

**Dr. Bob Davidov**

## **Использование общей памяти накопителя: текстовый файл, электронные таблицы**

*Цель работы:* Рассмотрение механизма обмена данными через общую память накопителя.

*Задача работы:* Построение канала обмена данными через общую память накопителя.

*Приборы и принадлежности:* Персональный компьютер, МатЛАБ.

### ВВЕДЕНИЕ

В среде МатЛАБ существует множество механизмов для приема - передачи данных. Которые обеспечивают программы МатЛАБ необходимыми данными и позволяют обмениваться данными с параллельными процессами МатЛАБ и других сред.

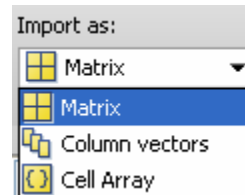
В рамках этой работы рассматриваются следующие вопросы.

- Импортирование данных электронной таблицы.
- Загрузка данных из текстового файла.
- Обмен данными МатЛАБ систем через бинарный файл.

### ОБЩИЕ СВЕДЕНИЯ

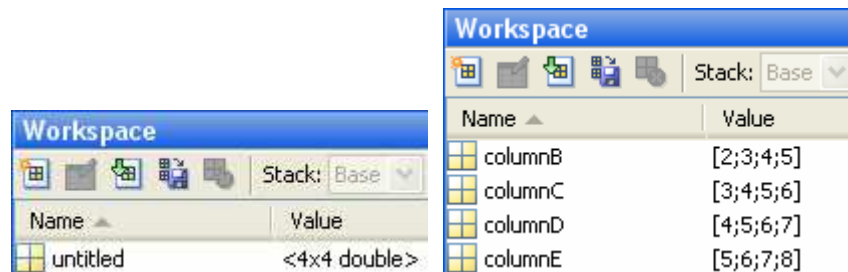
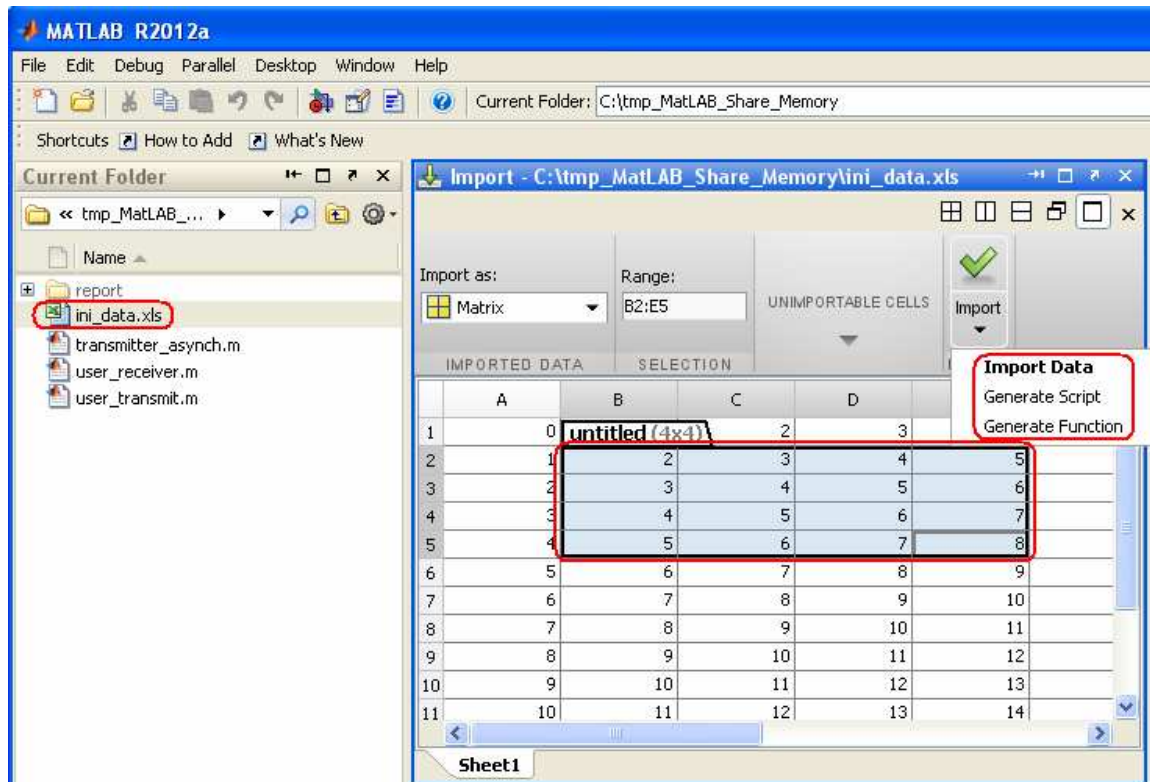
Цифровые и текстовые данные электронной таблицы Excel можно импортировать в Workspace МатЛАБ в виде:

