

Projet 2 : Détection des mots en utilisant edge impulse

Dans ce projet, nous allons construire un modèle à l'aide d'edge impulse et la carte arduino nano 32 BLE permettant de reconnaître les mots suivants : « FPGA », « Résistance », « Transistor » et « microcontrôleur »

Enoncé du projet : [Projets pratique TinyML - Google Docs](#)

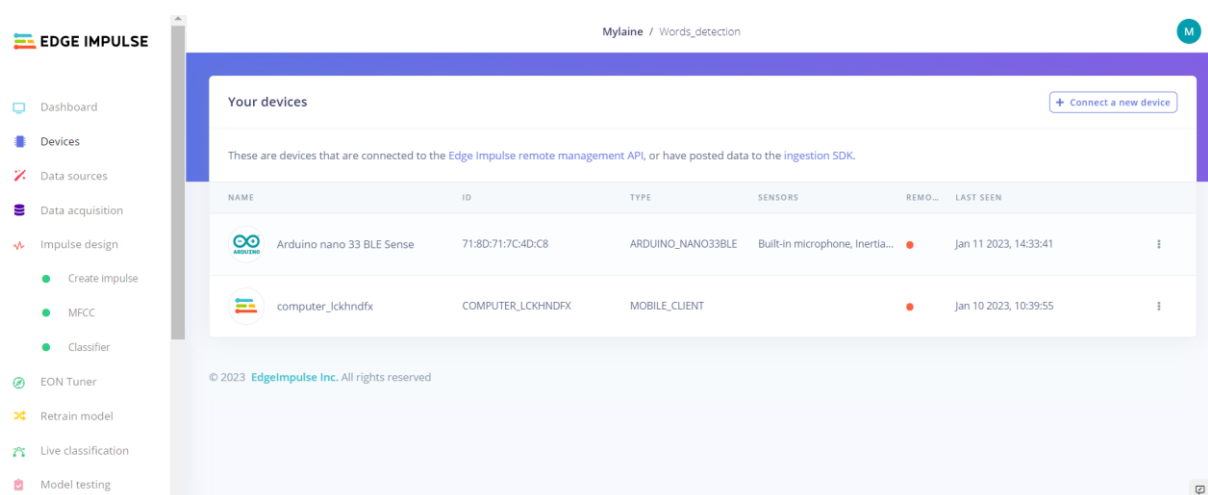
Liens utiles : [Edge Impulse with the Nano 33 BLE Sense | Arduino Documentation](#) | [Arduino Documentation](#)

Connect to device :

Nous devons dans un premier temps créer un compte sur edge impulse et ensuite connecter la carte à la plateforme.

Le lien suivant nous montre les étapes de connections d'une carte à l'interface edge impulse

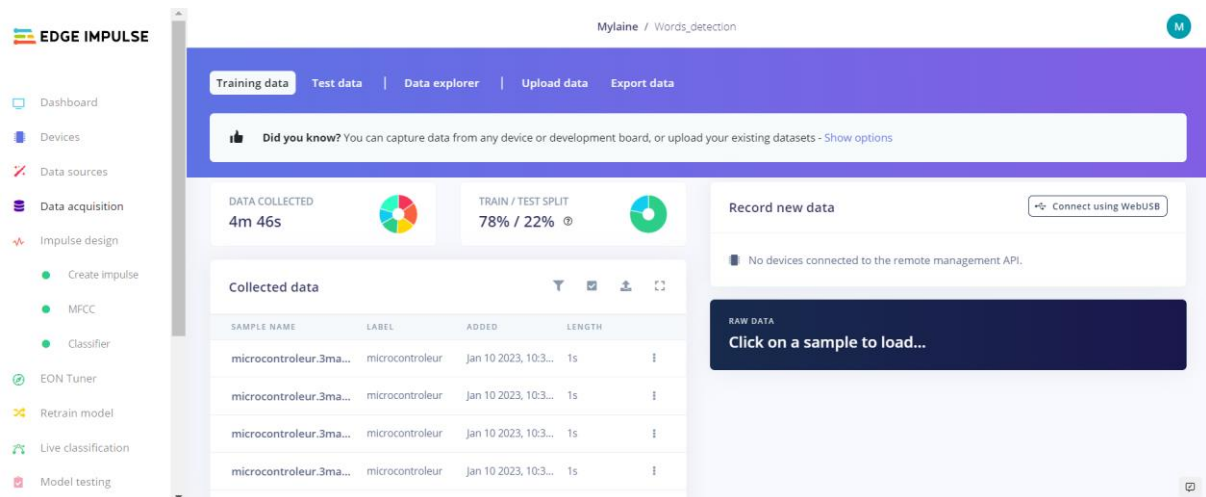
<https://docs.edgeimpulse.com/docs/development-platforms/fully-supported-development-boards>



