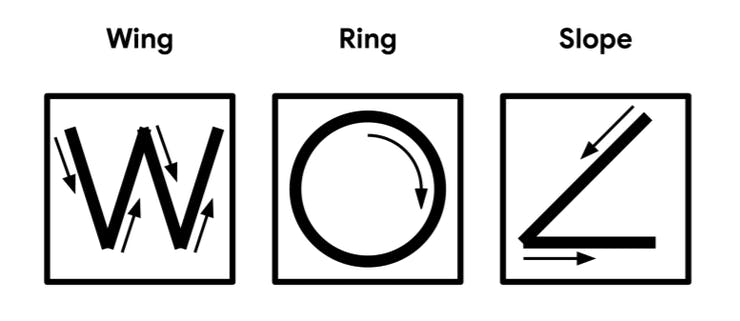
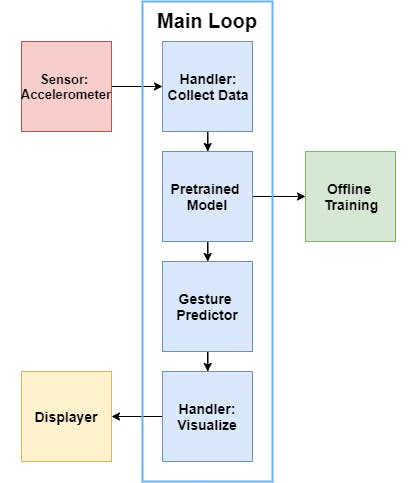
Human behavior is a rich data source that can be acquired in all kinds of daily scenes. It will be quite meaningful for agents to figure out what's going on with a specific person and further contribute to his or her next move. However, to deal with some complex data collected by the sensors, hand-crafted algorithms based on human expertise are required, which is not extendable between diverse application scenarios. Therefore, a machine learning algorithm is a good solution which can be easily applied under different scenarios, enabling the model to learn the relationship between complex data without high domain knowledge. To make machine learning applicable in resource-constrained platforms like smartphone, smartwatch or other daily-use devices, compression techniques like pruning and quantization and customized framework are necessary. That's exactly what TFLite does. In this project, we will build a demo, i.e. apply a machine learning algorithm on Arduino Nano 33 to build a magic wand based on TFLite framework, to show the potential of embedded machine learning in automated industry.

* **Objective**



3 kinds of gestures

* **Design** **Structure**



Design Structure

The 9 axis inertial sensor will send the current acceleration of the board to the handler that collects data for the Machine Learning Model. And the model has been already pre-trained offline, so it can predict the gesture directly by sending the extracted feature to the Gesture Predictor. Finally, the Handler for visualization could send the corresponding gesture to the Displayer, and we can see it from the terminal in Arduino IDE.

* **Deployment**

To run this project, we could follow the instructions below.

1. Add corresponding libraries

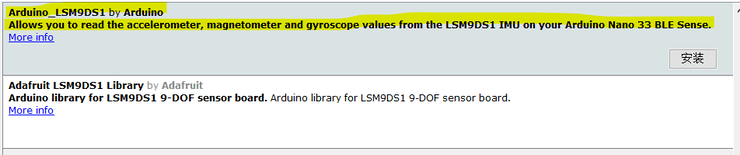
After installing Arduino IDE from [Arduino official website](https://www.arduino.cc/en/main/software), we click Tool -> Manage Libraries and search "Arduino\_TensorFlowLite" on the pop-up. It will show the following result.



search "Arduino\_TensorFlowLite"

After click "install", it will install this library for you automatically.

Then, repeat the above process to install "Arduino\_LSM9DS1" to handle the 9 axis inertial sensor in the Nano 33BLE. Please make sure the one you installed is provided by Arduino as shown by the following figure.



search "Arduino\_LSM9DS1"

2. Modify some details in the added libraries

To use the LSM9DS1 sensor in our project, some modification to the source file is needed.  
For Windows user, the source file could be found in "C:\Users\username\Documents\Arduino\libraries\Arduino\_LSM9DS1", for Linux/Unix user, the source file could be found in "/home/username/Arduino\_LSM9DS1/".  
  
Then open the "src/LSM9DS1.cpp" and find the "LSM9DS1Class::begin()", then insert the following code right before the "return".

// Enable FIFO (see docs https://www.st.com/resource/en/datasheet/DM00103319.pdf)  
writeRegister(LSM9DS1\_ADDRESS, 0x23, 0x02);  
// Set continuous mode  
writeRegister(LSM9DS1\_ADDRESS, 0x2E, 0xC0);

Also, please find the "LSM9DS1Class::accelerationAvailable()" and do some replacement as shown by the following code.

int LSM9DS1Class::accelerationAvailable()  
{  
 /\*  
 if (readRegister(LSM9DS1\_ADDRESS, LSM9DS1\_STATUS\_REG) & 0x01) {   
 return 1;   
 }  
 \*/  
  
 // Read FIFO\_SRC. If any of the rightmost 8 bits have a value, there is data  
 if (readRegister(LSM9DS1\_ADDRESS, 0x2F) & 63) {  
 return 1;  
 }  
 return 0;  
}

Please save it after finishing the above modification.

3. Load code for Magic Wand  
Then we can load the official example and play with it.  
In the Arduino IDE, click File-> Examples -> TensorFlowLite -> magic\_wand, then you can see the example c++ code for this application and no further modification is needed.  
Just go on and select the board we are using by click Tools -> Board -> Arduino Nano 33 BLE. Then click Tools -> Port -> COM\* or dev/ttyACM\* (varying on different computers) to set up the port for flashing code and serial communication.  
  
Then press the "upload" button in the Arduino IDE and click Tools -> Serial Monitor after the uploading to visualize the gesture prediction as shown by the following video demo.