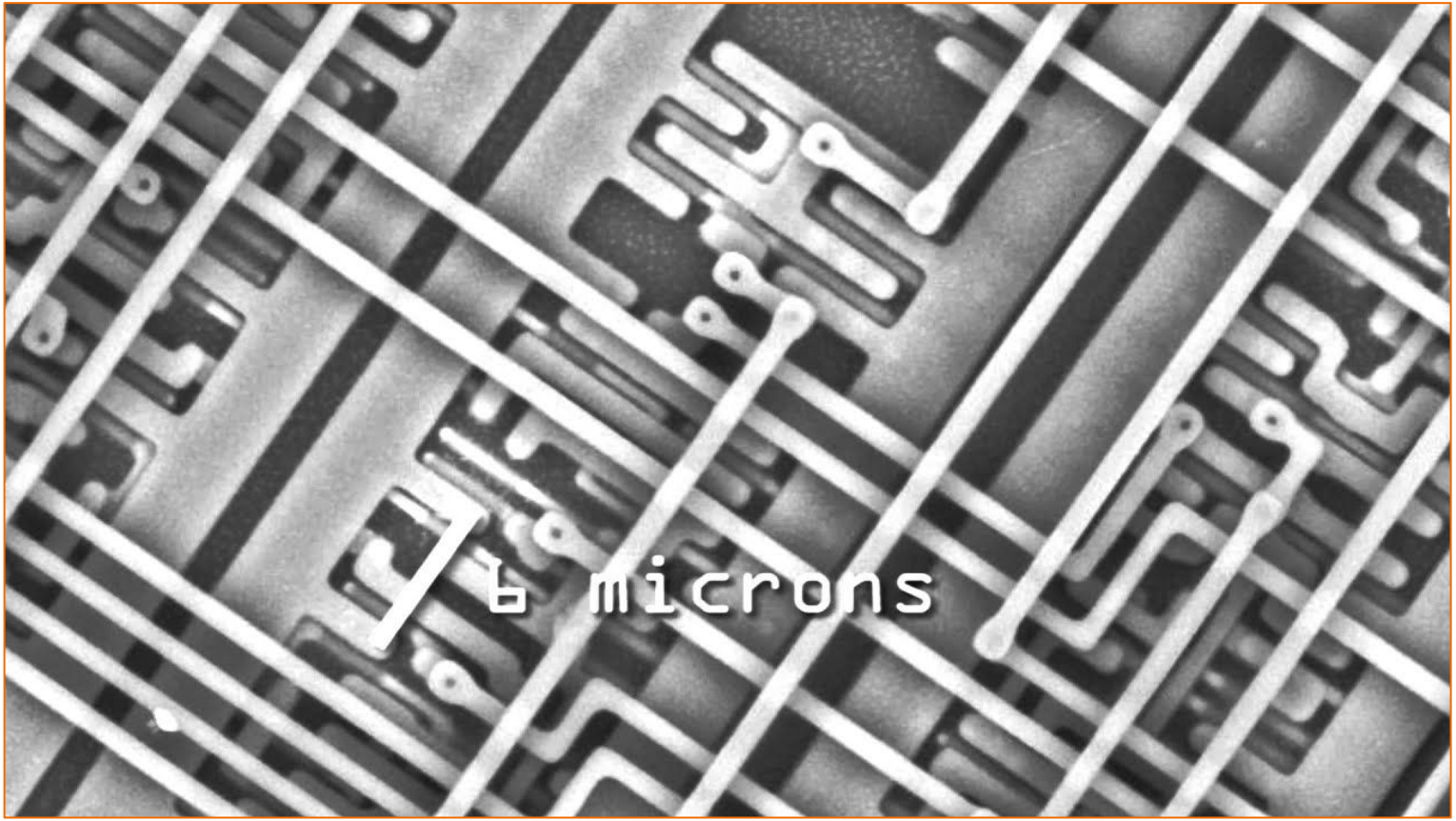




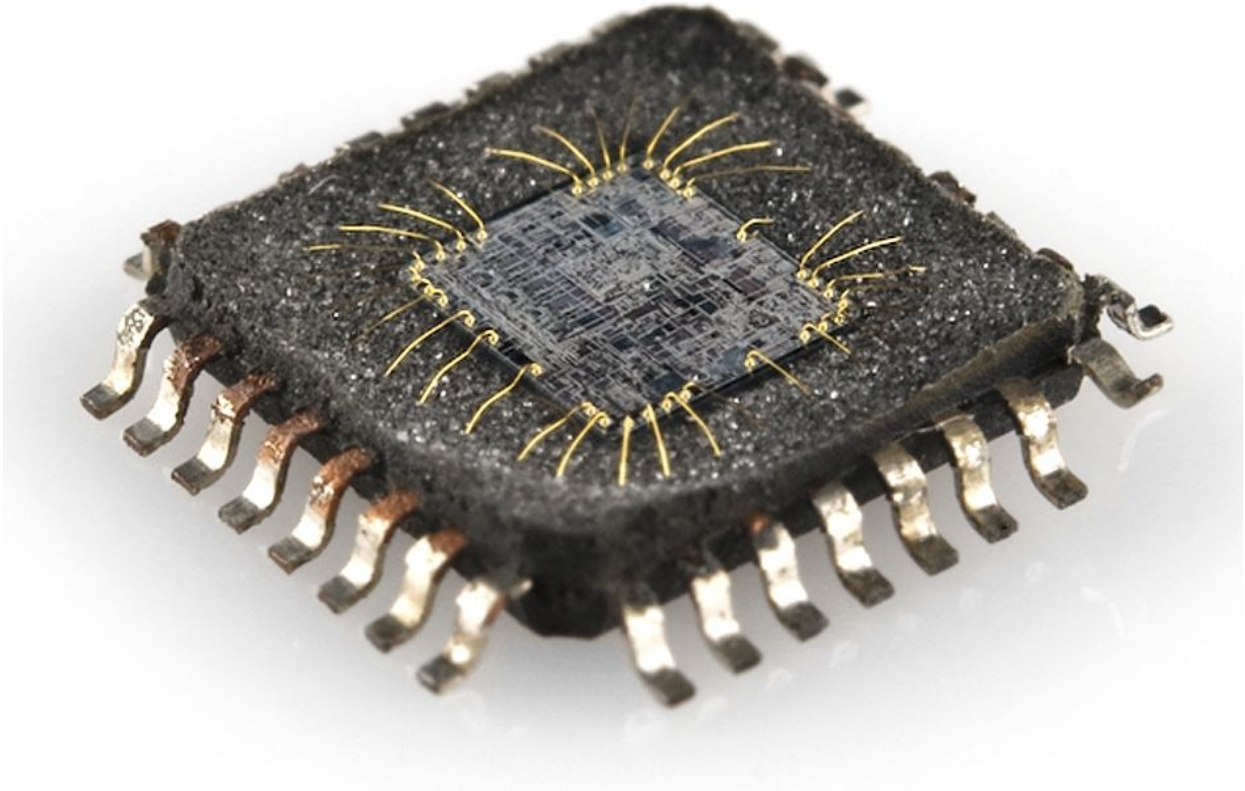
Flip-flop



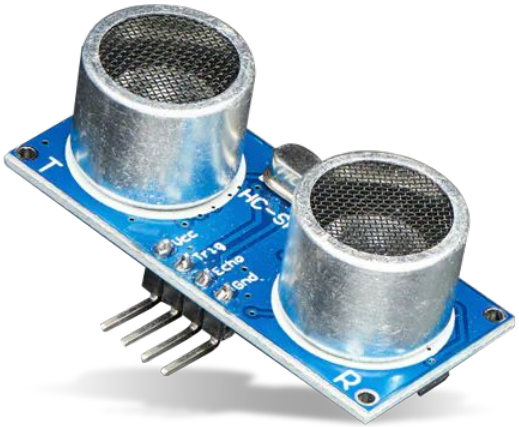
“Половин“ век по-късно



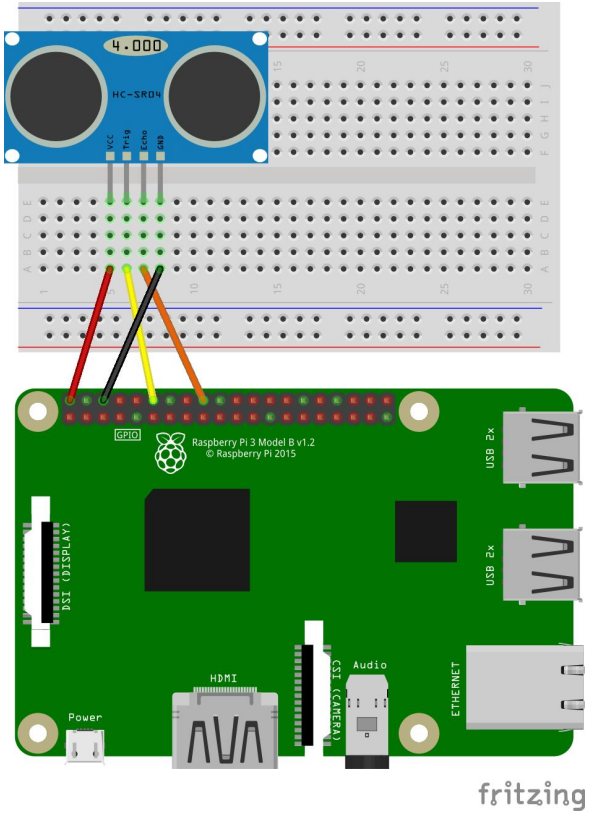
Integrated circuit



Да се върнем на сензора



- VCC
- GND
- TRIG
- ECHO



Кодът 1/2

```
import RPi.GPIO as GPIO
import time


class DistanceSensor:
    """HC-SR04 Ultrasonic distance sensor."""

    def __init__(self, trig, echo):
        """Initializator."""
        self._trig = trig
        self._echo = echo
        GPIO.setmode(GPIO.BCM)
        GPIO.setup(self._trig, GPIO.OUT, initial=GPIO.LOW)
        GPIO.setup(self._echo, GPIO.IN)

    def __del__(self):
        """Release the pins."""
        GPIO.cleanup()

    …
```

Кодът 2/2

```
…

def read(self):
    """Calculate distance in cm and return it."""
    GPIO.output(self._trig, GPIO.HIGH)
    time.sleep(0.00001)
    GPIO.output(self._trig, GPIO.LOW)
