



# Conception d'une carte à microcontrôleur ARM pour le robot Amphibot II

Alain Dysli

20 Juin 2005

Assistant: A. Crespi

Professeur: A.J. Ijspeert

# Plan de la présentation

- Introduction – Pourquoi une nouvelle carte pour Amphibot II
- Cahier des charges
- Microcontrôleur choisi
- Hardware – Architecture de la carte
- Implementation software et hardware des fonctions
- Améliorations
- Conclusion

# Introduction

Pourquoi avoir besoin d'une nouvelle carte pour Amphibot II?

- Le PIC n'est pas assez puissant pour en même temps contrôler les moteurs, communiquer avec des capteurs et communiquer avec d'autres modules
- Pour donner plus d'autonomie à Amphibot II

Il faut un microcontrôleur plus puissant capable de communiquer rapidement avec tous les modules!

# Cahier des charges

Concevoir une carte à microcontrôleur avec les spécificités suivantes:

- $\mu$ c plus rapide que le PIC
- Plus de mémoire que le PIC
- Bus CAN pour une communication haut débit entre les modules
- Bus I2C
- I/O utilisable pour modules externes
- Taille minimale
- Compatible avec les modules Amphibot II

# Choix du microcontrôleur

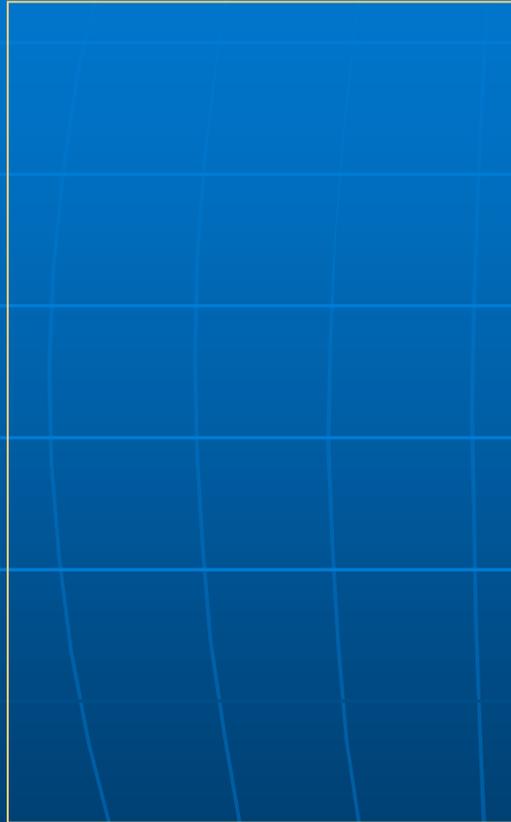
- Microcontrôleur Philips ARM
- Fréquence de 60 MHz
- 256 kB Flash
- 16 kB RAM
- CAN & I2C en hardware
- Convertisseur A/D
- Taille relativement faible



