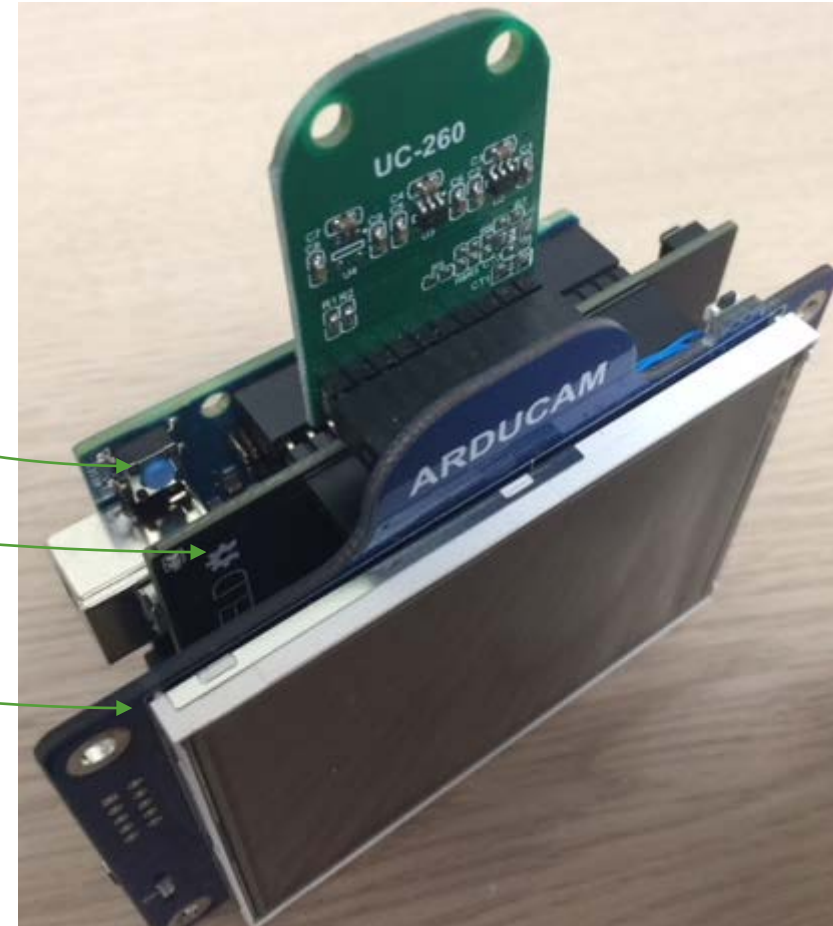


LIVE OBJECT TRACKING DEMO

Powered by ArduCam and a NeuroMem network

Assembly

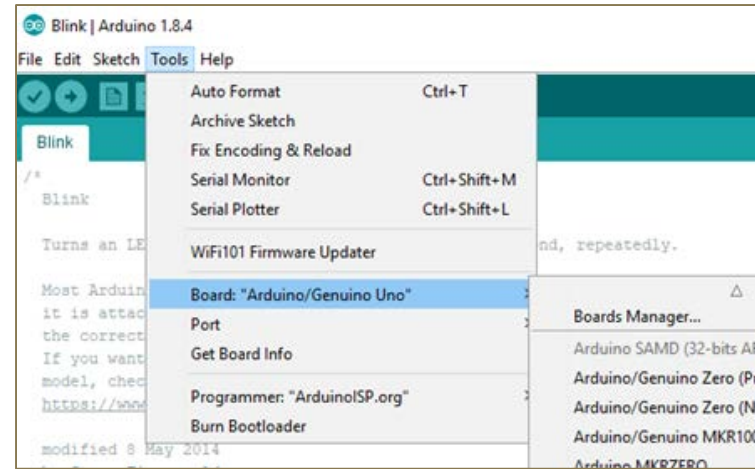
- Arduino microcontroller board with at least 5 KB memory
- NeuroShield featuring
 - InvenSense Accel/Gyro
 - 576 NeuroMem neurons
- ArduCam Shield V2 featuring
 - Color low-res CMOS sensor
 - LCD display
- Optional sets of spacers for Arduino
- Custom cable connecting the SPI 6-pin connector of the Arduino 101 to the ArduCam Shield



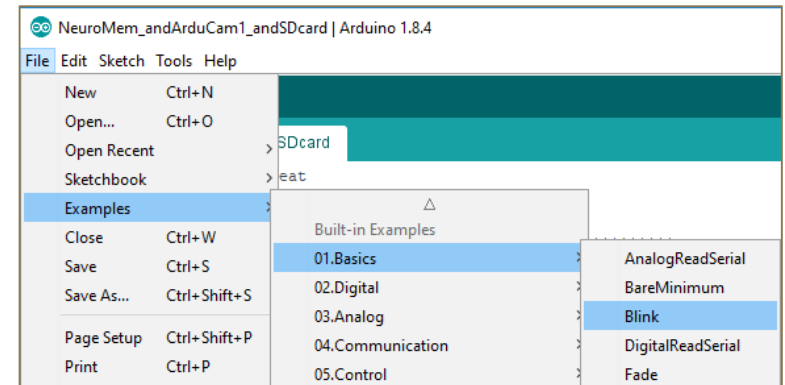
Installation

1. Requirements: Arduino IDE (you can download the latest version at <https://www.arduino.cc/en/Main/Software>)
2. Select your Arduino board under Tools\Board menu. If not in the list, select Board Manager and install its driver
3. Load the File\Examples\Basic\Blink script
4. Upload the script to your board
5. Verify that the LED of the microcontroller board is blinking