Après s'être connecter, nous devons recueillir les données en utilisant le microphone de la carte

- Data acquisition :

En utilisant le microphone de notre arduino nano BLE 33, nous devons enregistrer nos mots les uns après les autres en les catégorisant par label



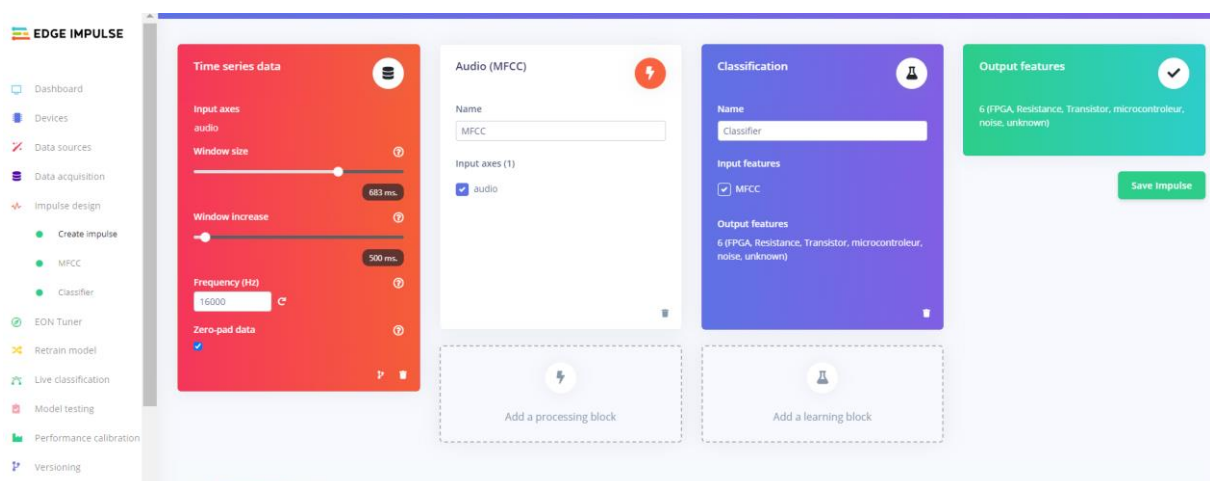
Nous devons nous rassurer que la quantité de données collectées est sensiblement la même pour tous les labels.

Avant de passer au test, nous devons diviser nos données en deux catégories (une pour l'apprentissage et l'autre pour le test).

