NeuroMem Console

Testing your NeuroMem Hardware at the Register Level

Version 2.5.1

Revised 08/23/2018

GENERALVISION

NeuroMem Console is a product of General Vision, Inc. (GV)

Contents

1 Getting Started

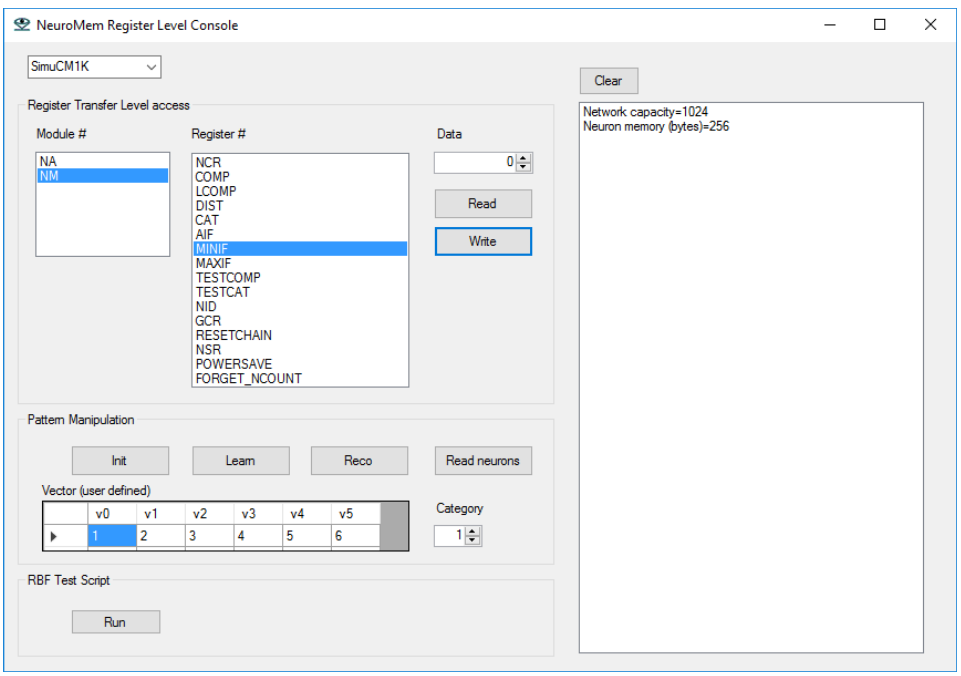
The NeuroMem Console is intended to test your NeuroMem hardware including proper connectivity, register level access to the NeuroMem network and other functionalities specific to the hardware. It is also useful to understand the behavior of the neurons and practice with their entry-level operations in a step by step mode. Higher-level functions allow to manipulate pattern vectors and run test scripts. The source code of the NeuroMem Console is supplied in the NeuroMem SDK for Windows.

1.1 Installation

- Double-click at the setup.exe file stored in the folder Install_NeuroMem Console
- Launch the program from the Start menu/General Vision/NeuroMem Console and wait until the panel below appears. This may take a few seconds.
- Connect your NeuroMem hardware to the USB connector of your PC.
- Select your NeuroMem platform and verify that the correct number of neurons is displayed in the Report Area

