

The introduction to the VR platform

Developed by [Xu Chun Lab](#) at ION/CEBSIT, Chinese Academy of Sciences

Questions and suggestions: chun.xu@ion.ac.cn

Available from https://github.com/XuChunLab/VR_platform

Created December 01, 2021, Modified Feb 07, 2022

A quick demo

- If you want to have a quick look how the VR program looks like without any hardware, please simply run <VRrun1.5demo>.
- This program will be run in the debug mode. You can use the arrow ↑ to move forward in the VR.
- You can pause the program by “Alt + Tab”.
- You can continue the program by switching back to the VR display. Otherwise, you can close the program by close the Cmd window.

A full run of the VR software (step 1)

- Please follow the following steps to run the main program <VRrun> with DAQ USB card and the laser mouse.
- For your first time of use this program, you should install the drivers in the following folders.
 - override the **driver** for the laser mouse such as Rapoo V310.
 - install **driver** for the DAQ USB card. You can skip this step if you only want to have a test with the laser mouse.
 - install the Windows **patches** only when you meet errors.

A full run of the VR software (step 2)

- Before you run, you should also have the following files in the same folder
 - Dll.dll (to communicate with the laser mouse)
 - ftd2xx.dll (for DAQ USB card)
 - Irrlicht.dll (engine to load the VR map).
 - run.txt (a proper setting file)
- After these initial steps, you are ready to run the VR software
 - Run the VR software <VRrun1.5demo> or <VRrun2>.

A customized run for context-dependent licking behavior

- Edit the parameters in the <run.txt>
 - p010=(./Maps/Test/VR_CXT.3ds) //file path for VR map
 - p011=(COM3) // USB COM for the DAQ USB card.
 - p022=(D:\data\) // folder to save the log files
 - p024=+00015 // maximal trials in this session
 - p028=+00000 // debug mode

- Run the VR software
 - <VRrun1.5demo>

Introduction to the <run.txt>

Developed by [Xu Chun Lab](#) at ION/CEBSIT, Chinese Academy of Sciences

Questions and suggestions: chun.xu@ion.ac.cn

Available from https://github.com/XuChunLab/VR_platform

Created December 01, 2021, Modified Feb 07, 2022

- To reuse parameters for specific experiment, you should save your own copy of <run.txt>.
- For those parameters you do not need, leave it blank after “=” (e.g. p027=).
- The <VRrun> programs read the settings defined in <run.txt>, generate the VR display, record IN (e.g. licking) and send OUT (e.g. water delivery) via DAQ card.

Parameters sorted by sections (1)

VR basics

- p000=+00001 //move_SpeedScale //
- p012=+09260 //resetPosition_X1, map boarder
- p010=(E:./../Maps/Test/VR_CXT.3ds) //Full file path for your map
- p011=(COM3) // USB for the digicard. Params_comNum
- p022=(D:\miniscope data\) // folder to save the log files

Start the trial

- p004=-16859 //startPosition_X
- p005=-00000 //startPosition_Y // view height
- p006=-01247 //startPosition_Z // keep it
- p008=+00090 //startAngle_Y // keep it
- p009=+00000 //startAngle_Z //keep it
- p024=+00015 // maximal trials in this session
- p018=+00060 //Buffer entering context (teleport threshold)
- p033=-15898 //Switch Context Trial Position

Default Context (context1)

- p016=-14600 //Context 1 Position X1 (context start)
- p017=-09639 //Context 1 Position X2 (context end)

- p027=-08700 // After Context 1, Return to start.
- p030=(001,022,104) // trial number in Context 1, separated by comma

Context 2

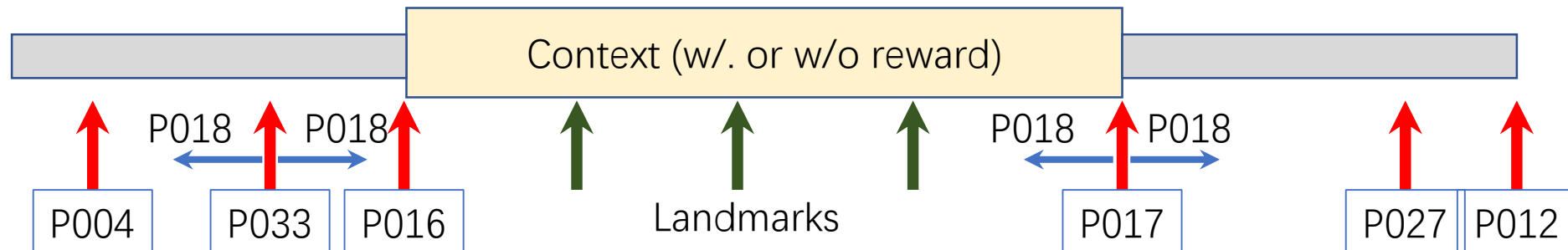
- p025=-15898 // Context 2 Position X1
- p026=-09553 // Context 2 Position X2
- p029=-00800 // After Context 2, Return to start.
- p031=(002,011,125) // trial number in context 2, three digit,