Dashboard -> perform split/train data

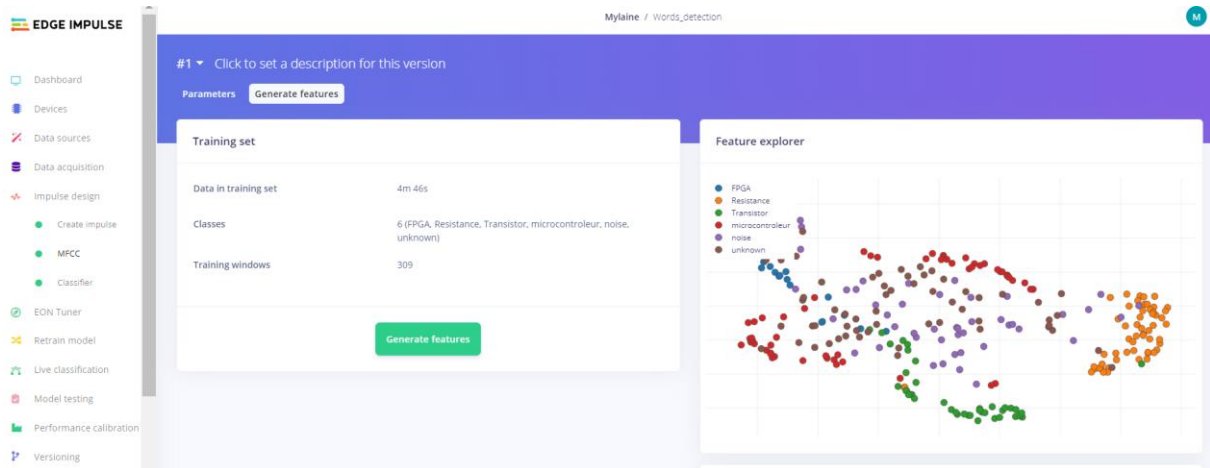
- Create impulse :

Choisir audio comme type de données et la classification comme méthode d'entraînement