# Hardware - Éléments

Battery

Motor



SALAMPower



SALAMCTRL



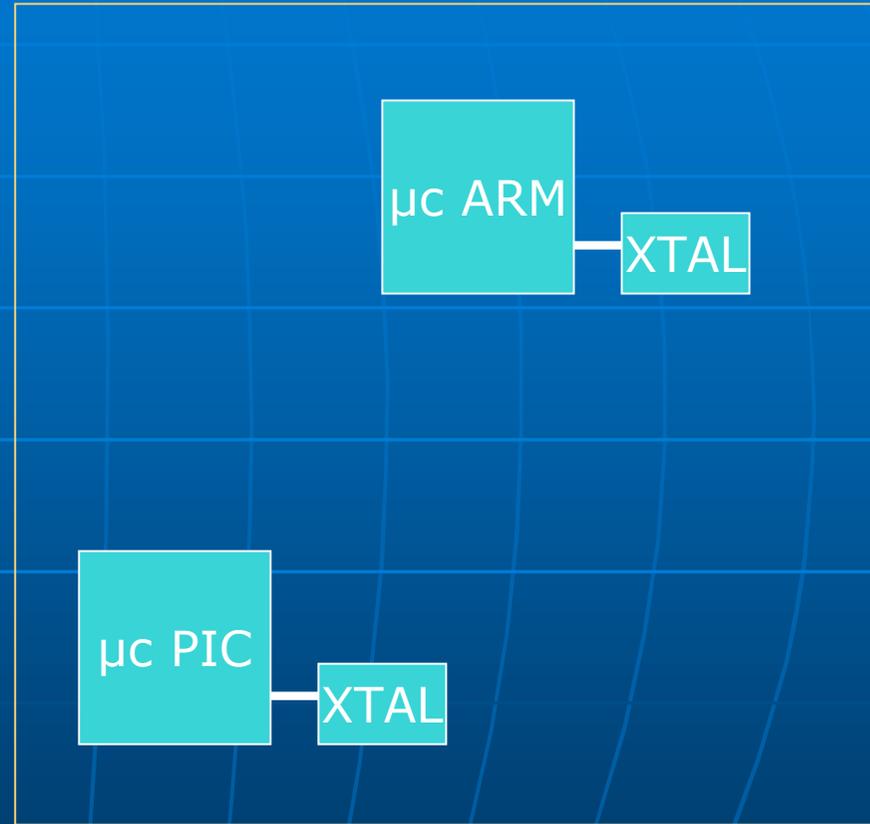
# Hardware - Microcontrôleurs

Battery

Motor

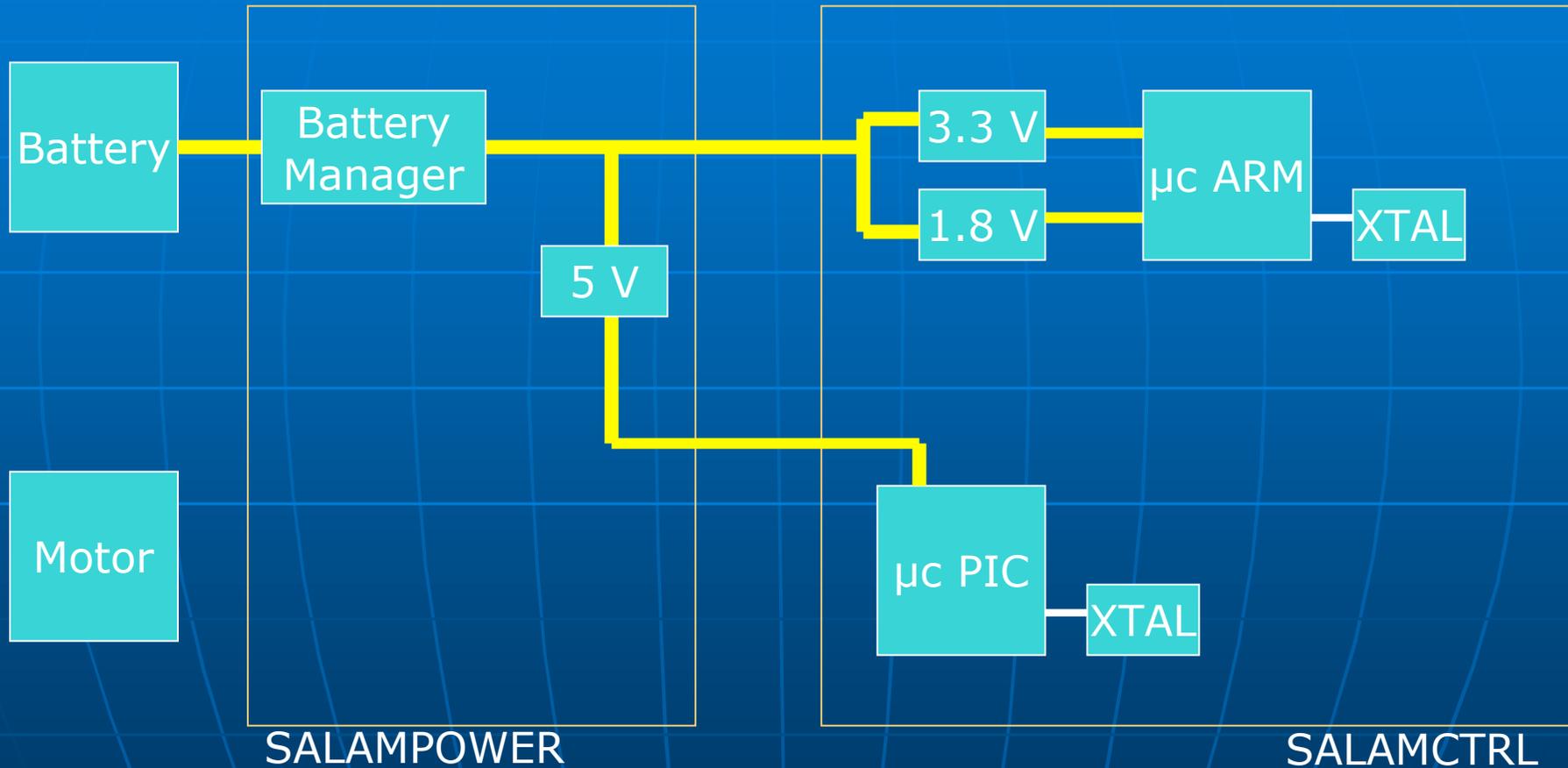


