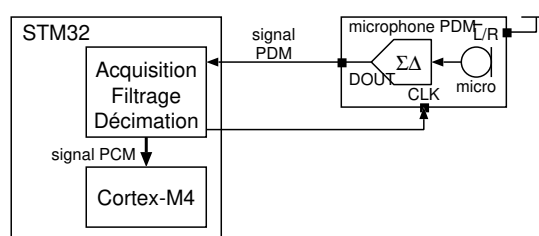
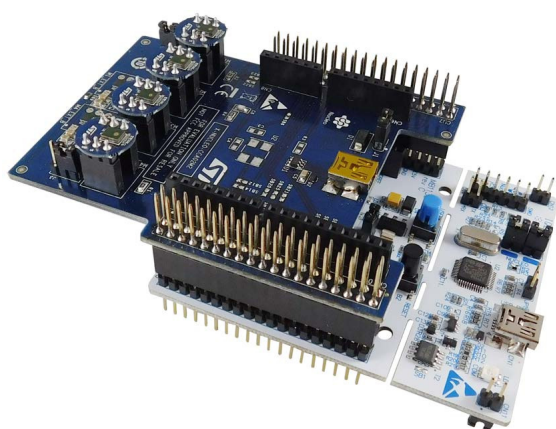
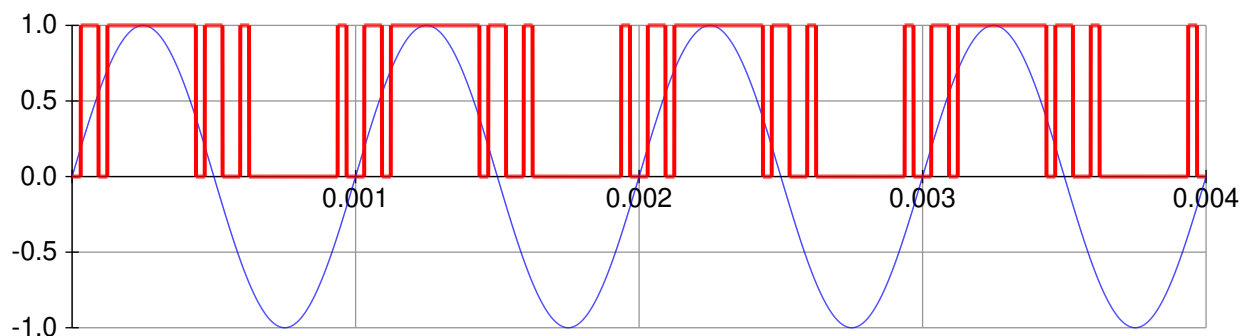


1. Introduction aux microphones numérique PDM	1
2. Aspect matériel : la carte X-NUCLEO-CCA02M2	2
2.1. Configuration I ² S, 2 microphones	3
2.2. Configuration I ² S, 4 microphones	4
2.3. Schéma et repérage des connexions	5
3. Aspect logiciel	6
3.1. Flux de données et buffers	6
3.2. Détails de la configuration matérielle	8

1. Introduction aux microphones numérique PDM



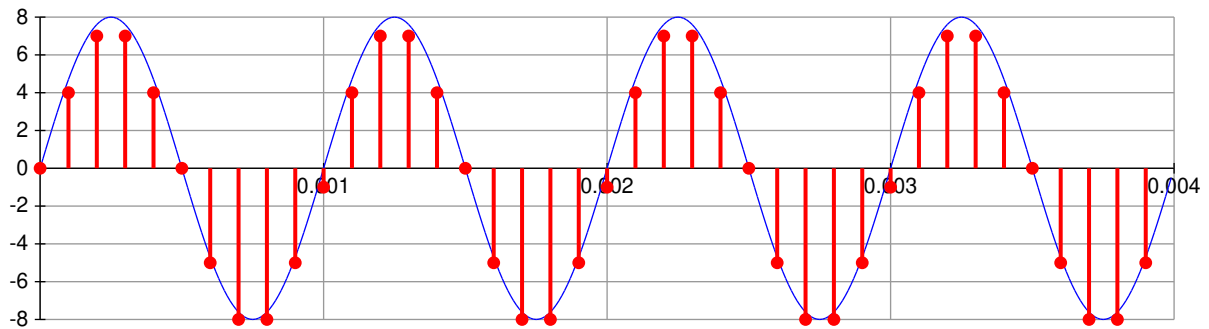
- PDM (Pulse Density Modulation) : format numérique dans lequel le signal issu du microphone est échantillonné sur 1 bit par un modulateur $\Sigma\Delta$. Le signal est en réalité suréchantillonné (fréquence d'échantillonnage entre 512 kHz et 4 MHz \gg 20 kHz pour la fréquence maximale du signal audio), ce qui permet d'avoir un excellent rapport signal sur bruit au niveau du signal final.



