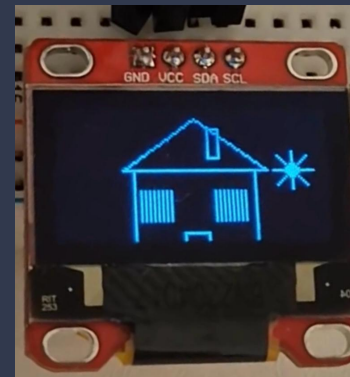


NEURAL SPEECH - CALCUL EMBARQUÉ

- Jérémy Populaire
- Angel Velasco
- Michael Adda
- Grégoire Marchal

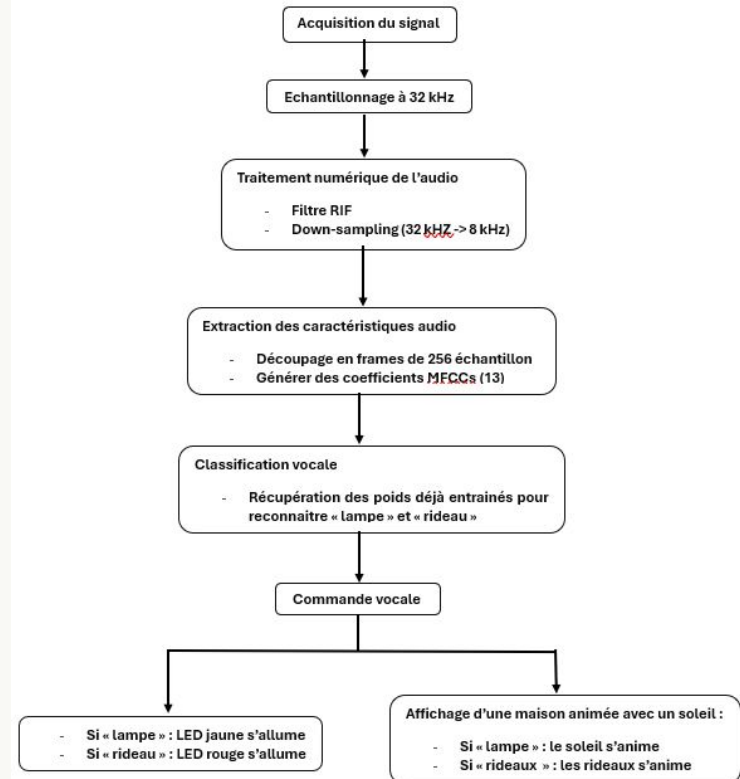


PRÉSENTATION USE CASE

Slide 2/11

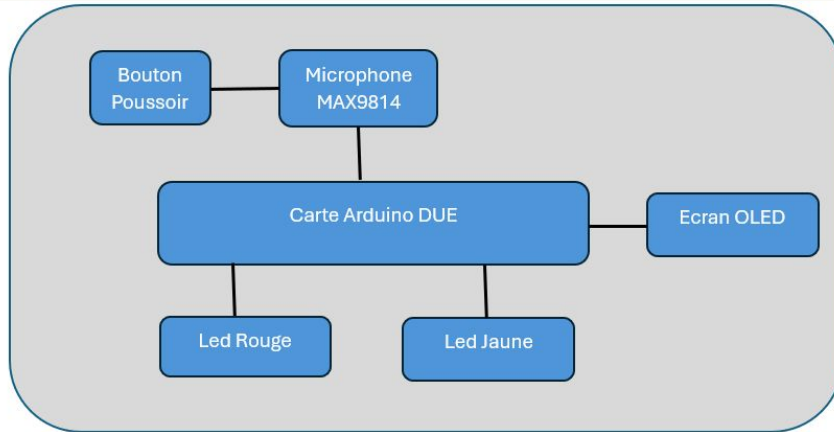
Choix de la Maison Connectée

- Allumage d'une LED rouge lorsque le mot "Lampe" est détecté et d'une LED blanche lorsque le mot "Rideaux" est détecté
- Affichage et animation sur écran Affichage d'une animation représentant une maison avec un soleil
 - Si le mot "Lampe" est reconnu, le soleil s'anime
 - Si le mot "Rideaux" est reconnu, les rideaux s'animent