    while not GPIO.input(self._echo):
        pulse_start = time.time()
    while GPIO.input(self._echo):
        pulse_end = time.time()
    pulse_duration = pulse_end - pulse_start
    distance = pulse_duration * 17165 # Half the speed of sound in cm/s
    return round(distance, 1)
```

Video Camera

Кодът 1/2

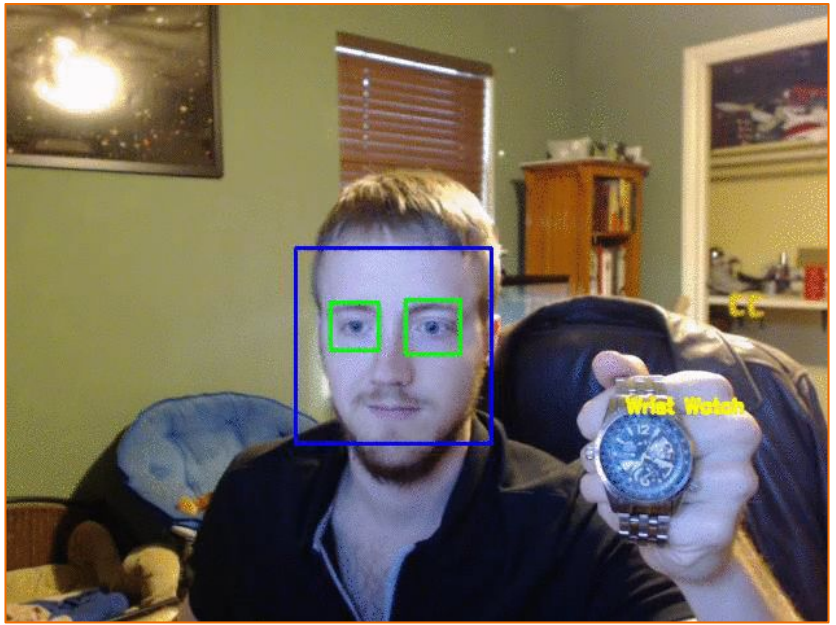
```
import cv2


class Camera:

    def __init__(self):
        self._video = cv2.VideoCapture(0)

    def __del__(self):
        self._video.release()
    …
```

Кодът 2/2

```
…
def _get_frame(self):
    success, image = self._video.read()
    if not success:
        print('Cannot read from video camera.')
        return False
    _, jpeg = cv2.imencode('.jpg', image)
    return jpeg.tostring()

def get_stream(self):
    while True:
        frame = self._get_frame()
        if frame:
            yield(b'--frame\n'
                  b'Content-Type: image/jpeg\n\n' + frame + b'\n\n')
```