Dans ce format, c'est la densité relative des impulsions (le nombre de bits à 1 successifs par rapport au nombre de bit à 0 successifs) qui représente l'amplitude du signal.

Les microphones PDM fournissent des données numériques dans ce format bien adapté une transmission de type série.

- PCM (Pulse Coded Modulation) : format numérique usuel dans lequel le signal est représenté par ses échantillons encodé sur n bits.



Ce format permet de réaliser des traitements plus aisément que le format PDM (filtrage numérique, ...). Il est donc nécessaire de transcoder le signal PDM en signal PCM.

- La conversion est réalisée par le microcontrôleur. Plusieurs options existent :
 - Utilisation d'un périphérique spécialisé (DFSDM = Digital Filter for $\Sigma\Delta$ Modulator) : réalise la conversion PDM \rightarrow PCM matériellement.
 - Utilisation d'un périphérique de transfert série (SPI ou I2S) pour faire l'acquisition du signal PDM, puis filtrage et décimation logiciels. C'est la seule alternative sur le STM32F411 utilisé en labo.
- Le dispositif fournit la possibilité d'échantillonner les données de plusieurs micros simultanément.
- L'ensemble modulateur $\Sigma\Delta$ + filtrage et décimation réalise la fonctionnalité classique d'un ADC $\Sigma\Delta$.

2. Aspect matériel : la carte X-NUCLEO-CCA02M2

La carte X-NUCLEO-CCA02M2 permet d'envoyer un flux audio PDM vers le microcontrôleur de la carte NUCLEO associée, provenant soit des deux microphones PDM soudés sur la carte, soit de deux ou 4 microphones PDM que l'on peut ajouter sur les connecteurs prévus à cet effet.

Un connecteur USB permet au microcontrôleur de diffuser éventuellement le flux vers un PC, l'ensemble étant alors vu comme un microphone à plusieurs voies.

La connectique de la carte supporte différents modes de transfert et de traitement du flux PDM (I^2S / SPI, SAI, DFSDM). L'adaptation pour ces différentes solutions est réalisée par la modification des connexions SBx sur la carte (Solder Bridge). La liste complète des connexions possibles, fournie par la table 1 p. 5 du document [UM2631](#), est reprise ci-dessous.

