

# Deploying the Magic Wand Project

In this document we are going to deploy the pre-trained magic wand project which can recognize gestures of the digits 0-9 by analyzing the time series data produced by the IMU!

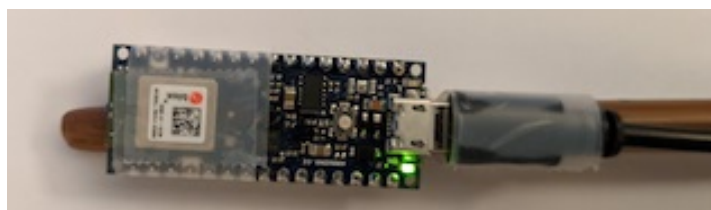
Screencast of Brian  
walking through this  
section goes here  
on the edX course

## Building the Wand

Note that you do not have to build the full magic wand and can simply hold the Arduino (with or without the shield) in your hand. However, you will likely find that when you deploy the example with Pete's pre-trained digit model, since he trained it using a 'wand,' it will be more accurate if you use a 'wand' too.

If you'd like to build a 'wand,' the 'wand' itself can be as simple as a stick. It doesn't need to do anything other than keep the board at its end as you hold the other end and wave it about. For fun, the course staff members have found cheap wands from online retailers and used them, but a simple piece of wood, a ruler, or even a thick piece of cardboard should work just as well.

You should place the board at the end of the wand, with the USB socket facing downwards, towards where you hold it, so that the cable can run down the handle. The sketch is designed to compensate for any rotation of the board around the wand's shaft, so as long as it's parallel to the wand's length, the board's twist won't matter. Use some sticky tape or some other easy-to-remove method to attach the board to the wand, and hold the cable in place along the shaft. The end result should look something like the image below.



If an ASCII-art diagram is more helpful, here's what you should aim for:

```

----
|      |<- Arduino board
|      |
| ( ) |  <- Reset button
|      |
- - - - <- USB port
  ||
  ||<- Wand
. . . .
  ||
  ||
  ( )
```

## Deploying the Magic Wand Project

1. Use a USB cable to connect the Arduino Nano 33 BLE Sense to your machine. You should see the green LED power indicator come on when the board first receives power.
2. Open the magic\_wand.ino sketch, which you can find via the File drop-down menu. Navigate, as follows: [File](#) → [Examples](#) → [Harvard\\_TinyMLx](#) → [magic\\_wand](#).
3. As always, use the Tools drop down menu to select appropriate Port and Board.
  - a. Select the Arduino Nano 33 BLE as the board by going to [Tools](#) → [Board:](#) [<Current Board Name>](#) → [Arduino Mbed OS Boards \(nRF52840\)](#) → [Arduino Nano 33 BLE](#). Note that on different operating systems the exact name of the board may vary but/and it should include the word Nano at a minimum. If you do not see that as an option then please go back to Setting up the Software and make sure you have installed the necessary board files.
  - b. Then select the USB Port associated with your board. This will appear differently on Windows, macOS, Linux but will likely indicate 'Arduino Nano 33 BLE" in parenthesis. You can select this by going to [Tools](#) → [Port:](#) [<Current Port \(Board on Port\)>](#) → [<TBD Based on OS> \(Arduino Nano 33 BLE\)](#). Where [<TBD Based on OS>](#) is most likely to come from the list below where [<#>](#) indicates some integer number
    - i. Windows → [COM<#>](#)
    - ii. macOS → [/dev/cu.usbmodem<#>](#)
    - iii. Linux → [ttyUSB<#>](#) or [ttyACM<#>](#)

4. Use the rightward arrow to upload / flash the code. Do not be alarmed if you see a series of orange warnings appear in the console. This is expected as we are working with bleeding edge code. You'll know the upload is complete when you see red text in the console at the bottom of the IDE that shows 100% upload of the code and a statement that says something like "Done in <#.##> seconds."

To help you debug other issues please check out our [FAQ appendix](#) with answers to the most common errors!

```

.....
.....
.....
#####
#      ##   ##
#      .##  #
###    ###   ##
#          ##
#       #     #
#       #     #
##      #     #
#         #
#         #
#         #
#         #
#         #
#         #
#         #
##        ##
##           ##
##      # #####
#####
.....

```

Found 0 (126)