Open CV

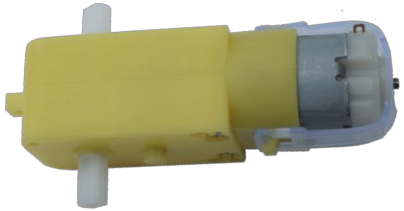
- pip install opencv-python
- CV - Computer Vision
- Реализирано е на C++
- Всевъзможна функционалност за обработване на изображения
- Включително ML неща като:
  - face recognition
  - Object detection
  - Motion tracking

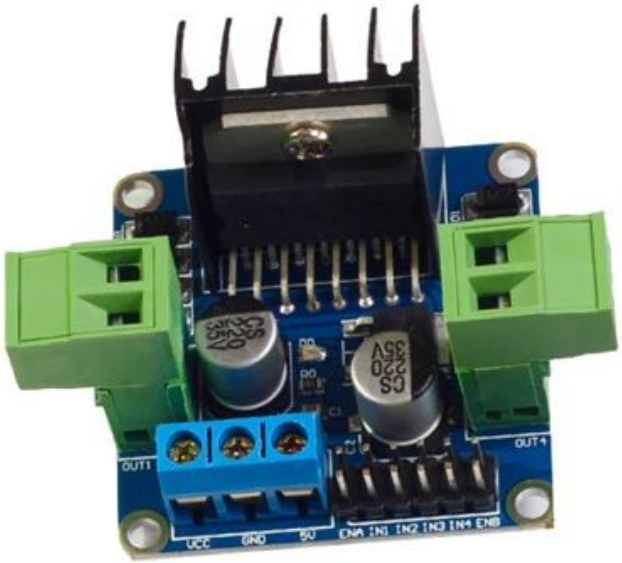


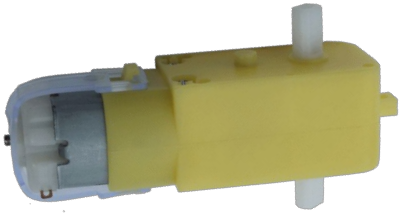
Да добавим face recognition и за нашия клас

```
# Machine learning магия, която си взимаш от нета:
face = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

def _add_faces(self, image):
    """Detect and highlight faces in an image."""
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    faces = face.detectMultiScale(gray, 1.1, 4)
    for (x, y, w, h) in faces:
        cv2.rectangle(image, (x, y), (x + w, y + h), (255, 0, 0), 2)
```

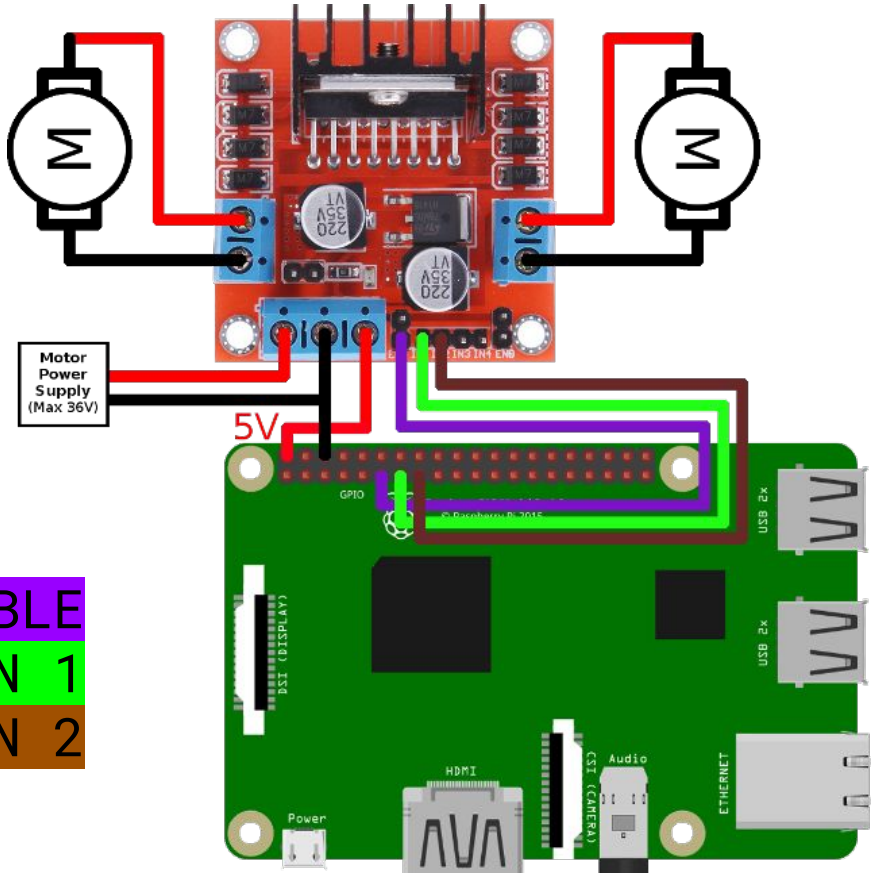
Motor Driver + Motors







Как работи?



- ENABLE
- DIRECTION 1
- DIRECTION 2

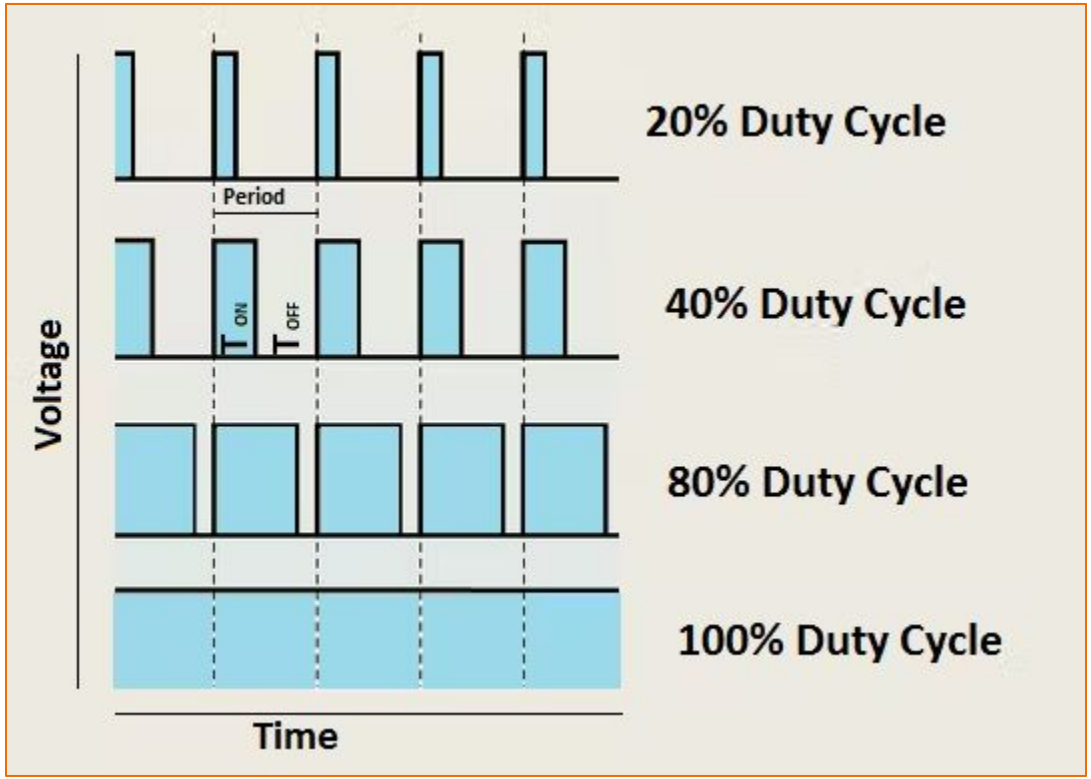
Кодът 1/2

```
import RPi.GPIO as GPIO


class Motor:
    "Basic motor, that can turn forwards and backwards."

    def __init__(self, fwd, bwd):
        """Initializator."""
        self._fwd = fwd
        self._bwd = bwd
        self._set_board()

    def _set_board(self):
        """Reserve and initiate the GPIO pins."""
        GPIO.setmode(GPIO.BCM)
        GPIO.setup(self._fwd, GPIO.OUT, initial=GPIO.LOW)
        GPIO.setup(self._bwd, GPIO.OUT, initial=GPIO.LOW)

    def __del__(self):
        """Release all GPIO pins."""
        GPIO.cleanup()
```