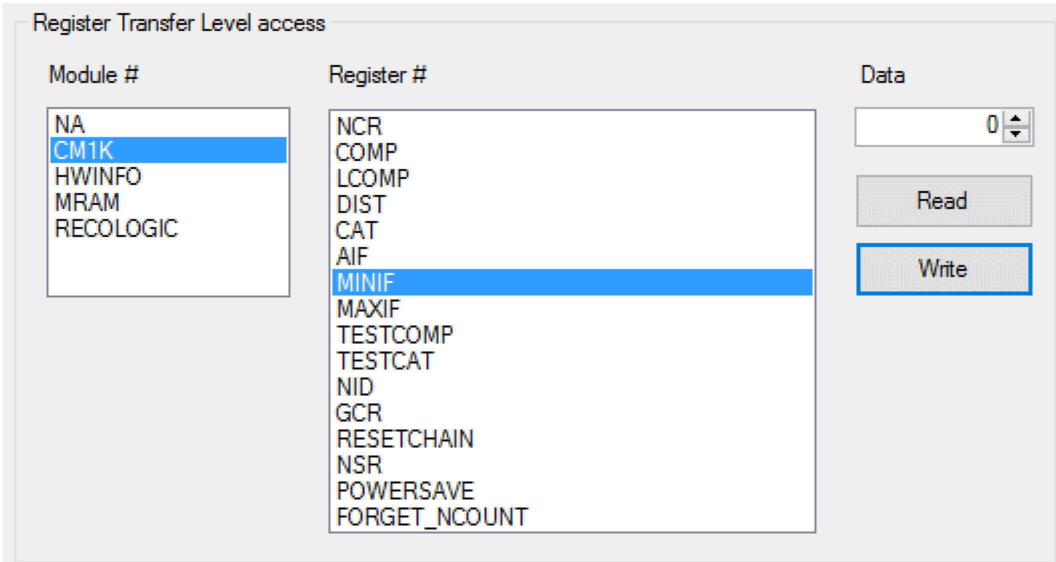
The panel features four zones:

- Register Transfer Level Access: To access the registers of the NeuroMem neural network (module NM) and other modules available on your hardware platform
- Pattern Manipulation area: To learn and recognize vector data edited on screen
- RBF Test Script: To execute a self-contained test documented later in this manual
- Log: To report all the executed functions



2 Register Transfer Level Access

Under the Register Transfer Level access, you can read or write single values manually to the registers of the different modules.



- Select a module
- Select a register of the module
- To execute a WRITE command, edit the value in the Data field. Then Click Write
- To execute a READ command, click Read and read the value in the Data field

All platforms have in common a NeuroMem neural network. Beyond that they can offer additional functionalities accessible through the same Register Transfer Level protocol. The following paragraph describes the registers of the NeuroMem network. Additional modules and registers specific to hardware platforms are listed in the Appendixes of this manual.

2.1 Module #1: the chain of NeuroMem chips

For a detailed description of the neuron's behavior and their interactions, please refer to the CM1K Hardware Manual and the NeuroMem Technology Reference Guide.

	Description	Addr 8-bit	Normal op	SR op	Default
NSR	Network Status Register	0x0D	RW	W	0x0000
GCR	Global Control Register	0x0B	RW		0x0001
MINIF	Minimum Influence Field	0x06	RW	RW	0x0002
MAXIF	Maximum Influence Field	0x07	RW		0x4000
NCR	Neuron Context Register	0x00		RW	0x0001
COMP	Component	0x01	W	RW	0x0000
LCOMP	Last Component	0x02	W		0x0000
INDEXCOMP	Component index	0x03	W	W	0x0000
DIST	Distance register	0x03	R	R	0xFFFF
CAT	Category register	0x04	RW	RW	0xFFFF
AIF	Active Influence Field	0x05		RW	0x4000
NID	Neuron Identifier	0x0A	R	R	0x0000
POWERSAVE	PowerSave	0x0E	W		n/a

	Description	Addr 8-bit	Normal op	SR op	Default
FORGET	Forget	0x0F	W		n/a
NCOUNT	Count of committed neurons	0x0F	R	R	0x0000
RESETCHAIN	Points to the first neuron	0x0C		W	n/a

3 Pattern Manipulation

3.1 Init script

Initialize the neural network and report the default global registers. The log should report a count of committed neurons equal to 0, a Minimum Influence Field of 2 and Maximum Influence Field of 16384. Any other value is incorrect.

3.2 Learn script

Broadcast the vector edited in the vector grid and learn it as belonging to the edited category, then report the number of committed neurons.

3.3 Reco script

Broadcast the vector edited in the vector grid and read the responses of all the firing neurons.
If the selected module is the RECOLOGIC module, the report is limited to the K top firing neurons, if applicable.

