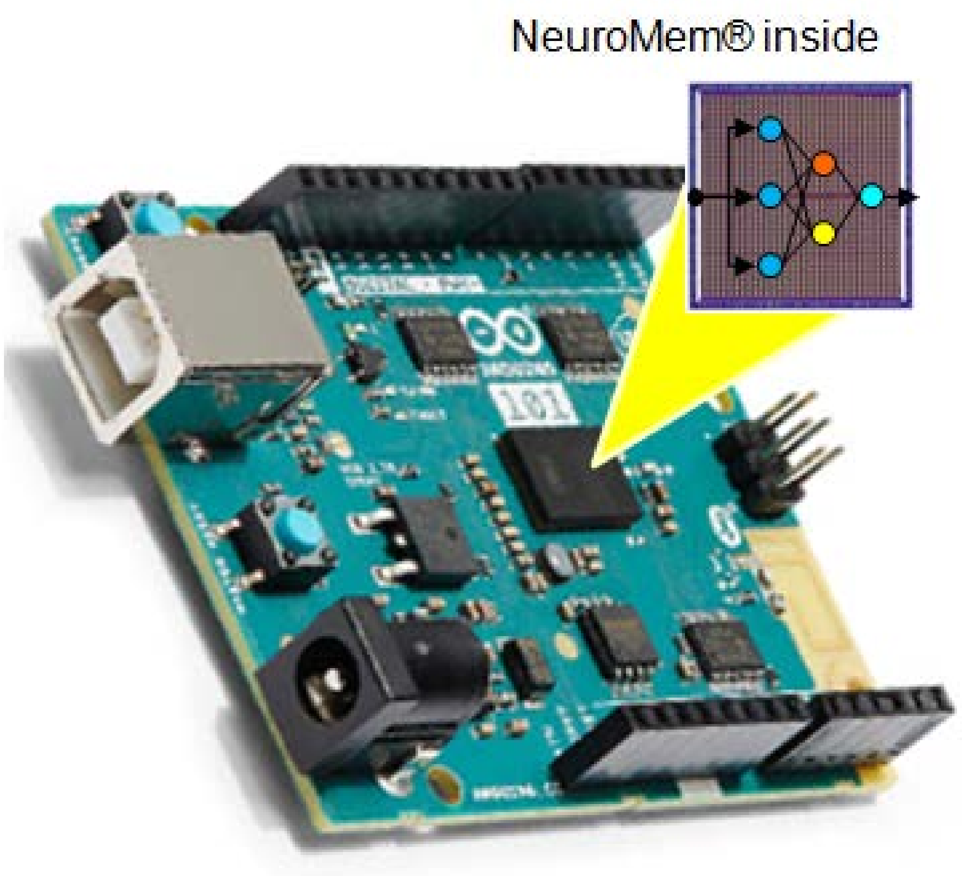
# Unleashing the neurons of the Intel® Curie module on the Arduino/Genuino 101 platform



Empower your project with neurons which can learn with the push of a button and immediately start recognizing.

Monitor signals and act only when significant events occur.



# Who is General Vision



RCE neurons on silicon

= NeuroMem technology

= Neuromorphic Memories

- Incorporation in 1987
- Using Restricted Coulomb Energy (*) neurons since 1988 (mostly for vision applications)
- Inventor of the 1st NeuroMem chip jointly with IBM (ZISC, 36 and later 78 neurons) in 1993
- Inventor of the 2nd NeuroMem chip (CM1K, 1K neurons) in 2007
- Intel rolls out the 1st SOC with NeuroMem inside

(*) RCE was invented by Pr. Leon N. Cooper and all, Physics Nobel Prize 1972

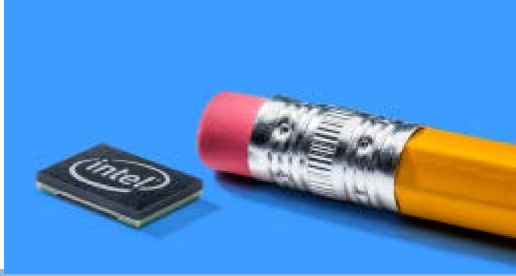
Director of the brain and neural systems center at Brown University.

