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

Teach the neurons with the push of a button or else, and immediately start recognizing

Monitor signals and act only when significant events occur.
What is NeuroMem?

- **NeuroMem**
  - = Neuromorphic Memories
  - = Digital neurons
  - = Trainable
  - = Parallel architecture

- 2015: Intel rolls out the QuarkSE, 1st SOC with NeuroMem inside (128 neurons with 128 bytes of memory per neuron)
- 2011: General Vision licenses its NeuroMem technology to Intel®
- 2007: General Vision introduces its NeuroMem CM1K chip (1024 neurons with 256 bytes of memory per neuron)
- 1993: IBM introduces the ZISC chip, ancestor of the NeuroMem chips (36 and 79 neurons of 64 bytes of memory per neuron)
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

<table>
<thead>
<tr>
<th>ANN Attributes</th>
<th>Quark SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neuron capacity</td>
<td>128</td>
</tr>
<tr>
<td>Neuron memory size</td>
<td>128 bytes</td>
</tr>
<tr>
<td>Categories</td>
<td>15 bits</td>
</tr>
<tr>
<td>Distances</td>
<td>16 bits</td>
</tr>
<tr>
<td>Contexts</td>
<td>7 bits</td>
</tr>
<tr>
<td>Recognition status</td>
<td>Identified, Uncertain or Unknown</td>
</tr>
<tr>
<td>Classifiers</td>
<td>Radial Basis Function (RBF)</td>
</tr>
<tr>
<td></td>
<td>K-Nearest Neighbor (KNN)</td>
</tr>
<tr>
<td>Distance Norms</td>
<td>L1 (Manhattan)</td>
</tr>
<tr>
<td></td>
<td>Lsup</td>
</tr>
</tbody>
</table>
A simple API

- **4 basic functions**
  - Learn/Recognize patterns (<=128 bytes)
  - Save / Restore knowledge

- **Additional settings**

---

- Learn pattern
- Recognize pattern
- Save Knowledge
- Load Knowledge
- RBF or KNN classifier
- Context segmentation
How to teach the neurons

- CurieNeurons libraries for real-time training
  - Data acquisition
  - Feature extraction
  - Broadcast to neurons for continuous recognition
  - User input to trigger a broadcast to neurons for learning, along with a category
  - The neurons build the knowledge autonomously

- Soon...Knowledge Builder apps for off-line training
  - Data collection and annotation
  - Learning of training sets, validation on testing sets
  - Export of the knowledge built by the neurons
Application deployment w/ live training

Training & Execution on Curie

- Knowledge: Built by the Curie neurons. Can be saved by the application to Flash, SD card, transmitted via BlueTooth, etc.
- Acquiring, Feature extractions
- Outputs to actuators, transmission, storage
- External input to trigger learning of a given category
Application deployment w/ off-line training

Knowledge Builder Training platform

- Annotate
- Collected Data
- Annotations
- Train and Validate
- Settings, Feature extractions
- Diagnostics

Execution platform

- Knowledge *.knf
- *.ino

General Vision 2/28/2017
CurieNeurons free library

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

```cpp
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[]);
}
```
CurieNeuronsPro library

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

```c
// 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);
```
Simple examples to get started

- **Simple script**
  - Understand the mechanism to learn, recognize user-generated vectors

- **Gesture recognition**
  - Using Curie’s 6-axis accelerometer/gyro

- **Video recognition**
  - Requires the ArduCam Shield board
**Stimuli** = A simple feature vector is assembled and normalized over n samples

\[ [ax_1, ay_1, az_1, gx_1, gy_1, gz_1, ax_2, ay_2, az_2, gx_2, gy_2, gz_2, \ldots ax_n, ay_n, az_n, gx_n, gy_n, gz_n] \]

**Category** = 1 for vertical, 2 for horizontal, 0 for anything else
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
Connecting the Intel Arduino/Genuino to the PC for demo of motion recognition

View this introduction on our youtube account
CurieNeurons w ArduCam (Pro only)

- Operation modes
  - Interlaced video display and recognition
  - User-Interrupt for learning
  - Optional Save of the knowledge

- Input
  - Shutter button
    - < 2 sec: learn a new category
    - > 2 sec: learn a background/null category
  - ROI is fixed and centered in video frame

- Output
  - LCD overlay after each frame capture
    - ROI rectangle
    - Text result
Knowledge Builder apps

- Off-line training and validation
- NeuroMem KB, generic and agnostic to data type
- Curie KB for acceleration and gyro
- Image KB for image and video
- More to come...
NeuroMem KB—Curie edition