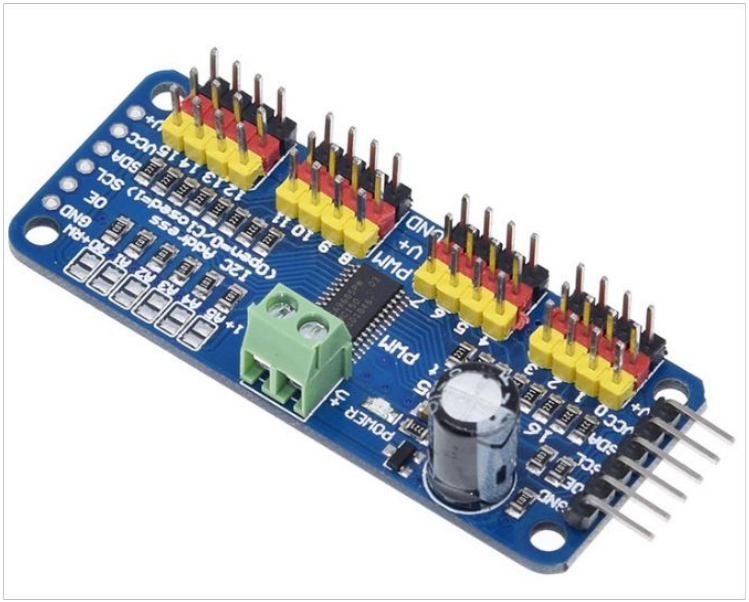
Кодът 2/2

```
def fwd(self):
    """Move forward."""
    GPIO.output(self._fwd, GPIO.HIGH)
    GPIO.output(self._bwd, GPIO.LOW)

def bwd(self):
    """Move backward."""
    GPIO.output(self._fwd, GPIO.LOW)
    GPIO.output(self._bwd, GPIO.HIGH)

def stop(self):
    """Stop moving."""
    GPIO.output(self._fwd, GPIO.LOW)
    GPIO.output(self._bwd, GPIO.LOW)
```

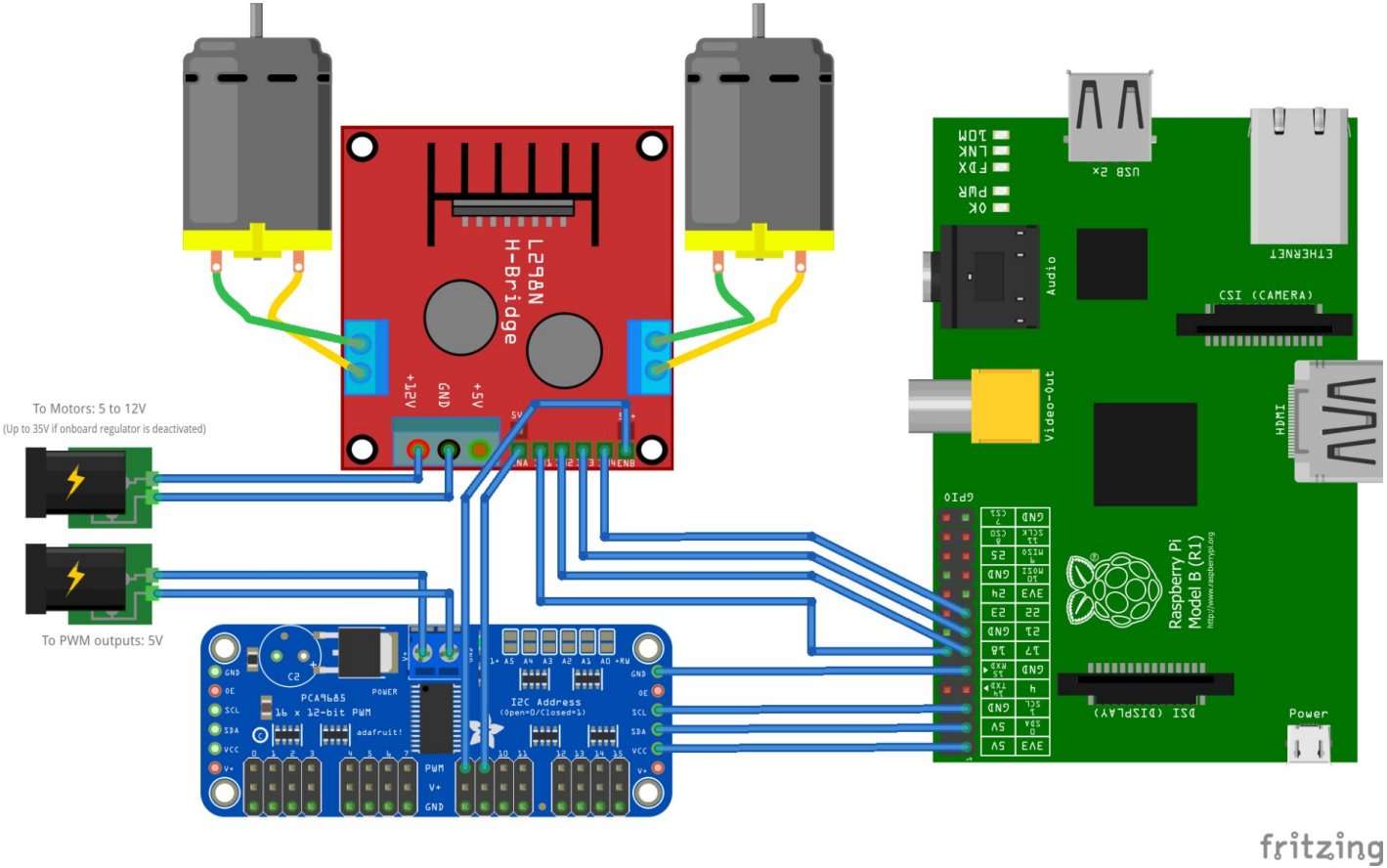
PWM (Pulse Width Modulation)



