

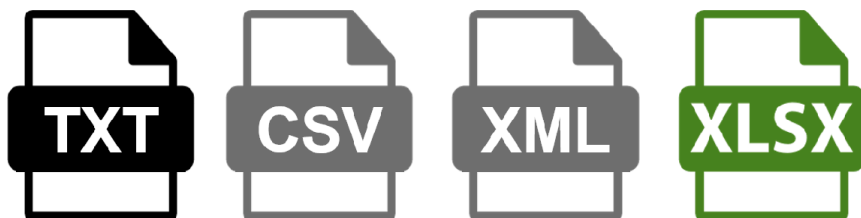
Импорт данных в MATLAB

Обзор различных методов получения данных непосредственно из MATLAB

Содержание

Работа с текстовыми файлами и таблицами.....	1
Интерактивный импорт таблиц.....	1
Импорт таблиц с помощью команды 'readtable'.....	2
Чтение бинарных файлов.....	2
Работа с медиа-файлами.....	3
Импорт изображений.....	3
Работа с вебкамерой.....	4
Подключение к базам данных.....	4
Получение данных из интернета.....	5
Базовые возможности.....	5
Продвинутая работа с WEB.....	6
Работа с оборудованием.....	6
Работа с измерительным оборудованием.....	6
Работа с устройствами сбора данных (DAQ).....	7
Сбор данных с микроконтроллеров.....	7
Работа с Arduino.....	7
Получение данных с датчиков телефона.....	9

Работа с текстовыми файлами и таблицами



- Поддерживаются любые текстовые файлы, в том числе специализированные форматы (CSV, XML, Excel, ...)
- Удобный импорт таблиц

Интерактивный импорт таблиц

В окне инструмента можно настроить параметры импорта и сгенерировать MATLAB код для автоматизации

```
uiimport('tsunamis.csv')
```

Импорт таблиц с помощью команды 'readtable'

Команда `readtable` работает с любыми текстовыми, CSV и Excel-файлами, в которых хранятся табличные данные. Имеет большое количество настроек импорта.

```
tsunamis = readtable('tsunamis.csv')
```

```
tsunamis = 162×20 table
```

...

	Latitude	Longitude	Year	Month	Day	Hour
1	-3.8000	128.3000	1950	10	8	3
2	19.5000	-156.0000	1951	8	21	10
3	-9.0200	157.9500	1951	12	22	NaN
4	42.1500	143.8500	1952	3	4	1
5	19.1000	-155.0000	1952	3	17	3
6	43.1000	-82.4000	1952	5	6	NaN
7	52.7500	159.5000	1952	11	4	16
8	50.0000	156.5000	1953	3	18	NaN
9	-2.4000	147.4000	1953	6	27	NaN
10	-18.3000	178.2000	1953	9	14	0

⋮

Чтение бинарных файлов



Зная формат хранения данных, можно прочитать любой бинарный файл

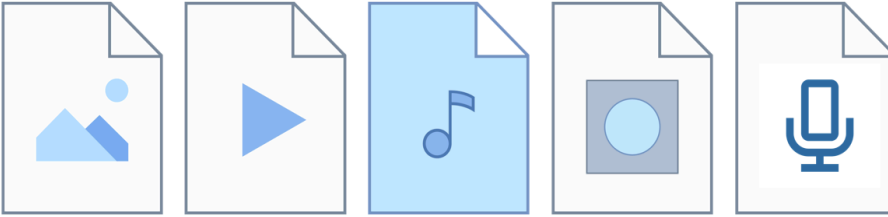
```
fileID = fopen('data.bin');  
data = fread(fileID, [4, Inf], 'double')
```

```
data = 4×4
```

```
16     2     3    13  
 5    11    10     8  
 9     7     6    12  
 4    14    15     1
```

```
fclose(fileID);
```

Работа с медиа-файлами



- Чтение видео, аудиофайлов изображений
- Поддерживается работа с камерами
- Запись звука с микрофона