- матрицы (Matrix)
- набора векторов – столбцов (Column vectors)
- в виде массива ячеек (Cell Array) в котором текстовые значения электронной таблицы замещаются в Workspace назначенными числовыми эквивалентами.



Для импортирования данных в Workspace (см. Рис. 1) необходимо:

- открыть в МатЛАБ соответствующую электронную таблицу,
- выделить диапазон данных,
- выбрать формат
- выбрать команду Import Data.



**Рис. 1.** Импорт данных электронной таблицы и форматы импортированных данных.

Процедура импортирования данных электронных таблиц в МатЛАБ позволяет создать m-программу (script) или m-функцию для автоматической загрузки выделенного диапазона электронной таблицы, например, для загрузки диапазона примера Рис. 1, можно получить следующие коды (комментарии сокращены):

m-программа:

```

%% Import the data
[~, ~, raw] =
xlsread('C:\tmp_MatLAB_Share_Memory\ini_data.xls', 'Sheet1', 'B2:E5');

%% Create output variable
untitled = cell2mat(raw);

%% Clear temporary variables
clearvars raw;

```

m-функция:

```
function data = importfile(workbookFile, sheetName, range)

% If no sheet is specified, read first sheet
if nargin == 1 || isempty(sheetName)
    sheetName = 1;
end

% If no range is specified, read all data
if nargin <= 2 || isempty(range)
    range = '';
end

%% Import the data
[~, ~, raw] = xlsread(workbookFile, sheetName, range);

%% Replace non-numeric cells with 0.0
R = cellfun(@(x) ~isnumeric(x) || isnan(x), raw); % Find non-numeric cells
raw(R) = {0.0}; % Replace non-numeric cells

%% Create output variable
data = cell2mat(raw);
```

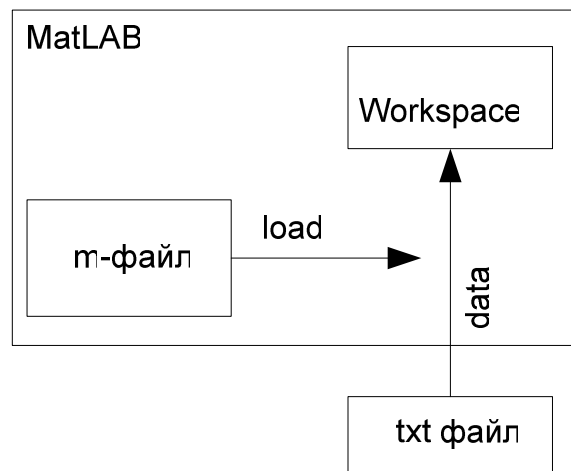
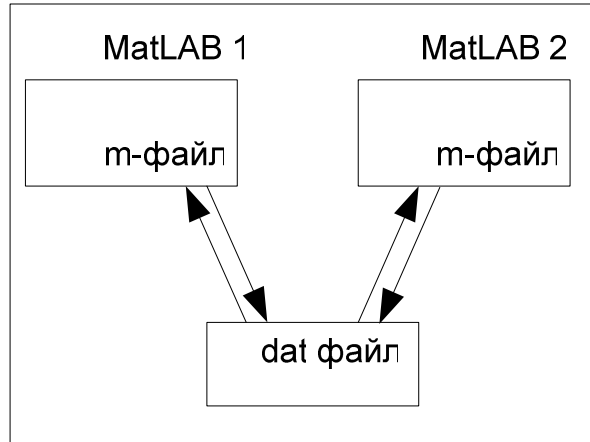


Рис. 2. Загрузка начальных параметров из текстового файла в Workspace.