3.4 Example

Init
Learn vector [1,2,3,4,5] with category =1
    Log should report NCOUNT=1
Reco vector [1,2,3,4,5]
    Log should report DIST=0, CAT=1, NID=1
    DIST is equal to 0 because there is an exact match between the learned vector and the current vector

Reco vector [9,2,3,4,5]
    Log should report DIST=8, CAT=1, NID=1
    DIST is equal to 8 due to the difference between the 1st component of the learned and current vectors

4 RBF Simple Script

4.1 Description of the script

The script illustrates the behavior of the neurons during learning and recognition operations. It is supplied in a variety of programming language including C/C++. C#, MatLab, Python, Arduino and more.

Step 1: Learn

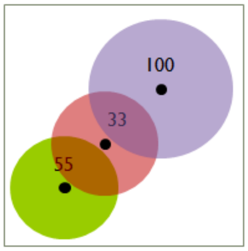
Learn vector1 [11,11,11,11] (1) with category 55
Learn vector2 [15,15,15,15] with category 33(2)
Leanr vector3 [20,20,20,20] with category 100

(1) The learned vectors are purposely set to arrays of constant values so their representation and relationship are easy to understand. The distance between two "flat" vectors is indeed the difference between their constant value times their number of components. For example the distance between [11,11,11,11] and [15,15,15,15] is equal to 4 * 4. This simple arithmetic makes it easy to understand the different cases of recognition illustrated in this test script.
(2) The category of the second neuron is purposely set to a lesser value than the category of the first neuron to verify that if both neurons fire with a same distance, the category of the neuron on the 2nd chip is still the first the be read out

Fig1 is a representation of the decision space modeled by the 3 neurons where Neuron1 is shown in red, Neuron2 in green and Neuron3 in blue. In the follwoing 2D graph, we limit the length of the models to 2 components instead of 4, so they can be positioned in an (X,Y) plot. X=1st component and Y=Last component, and a surrounding diamond shape with the side equal to their influence field.

Committed neurons= 3

NeuronID=1	Components=11, 11, 11, 11,	AIF=16,	CAT=55
NeuronID=2	Components=15, 15, 15, 15,	AIF=16,	CAT=33
NeuronID=3	Components=20, 20, 20, 20,	AIF=20,	CAT=100

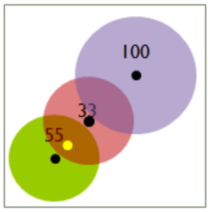


The influence fields of Neuron#0 and Neuron#1 overlap, as well as Neuron#1 and Neuron#2 overlap but differently since their distances from one another are different.

Step2: Recognition

The vectors submitted for recognition are selected purposely to illustrate cases of positive recognition with or without uncertainty, as well as cases of non recognition. The program reads the responses of all the firing neurons, which is until the distance register returns a value 0xFFFF or 65553.

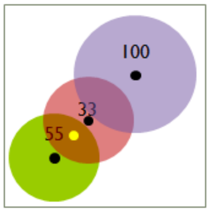
Case of uncertainty, closer to Neuron#1



Vector=12, 12, 12, 12
Neuron 1 and 2 fire. Vector is closer to Neuron1

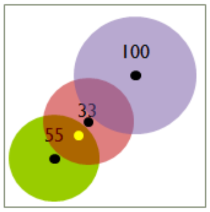
Response#1	Distance= 4	Category= 55	NeuronID= 1
Response#2	Distance= 12	Category= 33	NeuronID= 2
Response#3	Distance= 65535	Category= 65535	NeuronID= 65535

Case of uncertainty, equi-distant to Neuron#1 and Neuron#2



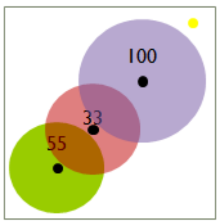
Vector=13, 13, 13, 13,
Neuron 1 and 2 fire. Vector is equi-distant to both neurons
Response#1 Distance= 8 Category= 33 NeuronID= 2
Response#2 Distance= 8 Category= 55 NeuronID= 1
Response#3 Distance= 65535 Category= 65535 NeuronID= 65535

Case of uncertainty, closer to Neuron#2



Vector=14, 14, 14, 14,
Neuron 1 and 2 fire. Vector is closer to Neuron2
Response#1 Distance= 4 Category= 33 NeuronID= 2
Response#2 Distance= 12 Category= 55 NeuronID= 1
Response#3 Distance= 65535 Category= 65535 NeuronID= 65535

Case of unknown

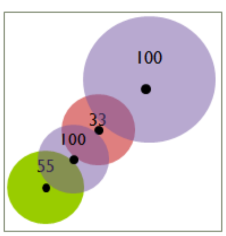


Vector=30, 30, 30, 30,
No neuron fire
Response#1 Distance= 65535 Category= 65535 NeuronID= 65535

Step3: Adaptive learning

Learning a new vector to illustrate the reduction of neurons'AIFs.

