

Обработка данных с датчиков, фильтрация СИГНАЛОВ

Изотов Илья, Кувшинов Олег

Ilya.izotov@urfu.ru, o.a.kuvshinov@urfu.ru

Принцип работы УЗ-датчика

Ультразвуковой дальномер HC-SR04

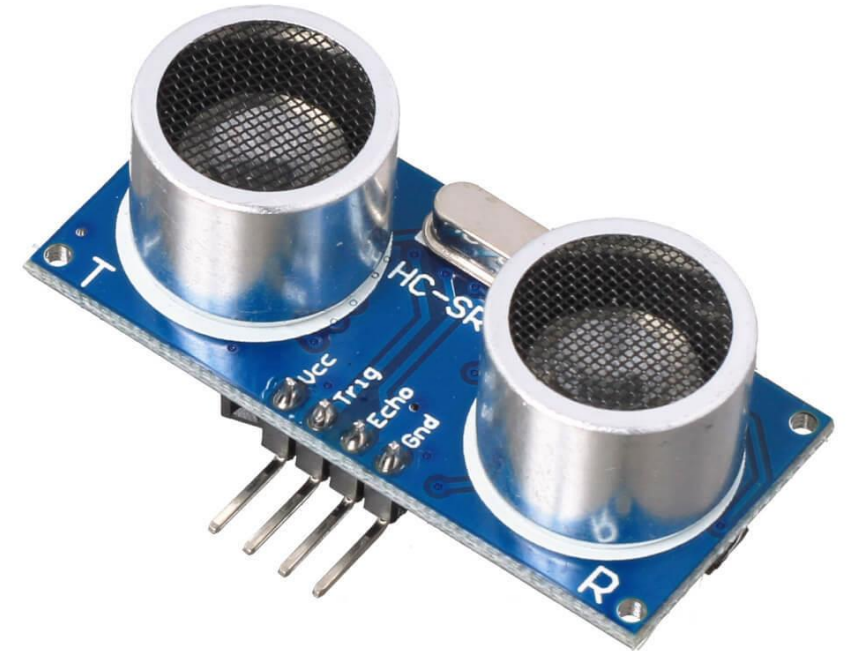
Диапазон расстояний: 2–400 см

Эффективный угол наблюдения: 15°

Напряжение питания: 5 В

Потребление в режиме тишины: 2 мА

Потребление при работе: 15 мА



Принцип работы УЗ-датчика

$$\text{Дистанция} = (t_2 - t_1) / 2 * 340 * 100$$

Датчик



Время

Препятствие

Получение значений УЗ-датчика

```
42 def get_distance(self):
43     """
44     Получите функцию ультразвукового расстояния с возвращаемым значением расстояния в см.
45     """
46     time_count = 0
47     time.sleep(0.01)
48     gpio.digital_write(gpio.TRIG, True) # Активация высокого уровня на контакте Trig.
49     time.sleep(0.000015) # Отправить прямоугольную волну высокого уровня 10-15 мкс
50     gpio.digital_write(gpio.TRIG, False) # Отключение
51     while not gpio.digital_read(gpio.ECHO): # Ожидание, пока вывод Echo перейдет с низкого уровня на высокий.
52         pass
53     t1 = time.time() # Время начала высокого уровня Echo
54     while gpio.digital_read(gpio.ECHO): # Ожидание, пока вывод Echo перейдет с высокого уровня на низкий уровень.
55         if time_count < 2000: # Обнаружение тайм-аута для предотвращения бесконечных циклов
56             time_count = time_count + 1
57             time.sleep(0.000001)
58         pass
59     else:
60         print("NO ECHO receive! Please check connection")
61         break
62     t2 = time.time() # Время окончания высокого уровня Echo.
63     distance = (t2 - t1) * 340 / 2 * 100 # Продолжительность высокого уровня вывода «Эхо» – это время
64     # от излучения до возврата ультразвуковой волны, то есть время x скорость звуковой волны/2
65     # равно значению одностороннего расстояния ультразвуковой волны от объекта.
66     # print("distance is %d" % distance) # Распечатать значение расстояния
67     if distance < 500: # Нормальное значение расстояния обнаружения
68         # print("distance is %d"%distance)
```

Обработка показаний датчика

1. Использование фильтров
2. Установка датчика на серводвигатель и измерение расстояний вокруг

Опрос датчика

Метод для опроса датчика: `ultrasonic.send_distance()`

Файл: `xr_startmain.py`

```
121         elif cfg.CRUIISING_FLAG == cfg.CRUIISING_SET['normal']:  
122             ultrasonic.send_distance()  
123             time.sleep(1)
```

Прием данных

```
1 import socket
2
3 host = "ip_address"
4 port = 2001
5 # Создаем подключение
6 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
7 s.connect((host, port))
8 try:
9     while True:
10         data = s.recv(1024)
11         if not data: break
12         print(list(data))
13 finally:
14     # Закрываем соединение
15     s.close()
16     print("Соединение закрыто")
```