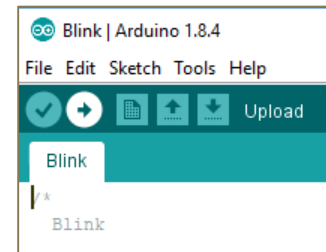
2



3



4



Setup

- Open the serial monitor
- Edit the NeuroMem platform
- Live video appears at the end of the setup
- The blinking rectangle at the center of the screen is the target being taught or monitored

NeuroShield_andArduCam1_andSDcard | Arduino 1.8.4

File Edit Sketch Tools Help

Verify

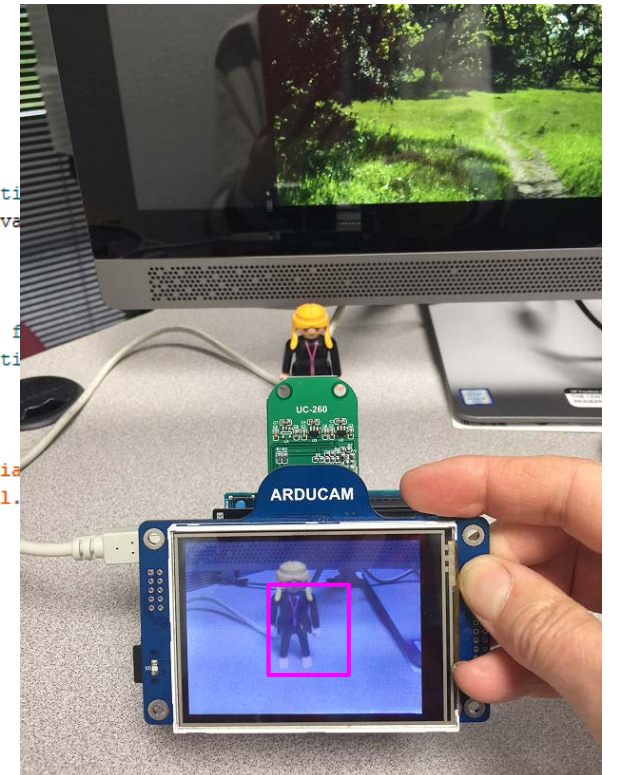
NeuroShield_andArduCam1_andSDcard \$

```
// Initialize the ArduCAM
myCAM.InitCAM();

// Check if the camera module type is OV2640
myCAM.wrSensorReg8_8(0xff, 0x01);
myCAM.rdSensorReg8_8(OV2640_CHIPID_HIGH, &vid);
myCAM.rdSensorReg8_8(OV2640_CHIPID_LOW, &pid);
if ((vid != 0x26) && ((pid != 0x41) || (pid != 0x42))) {
  Serial.println("Can't find OV2640 module!");
} else {
  Serial.println("OV2640 detected.");
}

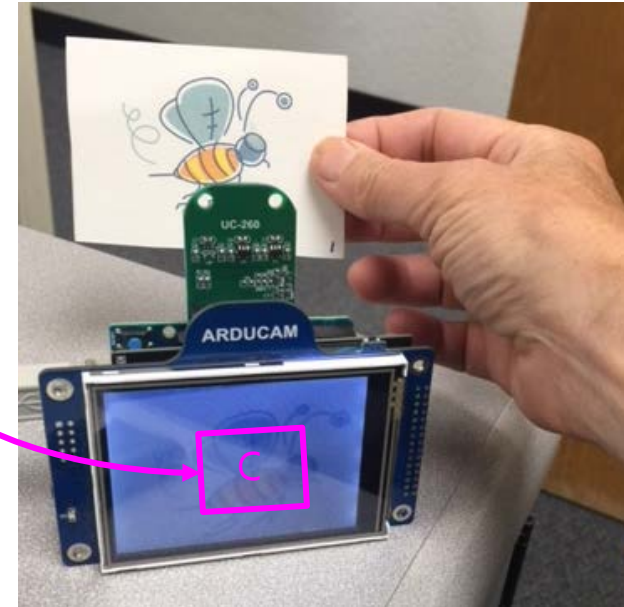
// Initialize the NeuroMem neural network
int NMplatform=2; //NeuroShield
if (hNN.begin(NMplatform) == 0)
{
  Serial.print("\nYour NeuroMem_Smart device is initialized");
  Serial.print("\nThere are "); Serial.print(hNN.nava
}
else
{
  Serial.print("\nYour NeuroMem_Smart device is NOT initialized");
  Serial.print("\nCheck your device type and connection");
  while (1);
}

Serial.print("Image width="); Serial.print(fw); Serial.print("\n");
Serial.print("ROI width="); Serial.print(rw); Serial.print("\n");
displayLCD_res("Ready", 10,10);
delay(100);
}
```



