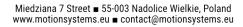


SOFTWARE

ForceSeatDI v.1.1 2023.12.06



Contents

1	General information 5
1.1	Introduction
1.2	ForceSeatMI vs ForceSeatDI 5
1.3	Features comparison 6
1.4	Documentation
2	How to activate the license 9
3	Cross compiling for Raspberry Pi 11
3.1	Introduction
3.2	New project
3.3	Project configuration
3.4	Build and run



General information

1.1 Introduction

ForceSeatDI (Direct Interface) is a lower level interface than ForceSeatMI. It controls hardware directly and ForceSeatPM is not required at all. All error handling and status checking have to be performed by the application. This interface allows to control more than one motion platform from the same PC and allows to create complex but fully synchronized movements of multiple motion platforms.



1.2 ForceSeatMI vs ForceSeatDI

ForceSeatMI (Motion Interface) is a programming interface that allows to add a motion platform support to any application or a game. The ForceSeatMI does not control hardware directly – it sends all data to ForceSeatPM. This approach delegates the responsibility of transforming telemetry data to a real motion from the application to ForceSeatPM. It means that application developers do not have to worry about things like platform disconnections, transmission errors, thermal protection warnings or signal filtering.

Direct Interface is less complex than ForceSeatMI. It manages hardware directly, thus ForceSeatPM is not essential. Checking errors or status is a duty of the application. It allows to control multiple platforms from the same computer. Thanks to such a solution it is possible to build complex and fully synchronized movements of multiple motion platforms at once.

WARNING

ForceSeatDI should be used only in very specific applications. For all other applications, ForceSeatMI is recommended.

1.3 Features comparison

	ForceSeatMI	ForceSeatDI
C/C++		
C#		
Python		
Unity 3D		
	(only Windows PC target plat- form)	(only Windows PC target plat- form)
Unreal Engine	● (only Windows PC target plat-	■ (only Windows PC target plat-
	form)	form)
Matlab/Simulink		
Microsoft Windows		
Linux		
(Ubuntu 16.04.3 LTS Desktop x64)		
Raspberry Pi 3		
(armv7l 4.9.35))		
Raspberry Pi 4 64-bit		
(armv72 6.1.5))		
Gear VR platforms		
(e.g. Oculus Go, Samsung Gear VR)		
Multiple platforms from single PC		•
over USB	(same data sent to all plat-	(separated control of each plat-
	forms)	form)
Multiple platforms from single PC		
over Ethernet		
Easy error handling		
	(by ForceSeatPM)	(by the application)
Diagnostic features		
	(by ForceSeatPM)	(by the application)
Requires ForceSeatPM		
Telemetry mode & scripting engine		
Motion profile selection by the user		
Inverse kinematics		
Fordward kinematics		
Motion compensation for VR		
	(by VR HeadWay in Force-	(by the application, e.g. camera
	SeatPM)	position correction)
Licensing	per motion platform	per motion platform
-	(license stored on PC)	(license stored on motion con- troller)

1.4 Documentation

Idea behind ForceSeatDI and its API structure is similar to ForceSeatMI and at this moment is it not described separately in any document. Please refer to ForceSeatMI documentation to get better understanding how the motion platform control works and examine examples delivered in ForceSeatDI SDK archive to see implementation details.

WARNING

6



7

ForceSeatDI should be used only in very specific applications. For all other applications, ForceSeatMI is recommended.



How to activate the license

ForceSeatDI license is kept in the motion controller so the activation process is different than for ForceSeatMI. Once you receive the activation code do following steps:

- 1. Install ForceSeatPM
- 2. Open ForceSeatPM main window
- 3. Click Tools and Diagnostic and then Devices
- 4. Click Quick Codes
- 5. Enter the activation code. The verification might take a while.
- 6. Once finish, close the Quick Codes window.
- 7. Turn off the device
- 8. Turn on the device
- 9. In Devices window you should see FSDI displayed in Features field

Devices					MotionSystems		
Below are all I	Below are all recognized devices including motion platforms and accessories.						
Device #1: Serial number: Status:	6DoF 	(copy to clipboa					
Firmware: Work time: Features: Offline:	105.2 25h 48m 20s default no	Temperature (C): Work days: Paused:	0, 0, 0, 0 25 no	Hardware ID: Up time: Parking:	54 Oh 04m 57s no		
Configu		ution Qu	ick Codes				



Cross compiling for Raspberry Pi

3.1 Introduction

This short tutorial shows the idea of using VisualGDB to build one of our examples for Raspberry Pi. This section does not describe the full tool chain and build system configuration, it only indicates a few important key points that should help you to configure your project:

- preprocessor directives
- copying .so file to destination board
- debugging the program as root.

For a more detailed description of how to use and configure VisualGDB, please refer to VisualGDB documentation.

TIP VisualGDB is not mandatory for ForceSeatDI to work on Raspberry Pi. You can compile ForceSeatDI using any other build system.