Learn vector[13,13,13,13] with category 100. This vector is equidistant to both Neuron1 and Neuron2. Learning it as a new category, will force Neuron1 and 2 to shrink from their AIF=16 to an AIF=8 to make room for a new neuron which will hold the new model and its category.



Committed neurons= 4

NeuronID=1	Components=11, 11, 11, 11,	AIF=8,	CAT=55
NeuronID=2	Components=15, 15, 15, 15,	AIF=8	CAT=33
NeuronID=3	Components=20, 20, 20, 20,	AIF=20	CAT=100
NeuronID=4	Components=13, 13, 13, 13,	AIF=8,	CAT=100

Note that if the vector to learn was [12,12,12,12], Neuron1 would shrink to 4 and Neuron2 to 12.

4.2 Running the RBF script

The RBF Test Script executes the scenario described in the previous paragraph, with the following addition: The learning is the last vector [13,13,13,…] commits a neuron on the last chip of the chain and allows verifying the proper interconnectivity between neurons across the entire chain of chips.

This is done by committing "dummy" neurons between the neuron #3 and the 1st neuron of the last chip. These neurons are assigned to a different context 99 so they will remain idle during any learning or recognition of the script which uses the default context of 1.

Vector w/ Comp	Learn as	Stored in NID	Reco Dist	Reco Cat	Reco NID	Comment
11	55	1				
15	33	2				
20	100	3				
12			10 30	55 33	1 2	The vector "12" triggers a recognition with uncertainty between Neuron 1 and 2, being closer to 1
13			20 20	33 55	2 1	The vector "13" triggers a recognition with uncertainty between Neuron 1 and 2, being closer to 2
21			10	100	3	The vector "21" triggers a positive recognition without uncertainty
5			0xFFFF	0xFFFF	0xFFFFFF	The vector "5" triggers an unknown recognition
Continue script to commit a neuron in the last chip…						
Vector w/ Comp	Learn as	Stored in NID	Reco Dist	Reco Cat	Reco NID	Comment
Chain of 1 chips						
13	100	4				Learning the vector "13" commits a fourth neuron and shrink the Influence Field of Neuron 1 and 2 from the value 40 down to 20.
12			10 10	55 100	1 4	The vector "12" is now recognized with uncertainty between Neuron 1 and 4 and is equi-distant to both
14			10 10	33 100	2 4	
13			0	100	4	
Chain of N chips						
13	100	**M** = (N-1) * 576 +1				Learning the vector "13" commits an Mth neuron and shrink the Influence Field of Neuron 1 and 2 from the value 40 down to 20.
12			10 10	55 100	1 M	The vector "12" is now recognized with uncertainty between Neuron 1 and 4 and is equi-distant to both
14			10 10	33 100	2 M	
13			0	100	M	

N=2 M=577
N=3 M= 1153
N=5 M= 2305

Etc.

Printout of the script for N=1

Learn vector 11 as CAT=55

Learn vector 15 as CAT=33

Learn vector 20 as CAT=100

Committed=3
NID 1 NCR=1 AIF=40 CAT=55 COMP=11,11,11,11,11,11,11,11,11,11,
NID 2 NCR=1 AIF=40 CAT=33 COMP=15,15,15,15,15,15,15,15,15,15,
NID 3 NCR=1 AIF=50 CAT=100 COMP=20,20,20,20,20,20,20,20,20,20,

Recognize vector 12 as:
CAT=55 at DIST=10, NID=1
CAT=33 at DIST=30, NID=2
NOTICE uncertainty; non equi-distant firing neurons