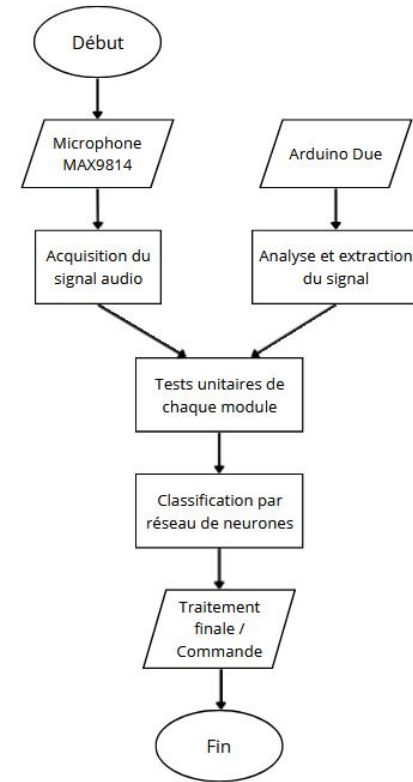
Diagrammes HW et SW

Slide 3/11



Légende :

Connecteur



NUMÉRISATION

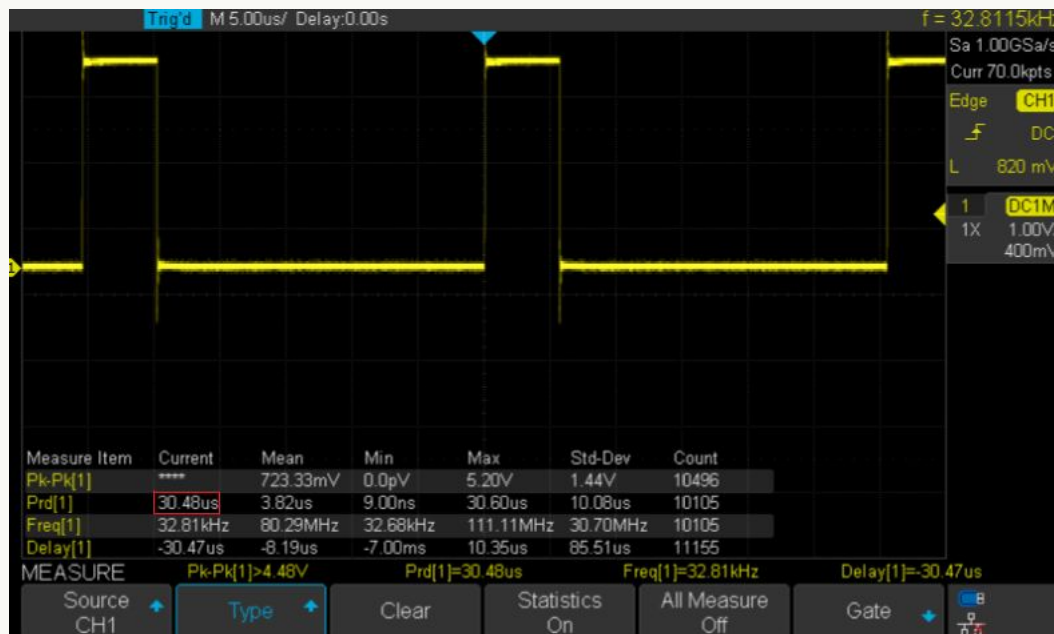
Slide 4/11

Configuration de l'ADC en mode interruption sur Timer

Période mesurée : 30.48 μ s

Fréquence mesurée : 32.81 kHz

Cette fréquence d'échantillonnage de 32,8 kHz permet une numérisation du signal, compatible avec l'application d'un filtre numérique