3.2 New project

Start Visual Studio and create a new Linux Project. Make sure to create it in the directory without spaces in the name, e.g. **C:\ForceSeatDI**



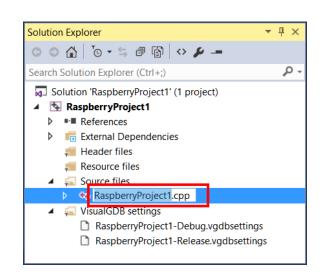
Interview of the project of the

Choose building under Raspberry Pi over network. In order to do it, create **new SSH connection**, then fill IP address and credentials. Make sure that your Raspberry Pi has **libusb** and compiler installed (**check ForceSeatDI readme.txt for details**).

Sector New Linux Project	
Person compares in the construction in the provided in the construction in the provided in the construction in the constr	
<pre>nu computer statucin were rooke acces</pre>	
In the second seco	
Were code acces	× 🖻
Version a New SSH Connection Create a New SSH Connection Provide default key location:	
Create a New SSH Connection Create a New SSH Connection OpenSSH Rey OpenSSH Rey Override default key location: Passphrase: Dublic kit compression (recommended for slow connections)	ΥM
The following scheme illustrates the involved computers and connections between them: Image: State in Vision Connection Sequences Sequences Image: State in Vision Connection Image: State Connection </td <td>_</td>	_
The following scheme illustrates the involved computers and connections between them: Image: State in Vision Connection Sequences Sequences Image: State in Vision Connection Image: State Connection </td <td></td>	
Create a New SSH Connection	
Setup new SSH connection Host Name: 192.168.2.141 User Name: pi Authentication method Password: Password: Password: OpenSSH key Override default key location: Passphrase: Passphrase: Instance Passphrase:	⊊and
Host Name: 192.168.2.141 User Name: pi PAuthentication method Password: Setup public key authentication (more secure than saving the password) Public key in Windows key store (associated with your user account): Auto RSA DSA OpenSSH key Override default key location: Passphrase: Use HTTP CONNECT proxy: Enable zlib compression (recommended for slow connections)	>
User Name: pi Authentication method Authentication method Sassword: Auto RSA DSA OpenSSH key Override default key location: Passphrase: Use HTTP CONNECT proxy: Enable zlib compression (recommended for slow connections)	
Authentication method Password: Setup public key authentication (more secure than saving the password) Public key in Windows key store (associated with your user account): Auto Auto RSA DSA OpenSSH key Override default key location: Passphrase: Use HTTP CONNECT proxy: Enable zlib compression (recommended for slow connections)	
Password: Password: Password: Public key authentication (more secure than saving the password) Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Passphrase: Use HTTP CONNECT proxy: Enable zlib compression (recommended for slow connections)	
Password: Password: Password: Public key authentication (more secure than saving the password) Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Public key in Windows key store (associated with your user account): Passphrase: Use HTTP CONNECT proxy: Enable zlib compression (recommended for slow connections)	
Setup public key authentication (more secure than saving the password) Public key in Windows key store (associated with your user account): Auto RSA DSA OpenSSH key Override default key location: Passphrase: Use HTTP CONNECT proxy: Enable zlib compression (recommended for slow connections)	
Public key in Windows key store (associated with your user account): Auto RSA DSA OpenSSH key Override default key location: Passphrase: Use HTTP CONNECT proxy: Enable zlib compression (recommended for slow connections)	
O OpenSSH key Override default key location: Passphrase: Use HTTP CONNECT proxy: Enable zlib compression (recommended for slow connections)	
Override default key location: Passphrase: Use HTTP CONNECT proxy: Enable zlib compression (recommended for slow connections)	
Passphrase: Use HTTP CONNECT proxy: Enable zlib compression (recommended for slow connections)	
Use HTTP CONNECT proxy:	
Enable zlib compression (recommended for slow connections)	ę
	ē
	¢
	e
	0
	0
Enable keep-alive packets every: 0 seconds	2

Choose Application as project type and MSBuild as tool chain.





After wizard creates project for you, rename RaspberryProject1.cpp to Main.cpp

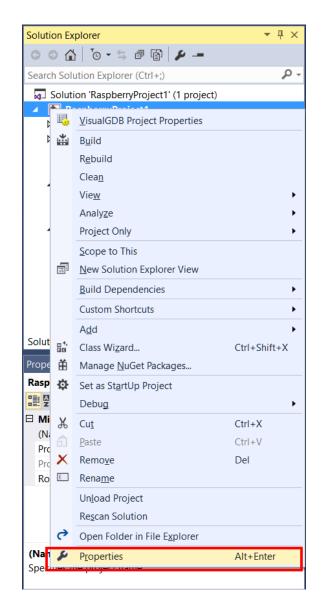
Copy following files from ForceSeatDI archive **code** directory to your project directory:

- 1. ForceSeatDI.h
- 2. ForceSeatDI_Defines.h
- 3. ForceSeatDI_Functions.h
- 4. ForceSeatDI_Loader.c
- 5. ForceSeatDI_Structs.cs

Add **ForceSeatDI_Loader.c** to your project and copy **Main.cpp** from **FastPos_CPP_Linux** example (overwrite the Main.cpp in your project).