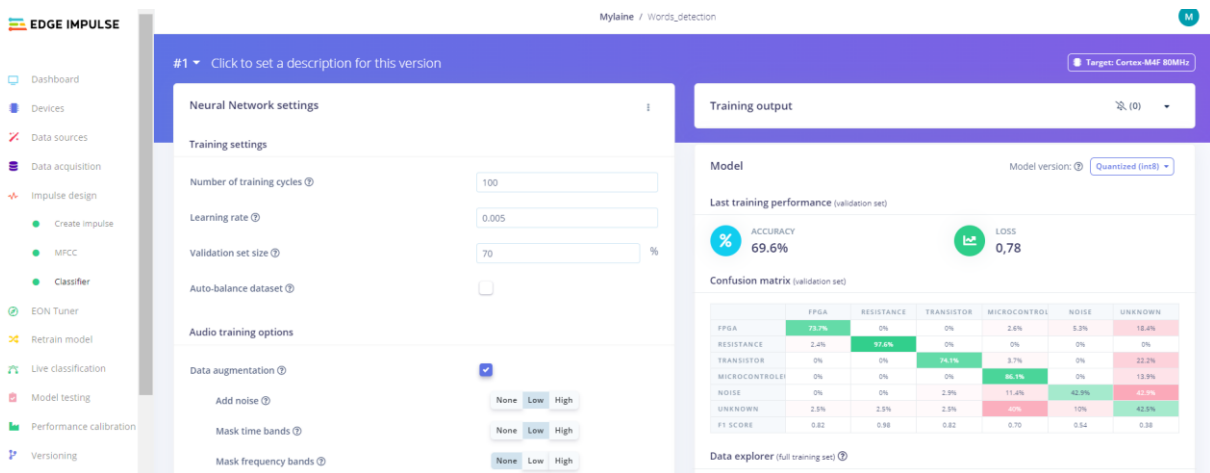


- MCCF

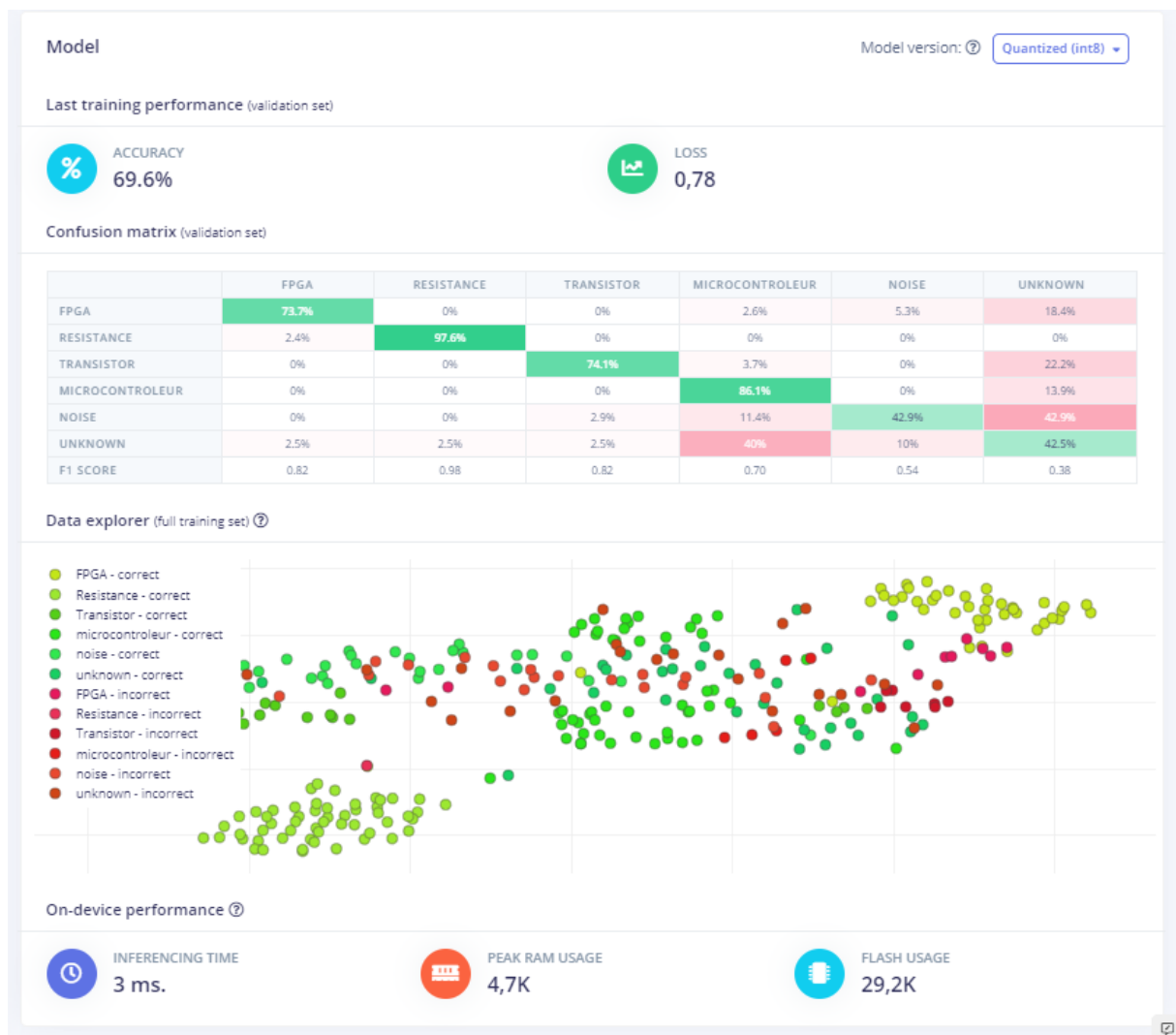
Enregistrer les paramètres -> generate features :