# A brief introduction



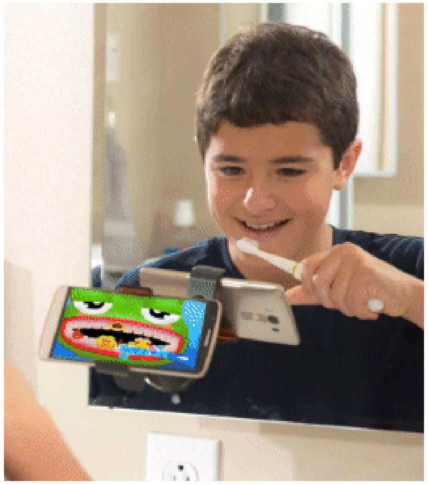

[View this introduction on our youtube account]

# What can I do with the Curie neurons?






Grush, the gaming toothbrush making sure the kids brush their teeth properly



Jagger & Lewis, smart collar monitoring well-being of dogs

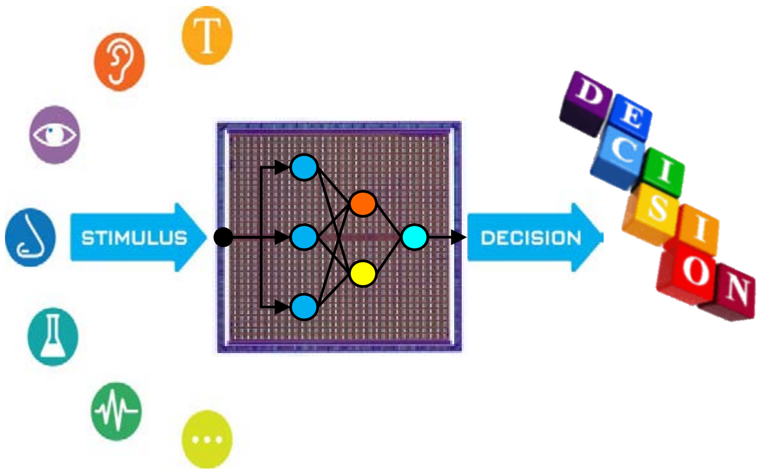

ShapeHeart, arm band with heart monitoring

# Benefits of the neurons



- The neurons learn by examples
  - No programming
  - Training can be done off-line or the fly
- Continuous monitoring at low-power
- Can detect novelty or anomaly
- Knowledge portability
- Knowledge expandability