PCA9685 16 Channel Servo I2C Module

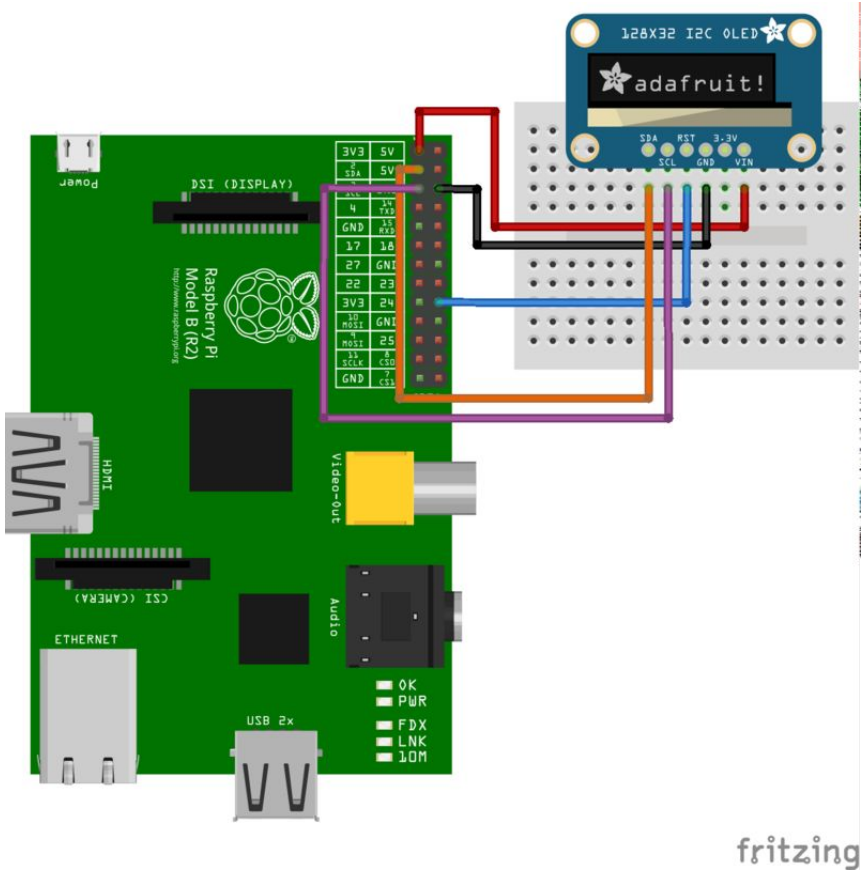


Как работи?

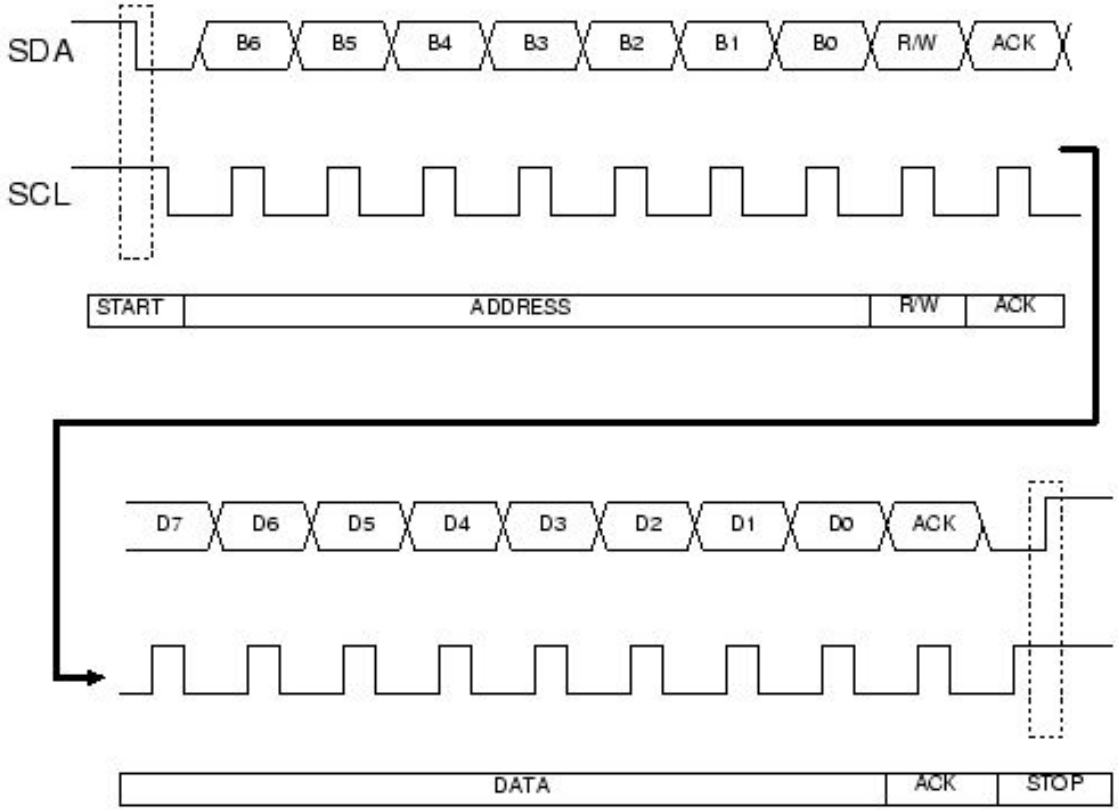

L298N
H-Bridge
+12V
GND
+5V
To Motors: 5 to 12V
(Up to 35V if onboard regulator is deactivated)
To PWM outputs: 5V
PCA9685
16 x 12-bit PWM
I2C Address
(Open=0/Closed=1)
Raspberry Pi
Model B (R1)
http://www.raspberrypi.org
USB 2x
ETHERNET
Audio
CSI (CAMERA)
Video-Out
HDMI
GPIO
DSI (DISPLAY)
Power
fritzing