Импорт изображений

Считаем изображение с диска и отобразим в MATLAB

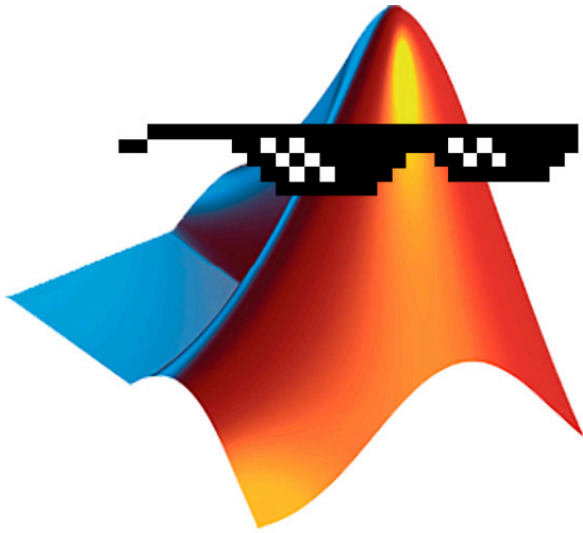
```
img = imread('MATLAB-logo.jpg')
```

`img = 564x614x3 uint8 array`

```
img(:, :, 1) =
```

[illegible]

```
imshow(img)
```



Работа с вебкамерой

Требуется установить [пакет поддержки](#).

Подключаемся к камере и выводим картинку с неё

```
cam = webcam()  
preview(cam)
```

Сделать снимок с камеры

```
img = snapshot(cam);  
imshow(img)
```

Подключение к базам данных



С [Database Toolbox](#) поддерживаются базы данных:

- Реляционные
- Графовые
- Документоориентированные

OPC Toolbox позволяет работать с промышленными БД.

Получение данных из интернета



Базовые возможности

MATLAB умеет:

- Скачивать файлы
- Обращаться к веб-сервисам (GET, POST, ...)
- Работать с FTP, Email

Обратимся к сервису [RESTCountries](https://restcountries.eu/), который бесплатно предоставляет информацию о странах.

Получим информацию о России

```
data = webread('https://restcountries.eu/rest/v2/name/russia')
```

```
data = struct with fields:
    name: 'Russian Federation'
    topLevelDomain: {'.ru'}
    alpha2Code: 'RU'
    alpha3Code: 'RUS'
    callingCodes: {'7'}
    capital: 'Moscow'
    altSpellings: {5x1 cell}
    region: 'Europe'
    subregion: 'Eastern Europe'
    population: 146599183
    latlng: [2x1 double]
    demonym: 'Russian'
    area: 17124442
    gini: 40.1000
    timezones: {9x1 cell}
    borders: {14x1 cell}
    nativeName: 'Россия'
    numericCode: '643'
    currencies: [1x1 struct]
    languages: [1x1 struct]
    translations: [1x1 struct]
```

```
flag: 'https://restcountries.eu/data/rus.svg'  
regionalBlocs: [1x1 struct]  
cioc: 'RUS'
```

Получим название и столицу каждой страны в мире (нужно передать дополнительный параметр `fields`)

```
data = webread('https://restcountries.eu/rest/v2/all', 'fields', 'name;capital')
```

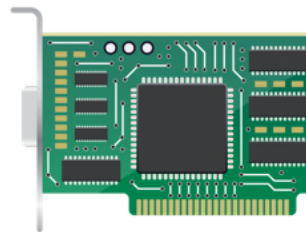
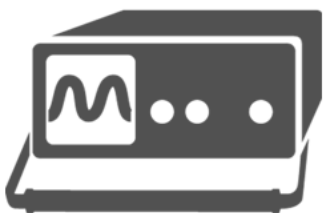
```
data = 250x1 struct array with fields:  
    name  
    capital
```

Продвинутая работа с WEB



Библиотека для работы с веб-сервисами: [MATLAB WEB API](#)

Работа с оборудованием



MATLAB может работать с большим количеством оборудования, в т.ч. с измерительными инструментами и платами сбора данных.

Работа с измерительным оборудованием

MATLAB поддерживает протоколы:

- Serial
- UDP, TCP/IP

С [Instrument Control Toolbox](#):

- IVI, VXI, GPIB, VISA, LXI, PXI
- I2C, SPI, MODBUS

- Bluetooth

С [Vehicle Network Toolbox](#):

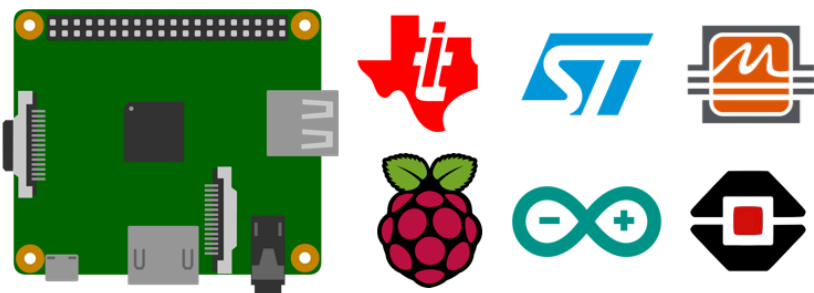
- CAN, CAN FD
- XCP, J1939

Работа с устройствами сбора данных (DAQ)