**Рис. 3.** Схема обмена данными параллельных процессов через dat файл.

## ПРИМЕРЫ ПОЛУЧЕНИЯ ПРОВЕРЕННЫХ РЕЗУЛЬТАТОВ И ВАРИАНТЫ ДЛЯ САМОКОНТРОЛЯ

**Задание 1.** Обмен данных параллельных процессов. Синхронная передача данных.

1. Загрузите МатЛАБ. Настройте его на рабочий каталог.
2. Напишите в МатЛАБ функцию проверки наличия `share_mem.dat` файла обмена данными (256 байт) в директории пользователя `C:\DOCUME~1\User\LOCALS~1\Temp\` и, в случае его отсутствия, создания файла.

```

function make_mmapfile
% make memory map file of zero x 256 byte
% in C:\DOCUME~1\User\LOCALS~1\Temp\
% if it is not created

filename = fullfile(tempdir, 'share_mem.dat');

if ~exist(filename, 'file')
    [f, msg] = fopen(filename, 'wb');
    if f ~= -1
        fwrite(f, zeros(1,256), 'uint8');
        fclose(f);
    else
        error('MATLAB:cannotOpenFile','Cannot open file "%s": %s.',
filename, msg);
    end
end
  
```

3. Напишите программу `transmitter.m` которая записывает в созданный файл общей памяти `share_mem.dat` собственные данные, причем каждый следующий блок данных

записывается только после подтверждения, что данные считаны другим процессом. Размер блока передается через первый байт файла памяти.

```
clear all
make_mmapfile;
m = memmapfile(fullfile(tempdir, 'share_mem.dat'), 'Writable', true,
'Format', 'uint8');

for i = 1:9;

    out_data(1:i) = i; % <= 254 byte
    s = length(out_data);
    m.Data(2:s+1) = out_data;
    m.Data(1)=s;

    % wait until the first byte is zero
    while (m.Data(1) ~= 0)
        pause(0.2);
    end

    disp(sprintf('loop number: %d',i));
end
m.Data(1) = 255; % end of transmission
```

4. Напишите программу `receiver.m` которая считывает данные из файла `share_mem.dat` и устанавливает в файле флаг подтверждения считывания.

```
make_mmapfile;
m = memmapfile(fullfile(tempdir, 'share_mem.dat'), 'Writable', true,
'Format', 'uint8');

while true
    % wait until the first byte is not zero.
    while m.Data(1) == 0
        pause(.25);
    end
    if m.Data(1) == 255 %transmission is off
        break
    end

    in_data = m.Data(2:1+m.Data(1));

    % display received data
    disp(sprintf(' %d',in_data));

    % signal: data is received
    m.Data(1) = 0;
end
disp ('all data is received');
```

5. Запустите `transmitter.m`

6. Откройте второе приложение МатЛАБ и запустите в нем программу `receiver.m`.
7. Убедитесь, что синхронная передача данных через файл данных `share_mem.dat` выполняется.

```
>> reseiver
1
22
333
4444
55555
666666
7777777
88888888
999999999
all data is received
```

8. Доработайте программы и определите максимальную скорость синхронного обмена данными.

**Задание 2.** Обмен данных параллельных процессов. Асинхронная передача данных..

1. Напишите m-программу которая записывает в созданный файл общей памяти `share_mem.dat` собственные блоки данных с собственной задержкой, например, 0.1 сек. Размер блока передается через первый байт файла памяти.

```
clear all
make_mmapfile;
m = memmapfile(fullfile(tempdir, 'share_mem.dat'), 'Writable', true,
'Format', 'uint8');

for i = 1:255;
    out_data(1:i) = i; % < 256 byte
    s = length(out_data);
    m.Data(2:s+1) = out_data;
    m.Data(1) = s;
    disp(sprintf('loop number: %d',i));
    pause(0.1);
end
disp('end of transmission');
```