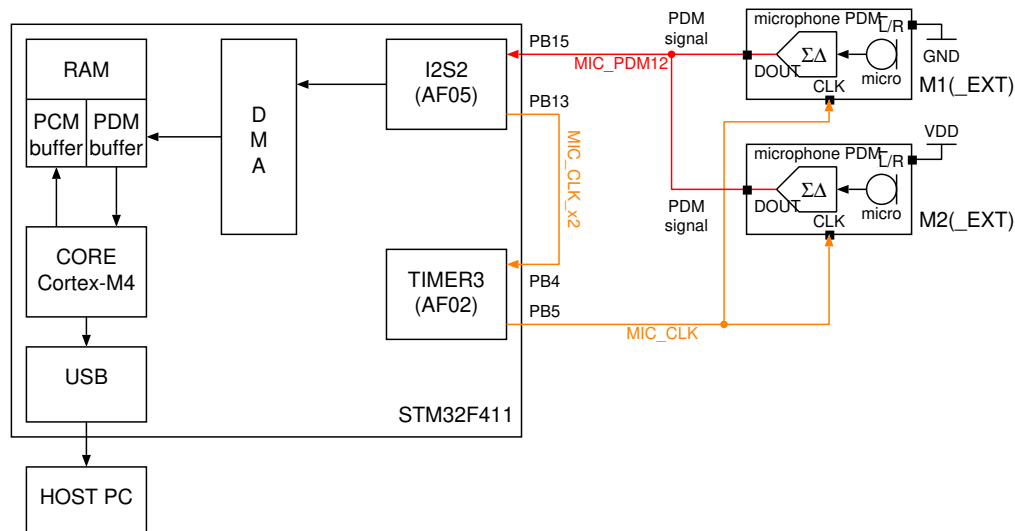
Solder bridge	Function
SB1	Connects USB DM pin from connector to MCU
SB2	Connects USB DP pin from connector to MCU
SB6	Routes on-board oscillator output to OSC_IN MCU pin
SB7	Connects MEMS clock to MCU timer output channel
SB8	Routes I2S clock to SPI clock
SB9	Merges on-board microphone PDMs to be acquired with one interface
SB10	Connects MIC34 PDM to MCU SPI
SB11	Connects MIC12 PDM to MCU I2S
SB12	Clock from the DFSDM peripheral
SB13	I2S clock from MCU
SB14	Connects I2S clock directly to MIC clock without passing through timer
SB15	Connects I2S clock to MCU timer input channel
SB16	Connects MIC12 PDM to MCU DFSDM
SB17	Connects MIC34 PDM to MCU DFSDM
SB24	Connects MIC34 PDM to MCU SAI
SB25	Connects MIC12 PDM to MCU SAI
SB26	Clock from the SAI peripheral

Les connexions configurées par défaut (■ pour l'USB, et ■ pour la gestion des micros PDM) permettent l'utilisation du mode I²S avec deux microphones (internes ou externes). On s'intéresse à ce mode en priorité, en considérant les périphériques utilisés sur un microcontrôleur STM32F411.

2.1. Configuration I²S, 2 microphones

- Organisation

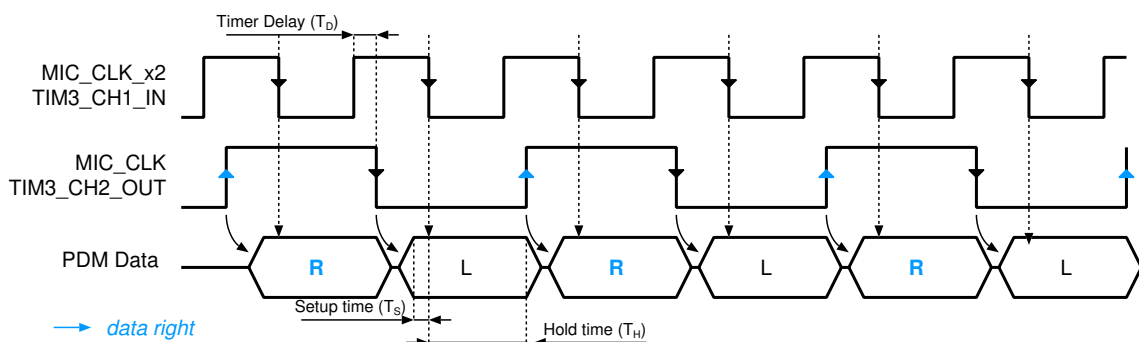
Le périphérique I2S fournit l'horloge MIC_CLK_x2 de fréquence double de celle de l'horloge d'échantillonnage des microphones MIC_CLK. Le Timer 3 réalise une division de fréquence par 2.



Les sorties DOUT des microphones sont reliées ensemble pour former le signal MIC_PDM12 composé pour chaque demi-période de MIC_CLK d'un échantillon 1 bit provenant alternativement d'un des microphones, puis de l'autre.