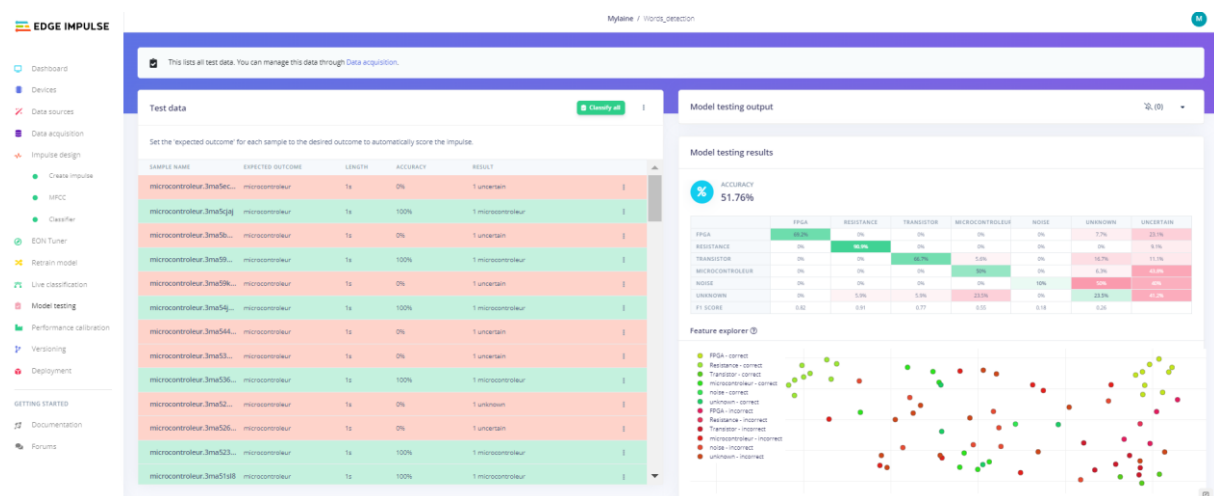
- Classifier :



Training output :



- Model testing : cliquer sur clasify all



Testing output :

