

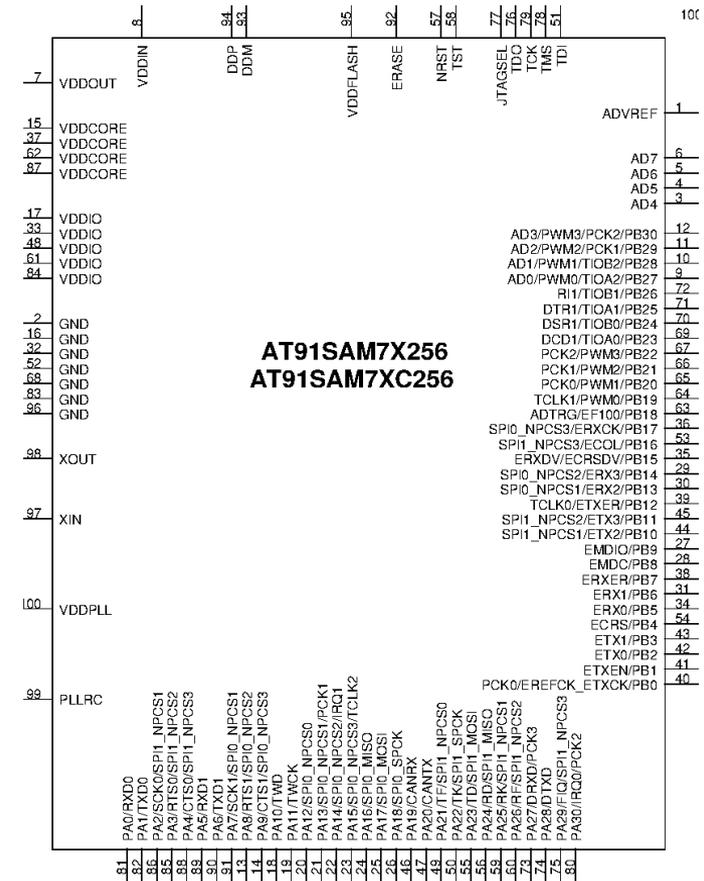
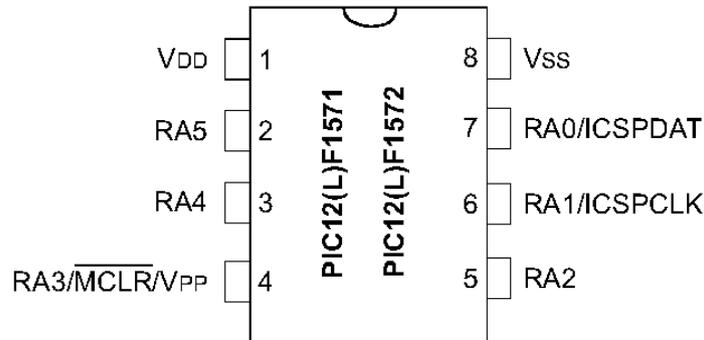
# Contrôle des I/Os

- Utilité d'un  $\mu$ C : piloter des composants externes
  - Langage de programmation = développement rapide
    - Réservé aux parties “organisation” ou “prise de décision”
    - Machines d'états complexes
    - ...
  - entrées / sorties simples et asynchrones :
    - Gérées par le programme
  - entrées / sorties complexes
    - Gérées par des modules spécifiques
    - => ces modules doivent pouvoir piloter les E/S

# Contrôle des I/Os

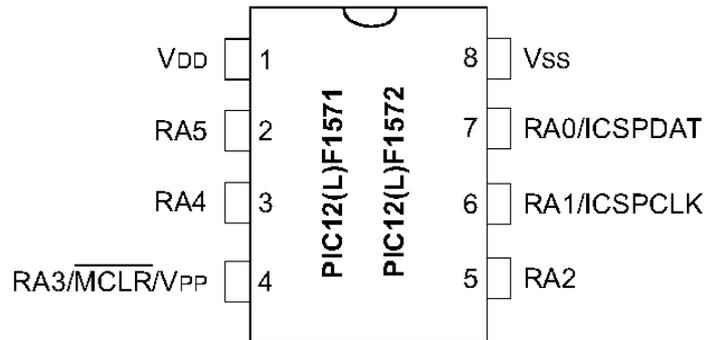
- nombre de fonctions disponibles supérieur au nombre d'I/Os