- Timing

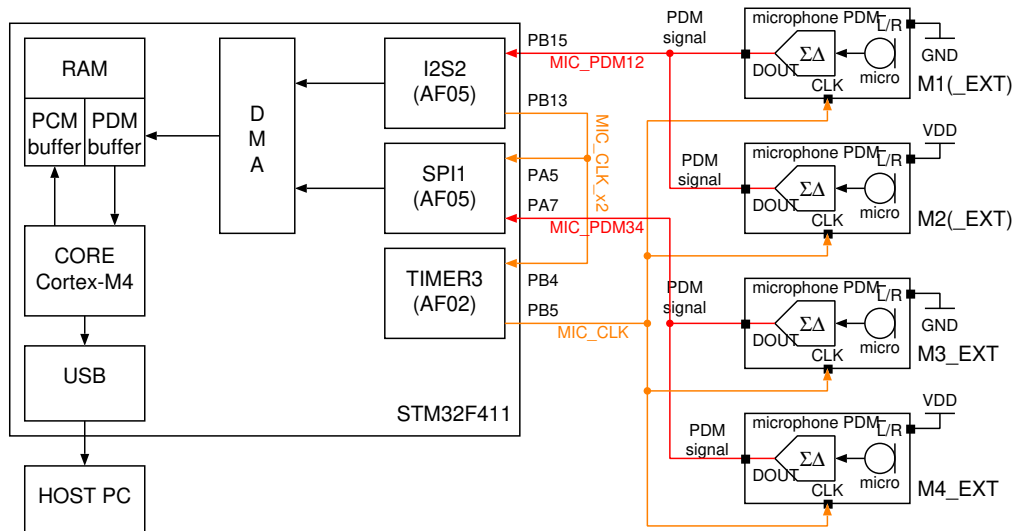
La division de fréquence par 2 réalisée par le Timer 3 introduit un décalage T_D par rapport à l'horloge d'entrée.



L'entrée \bar{L}/R de chaque microphone indique le front sur lequel le signal DOUT du microphone devient présent sur la ligne jusqu'au changement de front suivant : $\bar{L}/R = 0 \rightarrow$ front descendant, $\bar{L}/R = 1 \rightarrow$ front montant.