Model testing output

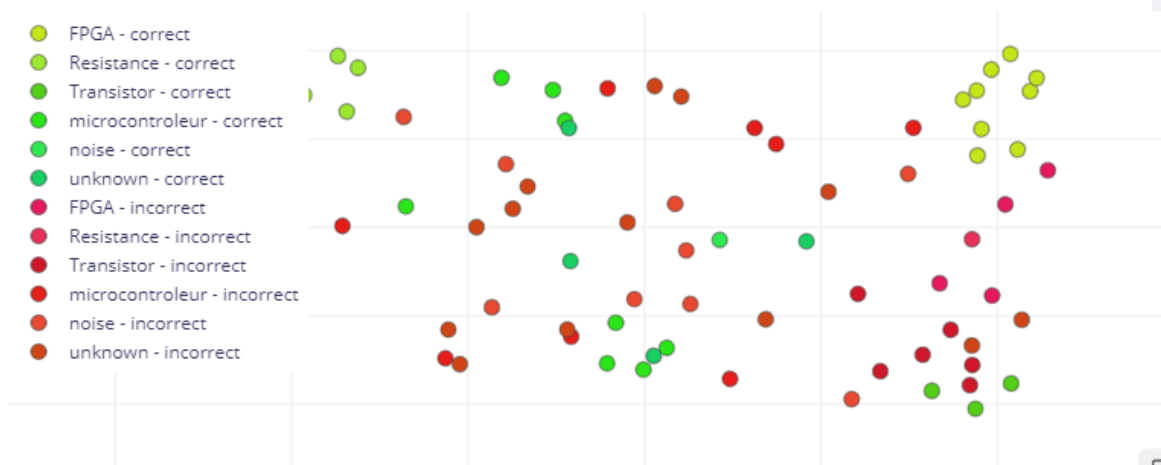
🔊 (0) ▼

Model testing results

% ACCURACY
51.76%

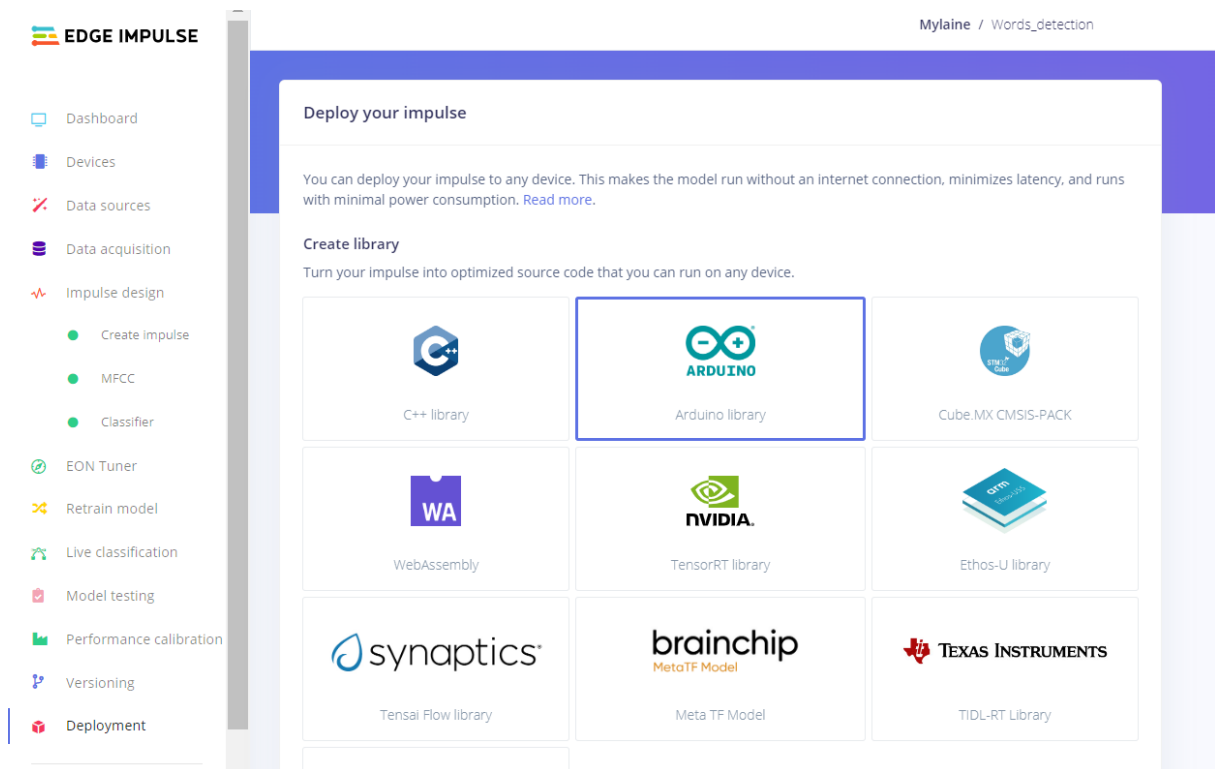
	FPGA	RESISTANCE	TRANSISTOR	MICROCONTR	NOISE	UNKNOWN	UNCERTAIN
FPGA	69.2%	0%	0%	0%	0%	7.7%	23.1%
RESISTANCE	0%	90.9%	0%	0%	0%	0%	9.1%
TRANSISTOR	0%	0%	66.7%	5.6%	0%	16.7%	11.1%
MICROCONTROLEUR	0%	0%	0%	50%	0%	6.3%	43.8%
NOISE	0%	0%	0%	0%	10%	50%	40%
UNKNOWN	0%	5.9%	5.9%	23.5%	0%	23.5%	41.2%
F1 SCORE	0.82	0.91	0.77	0.55	0.18	0.26	

Feature explorer ?



- Deployment :

Sélectionner arduino library pour une utilisation sur l'IDE arduino



Run optimization

Select optimizations *(optional)*

Model optimizations can increase on-device performance but may reduce accuracy. Click below to analyze optimizations and see the recommended choices for your target. Or, just click Build to use the currently selected options.






Enable EON™ Compiler

Same accuracy, up to 50% less memory. Open source.