Context 3

- p035=+00030 // Context 3 Position X1
- p036=+00030 // Context 3 Position X2
- p037=+00030 // Return position after Context 3
- p032=() // trial number in context 3, no space in between

Context 4

- p040=+00030 // Context 4 Position X1
- p041=+00030 // Context 4 Position X2
- p042=+00030 // Return position after Context 4
- p039=() // trial number in context 4, use 3 digits



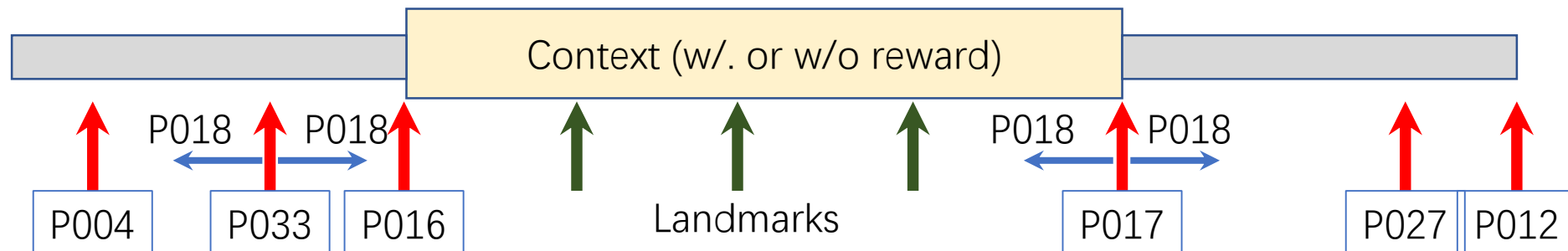
Parameters sorted by modules (2)

Context 5

- p043=-15898 // Context 5 Position X1
- p044=-09553 // Context 5 Position X2
- p045=-00841 // Return position after Context 5
- p049=(002,011,125) // trial list in context 5, three digit,

Context 6

- p046=-15898 // Context 6 Position X1
- p047=-09553 // Context 6 Position X2
- p048=-00841 // Return position after Context 6
- p050=(002,011,125) // trial list in context 6, three digit,



ALL parameter sorted by numbers (1)

- p000=+00001 //move_SpeedScale //original
p000=+00003
- p001=+00010 //move_DisturbanceThreshold
- p002=+00001 //turn_SpeedScale/=10
- p003=+00010 //turn_DisturbanceThreshold
- p004=-16687 //startPosition_X //p004=+75999
- p005=-00000 //startPosition_Y //need=0
- p006=-01247 //startPosition_Z p006=-01082
- p007=-00000 //startAngle_X //if==0,then
horizon
- p008=+00090 //startAngle_Y //original
p008=+02773
- p009=+00000 //startAngle_Z //need=0
- p010=(E:/../Maps/Test/VR_CXT.3ds) //Full file
path for your map
- p011=(COM3) //Params_comNum
- p012=-00060 //resetPosition_X1
- p013=+00000 //resetPosition_Y1//need=0
- p014=-01247 //resetPosition_Z1
- p015=+00600 //resetDistance
- p016=-07975 //Context 1 Position X1
- p017=-01630 //Context 1 Position X2
- p018=+00100 //Buffer Distance
- p019=+00003 //arrowMoveSpeed
- p020=+00000 //Params_autoTurnFlag = 1,0
- p021=+00020 //Params_autoTurnNum
- p022=(D:\miniscope data 2\Linear Track\) //
folder to save the data files



ALL parameter sorted by numbers (2)