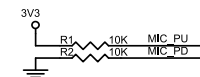
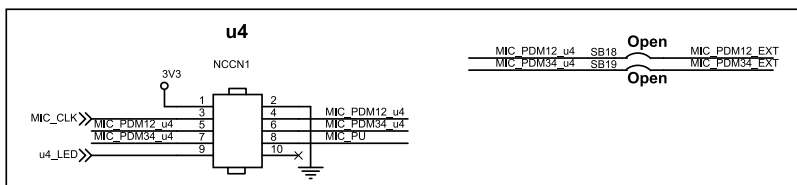
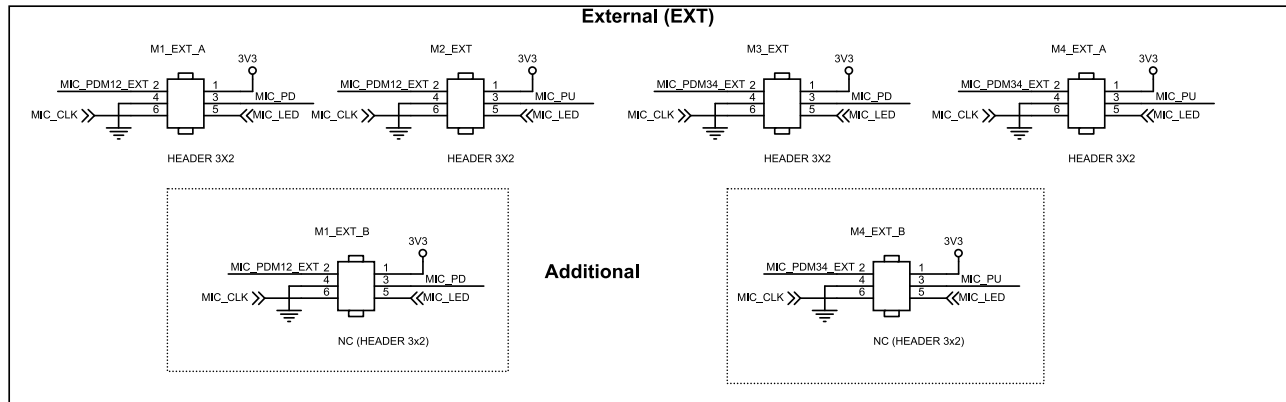
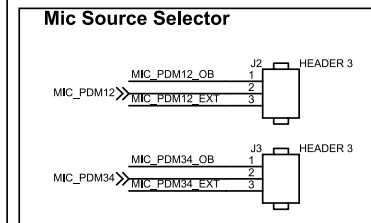
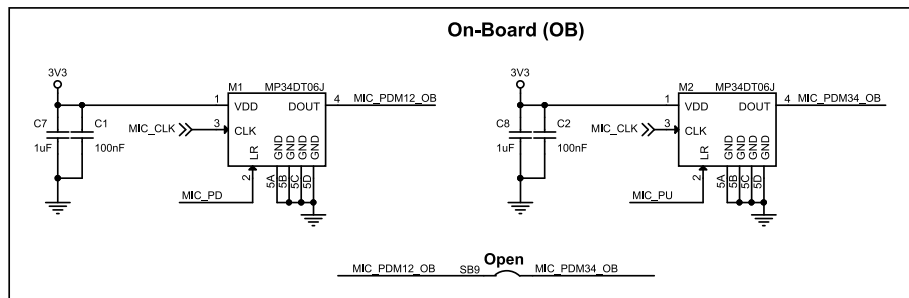
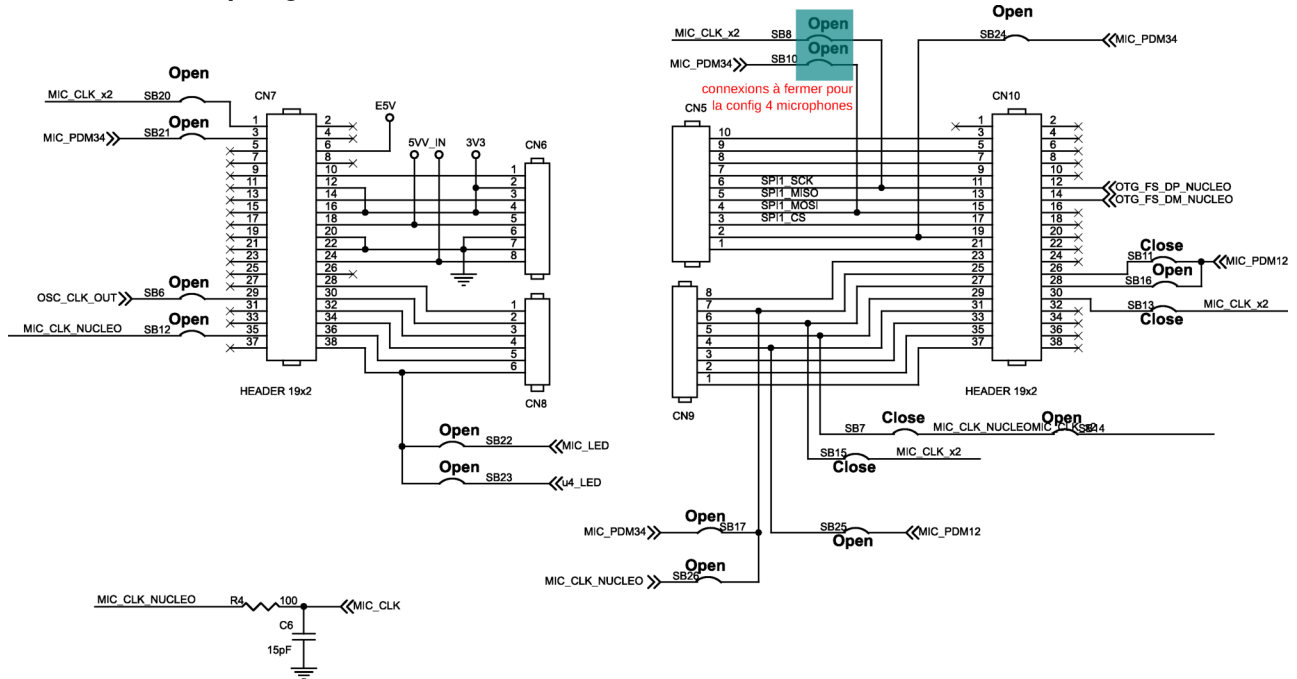
A chaque front de l'horloge MIC_CLK, un des microphones émet un 1 bit de donnée sur la ligne MIC_PDM12, l'autre ayant sa sortie en haute impédance. Le périphérique I2S, lui, fait l'acquisition sur les fronts descendants de l'horloge MIC_CLK_x2. Les données reçues par le périphérique I2S sont donc entrelacées (interleaved). Il sera nécessaire de les désentrelacer avant de réaliser la conversion PDM \rightarrow PCM.