3.3 Project configuration

Open the project Properties.

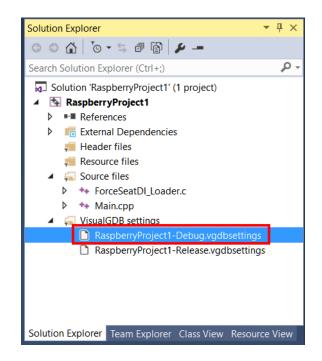




Configuration: Active(Debug) Platform: Active(VisualGDB) Configuration Manager... \sim Configuration Properties ICompile.PreprocessorDefinitions);DEBUG=1NO_IMPORT_FORCESEAT_D General Ignore Standard Include Paths Instrumentation Linux Project Remote Build Debugging ▲ C/C++ General Optimization Preprocessor Precompiled Headers Advanced Instrumentation Custom Step Output Command Line Preprocessor Definitions ▷ Linker Defines a preprocessing symbols for your source file. OK Anuluj Zastosuj

Go to **Preprocessor** configuration and add **NO_IMPORT_FORCESEAT_DI** to the list of definitions.

Go to VisualGDB settings and open Debug configuration (Release configuration for release build).



Go to **Debug Settings** and select run as root.

Raspberr	yProject1-Debug.vgdbsettings*	* X
Δ	Project settings	Debug settings
k	Synchronized Directories	Connection The following computers (configured on the first page) are involved in the debug setup:
Ж	Unit Tests	
1	MSBuild settings	SSHSSH
.∆	Debug settings	PKZ-PC 192.168.2.141
	Dynamic Analysis	Common Settings
	Code Coverage	Run debugger/gdbserver as: 📀 p 🚰 root (using 'sudo')
e T	Custom build steps	Underlying low-level debugger: GDB from the toolchain Custom
		Debug mode: 🔖 New Instance 🧔 Attach 🥐 Python script 📓 Custom GDB Stub 🎤 Fully Cu
1	Custom debug steps	Debugged executable: \$(TargetPath)
	Custom shortcuts	Executable arguments:
Ŵ	Raw Terminal	Custom working directory: \$(BuildDir)
	IntelliSense Directories	Extra LD_LIBRARY_PATH:
		Program output
1	IntelliSense Settings	Show program output in: The 'Program Console' tool window A terminal program on the target's screen
	Code Analyzers	Keep remote console window open after debugging ends
	Path mapping	Linux GUI (X11): 🛃 Forward to Windows machine (via XMing) 🔬 Show on target 🗙 Disable
		_

WARNING

If the application is not executed as root, it will not be able to detach the USB device from the kernel and connect to the motion platform.

Go to Custom build steps and add new After build step – Transfer a single file.

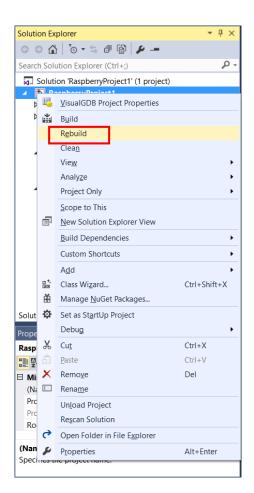
Raspber	Raspberry/Project1-Debug-vgdbsettings* 4ª ×						
Δ	Project settings	1	Custom build	teps			
12	Synchronized Directories	Bet	fore transferring	files: (empty)	\odot		
Z	Unit Tests		fore building: ter building:	(empty) (1 action)	⊙		
1	MSBuild settings			:\ForceSeatDI\ForceSeatDI_2.86\ForceSeatDI32.RSPi_3.so on local computer to \$(TargetDir)/ForceSeatDI32.so on BuildMachine			
4	Debug settings		Source path: Target path:	(C:\ForceSeatDIXForceSeatDI228)ForceSeatDI22RSP1.3.co 🔯 on 🗸 (local computer) 🗸 📲 Run command \$(TargetDir/)/forceSeatDI22.a)			
۲	Dynamic Analysis		Overwrite:	Aways Only when missing When time/size don't match x= Set an internal varia	able		
	Code Coverage			Reference a reusab Zitit selected action(s)	le action list		
Ť	Custom build steps			× Delete ⊮a Copy			
?Þ	Custom debug steps			 での ・ ・ ・			
	Custom shortcuts	Bet	fore cleaning:	(empty)	\odot		
Ţ	Raw Terminal	Aft	ter cleaning:	(empty)	\odot		

Browse and select source file ForceSeatDI32.RSPi_3.so on your PC disk.

Type: \$(TargetDir)/ForceSeatDI32.so into Target path.

3.4 Build and run

Rebuild the project.



Run the application and check the Output console.