[Data Acquisition Toolbox](#) позволяет писать свои драйвера и работать с интерфейсами:

- USB
- PCI/PCI-Express
- PXI/PXI-Express

Сбор данных с микроконтроллеров



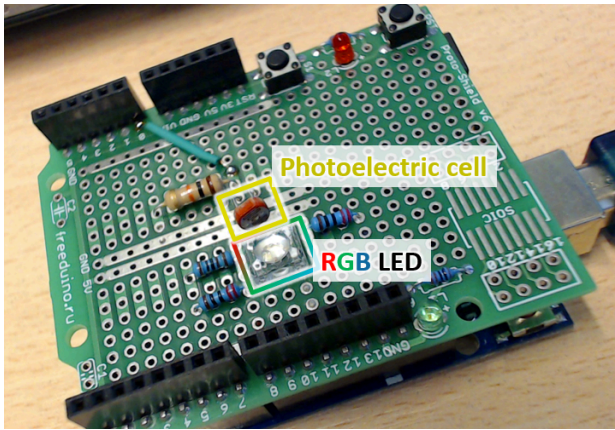
MATLAB работает с большим количеством микроконтроллеров от разных производителей.

Для работы с профессиональными контроллерами требуется [Embedded Coder](#).

Работа с Arduino

Для работы с Arduino необходимо установить [пакет поддержки](#).

Получим данные с Arduino Uno. На базе этой платы реализовано устройство с RGB-светодиодом и датчиком освещенности. [Подробнее об устройстве](#).



Подключимся к плате

```
a = arduino
```

```
a =
  arduino with properties:
    Port: 'COM3'
    Board: 'Uno'
    AvailablePins: {'D2-D13', 'A0-A5'}
    Libraries: {'I2C', 'Servo', 'SPI'}
```

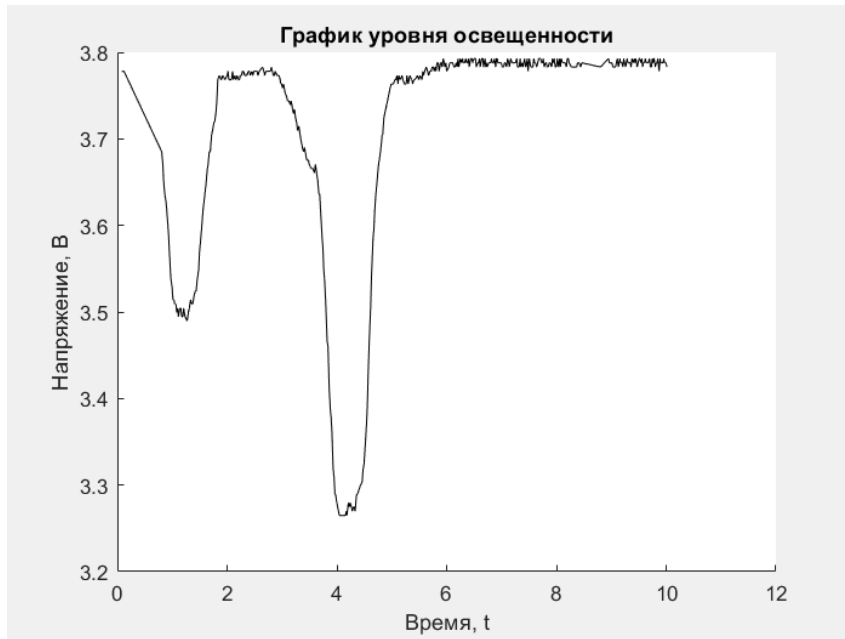
Снимем значение напряжения с датчика освещенности

```
a.readVoltage('A0')
```

```
ans = 3.7488
```

Построим график освещенности в реальном времени

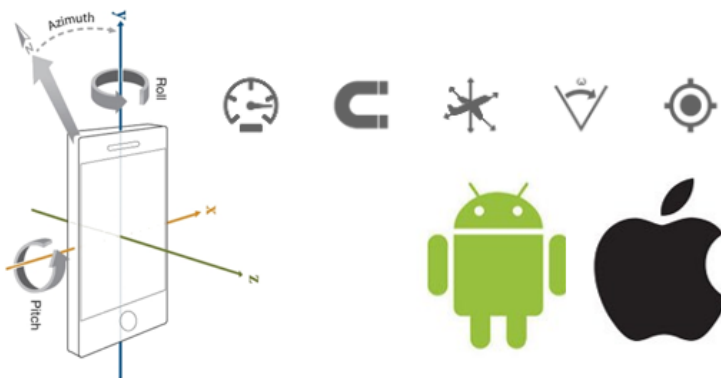
```
f = figure('visible','on');
h = animatedline();
tic();
t = 0;
title('График уровня освещенности')
xlabel('Время, t')
ylabel('Напряжение, В')
while isValid(f) && t < 10
    t = toc();
    v = a.readVoltage('A0'); % Аналоговый вход A0
    addpoints(h, t, v);
    drawnow limitrate
end
```