2.2. Configuration I²S, 4 microphones

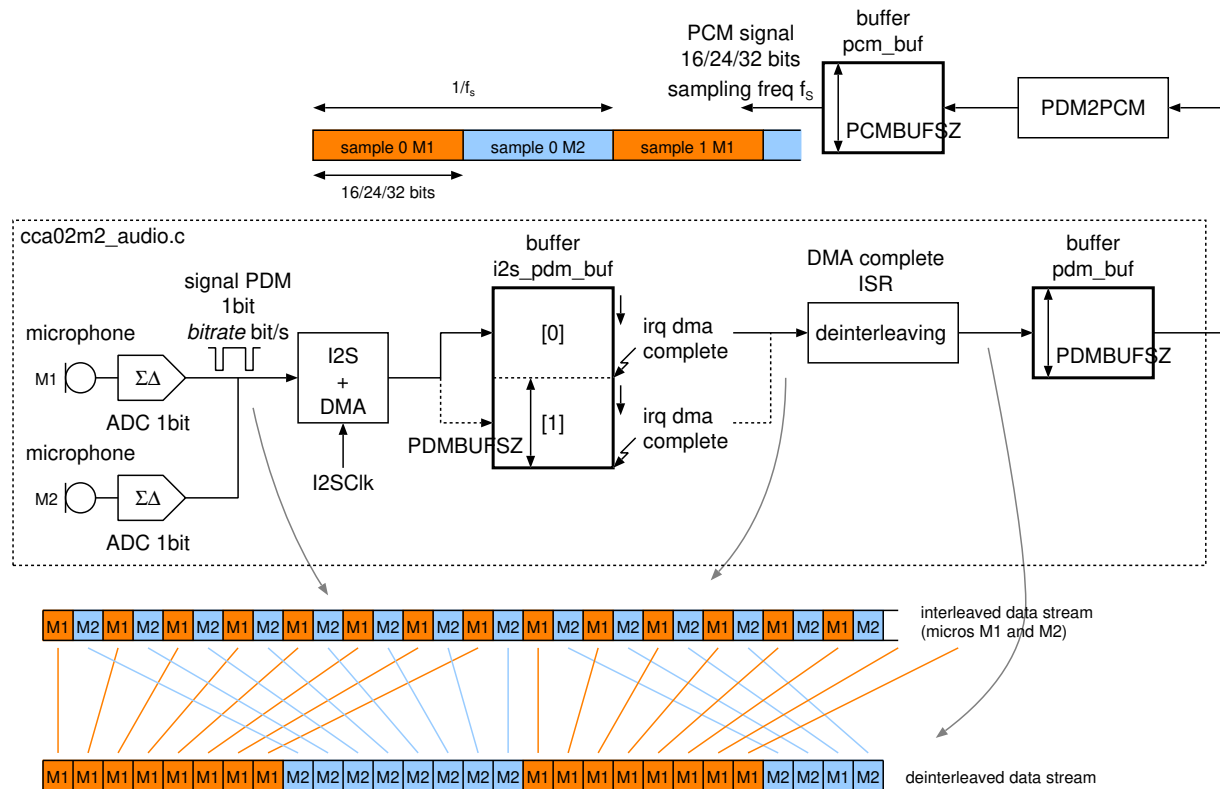


Même principe que précédemment pour les microphones M1(_EXT) et M2(_EXT). On utilise le périphérique SPI1, avec lequel l'horloge MC_CLK_x2 issue du périphérique I2S2 est partagée, pour acquérir les données des microphones M3_EXT et M4_EXT.

2.3. Schéma et repérage des connexions



Le flux de données, ainsi que les buffers nécessaires pour le stocker sont représentés sur la figure ci-dessous.



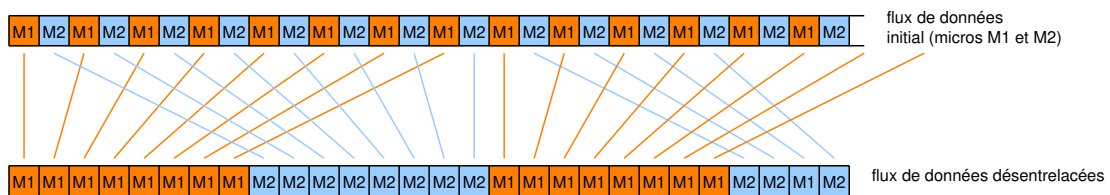
- Le périphérique I2S est configuré pour remplir au rythme de *bitrate* alternativement deux buffers (*i2s_pdm_buf[0]* et *i2s_pdm_buf[1]*) pendant une durée *NMS_PER_INTERRUPT* fixée à 1 ms par défaut. Le flux de données est un constitué alternativement d'1 bit provenant de chacun des microphones. La taille nécessaire au stockage des données sur chaque intervalle de temps est

$$\text{PDMBUFSZ} = \text{bitrate_kHz} / 8 \times \text{channel_nb} \times \text{NMS_PER_INTERRUPT}$$

La buffer sera déclaré de la manière suivante

```
uint8_t pdm_buf[PDMBUFSZ];
```

- Une fois un des buffers rempli, le périphérique DMA génère une demande d'interruption dont le traitement permet de désentrelacer les données et de remplir le buffer *pdm_buf* de taille *I2SPDMBUFSZ* qui contient alors les données correspondant à l'intervalle de temps *NMS_PER_INTERVAL*.



À l'issue de cette opération, un flag est positionnée par la callback passée en paramètre à l'initialisation. En parallèle avec cette opération, le périphérique I2S continue à remplir le second buffer par DMA avec les données des microphones, puis générera une IRQ quand le buffer sera complet ...

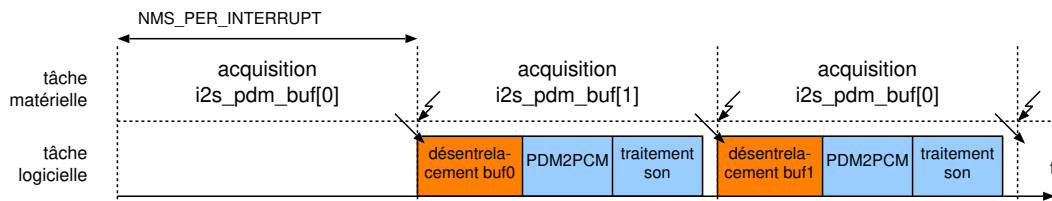
- Le flag permet de déclencher la conversion du buffer d'échantillon PDM organisés en octets en un signal PCM à fournissant les échantillons à la fréquence f_s encodés sur 16/24/32 bits selon le format sélectionné à l'initialisation. LA taille du buffer PCM doit être

$$\text{PCMBUFSZ} = \text{CHANNEL_NB} \times \text{SAMPLING_FREQ} / 1000 \times \text{N_MS_PER_INTERRUPT}$$