ИК-датчик препятствий

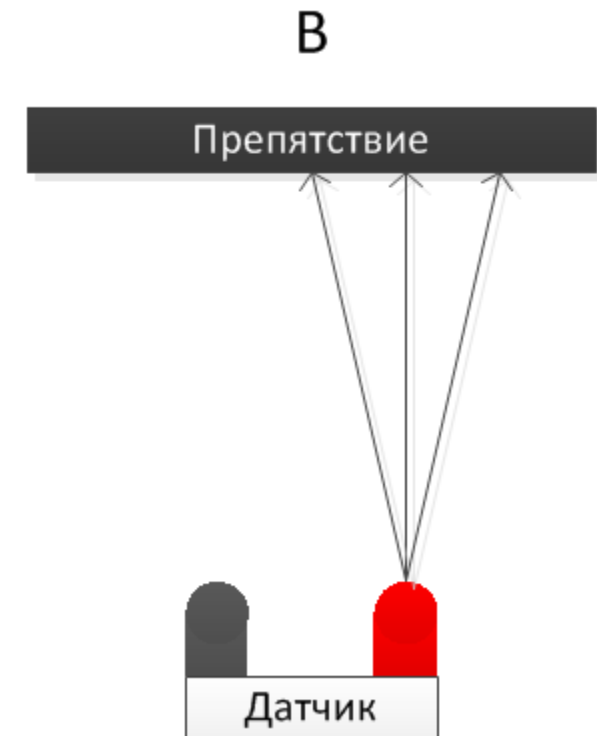
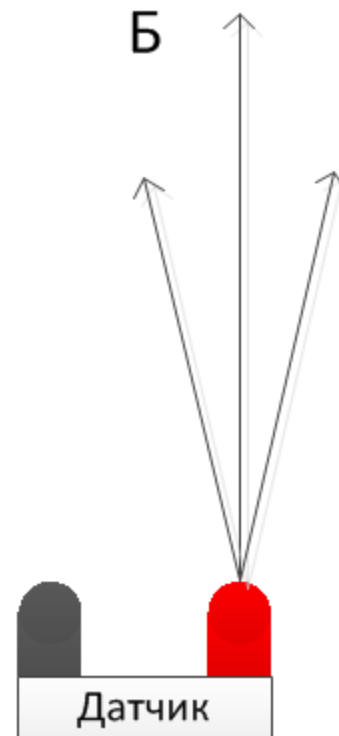
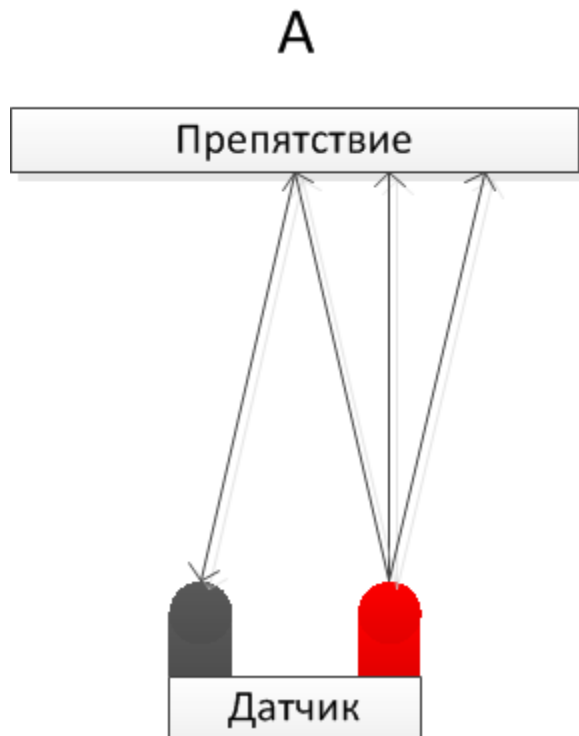
Расстояние срабатывания: 3–80 см (регулируется)

Напряжение питания: 5 В

Потребляемый ток: 100 мА



Принцип работы ИК-датчика



Опрос датчиков

Создадим метод для опроса датчика: `infrared.send_ir()`

Файл: `xr_infrared.py`: импорт класса `Socket()`

```
21 from xr_socket import Socket
22 socket = Socket()
23
```

Опрос датчиков

Файл: xr_infrared.py: создание метода send_ir()

```
31     def send_ir(self):
32         buf = bytes([gpio.digital_read(gpio.IR_L),
33                     gpio.digital_read(gpio.IR_R),
34                     gpio.digital_read(gpio.IRF_L),
35                     gpio.digital_read(gpio.IR_M),
36                     gpio.digital_read(gpio.IRF_R)])
37
38         try:
39             socket.sendbuf(buf)
40         except Exception as e: # Обработка исключений
41             print('send_distance error:', e) # Вывод в консоль ошибки
42         else:
43             buf = []
```

Опрос датчиков

Файл `xr_startmain.py`: вызов нового метода

```
121     elif cfg.CRUIISING_FLAG == cfg.CRUIISING_SET['normal']:
122         ultrasonic.send_distance()
123         infrared.send_ir()
124         time.sleep(1)
```

Задание

1. Определить, для каких целей может использоваться каждый датчик на площадке
2. Откалибровать ИК-датчик по дальности срабатывания
3. Разработать алгоритм, объединяющий показания с ИК- и УЗ-датчиков для повышения точности определения препятствий

Понравилось занятие?

Сканируй QR код и оставляй
свой отзыв



Спасибо за внимание

Изотов Илья, Кувшинов Олег

Ilya.izotov@urfu.ru, o.a.kuvshinov@urfu.ru