TEMPLE UNIVERSITY

Sean P Martin <tuf53749@temple.edu>

Re: Simulink Stateflow for Serial Communications [ref:_00Di0Ha1u._5000ZsGaOI:ref]

9 messages

Julia Antoniou <support@mathworks.com>
To: "spmartin@temple.edu" <spmartin@temple.edu>

Tue, Sep 26, 2017 at 11:32 AM

Hi Sean,

My name is Julia, and I am writing in reference to your Technical Support Case #02765237 regarding 'Simulink Stateflow for Serial Communications'.

Based on my understanding, you have a MATLAB function that interacts with an instrument connected to a serial port. The output of this function is a serial object. You would like to use this serial object output in Simulink, specifically with multiple Stateflow charts. Is my understanding correct?

First, would you be able to send me the complete output displayed after entering the following command?

```
>> ver
```

This will provide me with some license information, as well as a list of the toolboxes you have installed.

Would it also be possible for you to send me your function? This would help me better understand how this object might interact with Simulink. If you also have a Simulink model already created that you would like to pass this object into, it would be helpful to send me that model as well.

Please feel free to reply to this email if you have any questions about this case.

Sincerely,
Julia Antoniou
MathWorks Technical Support Department

Please preserve the Reference ID in further correspondence on this query. This allows our systems to automatically associate your reply to the appropriate Case.

--------------- Original Message ---------------
From: MathWorks Customer Support Auto Response [tsdonotreply@mathworks.com]
Sent: 9/25/2017 7:29 PM
To: spmartin@temple.edu
Subject: MathWorks Inc Case # 02765237: Simulink Stateflow for Serial Communications -

The MathWorks Technical Support team has received your request for technical assistance and Case #02765237 was created for it. Our goal is to contact you within 1 business day.

Use the following link to view and update your Case online:
http://www.mathworks.com/support/servicerequests/index.html

This is an auto-generated reply to your request. If you have concerns about our technical support services, please contact tsmanagers@mathworks.com.

For MathWorks Technical Support contacts by country, visit http://www.mathworks.com/support/contact_us/index.html.

Sincerely,
MathWorks Technical Support

www.mathworks.com/support/contact_us

ref:_00Di0Ha1u._5000ZsGaOI:ref

Sean P Martin <spmartin@temple.edu> Wed, Sep 27, 2017 at 12:24 AM
To: Julia Antoniou <support@mathworks.com>

Hello Julia,

Thank you for the reply! Your understanding of my problem is correct. Below is the output of the ver command.

```
>> ver
-----------------------------------------------------------------------------------------------------
MATLAB Version: 9.2.0.556344 (R2017a)
MATLAB License Number: 637718
Operating System: Microsoft Windows 10 Home Version 10.0 (Build 15063)
Java Version: Java 1.7.0_60-b19 with Oracle Corporation Java HotSpot(TM) 64-Bit Server VM mixed mode
-----------------------------------------------------------------------------------------------------
MATLAB                                                Version 9.2         (R2017a)
Simulink                                              Version 8.9         (R2017a)
Bioinformatics Toolbox                                Version 4.8         (R2017a)
Communications System Toolbox                         Version 6.4         (R2017a)
Computer Vision System Toolbox                        Version 7.3         (R2017a)
Control System Toolbox                                Version 10.2        (R2017a)
Curve Fitting Toolbox                                 Version 3.5.5       (R2017a)
DSP System Toolbox                                    Version 9.4         (R2017a)
Data Acquisition Toolbox                              Version 3.11        (R2017a)
Database Toolbox                                      Version 7.1         (R2017a)
Econometrics Toolbox                                  Version 4.0         (R2017a)
Embedded Coder                                        Version 6.12        (R2017a)
Filter Design HDL Coder                               Version 3.1.1       (R2017a)
Financial Instruments Toolbox                         Version 2.5         (R2017a)
Financial Toolbox                                     Version 5.9         (R2017a)
Fixed-Point Designer                                  Version 5.4         (R2017a)
Fuzzy Logic Toolbox                                   Version 2.2.25      (R2017a)
Global Optimization Toolbox                           Version 3.4.2       (R2017a)
HDL Coder                                             Version 3.10        (R2017a)
Image Acquisition Toolbox                             Version 5.2         (R2017a)
Image Processing Toolbox                              Version 10.0        (R2017a)
Instrument Control Toolbox                            Version 3.11        (R2017a)
MATLAB Coder                                          Version 3.3         (R2017a)
MATLAB Compiler                                       Version 6.4         (R2017a)
MATLAB Compiler SDK                                   Version 6.3.1       (R2017a)
MATLAB Report Generator                               Version 5.2         (R2017a)
Mapping Toolbox                                       Version 4.5         (R2017a)
Model Predictive Control Toolbox                      Version 5.2.2       (R2017a)
Neural Network Toolbox                                Version 10.0        (R2017a)
Optimization Toolbox                                  Version 7.6         (R2017a)
Parallel Computing Toolbox                            Version 6.10        (R2017a)
Partial Differential Equation Toolbox                 Version 2.4         (R2017a)
RF Blockset                                           Version 6.0         (R2017a)
RF Blockset                                           Version 6.0         (R2017a)
RF Toolbox                                            Version 3.2         (R2017a)
Robotics System Toolbox                               Version 1.4         (R2017a)
Robust Control Toolbox                                Version 6.3         (R2017a)
Signal Processing Toolbox                             Version 7.4         (R2017a)
SimBiology                                            Version 5.6         (R2017a)
Simscape                                              Version 4.2         (R2017a)
Simscape Electronics                                  Version 2.11        (R2017a)
Simscape Multibody                                    Version 5.0         (R2017a)
Simscape Power Systems                                Version 6.7         (R2017a)
Simulink Coder                                        Version 8.12        (R2017a)
Simulink Control Design                               Version 4.5         (R2017a)
Simulink Real-Time                                    Version 6.6         (R2017a)
Spreadsheet Link                                      Version 3.3.1       (R2017a)
Stateflow                                             Version 8.9         (R2017a)
Statistics and Machine Learning Toolbox               Version 11.1        (R2017a)
Symbolic Math Toolbox                                 Version 7.2         (R2017a)
```

```
System Identification Toolbox                   Version 9.6        (R2017a)
Wavelet Toolbox                              Version 4.18       (R2017a)
>>
```