=> nécessité de multiplexage des broches



# Contrôle des I/Os

- Zoom sur le nom de broches



AD3/PWM3/PCK2/PB30	12
AD2/PWM2/PCK1/PB29	11
AD1/PWM1/TIOB2/PB28	10
AD0/PWM0/TIOA2/PB27	9
RI1/TIOB1/PB26	72
DTR1/TIOA1/PB25	71
DSR1/TIOB0/PB24	70
DCD1/TIOA0/PB23	69
PCK2/PWM3/PB22	67
PCK1/PWM2/PB21	66
PCK0/PWM1/PB20	65
TCLK1/PWM0/PB19	64
ADTRG/EF100/PB18	63
SPI0_NPCS3/ERXCK/PB17	36
SPI1_NPCS3/ECOL/PB16	53
ERXDV/ECRSDV/PB15	35
SPI0_NPCS2/ERX3/PB14	29
SPI0_NPCS1/ERX2/PB13	30
TCLK0/ETXER/PB12	39
SPI1_NPCS2/ETX3/PB11	45
SPI1_NPCS1/ETX2/PB10	44
EMDIO/PB9	27
EMDC/PB8	28
ERXER/PB7	38
ERX1/PB6	31
ERX0/PB5	34
ECRS/PB4	54
ETX1/PB3	43
ETX0/PB2	42
ETXEN/PB1	41
PCK0/EREFCK_ETXCK/PB0	40

# Contrôle des I/Os

- Stratégies de multiplexage:
  - n'importe quelle fonction peut être appliquée à n'importe quelle broche (rare)
  - Une fonction donnée n'est disponible que sur un nombre limité de broches
  - Certaines fonctions sont attribuées à 2 ou 3 broches maxi (gros  $\mu$ C, fonctions secondaires)
  - Quelques fonctions bien spécifiques sont attribuées à des broches fixes (fonctions principales / spécifiques)
- Configuration
  - Au niveau des ports d'E/S
  - On attribue une broche à une fonction ou une fonction à une broche
  - spécifique au fabricant / à la série

# Multiplexage sur PIC32

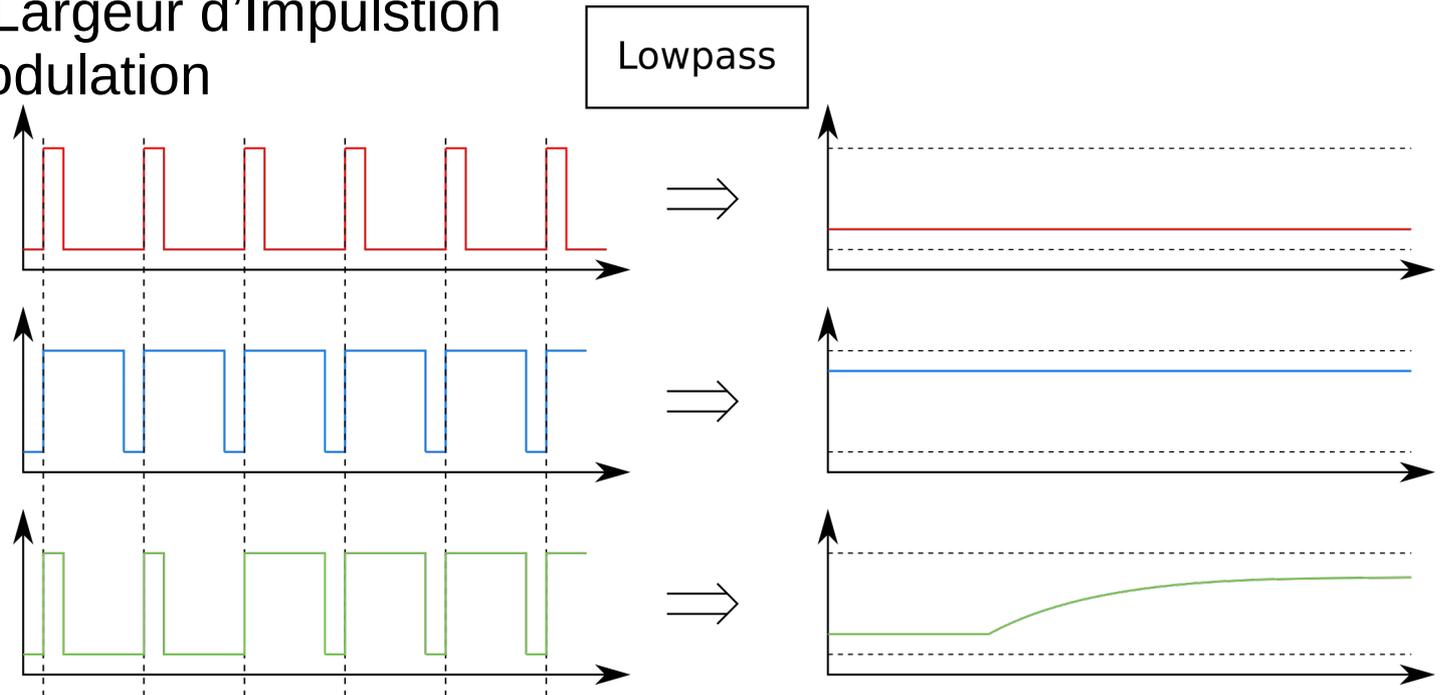
- Certaines fonctions sont multiplexées
  - Choix jusqu'à 1 broche parmi 16
- Les autres sont figées
  