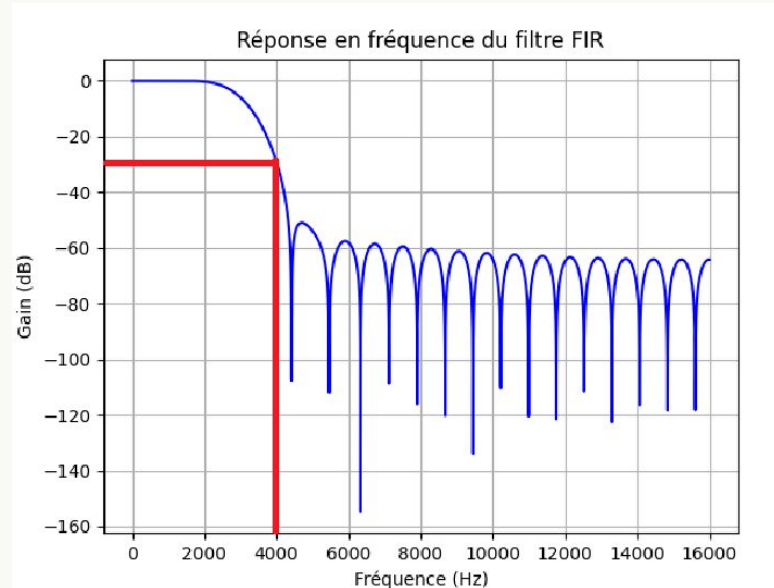


Signal de sortie du DAC

CONDITIONNEMENT

Slide 5/11

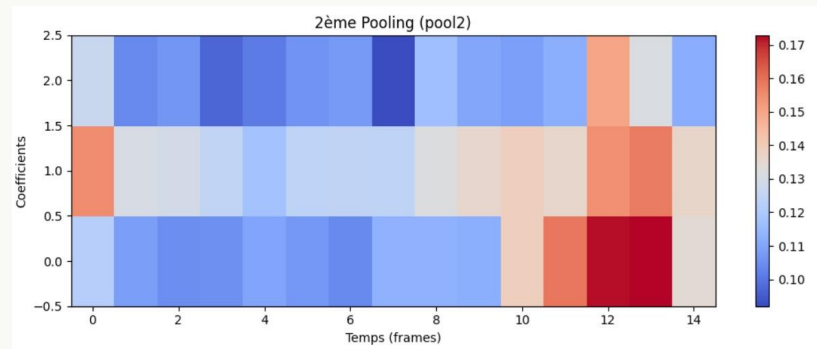
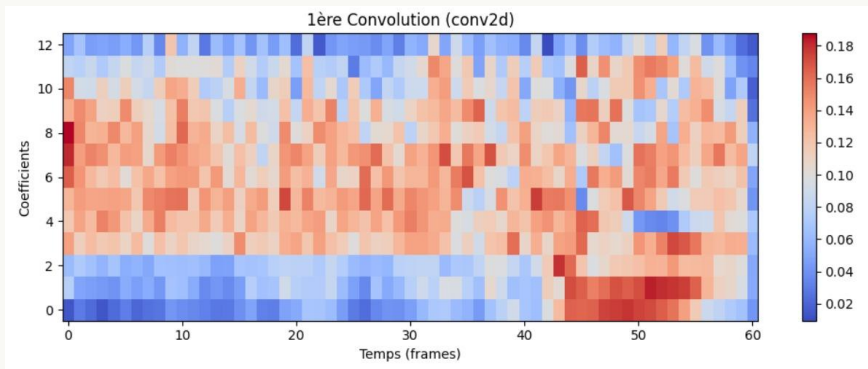
- Filtre passe-bas RIF conçu pour atténuer les fréquences au-delà de 4 kHz d'au moins 30 dB
- Échantillonnage initial à 32 kHz, réduit ensuite à 8 kHz pour limiter l'aliasing
- Traitement temps réel assuré avec un temps de calcul moyen de 21 μ s, inférieur à la période d'échantillonnage (31 μ s)



Représentation du Gain du filtre RIF en fonction de la fréquence

CARACTÉRISATION

Slide 6/11



Spectre du signal après la convolution

[electronique.wav](#)

Spectre des coefficients après le
deuxième Pooling


ENTRAÎNEMENT

Slide 7/11

- Calcul des coefficients MFCC sur Python
- Entraînement du réseau sur un jeu de données de 100 échantillons
- Vérification des coefficients sur 10 données test
- MSE voulue inférieure à 0.05

```
MSE test = 0.030003
```

```
--- Classification testdata détaillée ---
```

```
1/1  0s 89ms/step
```

```
idx= 0 pred=0.000508 → Rideau (attendu=0.0)
```

```
idx= 1 pred=0.000645 → Rideau (attendu=0.0)
```

```
idx= 2 pred=0.000165 → Rideau (attendu=0.0)
```

```
idx= 3 pred=0.001355 → Rideau (attendu=0.0)
```

```
idx= 4 pred=0.004000 → Rideau (attendu=0.0)
```

```
idx= 5 pred=0.999590 → Lampe (attendu=1.0)
```

```
idx= 6 pred=0.997327 → Lampe (attendu=1.0)
```

```
idx= 7 pred=0.997935 → Lampe (attendu=1.0)
```

```
idx= 8 pred=0.999448 → Lampe (attendu=1.0)
```

```
idx= 9 pred=0.998768 → Lampe (attendu=1.0)
```

*Résultats de la classification du modèle avec les prédictions
sur Python*

CLASSIFICATION

Slide 8/11

Classification des mots “Rideau” et “Lampe” :

- Modèle intégré à l’Arduino pour une classification en temps réel
- Prédictions réalisées sur les données audio reçues

