

Hi! Here I left the last version of the v_repExtShareMemory plugin for Vrep. This Plugin allows fast inter process communication of V-REP. It's particularly useful to communicate V-REP with Matlab or Simulink. Here some information:

¿What is Share Memory?

"In computer programming, shared memory is a method by which program processes can exchange data more quickly than by reading and writing using the regular operating system services"

¿Why use Share Memory instead V-REP communication Plugin?

Share Memory plugin is the fastest and simplest interprocess communication method. It doesn't make any synchronization; it has not transfer delay, death times or transfers buffers/queues.

¿It is secure?

In order to ensure top speed communication this plugin don't use any inter process synchronization method, so NO, IT's NOT SECURE. Lossless communication is not ensured. If a single program writes the memory, in practice it will not crash. For for bilateral communication of two programs create two separate share memory's! The Library also includes the CEvent functionality that could be used to synchronize services, but it just work on Windows!

¿For what it would be used?

It allows communicating V-REP with external process (running in the same PC), so you could communicate with almost any program developed in any language that allows Share Memory communication.

For example you could communicate V-REP with C/C++ custom program ,Matlab m-function , Simulink block, LabView scheme, and any other program/language that can access share memory.

You could, for example , make bridge between your robots hardware and Vrep.

Event functionality could be use for fast synchronization of V-REP with external program too.

Files

To install copy the dll on V-REP main folder (\Program Files (x86)\V-REP3\V-REP_PRO_EDU)

Main Functions Prototype:

CODE: [SELECT ALL](#)

```
number result,number ID=simExtShareMemoryOpen(string Name,int size)
```

Creates and open a new shared memory

ID it's the number assigned to the share memory, it is his main identification but not the only one.

Name is the Share memory name, it must be unique in all system. All process that access to a share memory with this name could read and write it!

Size is the size in bytes of the share memory, take care of the dimension assigned, if you try to read or write out of this area you will get an error.

CODE: [SELECT ALL](#)

```
number result=simExtShareMemoryClose(number ID)
```

Close specific share memory by its **ID**. If you let this empty, it will try to close any opened memory.

Be aware: If you don't close the memory at the end of the simulation, it will remain opened. If try to open it again, you will get an error. When the program is closed all instances of share memories are going to get closed.

In fact all share memory's remains opened until all programs close it. It's a good practice to close the share memory in the same script that you create it.

CODE: [SELECT ALL](#)

```
number result,charbuff data=simExtShareMemoryRead(int ID)
```

Reads a share memory by its **ID**.

charbuff data should has full size of the memory , so be aware to create a sufficient big variable to contain the hole memory. After you may format data with **simUnpackFloats()** /**simUnpackInts()** functions.

CODE: [SELECT ALL](#)

```
number result=simExtShareMemoryWrite(intId,charbuff data)
```

It writes the Idmemory **data** should not exceed the size of the memory .Before you may format your data package with **simPackFloats ()/simPackInts()**.

Auxiliary functions

CODE: [SELECT ALL](#)

```
number result, string name=simExtShareMemoryGetName(int ID)
```

Given a **ID** It returns the memory **name**.

CODE: [SELECT ALL](#)

```
numberresult,number ID=simExtShareMemoryGetId(string Name)
```

Given a **Name** it returns the memory **ID**, you may use it to read/write a memory created in other script.

CODE: [SELECT ALL](#)

```
tableID,tablename,table sizes=simExtShareMemoryList()
```

It returns all memories **ID, name and size**. It is a way to find an specific memory created in any Script.