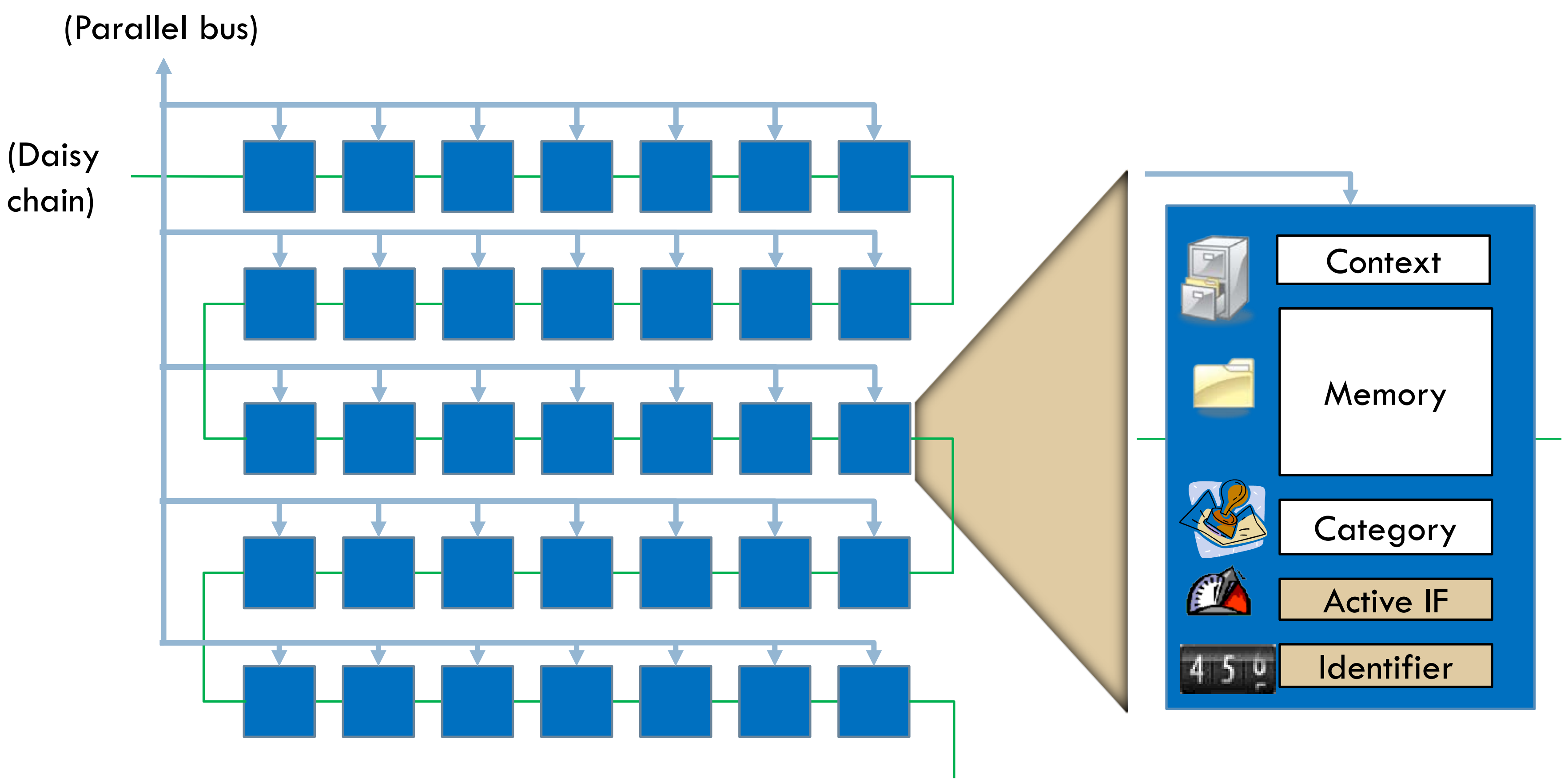
- Input= Stimuli
- Output=Decision



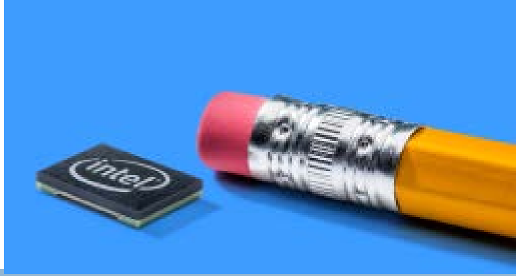
# About the neurons



**Chain of identical neuron cells, no supervisor, low clock, low power**



# Curie Neurons attributes



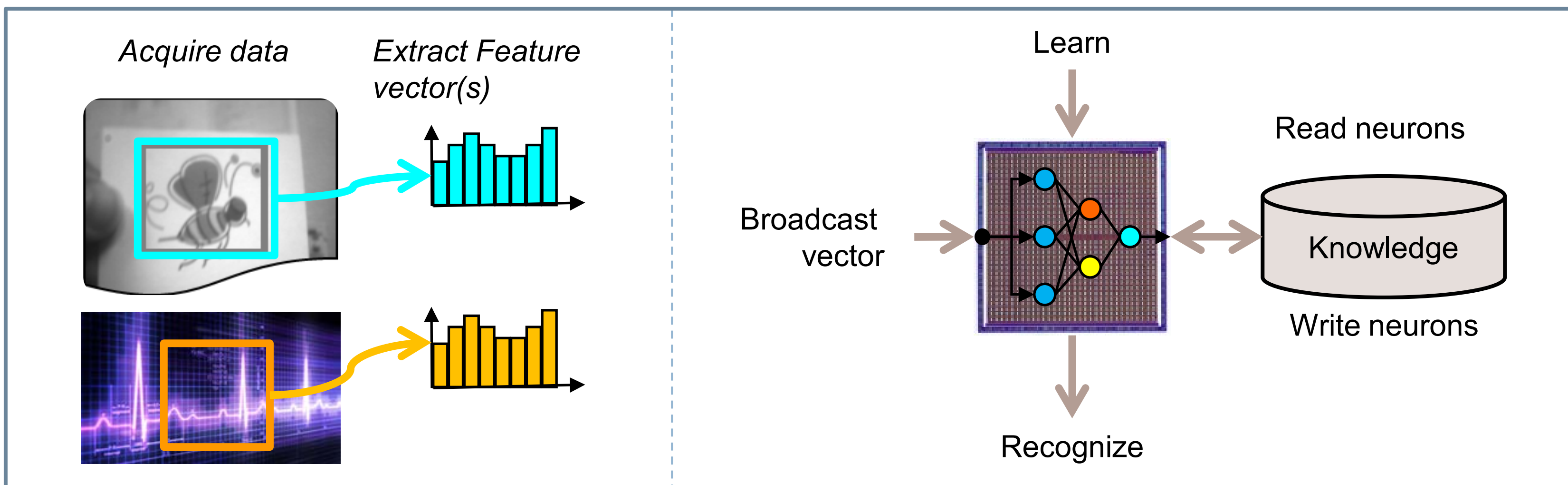



| ANN Attributes | CM1K |
|---|---|
| Neuron capacity | 128 |
| Neuron memory size | 128 bytes |
| Categories | 15 bits |
| Distances | 16 bits |
| Contexts | 7 bits |
| Recognition status | Identified, Uncertain or Unknown |
| Classifiers | Radial Basis Function (RBF)<br>K-Nearest Neighbor (KNN) |
| Distance Norms | L1 (Manhattan)<br>Lsup |

# How to interface to the neurons



- 4 basic functions
  - Learn / Recognize
  - Save / Restore knowledge
- Tuning and expansion options