Recognize vector 13 as:
CAT=33 at DIST=20, NID=2
CAT=55 at DIST=20, NID=1
NOTICE uncertainty; equi-distant firing neurons

Recognize vector 21 as:
CAT=100 at DIST=10, NID=3
NOTICE positive identification

Recognize vector 5 as:
NOTICE non recognition

Learn vector 13 as CAT=100
NOTICE AIF of Neuron 1 and 2 have shrunk

Committed=4
NID 1 NCR=1 AIF=20 CAT=55 COMP=11,11,11,11,11,11,11,11,11,11,
NID 2 NCR=1 AIF=20 CAT=33 COMP=15,15,15,15,15,15,15,15,15,15,
NID 3 NCR=1 AIF=50 CAT=100 COMP=20,20,20,20,20,20,20,20,20,20,
NID 4 NCR=1 AIF=20 CAT=100 COMP=13,13,13,13,13,13,13,13,13,13,

Recognize vector 12 as:
CAT=55 at DIST=10, NID=1
CAT=100 at DIST=10, NID=4

Recognize vector 14 as:
CAT=33 at DIST=10, NID=2
CAT=100 at DIST=10, NID=4

Recognize vector 13 as:
CAT=100 at DIST=0, NID=4

4.3 Test “Interchip” option

If the platform has a NeuroMem network composed of more than one chip, the Interchip option is enabled.

This option runs the RBF Test Script script with the following addition: each time a new neuron is committed, the rest of the neurons of the same chip are automatically assigned to a dummy context 99 and committed with a dummy model and dummy category 1.

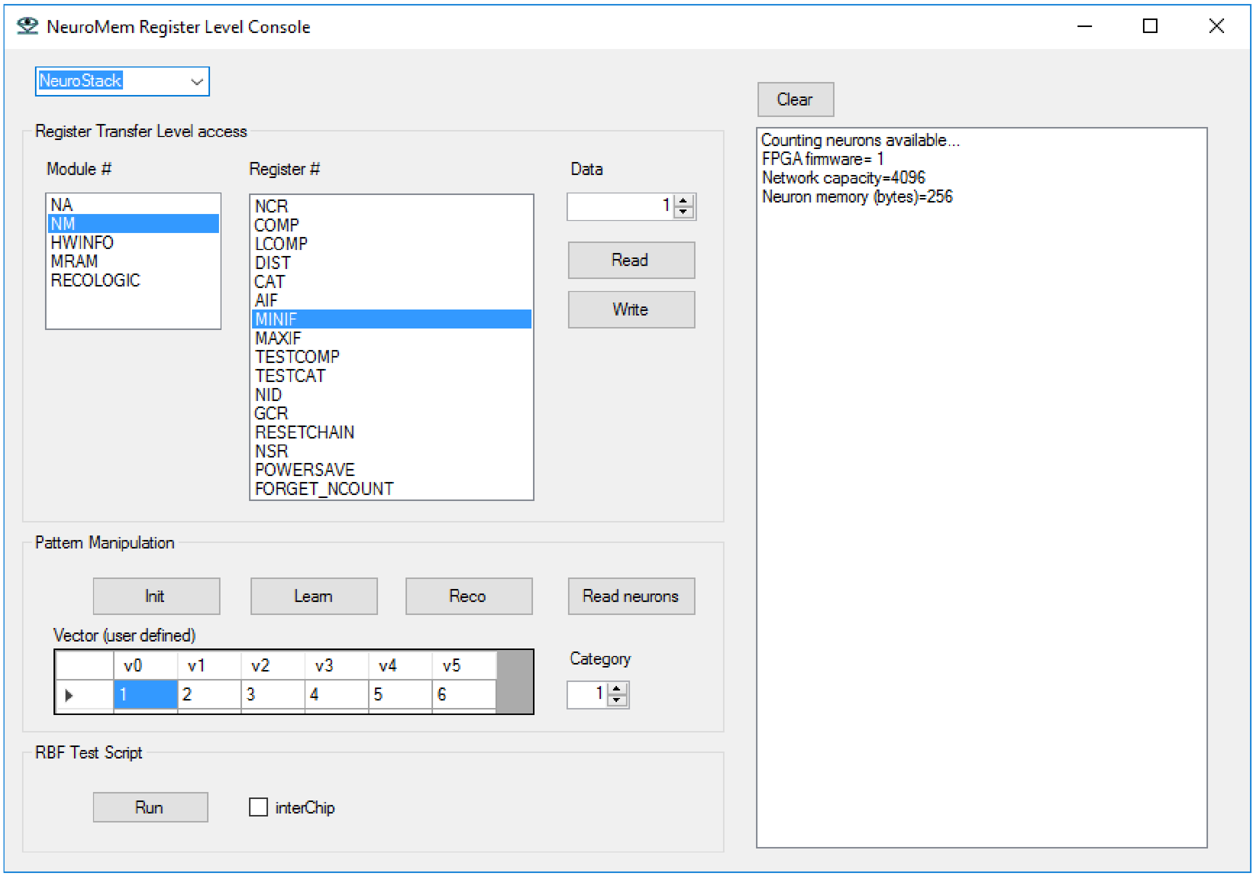
The recognized distances and categories remain the same, but the identifiers of the firing neurons (NID) are different.