I²C (Inter-Integrated Circuit) Interface

- VCC
- GND
- RST
- SCL
- SDA



I²C (Inter-Integrated Circuit) Interface

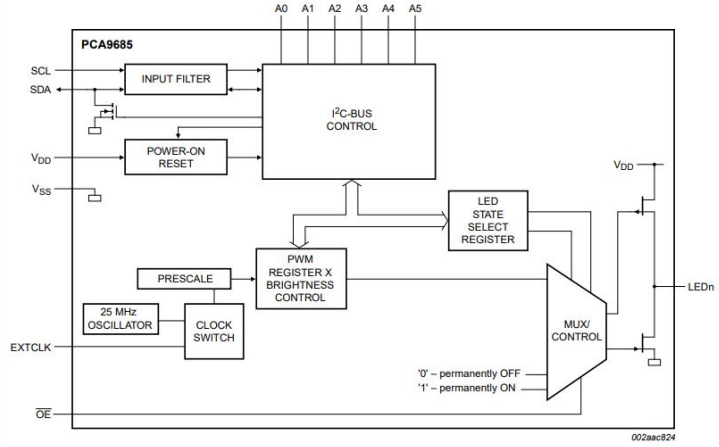


WTF?

PCA9685

16-channel, 12-bit PWM Fm+ I²C-bus LED controller

Rev. 4 — 16 April 2015 Product data sheet



Remark: Only one LED output shown for clarity.

Fig 1. Block diagram of PCA9685

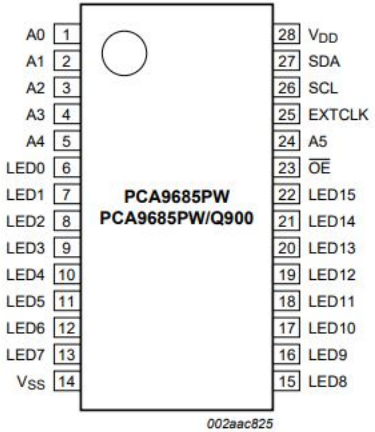


Fig 2. Pin configuration for TSSOP28

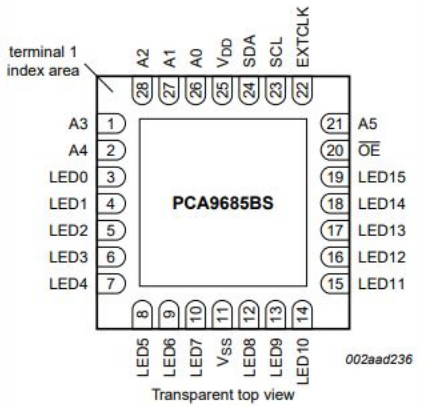


Fig 3. Pin configuration for HVQFN28

WTF?

Table 4. Register summary

Register# (decimal)	Register# (hex)	D7	D6	D5	D4	D3	D2	D1	D0	Name	Type	Function
0	00	0	0	0	0	0	0	0	0	MODE1	read/write	Mode register 1
1	01	0	0	0	0	0	0	0	1	MODE2	read/write	Mode register 2
2	02	0	0	0	0	0	0	1	0	SUBADR1	read/write	I^2C-bus subaddress 1
3	03	0	0	0	0	0	0	1	1	SUBADR2	read/write	I^2C-bus subaddress 2
4	04	0	0	0	0	0	1	0	0	SUBADR3	read/write	I^2C-bus subaddress 3
5	05	0	0	0	0	0	1	0	1	ALLCALLADR	read/write	LED All Call I^2C-bus address
6	06	0	0	0	0	0	1	1	0	LED0_ON_L	read/write	LED0 output and brightness control byte 0
7	07	0	0	0	0	0	1	1	1	LED0_ON_H	read/write	LED0 output and brightness control byte 1
8	08	0	0	0	0	1	0	0	0	LED0_OFF_L	read/write	LED0 output and brightness control byte 2
9	09	0	0	0	0	1	0	0	1	LED0_OFF_H	read/write	LED0 output and brightness control byte 3
10	0A	0	0	0	0	1	0	1	0	LED1_ON_L	read/write	LED1 output and brightness control byte 0

Кодът

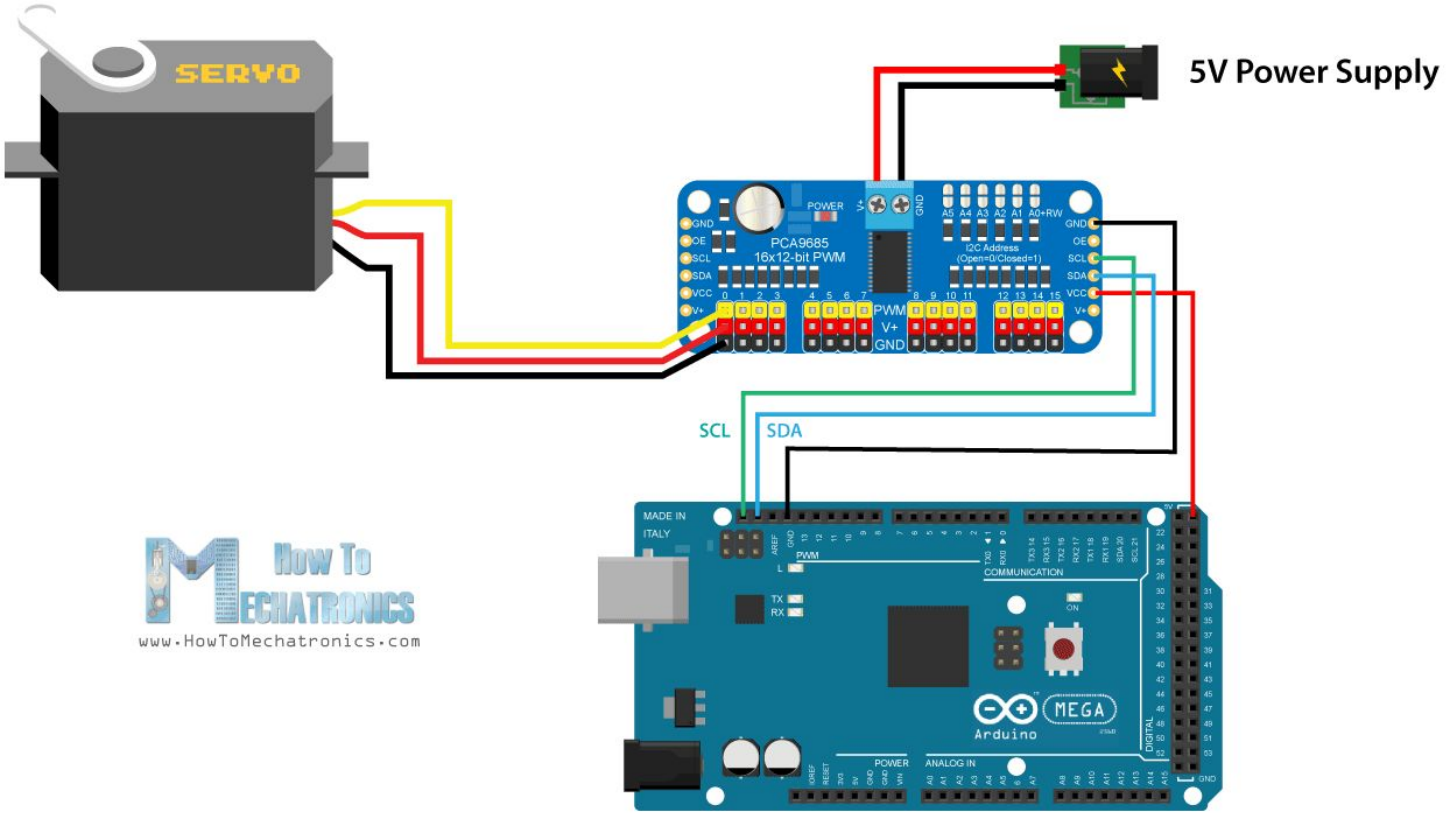
Имплементацията

```
from .hardware.pwm import Pwm

class Engine:
    "Engine of the car. It controls the motors' speed."

    def __init__(self, left, right):
        """Initializator."""
        self._pwm = Pwm()
        self._motor_left = left
        self._motor_right = right

    def set_speed(self, speed):
        """Set speed of both motors."""
        self._pwm.set_value(self._motor_left, 0, speed)
        self._pwm.set_value(self._motor_right, 0, speed)
```