Available optimizations for Classifier

<div>Quantized (int8) </div> <div>Currently selected</div> <div>This optimization is recommended for best performance.</div>	RAM USAGE	LATENCY	CONFUSION MATRIX 																																										
	4,7K	3 ms	<table><tr><td>69.2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>7.7</td><td>23.1</td></tr><tr><td>0</td><td>90.9</td><td>0</td><td>0</td><td>0</td><td>0</td><td>9.1</td></tr><tr><td>0</td><td>0</td><td>66.7</td><td>5.6</td><td>0</td><td>16.7</td><td>11.1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>50</td><td>0</td><td>6.3</td><td>43.8</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>10</td><td>50</td><td>40</td></tr><tr><td>0</td><td>5.9</td><td>5.9</td><td>23.5</td><td>0</td><td>23.5</td><td>41.2</td></tr></table>	69.2	0	0	0	0	7.7	23.1	0	90.9	0	0	0	0	9.1	0	0	66.7	5.6	0	16.7	11.1	0	0	0	50	0	6.3	43.8	0	0	0	0	10	50	40	0	5.9	5.9	23.5	0	23.5	41.2
	69.2	0	0	0	0	7.7	23.1																																						
	0	90.9	0	0	0	0	9.1																																						
0	0	66.7	5.6	0	16.7	11.1																																							
0	0	0	50	0	6.3	43.8																																							
0	0	0	0	10	50	40																																							
0	5.9	5.9	23.5	0	23.5	41.2																																							
FLASH USAGE	ACCURACY																																												
29,2K	51.76%																																												

<div>Unoptimized (float32)</div> <div>Click to select</div>	RAM USAGE	LATENCY	CONFUSION MATRIX 																																										
	6,8K	24 ms	<table><tr><td>69.2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>7.7</td><td>23.1</td></tr><tr><td>0</td><td>90.9</td><td>0</td><td>0</td><td>0</td><td>0</td><td>9.1</td></tr><tr><td>0</td><td>0</td><td>66.7</td><td>5.6</td><td>0</td><td>16.7</td><td>11.1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>50</td><td>0</td><td>6.3</td><td>43.8</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>10</td><td>50</td><td>40</td></tr><tr><td>0</td><td>5.9</td><td>5.9</td><td>23.5</td><td>0</td><td>23.5</td><td>41.2</td></tr></table>	69.2	0	0	0	0	7.7	23.1	0	90.9	0	0	0	0	9.1	0	0	66.7	5.6	0	16.7	11.1	0	0	0	50	0	6.3	43.8	0	0	0	0	10	50	40	0	5.9	5.9	23.5	0	23.5	41.2
	69.2	0	0	0	0	7.7	23.1																																						
	0	90.9	0	0	0	0	9.1																																						
0	0	66.7	5.6	0	16.7	11.1																																							
0	0	0	50	0	6.3	43.8																																							
0	0	0	0	10	50	40																																							
0	5.9	5.9	23.5	0	23.5	41.2																																							
FLASH USAGE	ACCURACY																																												
24,6K	51.76%																																												

Estimate for Cortex-M4F 80MHz.

En enfin build.

Celui-ci nous génère un dossier.zip que nous devons inclure dans l'IDE arduino en tant que librairie et utiliser des exemples à nos intérêts.

Ouvrir le fichier exemple d'utilisation continue à l'aide d'un microphone et ajouter les débris de code suivant :

```
|
#define Ledpin 0 // definition du port

void setup()
{
    // put your setup code here, to run once:
    Serial.begin(115200);
    // comment out the below line to cancel the wait for USB connection (needed for native USB)
    while (!Serial);
    Serial.println("Edge Impulse Inferencing Demo");

    /*****
    pinMode(Ledpin, OUTPUT); // ledpin connecter en sortie
    *****/

    for (size_t ix = 0; ix < EI_CLASSIFIER_LABEL_COUNT; ix++) {
        ei_printf("    %s: %.5f\n", result.classification[ix].label,
            result.classification[ix].value);

        /*****
        // definition du clignotement de la led
        if(result.classification[ix].value <0.7){
            digitalWrite(Ledpin, HIGH); // la led s'allume
            delay(1000); // reste allumer pendant 1s
            digitalWrite(Ledpin, LOW); // la led s'eteint
            delay(1000); // reste éteinte pendant 1s
        }
        *****/
    }
    *
```

Dans l'onglet outils, sélectionner le type de carte et le port et enfin compiler et televerser le programme sur la carte