Le buffer sera déclaré en `uint16_t` pour une résolution 16 bits et en `uint32_t` pour une résolution 24 ou 32 bits.

```
uint16_t pcm_buf[CHANNEL_NB*SAMPLING_FREQ/1000*N_MS_PER_INTERRUPT];
```

Du point de vue exécution temporelle, on doit avoir le fonctionnement suivant :

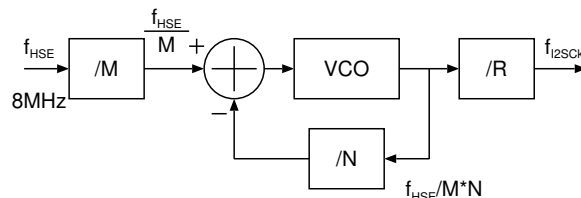


3.2. Détails de la configuration matérielle

• Configuration I2S – horloge

– Configuration de l'horloge I2S

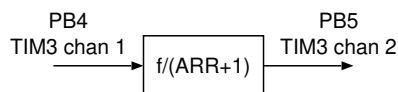
Le périphérique I2S reçoit une horloge spécifique fournie par une PLL (PLLI2S) alimentée par le même oscillateur principal (interne 16MHz ou **externe 8MHz**) que la PLL qui fournit l'horloge système. La PLLI2S est configurée de manière à fournir une fréquence d'horloge qui permettra d'obtenir exactement le *bitrate* voulu.



– Configuration du Timer 3 en diviseur de fréquence

Le Timer 3 n'est utilisé que dans le cas d'une configuration 2 ou 4 microphones. Le *bitrate* du périphérique I2S est alors doublé pour permettre à celui-ci d'échantillonner alternativement les signaux des deux microphones sur le front descendant de son horloge (voir le paragraphe Timing dans la section 2.1).

Le Timer 3 est simplement utilisé pour réaliser une division de la fréquence du signal arrivant sur PB4 (TIM3 channel 1).



Le signal avec la fréquence divisé est fourni sur la sortie PB5 (TIM3 channel 2).

- Configuration des broches : PB4 TIM3 channel 1 et PB5 TIM3 channel 2.
- Configuration de la broche channel 1 en entrée, et connexion à l'horloge (voir la partie External clock source mode 1 dans la section 13.3.3 Clock source selection du manuel du STM32F411) + configuration de la prise en compte du front montant dans le registre CCER.
- Configuration de la broche channel 2 en sortie (reg CCMR mode PWM1, reg CCER output enable, reg CCR2 = 1)
- Configuration de PSC=0, (ARR+1) permet de fixer le rapport de division.

• Configuration I2S+DMA

La configuration permet :

- de définir le bit rate à partir de l'horloge fournie par la PLLI2S,

audio freq [kHz]	bitrate [kbit/s]	oversampling	PLLI2S			I2S DIV	Commentaire
			M	N	R		
8	1280	160	4	96	5	30	cfg 1 micro
8	2560	160	4	96	5	15	cfg 2 micros
16	1280	80	4	96	5	30	cfg 1 micro
16	2560	80	4	96	5	15	cfg 2 micros
32	2048	64	5	192	3	50	cfg 1 micro
32	4096	64	5	192	3	25	cfg 2 micros
48	3072	64	4	192	5	25	cfg 1 micro
48	6144	64	4	192	5	10	cfg 2 micros

- de lier le périphérique au DMA1, Stream 3, Channel 0 qui permettra de copier chaque paquet de 16bits acquis dans la mémoire.

On utilise un double buffer circulaire permettant pour chacun d'eux de faire l'acquisition d'1ms de signal audio sur chacune des deux voies. Les données sont entrelacées (1 bit voie gauche, 1 bit voie droite). Lorsqu'un des buffers est complet, le DMA bascule automatiquement sur l'autre buffer et déclenche une interruption qui permettra de désentrelacer les données des deux microphones.

- **Configuration SPI+DMA** : cette configuration est nécessaire pour pouvoir utiliser 4 microphones. Cette configuration n'est actuellement pas supportée par le projet.