SALAMPower

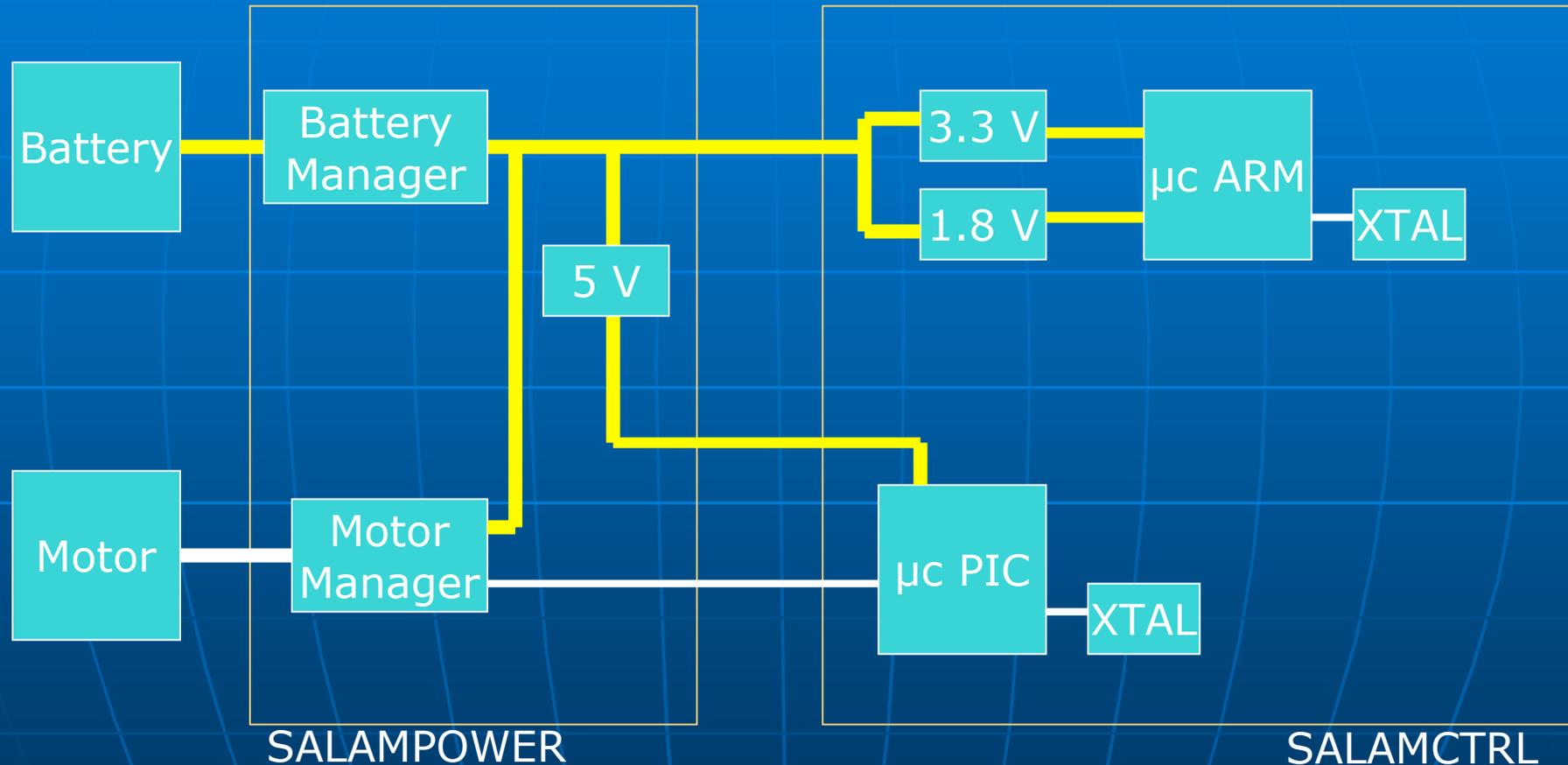


SALAMCTRL

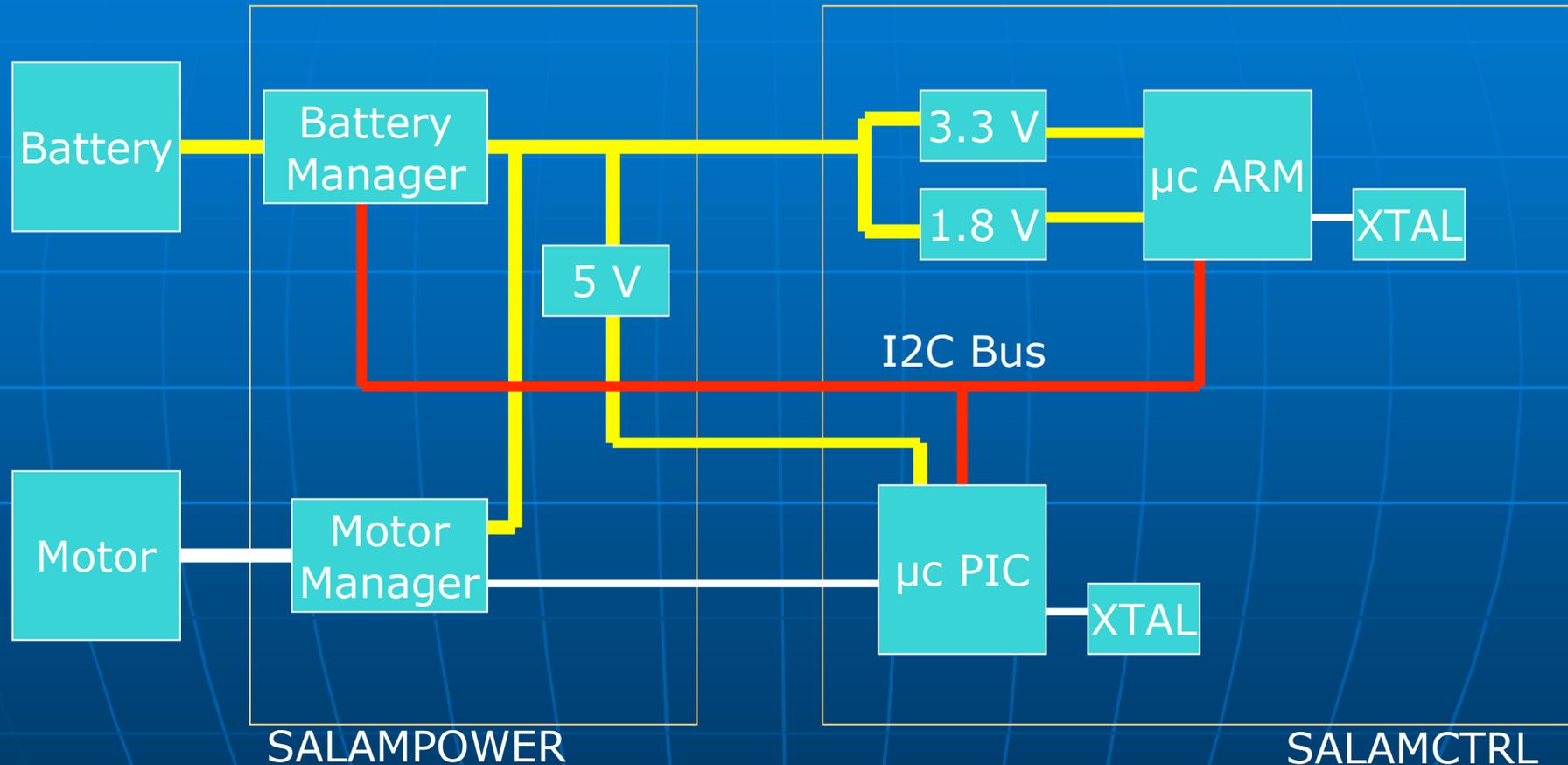
# Hardware - Énergie



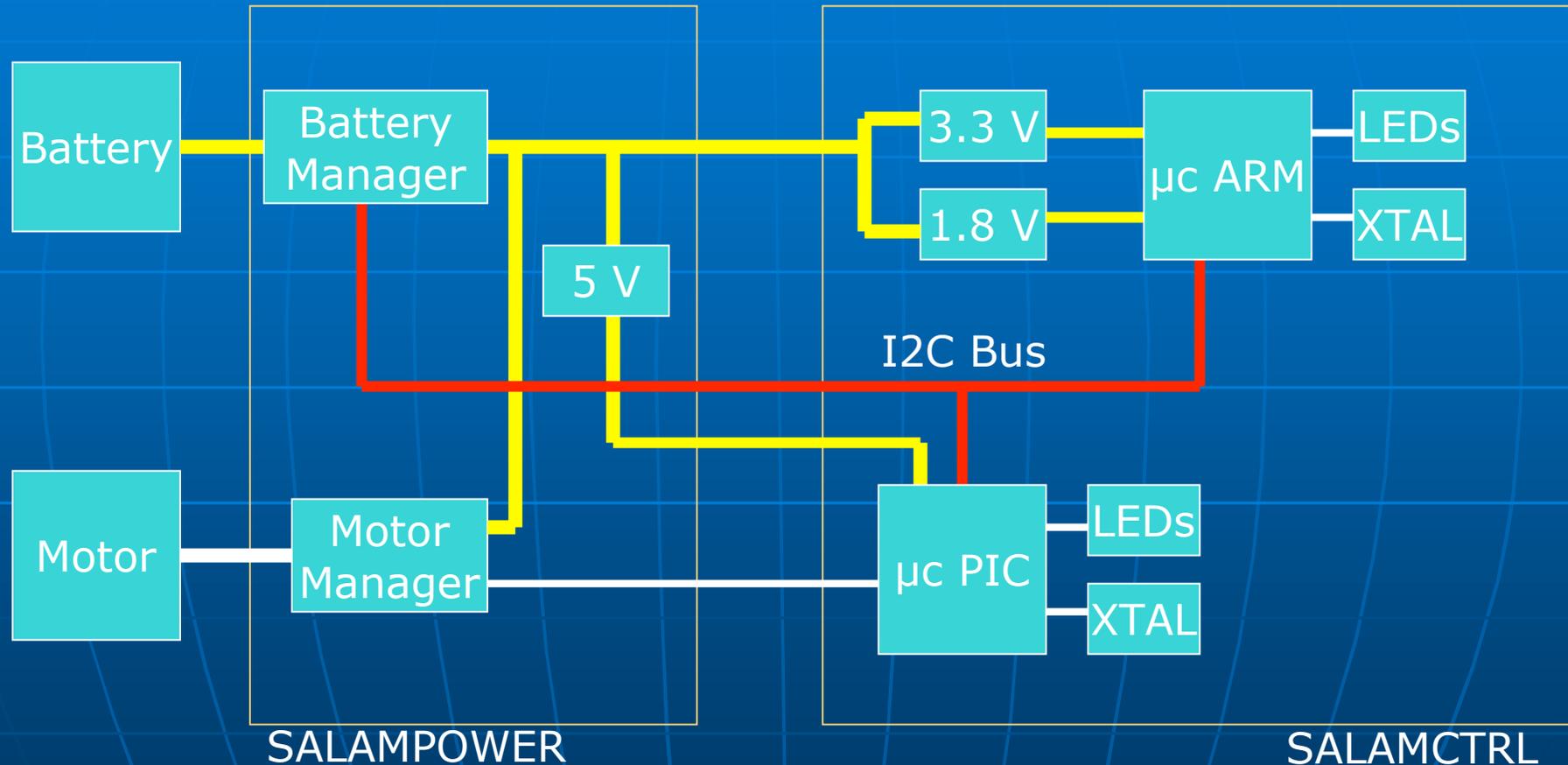