2. Напишите m-программу которая считывает данные из файла `share_mem.dat` без подтверждения и запоминает размеры принятых блоков данных.

```
make_mmapfile;
m = memmapfile(fullfile(tempdir, 'share_mem.dat'), 'Writable', true,
'Format', 'uint8');

i = 1;
while true
    if m.Data(1)>0
```

```

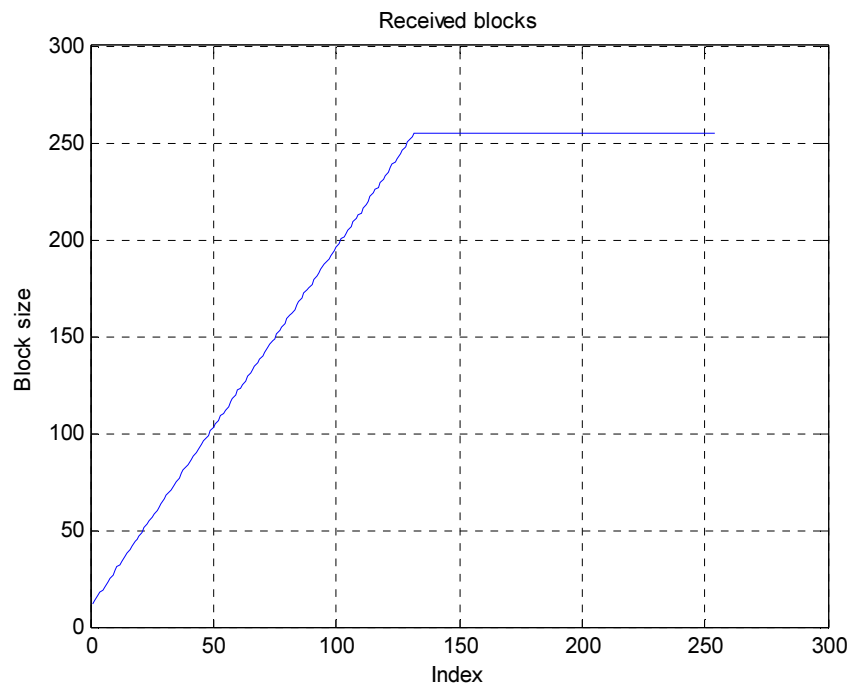
        in_data = m.Data(2:1+m.Data(1));
        block_size(i) = m.Data(1);
        i = i+1;
    end

    % display received data
    disp(sprintf(' %d',in_data));

    pause(0.2);
    if i == 255
        break
    end
end
disp ('stop');
figure
plot (block_size);
grid on
xlabel('Index');
ylabel('Block size');
title('Received blocks');

```

3. Откройте второе приложение МатЛАБ
4. В первом приложении МатЛАБ запустите программу записи данных.
5. Во втором приложении МатЛАБ запустите программу чтения данных.
6. Проанализируйте принятые блоки данных по их размерам. Почему не все переданные блоки данных получены? Почему считано множество блоков по 255 байт тогда как был передан всего один блок в 255 байт?



### Задание 3. Обмен данными пользователей нескольких сред МатЛАБ.

1. Напишите m-программу которая передает другой среде МатЛАБ текстовые сообщения набранные в командной строке первой среды МатЛАБ. Следующую строку можно передать только после получения ответа, что переданное сообщение принято. Сообщения больше 255 байт должны быть уменьшены до 255 байт с выдачей предупреждения. При передачи пустой строки процесс передачи заканчивается.

```
make_mmapfile;
m = memmapfile(fullfile(tempdir, 'share_mem.dat'), 'Writable', true,
'Format', 'uint8');

while true
    str = input('Enter the string (Empty string STOPS): ', 's');

    len = length(str);
    if (len == 0)
        disp('Transmission is terminated')
        break;
    end

    if len > 255
        warning('ml:ml', 'String is decreased to 255 characters. ');
        str = str(1:min(len,255)); % Limit message
        len = 255;
    end

    % write message to the memory map file
    m.Data(2:len+1) = str;
    m.Data(1)=len;

    % wait until the receiver reset the first byte
    while (m.Data(1) ~= 0)
        pause(0.2);
    end

    disp('String has been got')
end
```

2. Напишите m-программу которая принимает сообщения другой среды МатЛАБ и отправляет подтверждение, что сообщение принято.

```
make_mmapfile;
m = memmapfile(fullfile(tempdir, 'share_mem.dat'), 'Writable', true,
'Format', 'uint8');

while true
    % wait message
    while m.Data(1) == 0
        pause(0.2);
    end
```



```

msg = char(m.Data(2:1+m.Data(1)))';
disp('Received message:')
disp(msg)

% inform the transmitter its message has been got
m.Data(1) = 0;

```

end

3. Запустите программу передачи сообщений и введите сообщение.
4. В другой среде МатЛАБ запустите программу приема сообщений.
5. Введите несколько сообщений. Убедитесь что они принимаются.