Min Nbr of chips	Vector w/ Comp	Learn as	Stored in NID		Reco Dist	Reco Cat	Reco NID	
			normal	interchip			normal	interchip
1	11	55	1	1				
2	15	33	2	577				
3	20	100	3	1153				
	12				10 30	55 33	1 2	1 577
	13				20 20	33 55	2 1	577 1
	21				10	100	3	1153
	5				0xFFFF	0xFFFF	0xFFFFFF	
N	**13**	**100**	**M = (N-1) * 576 +1**					
	12				10 10	55 100	1 M	1 M
	14				10 10	33 100	2 M	577 M
	13				0	100	M	M

N=4 M= 1729
N=5 M= 2305
N=7 M=3457
Etc.

4.4 NeuroStack Specific Modules and Registers

If you are using the NeuroMem Console with a stack of NeuroStack, the selection of the platform triggers the detection of the stacked boards and the sizing of the NeuroMem network. Please allow up to 4 seconds per board to be accounted for.



Module #2: Hardware Settings

Register	Description	Addr 8-bit	Op	Default
HW_REV	Revision number of the board, including its USB protocol.	0x01	R	0x0000
CLK_CNT	Clock counter - Write : resets its value to 0 - Read: number of clock cycles spent at execution of the USB commands since the last reset	0x02	RW	0x0000

Module #4: Recognition Logic module

> *A recognition logic module is supplied in the FPGA firmware version 1 or greater. This means that if the NeuroStack is programmed with a firmware version 0, the integrated functions delivered in the NeuroMem API will not execute at optimum speed because they heavily reply on USB I/O transactions to read and write registers. Contact General Vision to obtain information about updates.*

Access to the Recognition Logic is made through the module 4, but it is easier to test it through the “Integrated Functions” of the Diagnostic Utility described in the next chapter.

Register	Description	Addr 8-bit	Operation	Data 16-bit/ Default
VERSION	Version number of the recognition logic module. 0=bare minimum, no module 4 1= vector recognition with automatic K readout Refer to next chapter on configuration files for information on additional Recognition Logic features or versions.	0x00	R	0x0001
RECO	Write: Recognize the input vector of length of up to 256 bytes. The responses of the top K firing neurons are accumulated in a Result buffer. Read: Read the Length of the Result buffer in words.	0x01	W R	0x0000

RESULTS	Read: output the Result buffer with the following format
	
	Format for a recognized vector:
	Up to **K** series of
	{
	- Distance value (16-bit value),
	-Category value (16-bit value)
	-Identifier value { 0x00, 24-bit value}
	}
	Ended with 0xFFFF, delimiter between consecutive RECO
	
	Format for a non recognized vector:
	Byte 1-2= 0xFFFF
	
	Write: Clear the Result buffer
		0x02	R
			W
				0x0000

K	Write the value K or maximum number of responses to read if applicable Read the value K	0x03	W R	0x0001