I've attached two .m files and two .slx files to this email. Basically, one .m file initializes the serial port with a tag and the other sends an arbitrary opcode to a serial port object. Since I could not figure out how to use Simulink/Stateflow to pass around serial object data types, I just initialize the serial port object with a tag and use instrfind to find the serial port object.

Right now, if I run the two .slx files separately, I can initialize the serial port and then send it commands. If I run them as subsystems inside one .slx file, writing to the serial port does not work. I am sure to give the serial_init priority in this case.

Any help you can give would be great. Particularly, how can I manipulate serial port objects in Simulink/Stateflow.

Thank you!
Sean

Sean P. Martin
Electrical Engineering, Temple University 2018
[Quoted text hidden]

4 attachments

test_cmd.m
1K

test_serial.m
1K

init_serial.slx
24K

sendcmd_test.slx
22K

Julia Antoniou <support@mathworks.com> Wed, Sep 27, 2017 at 9:12 AM
To: "spmartin@temple.edu" <spmartin@temple.edu>

Hi Sean,

I am writing in reference to your Technical Support Case #02765237 regarding 'Simulink Stateflow for Serial Communications'.

Thanks for sending all the information! Typically, the "serial" data type is not supported in Simulink models. I think you were able to get around this for one of your models by using "coder.extrinsic" in your Stateflow chart. However, I would advise against using the MATLAB serial functions with your Simulink model.

Instead, I would recommend using the Instrument Control Toolbox. This toolbox has Simulink blocks for sending and receiving live data between instruments and models, including blocks specifically for serial communication modeling. Since you have a license for the Instrument Control Toolbox, you could use these built-in Simulink blocks for any serial communication modeling instead of bringing serial objects in from MATLAB.

Below is a link to one of our documentation pages that has a step-by-step example on how the Instrument Control Toolbox can be used to communicate with a device over a serial connection. I think this would be a great place to start learning about how you can model your serial communication system using this toolbox.
https://www.mathworks.com/help/releases/R2017a/instrument/building-simulink-models-to-send-and-receive-data.html

If you have any questions about this case, please feel free to reply to this email. I will happy to re-open this case and assist you further.

Sincerely,
Julia Antoniou
MathWorks Technical Support Department

Please preserve the Reference ID in further correspondence on this query. This allows our systems to automatically associate your reply to the appropriate Case.

If you have a new technical support question, please submit a new request here:
http://www.mathworks.com/support/servicerequests/create.html
[Quoted text hidden]

ref:_00Di0Ha1u._5000ZsGaOI:ref

Sean P Martin <spmartin@temple.edu> Wed, Sep 27, 2017 at 10:29 AM
To: Julia Antoniou <support@mathworks.com>

Hi Julia,

This approach was my first thought. However, I cannot send different sized arrays of data using the Serial Send block. That is, sometimes I need to send an array like [128 132 76 54] to the instrument and other times something like [128 44]

Is there a way to send variable sized arrays with the instrument control box? Moreover, the reason I want to use Stateflow is so I can read in data from my instrument and act on it in a defined way in a finite state machine.

Again, thank you for the help!
[Quoted text hidden]

Julia Antoniou <support@mathworks.com> Wed, Sep 27, 2017 at 4:37 PM
To: "spmartin@temple.edu" <spmartin@temple.edu>

Hi Sean,

I am writing in reference to your Technical Support Case #02765237 regarding 'Simulink Stateflow for Serial Communications'.

I am glad to hear the Instrument Control Toolbox was your first thought! It is what we generally recommend when users want to do serial communication in Simulink.

Would you be able to tell me a bit more about why you would like to send different sized arrays? Do the arrays vary between a few different sizes, or is it possible for them to be many different sizes? Is there a maximum size array you are expecting? This kind of information will help me find the best workaround for your situation.