# Hardware – Contrôle du moteur



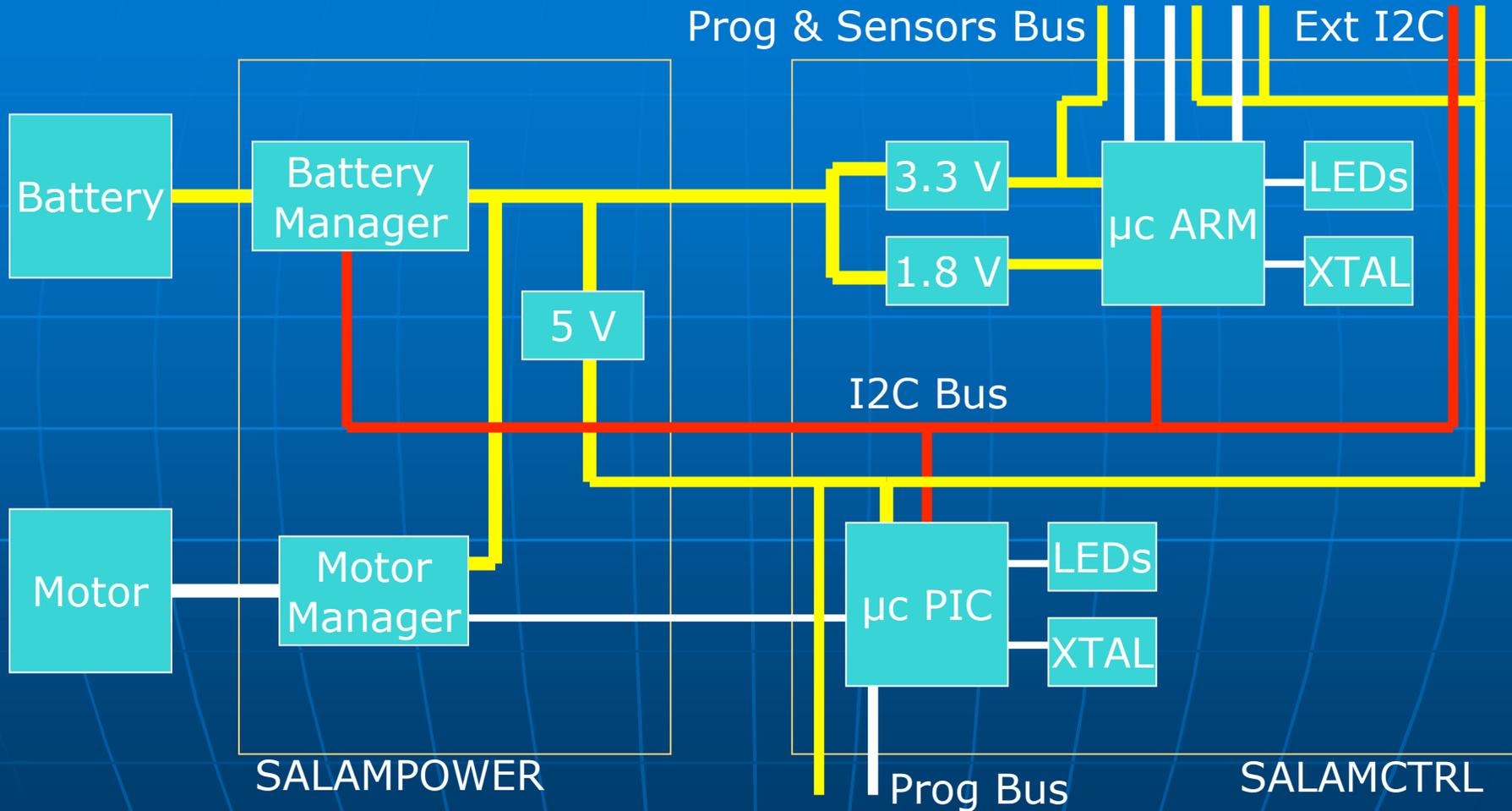
# Hardware – Moteur & Batterie



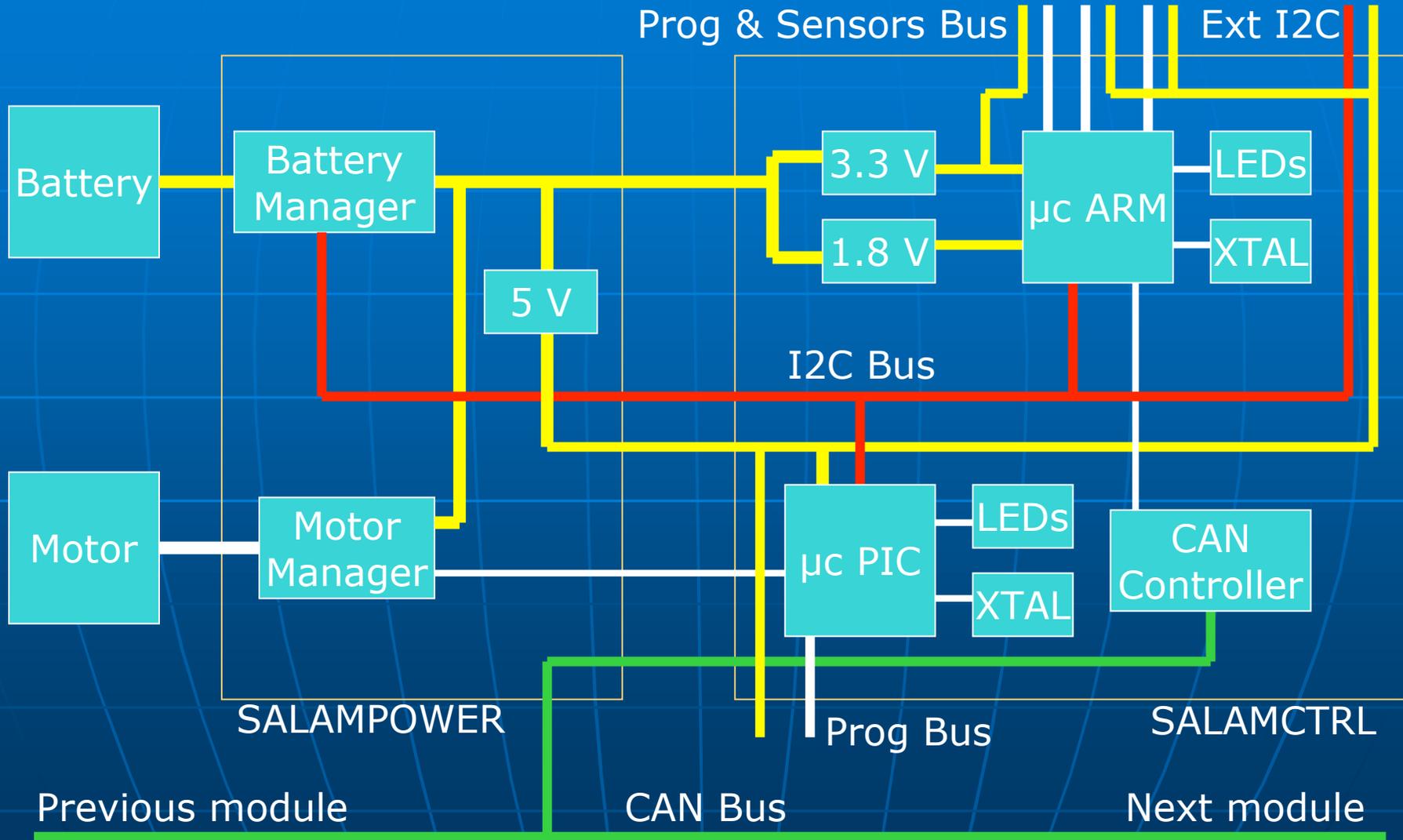
# Hardware - Communication



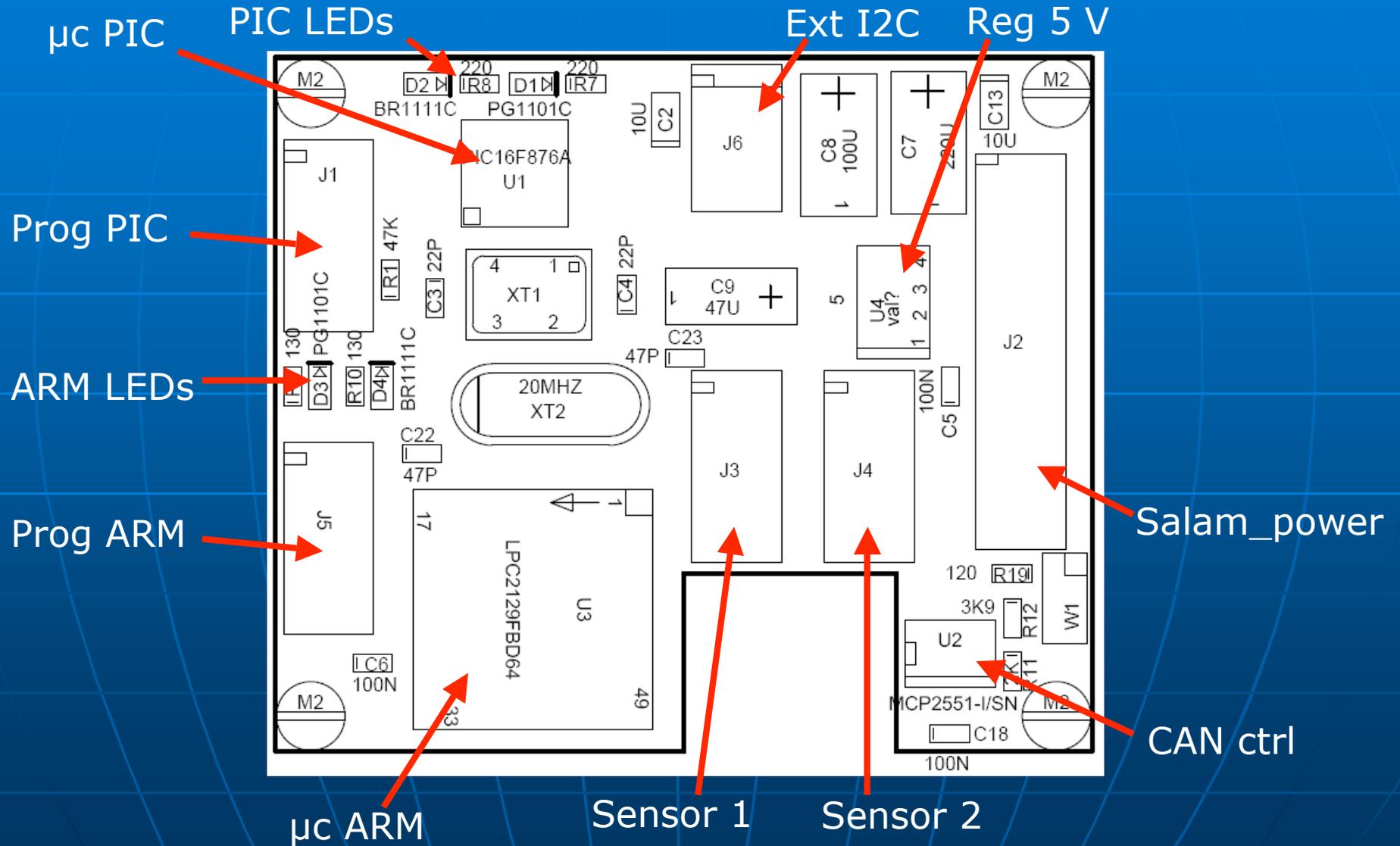