# How to teach the neurons

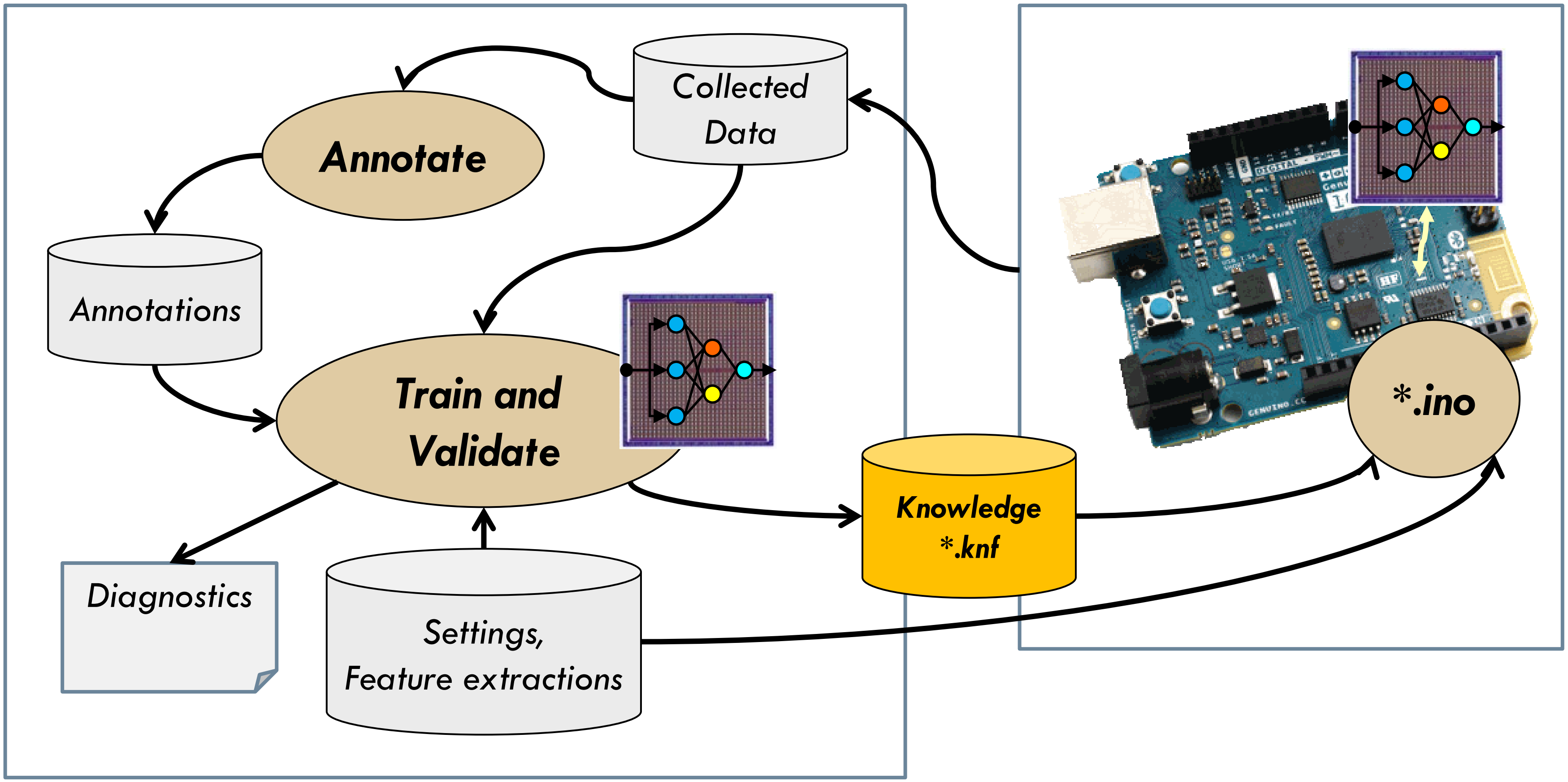


- Simple workflow
  - Data collection and annotation
  - Feature extraction
  - Broadcast to neurons with annotated category
  - The neurons build the knowledge autonomously
- Knowledge Builder suite for off-line training
  - UI for training and validation per sensor type
    - Curie KB for acceleration and gyro signals
    - Image KB for image data
    - More to come…
- CurieNeurons libraries for real-time training on Arduino/Genuino 101

# Application deployment w/ off-line training



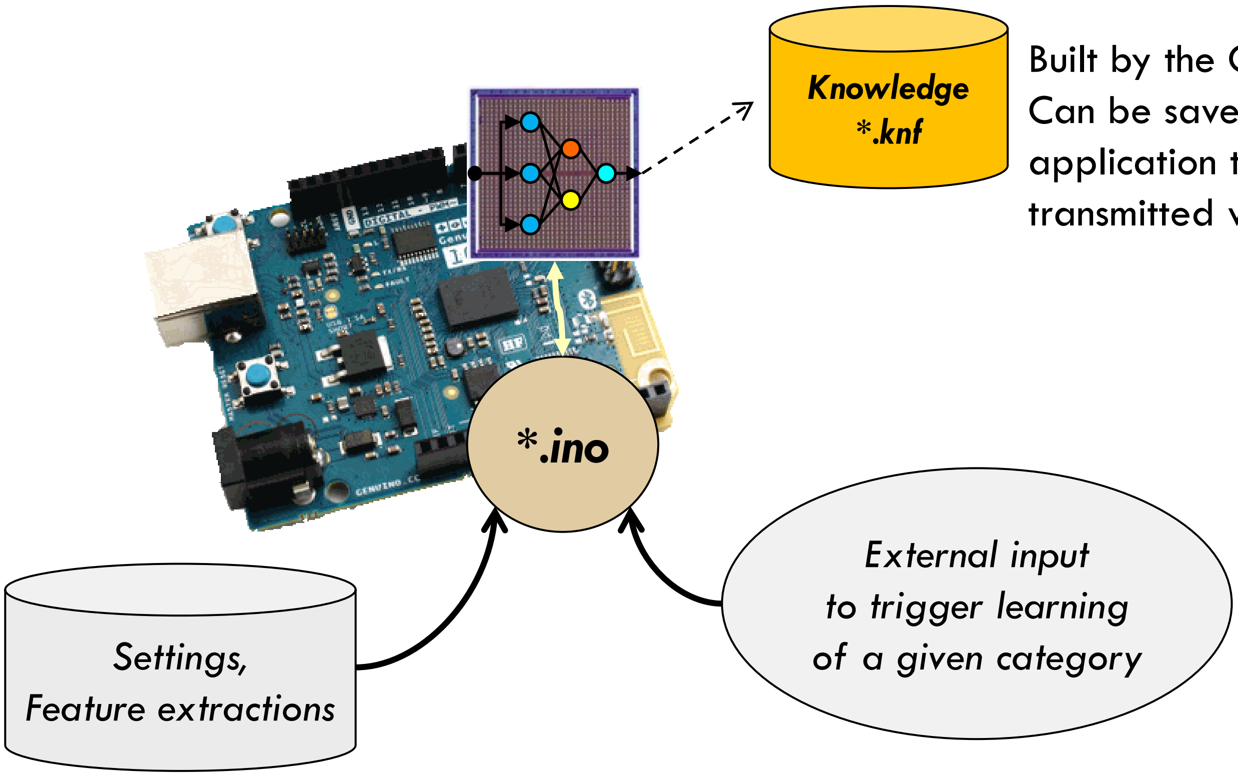
**Knowledge Builder Training platform** **Execution platform**