- p023=+00030 //Angle_Y
- p024=+00030 // maximal trials in this session
- p025=-15898 // Context 2 Position X1
- p026=-09553 // Context 2 Position X2
- p027=-00841 // Return position after Context 1
- p028=+00001 // debug mode
- p029=-08764 // Return position after Context 2
- p030=() // trial number in Context 1,separated by comma, no space in between
- p031=(002,011,125) // trial number in context 2, three digit,
- p032=() // trial number in context 3
- p033=-15898 //Switch Context Trial Position
- p034=+00002 // sample interval to save VR data
- p035=+00030 // Context 3 Position X1
- p036=+00030 // Context 3 Position X2
- p037=+00030 // Return position after Context 3
- p038=(D:\miniscope data 2\P20200509160924.txt) // Pfile for Replay
- p039=(002,011,125) // trial number in context 4, three digit,
- p040=-15898 // Context 4 Position X1
- p041=-09553 // Context 4 Position X2
- p042=-00841 // Return position after Context 4
- p043=-15898 // Context 5 Position X1
- p044=-09553 // Context 5 Position X2
- p045=-00841 // Return position after Context 5



ALL parameter sorted by numbers (3)

- p046=-15898 // Context 6 Position X1
- p047=-09553 // Context 6 Position X2
- p048=-00841 // Return position after Context 6
- p049=(002,011,125) // trial list in context 5,
three digit,
- p050=(002,011,125) // trial list in context 6,
three digit,
- p051=+00003 // auto speed (valid in
Version1.3.2)
- p052=+03840 // 3840 ScreenWidth (v2.3)
- p053=+02160 // 2160 ScreenHeight (v2.3)
- end //parameter loading stop here



Introduction to the VR data files

Developed by [Xu Chun Lab](#) at ION/CEBSIT, Chinese Academy of Sciences

Questions and suggestions: chun.xu@ion.ac.cn

Available from https://github.com/XuChunLab/VR_platform

Created December 01, 2021, Modified Feb 07, 2022

- Log.txt (Record experimental info for every run of Main Program)
- Pxxx file (Spatial positions and event markers in the VR)
- Lxxx file (Data saved from DAQ card)

Pxxx file (Spatial positions and event markers)

- There are three columns in this file.
 - The 1st is the timestamp for each row
 - The 2nd is the VR spatial position
 - The 3rd is the event markers.
 - “6” is for entry of context 1
 - “8” is for entry of context 2
 - “9” is for entry of context 3
 - “4” is for entry of context 4
 - “1” is for entry of context 5
 - “2” is for entry of context 6
 - “7” is for exit of any context
 - “5” is for entry of zone 1 (reward/punishment)
 - “3” is for entry of zone 2 (reward/punishment)

EXAMPLE

0	1	2
550	-12580	0
580	-12580	0
590	-12573	0
590	-12570	0
590	-12567	0
590	-12567	0
610	-12562	0
610	-12562	0

Lxxx file (data from DAQ card)

- There are eight columns in this file. Depending on how the I/O pins of DAQ are connected (a connection example provided in the next page), each column represents corresponding data.
 - The 1st is the timestamp for each row
 - The 2nd is the lick detection (1 for yes, 0 for no)
 - The 3rd is the lick detection at another port.
 - The 4th is the trigger for water delivery.
 - The 5th is the trigger for optogenetic laser.
 - The 6th, TBA.
 - The 7th, TBA.

EXAMPLE							
0	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0

Connect Arduino-based modules with DAQ

- There are 4 inputs to DAQ. For example,
 - PI0, connected with a left lick port, water delivery/air puff would be available when the animal enter zone #1.
 - PI1, connected with a right lick port, water delivery/air puff would be available when the animal enter zone #2.
 - PI2, connected to the trigger for water pump.
 - PI3, connected to the trigger for the laser stimulation.
- There are 4 outputs from DAQ. For example,
 - PO0, connected with a left lick port, water delivery/air puff would be available when the animal enter zone #1.
 - PO1, connected with a left lick port, water delivery/air puff would be available when the animal enter zone #2.
 - PO2, connected to the trigger for the optogenetics.
 - PO3, connected to the trigger for the synchronized neural recording.

Contacts

- XU Chun, Ph.D. Principal Investigator.

Lab of Neurobiology of Context and Behavior, Institute of Neuroscience (ION)

Center for Excellence in Brain Science and Intelligence Technology (CEBSIT)

Chinese Academy of Sciences (CAS)

320 Yue Yang Road, Shanghai 200031, China

- E-mail: chun.xu@ion.ac.cn
- <http://english.cebsit.cas.cn/lab/xuchun/>