Sincerely,
Julia Antoniou
MathWorks Technical Support Department

Please preserve the Reference ID in further correspondence on this query. This allows our systems to automatically associate your reply to the appropriate Case.

--------------- Original Message ---------------
[Quoted text hidden]

ref:_00Di0Ha1u._5000ZsGaOI:ref

Sean P Martin <spmartin@temple.edu> Wed, Sep 27, 2017 at 7:26 PM
To: Julia Antoniou <support@mathworks.com>

Hi Julia,

The arrays I am sending are opcodes for an iRobot Create2. Basically, I am sending different opcodes via a serial port. They are arrays of varying sizes, so many different sizes.

https://cdn-shop.adafruit.com/datasheets/create_2_Open_Interface_Spec.pdf

The above link is a manual for the robot operation. This will detail the different size opcodes I plan to send in these arrays with serial.

Thanks!
Sean

Sean P. Martin
Electrical Engineering, Temple University 2018

[Quoted text hidden]

Julia Antoniou <support@mathworks.com> Thu, Sep 28, 2017 at 9:50 AM
To: "spmartin@temple.edu" <spmartin@temple.edu>

Hi Sean,

I am writing in reference to your Technical Support Case #02765237 regarding 'Simulink Stateflow for Serial Communications'.

Thanks for sending along the PDF - it really helped to clear up what exact signals you are trying to send. I see there are about 6 standard sizes for the opcodes (1x1, 1x2, 1x3, 1x4, 1x5, 1x16), and a few opcodes that could be any length based on the command you would like to send to the Roomba.

Do you have a more recently updated model? It would be helpful to know how you are trying to send the correct opcode to the serial send block. Are you only sending one opcode per simulation? If not, how are you cycling between opcodes during the simulation? Are you getting any specific errors with your current model that does not let you send variable sized arrays?

Sincerely,
Julia Antoniou
MathWorks Technical Support Department

Please preserve the Reference ID in further correspondence on this query. This allows our systems to automatically associate your reply to the appropriate Case.

--------------- Original Message ---------------
From: [spmartin@temple.edu]
[Quoted text hidden]

ref:_00Di0Ha1u._5000ZsGaOI:ref

Sean P Martin <spmartin@temple.edu> Thu, Sep 28, 2017 at 1:05 PM
To: Julia Antoniou <support@mathworks.com>

The serial send block will not accept variable sized arrays. I've tried padded the arrays with zeros to make them fixed size but these commands do not register with the Roomba. I've also tried switching between various sized arrays using a combination of a multiport switch and a stair generator. I've attached models which do this.

I would like to create a library of Simulink blocks. One which initializes the connection to the Roomba, one to read in sensor data, one to spin the wheels, etc. This is why I am leaning towards using the coder.extrinsic functionality in Stateflow. There are Matlab functions existing to spin the Roomba wheels, etc. If there is a way for two Simulink blocks to access an already existing serial port object this would be the ideal solution. (When I do this I am prompted that my serial object is invalid)

In the same .slx file, can two blocks share access to a serial port object?

Again, thanks for the dialogue and help.

Sean

Sean P. Martin
Electrical Engineering, Temple University 2018

[Quoted text hidden]

2 attachments

test2_roombie.slx
20K

roombastate.slx
24K

Julia Antoniou <support@mathworks.com> Thu, Sep 28, 2017 at 4:36 PM
To: "spmartin@temple.edu" <spmartin@temple.edu>

Hi Sean,

I am writing in reference to your Technical Support Case #02765237 regarding 'Simulink Stateflow for Serial Communications'.

You have already tried a couple of the workarounds I was going to mention - padding with zeros, and switching between various sized arrays. I do believe that two blocks can share access to a serial port object, but this is not a functionality that is well documented. If not, you could always use model reference blocks which contain the blocks in separate ".slx" files to get past this limitation.

I was able to find a couple additional workarounds that have been suitable for users with a similar variable-size-signal serial issues in the past. The simplest workaround is to avoid using Simulink Serial blocks and instead, similarly to your original model, open a serial object through Stateflow with MATLAB functions. From there, you could write the data as necessary using the "fwrite" function. Make sure to properly use the "fclose" function along with "fwrite" to avoid any errors. Documentation for both the "fwrite" and "fclose" functions are linked below.
https://www.mathworks.com/help/matlab/ref/fwrite.html
https://www.mathworks.com/help/matlab/ref/fclose.html

A more complicated workaround is required to use the Simulink Serial blocks. One could encapsulate the Serial Send block in a Function Call block with one input. Then, in the Stateflow chart, create two parallel states: one for the controller, and one for a serial interface. The controller adds data into a buffer, and the serial interface reads the buffer and sends data out one at a time along with an output event to activate the Serial Send block.

If you run into any issues with these workarounds, feel free to reply to this email and I will assist you further. Until then, I will close this case for better organization on our end.

If you have a new technical support question, please submit a new request here:
http://www.mathworks.com/support/servicerequests/create.html

Sincerely,
Julia Antoniou
MathWorks Technical Support Department

--------------- Original Message ---------------
From: [spmartin@temple.edu]
[Quoted text hidden]

ref:_00Di0Ha1u._5000ZsGaOI:ref