#### Задание 4. Загрузка начальных данных в Workspace.

1. Создайте m-файл, содержащий начальные параметры в виде структурной переменной, например

```

%{{{
%***** THIS IS A GUI GENERATED FILE *****
% This file of user parameters has been generated by the StatEye GUI.      %
% If you wish to enter parameters manually and not through the GUI then copy %
% user_parameters_manual_template.m to user_parameters.m, and then run main.m %
%                                     %
param.bps = 1e+006;
param.bps_max = 1.1e+010;
param.bps_scan = 'off';
param.txdj = 0.15;
param.tx = 'gaussian_pulse';
param.tx_spectrum_data = ['./Models/Tx_spectrum/ideal_1Gbps_20GHz.txt'];
param.deemphasis_mode = 'manual';
param.deemphasis_start = [0.1 0.1 0.1 0.1];
param.txpre_num = 1;
param.nonlinear_tosa_data = [];
param.p4_s2p_channel_data = ['./Models/Channels/p4_s2params/thru_TXP_040309/'];
param.rxFilterParam = [0.75 0.75];
param.plot_stat_eye = 'on';
param.plot_bathtab = 'off';
param.resultsfile = ['./Results/example.mat'];
%                                     %
%***** END OF GUI GENERATED FILE *****
%                                     %
%}}}}

```

2. Загрузите m-программой начальные параметры в Workspace, например, как это сделано фрагментом программы StatEye

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% main.m   v3.1 Beta D
% Matlab v6.5

```

```

% 23 May 2005
%
% Execution script to enable easy execution of the scripts.
% Scripts runs all required interpolation plus optimisation,
% followed by Analysis and then dumps results into excel
% compatible output file
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
% This is the root, ie the prime routine, for the StatEye script.
% It calls User.Params, creates the global internal params and
% calls the script functions.

run('User_parameters'); % This sucks in all User Defined parameters
param.version = '3.1 Beta D'; % For clarity, confirm StatEye version
disp(['Running main. StatEye version ' param.version]);

if strcmp(param.deemphasis_mode, 'off') || param.txpre_num == 0 || param.txpre_num == 0
    disp(['Tx signal is "' param.tx '"']); % type of the Tx signal
else
    param.tx = 'ideal';
    disp(['Tx signal is "' param.tx '"']);
    if strcmp(param.deemphasis_mode, 'manual')
        disp('Manual deemphasis is on');
    else
        disp('Automatic deemphasis is on');
    end
end

% End of main.m

```

3. Обратите внимание на пример использования основной программной загруженных параметров.

**Задание 5.** Чтение/запись данных электронных таблиц в МатЛАБ.

1. Проверьте механизм загрузки в Workspace данных электронных таблиц показанный на Рис. 1 .
2. Получите информацию о типе и страницах таблицы  
`[type, sheets] = xlsfinfo('ini_data.xls')`
3. Считайте данные таблицы  
`[type, sheets] = xlsread('ini_data.xls')`

4. Введите числовые данные всех страниц таблицы

```
S = importdata('ini_data.xls')
```

5. Запишите данные d1 и d2 на 1-ю и 2-ю страницы новой таблицы.

```
>> p1 = {'Time','Pressure'; 10 1.23; 11 1.32; 12 1.45};
```

```
>> p2 = {'Time','Pressure'; 10 2.45; 11 2.23; 12 2.32};
```

```
>> xlswrite('ini_data_1.xls',p1,1);
```

```
>> xlswrite('ini_data_1.xls',p2,2);
```

6. Введите данные таблицы и рассмотрите структуру введенных данных.

```
pr = importdata xlsread('ini_data_1.xls')
```

## **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Приведите примеры обмена данными параллельных процессов МатЛАБ через общую память.
2. Какова максимальная скорость синхронного обмена данными через dat файл?

## **БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

1. Data Import and Export, MATLAB R2014a, import\_export.pdf