# Application deployment w/ live training



## Training & Execution on Curie




Built by the Curie neurons. Can be saved by the application to Flash, SD card, transmitted via BlueTooth, etc

# NeuroMem Knowledge Builder – Curie edition



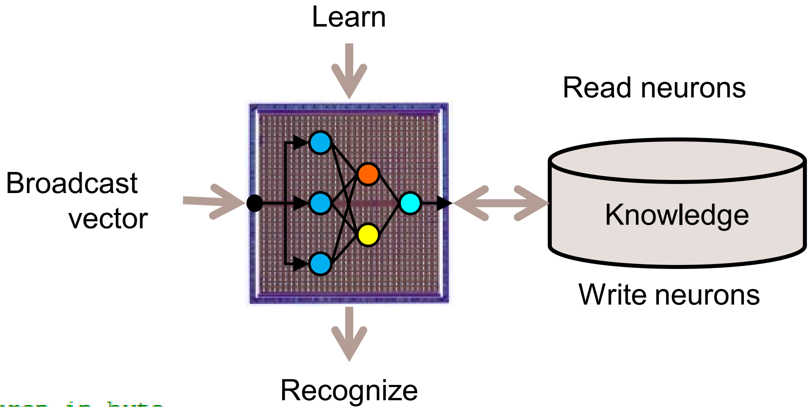


General Vision

# CurieNeurons library



- RBF classifier
- Single context
- No access to the neurons' registers




```
class CurieNeurons
{
    public:

        # define NEURONSIZE     128 //memory capacity of each neuron in byte
        # define MAXNEURONS     128 // number of silicon neurons

        CurieNeurons();
        void Init();
        void getNeuronsInfo(int* neuronSize, int* neuronsAvailable, int* neuronsCommitted);
        void Forget();
        void Forget(int Maxif);

        int Learn(unsigned char vector[], int length, int category);
        int Classify(unsigned char vector[], int length);
        int Classify(unsigned char vector[], int length, int* distance, int* category, int* nid);
        int Classify(unsigned char vector[], int length, int K, int distance[], int category[], int nid[]);

        void ReadNeuron(int nid, int* context, unsigned char model[], int* aif, int* category);
        void ReadNeuron(int nid, unsigned char neuron[]);
        int ReadNeurons(unsigned char neurons[]);
        int WriteNeurons(unsigned char neurons[]);
```