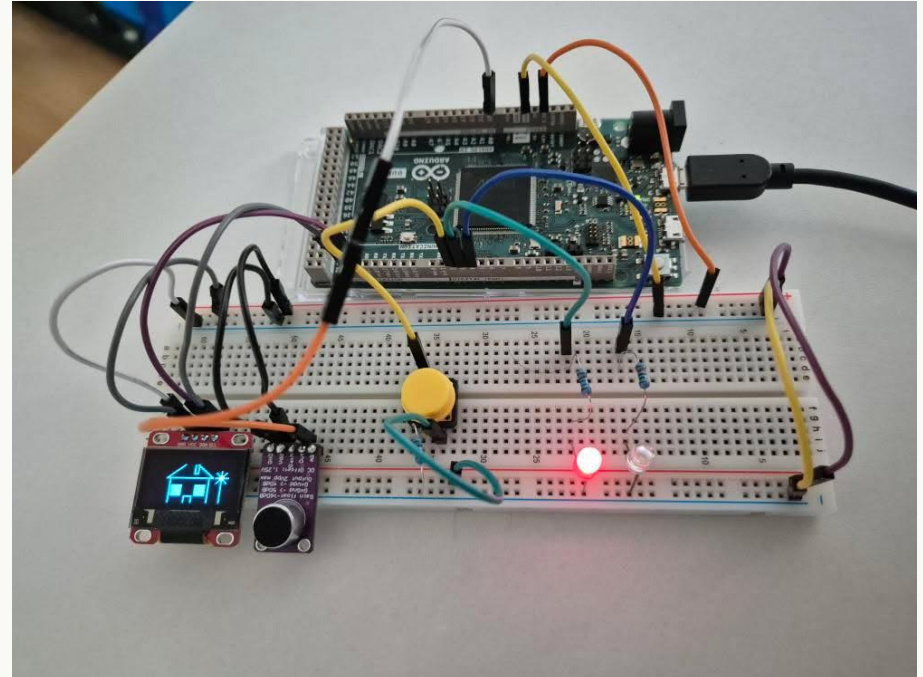
```
Sample 0 | Prédiction: 0.7985 | Attendu: 0.00 (Rideau) | MSE: 0.637610
Sample 1 | Prédiction: 0.8021 | Attendu: 0.00 (Rideau) | MSE: 0.643346
Sample 2 | Prédiction: 0.8095 | Attendu: 0.00 (Rideau) | MSE: 0.655236
Sample 3 | Prédiction: 0.8129 | Attendu: 0.00 (Rideau) | MSE: 0.660777
Sample 4 | Prédiction: 0.8053 | Attendu: 0.00 (Rideau) | MSE: 0.648527
Sample 5 | Prédiction: 0.8259 | Attendu: 1.00 (Lampe) | MSE: 0.030299
Sample 6 | Prédiction: 0.8079 | Attendu: 1.00 (Lampe) | MSE: 0.036912
Sample 7 | Prédiction: 0.8029 | Attendu: 1.00 (Lampe) | MSE: 0.038843
Sample 8 | Prédiction: 0.8032 | Attendu: 1.00 (Lampe) | MSE: 0.038723
Sample 9 | Prédiction: 0.7950 | Attendu: 1.00 (Lampe) | MSE: 0.042032
=====
MSE finale sur testData[] : 0.343230
```

*Résultats de la classification du modèle avec les prédictions
sur Arduino*

BILAN USE CASE

Slide 9/11

- Choix entre allumer la **Lampe** ou fermer les **Rideaux**
- Mot "Lampe" prononcé : Led jaune s'allume et animation sur l'écran OLED
- Mot "Rideaux" prononcé : Led rouge s'allume et animation sur l'écran OLED



FA (Performance/Précision)

Slide 10/11

Performance :

- Réduction de la taille de l'enregistrement qui fait baisser le temps d'exécution

Précision :

- Entrainement de l'ia avec Tensorflow
- Acquisition des poids et biais pour implémentation CNN

```
Appuyez sur le bouton pour commencer l'enregistrement...  
  
Enregistrement et extraction des MFCCs...  
Début de l'enregistrement...  
Enregistrement terminé !  
Extraction MFCCs terminée.  
MFCCs transférés au réseau neuronal  
Calcul de la prédiction...  
Lampe détectée !  
Temps d'exécution : 2366 ms  
En attente d'un nouvel enregistrement...
```

Étape d'exécution sur la console

BILAN

Slide 11/11

Merci de nous avoir écouté !

FP1	FP2	FP3	FP4	FP5	FP6
<ul style="list-style-type: none">- Échantillonnage via ADC à 32 kHz	<ul style="list-style-type: none">- Filtrage numérique- Down-sampling à 8 kHz- Temps de traitement < 31µs par échantillon	<ul style="list-style-type: none">- Une lecture sur Audacity claire du mot enregistré	<ul style="list-style-type: none">- Découpage en frames de 256 échantillons- Extraction de 13 MFCCs par frame	<ul style="list-style-type: none">- Entraînement d'un CNN- MSE < 0,05- Création de notre jeu d'entraînements	<ul style="list-style-type: none">- Implémentation sur Arduino pour la détection vocale et déclenchement des commandes