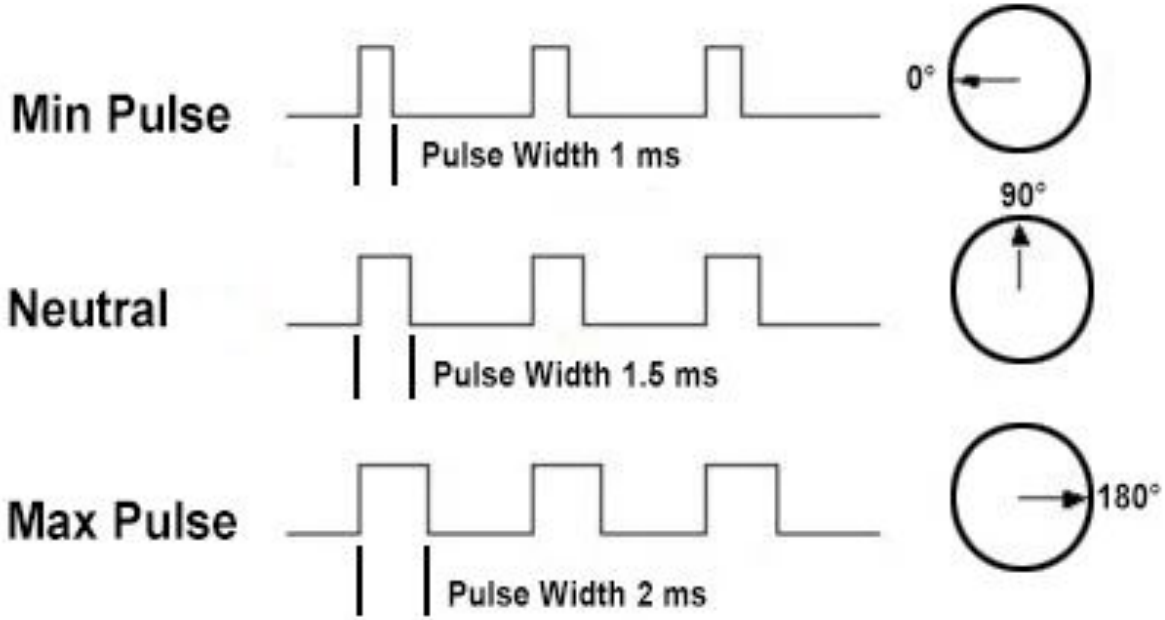
SG90 180° Servo

Как се свързва?

- VCC
- GND
- PWM



Как работи?


Min Pulse
Pulse Width 1 ms
0°
Neutral
Pulse Width 1.5 ms
90°
Max Pulse
Pulse Width 2 ms
180°