Выберем цвет

```
color = 'white';
c_list = struct('red', [1 0 0], 'green', [0 1 0], 'blue', [0 0 1], 'white', [1 1 1], 'black', [0 0 0]);
color = c_list(color);
a.writePWMDutyCycle('D9', 1 - color(1)); % Цифровой выход D9
a.writePWMDutyCycle('D10', 1 - color(2)); % Цифровой выход D10
a.writePWMDutyCycle('D11', 1 - color(3)); % Цифровой выход D11
```

Получение данных с датчиков телефона



На телефон нужно установить приложение [MATLAB Mobile](#). На компьютер нужно установить пакет поддержки для датчиков телефона: [Android](#), [iOS](#).

Компьютер и смартфон должны находиться в одной сети (например, подключены к одному WiFi-роутеру).

Запускаем коннектор

```
connector on password
```

```
DNS name: WINDOWSPCIP
IP address: 192.168.56.110
Use this link to test the MATLAB Connector:
http://WINDOWSPCIP:31415/
```

If the test is successful, but MATLAB Mobile cannot connect, your computer might have multiple IP addresses. To determine the correct one, see [Determining the DNS Name or IP Address of a Computer](#).

На телефоне в MATLAB Mobile нужно подключиться к компьютеру по IP с паролем *password*.

После этого подключаемся к датчикам телефона

```
m = mobiledev
```

```
m =
mobiledev with properties:

    Connected: 1
    Logging: 0
    InitialTimestamp: ''

    AccelerationSensorEnabled: 0
    AngularVelocitySensorEnabled: 0
    MagneticSensorEnabled: 0
    OrientationSensorEnabled: 0
    PositionSensorEnabled: 0

Supported functions
```

На телефоне выбираем датчик **Orientation** и нажимаем **START**.

Считываем углы ориентации телефона с гироскопа

```
angles = m.Orientation
```

```
angles = 1x3
-169.4065 -28.4672 8.5426
```

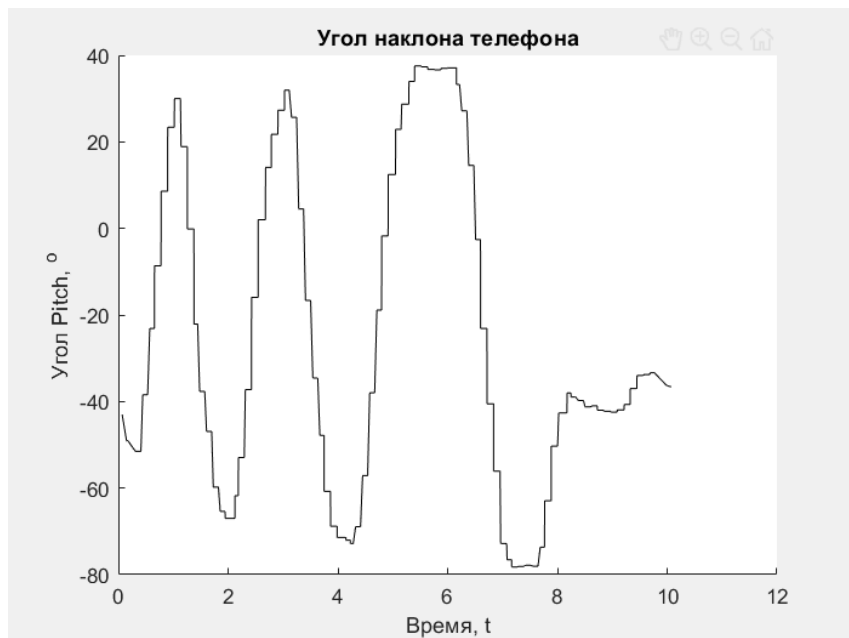
Строим график угла наклона в реальном времени

```
f = figure('visible','on');
h = animatedline();
tic();
t = 0;
title('Угол наклона телефона')
xlabel('Время, t')
```

```

ylabel('Угол Pitch, ^o')
while isvalid(f) && t < 10
    t = toc();
    angle = m.Orientation(2); % Pitch
    addpoints(h, t, angle);
    drawnow limitrate
end

```



Pavel Roslovets, [ETMC Exponenta](#) © 2018