# CurieNeurons Geek library



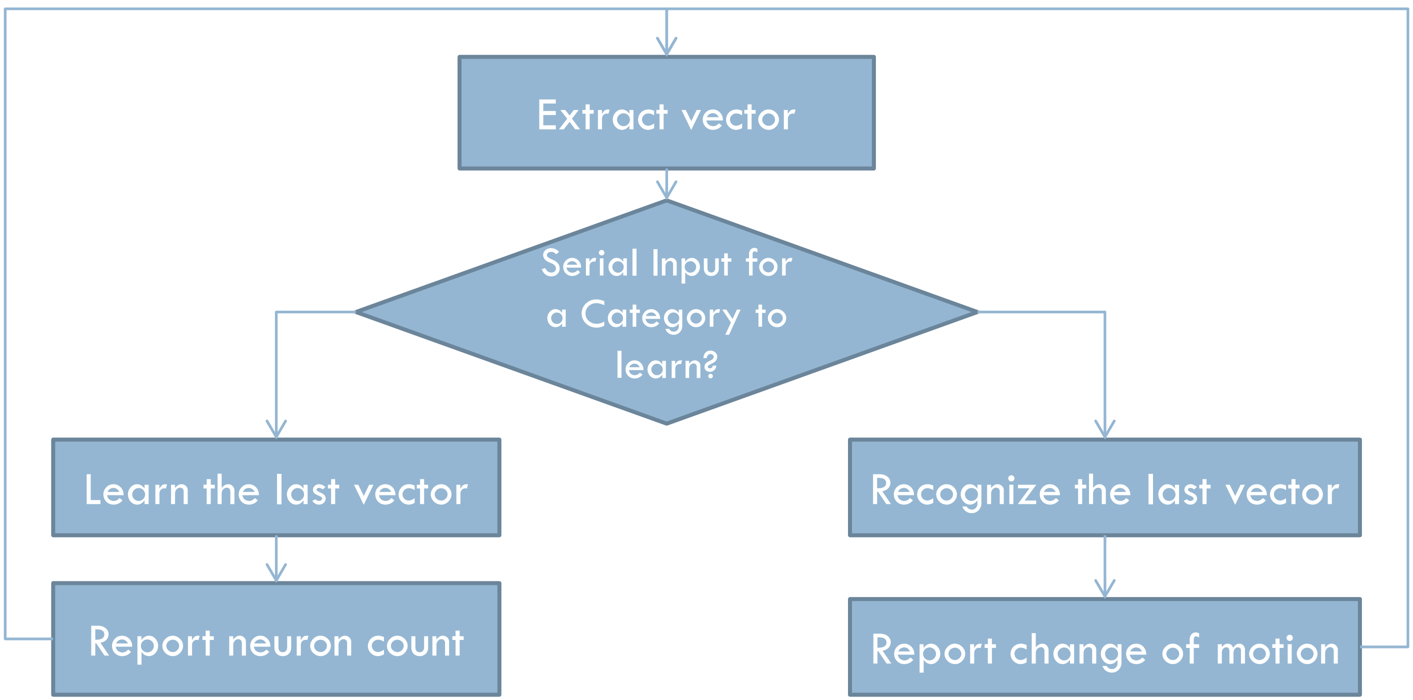
- Full access to the neurons' register
- Access to both RBF and KNN classifiers
- Access to multiple contexts
  - Sensor fusion
  - Cascade classifiers

```
//Functions available in the Geek Library
//---------------------------------------------

void SetContext(int context, int minif, int maxif);
void GetContext(int* context, int* minif, int* maxif);
void SetRBF();
void SetKNN();

int NCOUNT();
void NSR(int value);
int NSR();
void MINIF(int value);
int MINIF();
void MAXIF(int value);
int MAXIF();
void GCR(int value);
int GCR();
int DIST();
void CAT(int value);
int CAT();
void NID(int value);
int NID();
void RSTCHAIN();
void AIF(int value);
int AIF();
void IDX(int value);
```

# CurieNeurons_IMU Example



**Stimuli** = A simple feature vector is assembled and normalized over n samples
[ax1, ay1, az1, gx1,gy1, gz1, ax2, ay2, az2, gx2, gy2, gz2, ... axn, ayn, azn, gxn, gyn, gzn]
**Category**= 1 for vertical, 2 for horizontal, 0 for anything else