Кодът

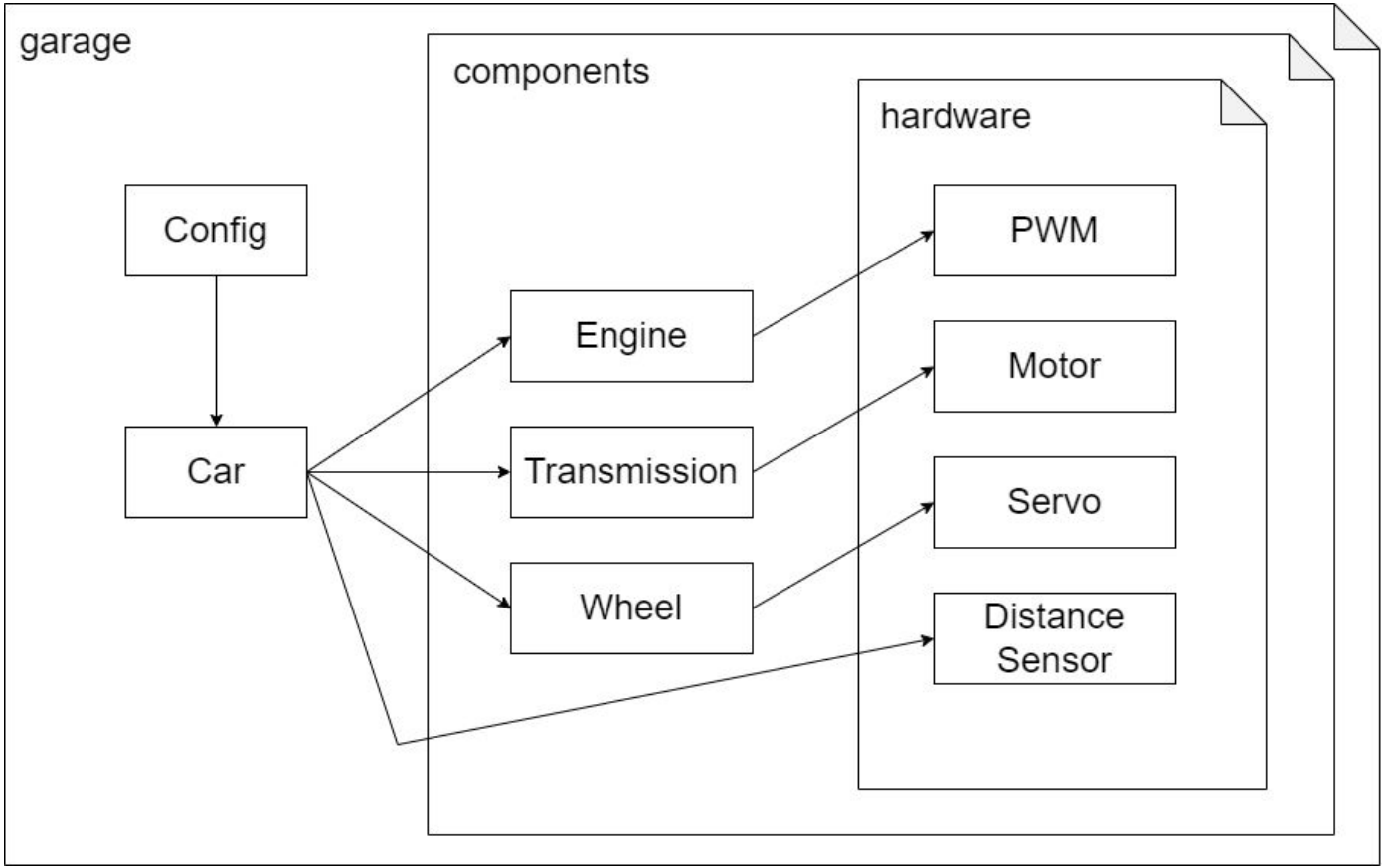
```
from adafruit_servokit import ServoKit


class Servo:
    "Servo that can rotate from about 0 to 180 degrees"

    def __init__(self, port, min, max):
        """Initializator."""
        self._port = port
        self._iface = ServoKit(channels=8).servo[port]
        self._min = min
        self._max = max

    def set(self, value):
        """Set rotation based on an angle value."""
        value = max(value, self._min)
        value = min(value, self._max)
        self._iface.angle = value
```

Имплементация

```
class Wheel:
    "Wheel that can make the car turn left/right"

    def __init__(self, port, min, max):
        """Initializator."""
        self._min = min
        self._max = max
        self._servo = Servo(port, min, max)

    def left(self):
        """Turn left."""
        self._servo.set(self._min)

    def right(self):
        """Turn right."""
        self._servo.set(self._max)

    def center(self):
        """Turn center."""
        self._servo.set(self._min + (self._max - self._min) // 2)
```

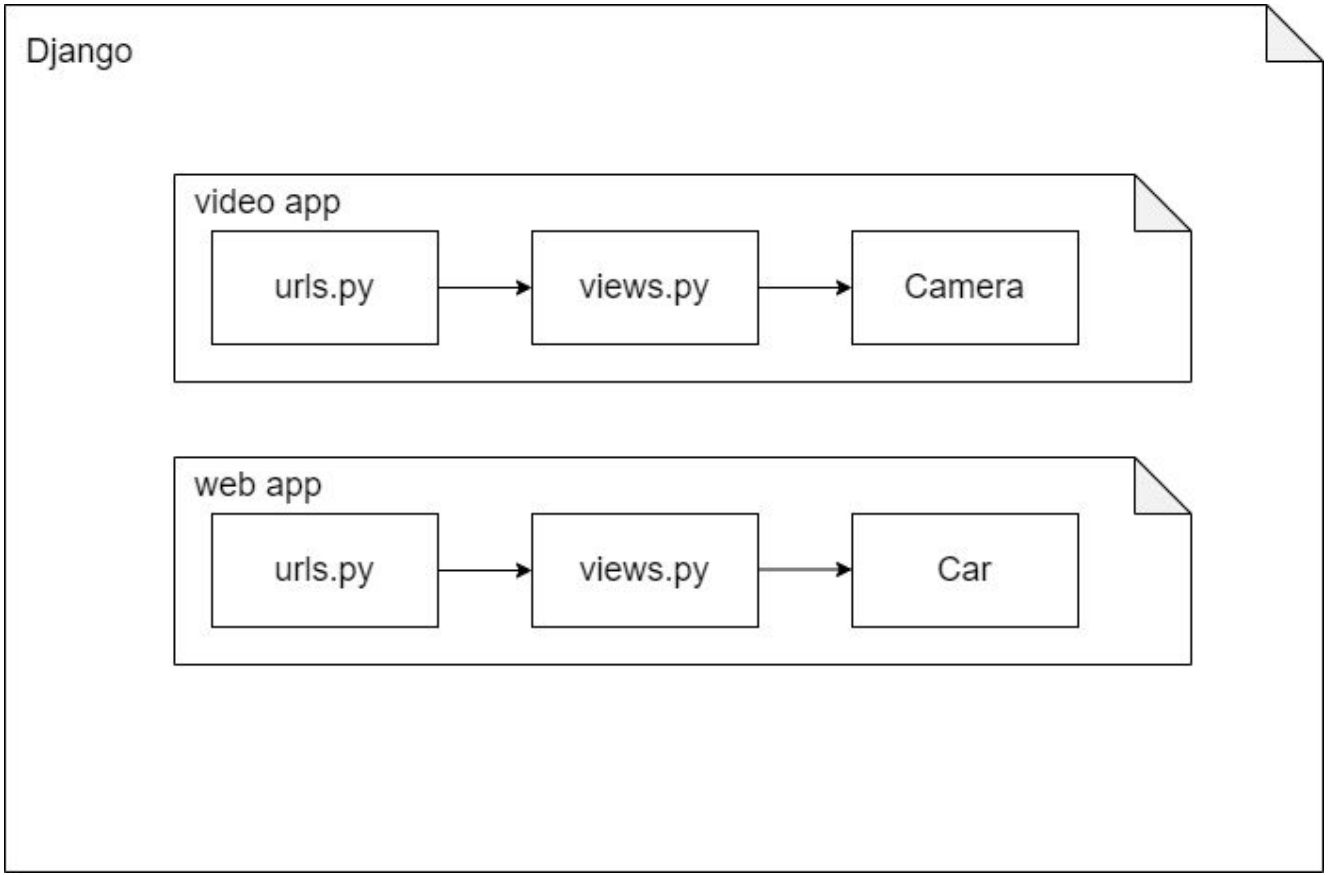
Дизайн

https://github.com/gvkunchev/fmi-robotics/tree/main/car/web/garage



Interface

https://github.com/gvkunchev/fmi-robotics/tree/main/car



Въпроси?