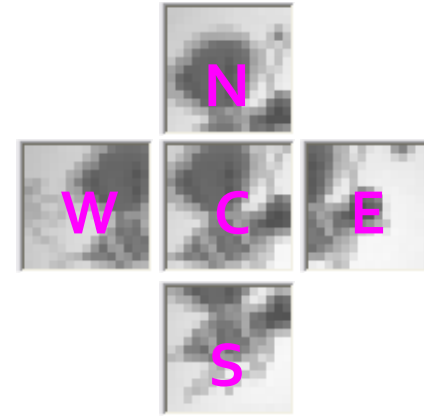
Learning the target

- Move camera so the target appears within the blinking rectangle
- Press the upper right button for 2 seconds
- Unless you moved, the letter "C" for Centered should appear in the middle of the screen



Learning logic

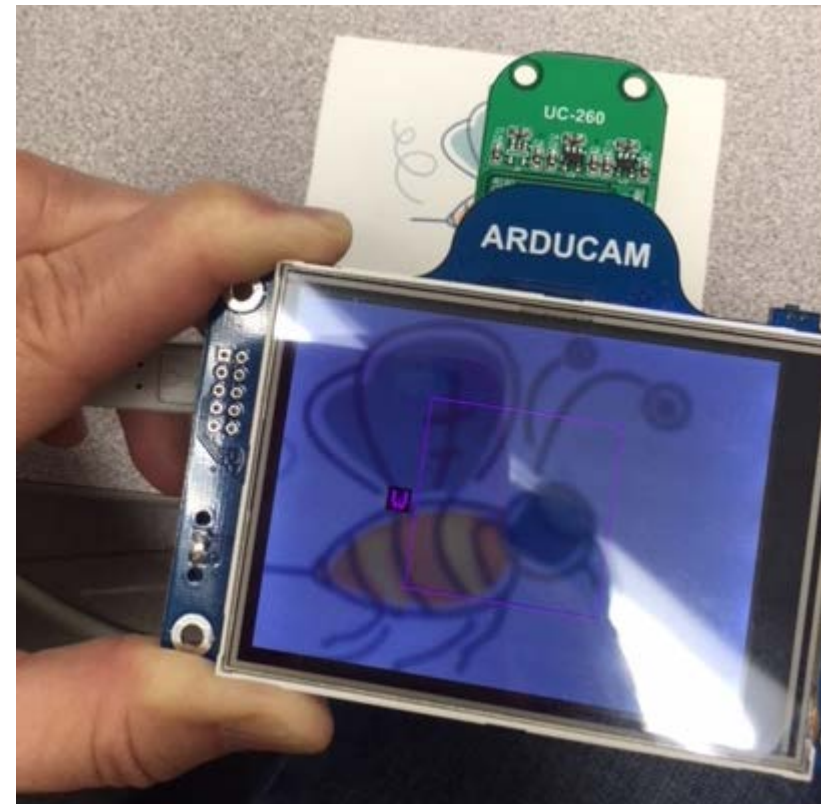
- Learn the central pattern as “Centered”
- Learn the surrounding cardinal patterns as “North, East, South, West”
- 1 learning can commit up to 5 neurons



```
NeuroShield_andArduCam_Tracker §  
  
// Supervised Learning  
getSubsample(left, top);  
hNN.learn(vector, vlen, 1); // center or category 1  
getSubsample(left, top - offsetY) ;  
hNN.learn(vector, vlen, 2); // north or category 2  
getSubsample(left + offsetX, top) ;  
hNN.learn(vector, vlen, 3); // east or category 3  
getSubsample(left, top + offsetY) ;  
hNN.learn(vector, vlen, 4); // south or category 4  
getSubsample(left - offsetX, top) ;  
hNN.learn(vector, vlen, 5); // west or category 5
```

Tracking report

- Move camera left until the reported category "W" appears
- Move camera right until the reported category "E" appears
- Move camera Up until the reported category "N" appears
- Move camera down until the reported category "S" appears



Summary: A Simple and limited UI

- Operation modes
 - Interlaced video display and recognition
 - User-Interrupt for learning the target
 - Optional Save of the knowledge
- Input
 - Shutter button used to trigger learning
 - < 2 sec : learn a new category
 - ROI is fixed and centered in video frame
 - The Center, East, West, North and South ROIs are learned automatically
- Output
 - LCD overlay after each frame capture
 - ROI rectangle
 - Abbreviation of the recognized category : C, W, E, N, S

What is next ?

- Mount the camera on a motor and use the output of the neurons to move the camera and re-center the target in the field of view
- Optimize the speed performance by displaying only 1 frame every N on the LCD (if possible)
- Extract more advanced feature(s) depending on the scenery
- And more....