# The movie



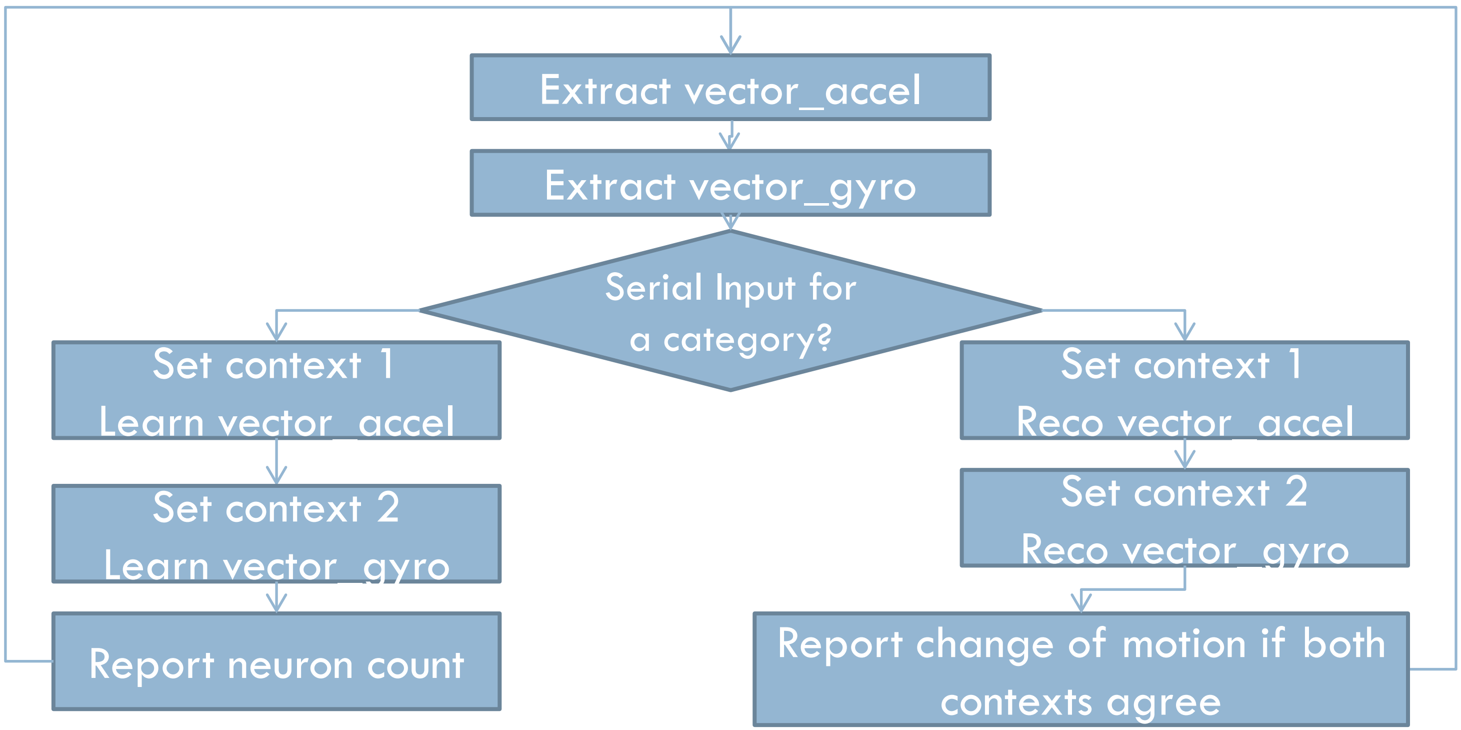

[View this introduction on our youtube account]

General Vision

# CurieNeurons_IMU2: 2 contexts



**Stimuli** = 2 simple feature vectors assembled and normalized over n samples
context 1, vector_accel= [ax1, ay1, az1, ax2, ay2, az2,… axn, ayn, azn]
context 2, vector_gyro= [gx1,gy1, gz1, gx2, gy2, gz2, … gxn, gyn, gzn]
**Category**= 1 for vertical, 2 for horizontal, 0 for anything else
**Observation**= commits more neurons, but less false hits



# What next?



- Free CurieNeurons library
- CurieNeuronsGeek library
- NeuroMem Knowledge Builder(Curie edition)
- Training courses on NeuroMem
- Thank you and visit us at www.general-vision.com