- Broches en entrée :
  - 1 registre par fonction permet de choisir la broche à écouter
- Broches en sortie :
  - 1 registre par broche permet de choisir la fonction qui accède à la broche

# Contrôle simple des LEDs

- On va faire clignotter les LEDs sans utiliser le processeur
  - Le programme ne sert QUE pour la configuration
    - Ou à la modifier selon certains critères
- Utilisation du module PWM pour le clignotement

# MLI / PWM

- Modulation en Largeur d'Impulsion  
Pulse Width Modulation



- Sortie numérique oscillante + filtre passe bas =  
sortie analogique (faible qualité)

# MLI / PWM

- On garde la valeur moyenne, les résidus fréquentiels dépendent du filtre/fréquence de modulation
- En numérique le temps aussi est discret :  
relation entre  $F_{\text{modulation}}/F_{\text{système}}$  et la dynamique
  - Ex: quelle fréquence système pour avoir une sortie qualité audio ?
- ATTENTION :  $F_m \neq F_s$

# MLI / PWM

- En techno FPGA : pas de souci (m'enfin, pas trop)
- En C : très difficile (bit banging)
  - Nécessité d'une base de temps fiable (timer)
  - Temps de calcul non maîtrisé =>  $F_m \ll$  temps de calcul
- En  $\mu$ C : utilisation d'un périphérique dédié
  - La difficulté devient l'usage du module

# MLI / PWM

- Techno *lowcost* :
  - Timer classique avec option comparateur
  - La valeur mise dans le comparateur définit la valeur moyenne de la sortie (qui deviendra une valeur analogique après le filtre passe-bas)
- Le timer évolue normalement, éventuellement ralentit par son *prescaler*
- Si  $\text{timer} < \text{comparateur} \Rightarrow \text{sortie} = 1$   
sinon  $\Rightarrow \text{sortie} = 0$
- Inconvénients :
  - le timer n'est pas utilisable pour une autre application
  - Fm est directement liée au prescaler et à la valeur max du timer

# MLI / PWM

- Techno *highperf* : Toutes les ressources sont dédiées
  - Le principe reste basé sur un timer + comparateur, mais pas/peu de mutualisation
- Horloge de base entièrement configurable
  - Diviseur configurable + prediviseur éventuel
  - Valeur max du compteur au choix
  - $F_m = \text{horloge de base} / \text{valeur max}$
- Possibilité de définir des polarités / déphasages / ...
- Inconvénients :
  - Chaque valeur à configurer complexifie le module, et sa compréhension
  - Ben le prix accessoirement !

# MLI / PWM sur PIC32

- Module comparateur associé à un timer
  - Timer2 ou Timer3 (pas le choix)
  - 5 modules comparateur (5 sorties PWM max)
- Plusieurs modes de fonctionnement
  - Statique
  - échantillonné