Communicating with the Bitsi in Matlab/Psychtoolbox

Matlab/Psychtoolbox, has no specific settings to create a serial communication channel like Presentation has. We have to set it up ourselves. However, there is a bitsi.m file available that takes care of most the setup and offers some extra functionality.

The bitsi.m file is created quite a few years ago, at the Donders Institute, to aid in a simpler use of the bitsi. Some people adapted the file to their own needs but not always making notes in the documentation and still using the same file name. That is why you have to be careful with just copying any bitsi.m file. I will leave a link here to an 'original' bitsi.m file, or at least one that is checked and known to be working correctly.

I will first give an example without the bitsi.m file though, because it helps in better understanding what you are doing. After that, I will give an example of the same code, using the bitsi.m file.

Read a button Push in Matlab/Psychtoolbox (Windows)

```
% Open serial connection to the bitsi
s = serial('com1');        % create serial object
set(s,'BaudRate',115200);  % set the baud rate
fopen(s);                  % create connection

% set bitsi in default mode
fwrite(s,0);
fwrite(s,2);
fwrite(s,0);

% read button press
fprintf('Press a button on the buttonbox.\n');
fprintf('Pressing "A" or "E" will quit the program.\n');

quit = false;
response = 0;
while quit == false
   if response == 0 && s.BytesAvailable > 0
      response = fread(s, 1, “char”);
      fprintf('Button pressed is: %c\n', response);
   end
   if  response == 'A' || response == 'E'
       quit = true;
   end
   response = 0;
end

fclose(s);                    % close connection
delete(s);                    % clear memory
clear s;                      % clear workspace
```

In the code above, we create a serial object and set the baudrate to 115200 bps. All other default parameters don’t need any change. Next, we open the serial port, so it’s ready for use.

The following three lines of code will reset the bitsi software to make sure we operate in trigger mode with 30ms trigger time and no output mask set.

Then we print some instructive lines on the screen and create a loop where we read the incoming serial characters arriving from the button box. The program quits when an ‘A’ or ‘E’ character is received. This corresponds to releasing the right or the left index button (form the button boxes) respectively. This enables us to check both button boxes.

Before we finish the program, we must close and delete the serial object. If we don’t, we might not be able to open the serial port the next time.

We can check what serial ports are available and whether they are open or not, with the command ‘instrfindall’. You can delete all possible serial ports in one command, with: delete(instrfindall).

Read a button Push with bitsi.m script (Windows)

```
% create a serial object
b1 = Bitsi('com1');

fprintf('Press a button on the buttonbox.\n');
fprintf('Pressing "A" or "E" will quit the program.\n');

quit = false;

while quit == false
  [resp, time_resp] = b1.getResponse(0.001, true);
  if resp > 0
    fprintf('Button pressed is: %s\n', char(resp));
  end;
  if resp == 'A' || resp == 'E'
     quit = true;
  end;
end;

b1.close;
clear b1;
```

The above code uses the bitsi.m script but works about the same as the first code we without the bitsi.m script. There are three main differences.

- Creating the serial object is done in one line,
- Getting the response takes two arguments,
- Closing and deleting the serial object is done in one line.

Much can be said about the bitsi.m script but for now, I want to leave it with just a working example.

Send code in Matlab/Psychtoolbox (Windows)

Sending a code in Matlab is a lot like receiving data. We again have to create a serial object and close it at the end. In between, we send data, instead of receiving it. An example script is given below, where we send code 255 to the bitsi.

```
% Open serial connection to the bitsi
s = serial('com1');            % create serial object
set(s,'BaudRate',115200);      % config

fopen(s);                      % open connection

% set bitsi to level mode
fwrite(s,0);                   % software reset
fwrite(s,2);
fwrite(s,0);

% write in trigger mode
fwrite(s,255);                 % trigger all outputs

% wait
pause(1);                      % pause for 1 second

% close serial connection
fclose(s);                     % close connection
delete(s);                     % clear memory
clear s;                       % clear workspace
```

Send code in Matlab/Psychtoolbox with bitsi.m script (Windows)

The same code as above but with the use of the bitsi.m script, looks like printed below.

```
% Open serial connection to the bitsi
s = Bitsi('com1');             % create serial object
set(s,'BaudRate',115200);      % config

% write in trigger mode
s.sendTrigger(255);            % trigger all outputs

% wait
pause(1);                      % pause for 1 second

s.sendTrigger(255);            % trigger all outputs

% close serial connection
s.close();
```