# Hardware - Communication



# Hardware - Communication



# Hardware



# Systeme - Software

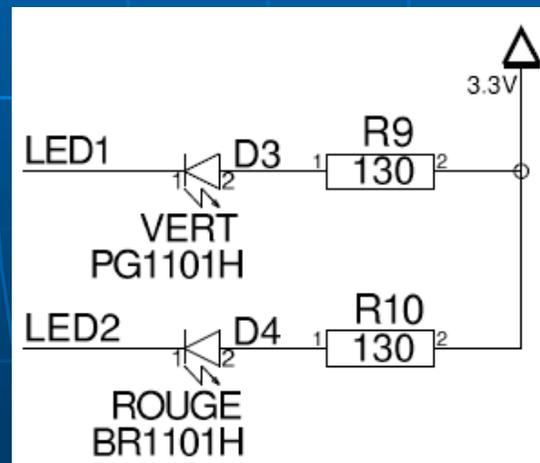
Set complet de routines pour:

- Initialiser le microcontrôleur pour qu'il tourne à 60 MHz
- Initialiser l'horloge des périphériques internes à 60 MHz
- Enclencher le Memory Accelerator Module

# LEDs - Hardware

- Affichage à 2 LEDs rouge et verte

Utile pour déboguer!



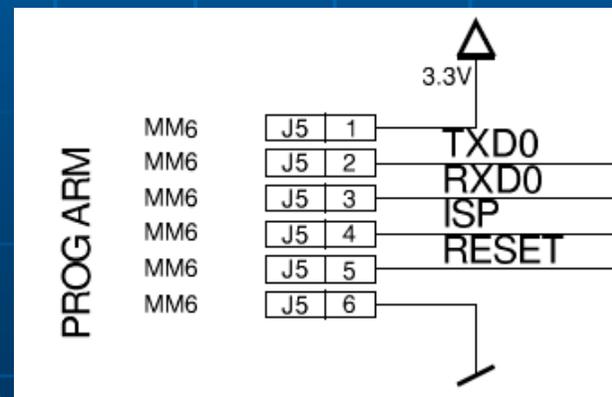
# LEDs - Software

Set complet de routines pour:

- Initialiser l'accès aux LEDs
- Allumer les LEDs
- Eteindre les LEDs
- Faire clignoter les LEDs

# Programmation - Hardware

- Port de programmation du PIC identique à la version précédente
- Programmation du ARM grâce à un programmeur brancher sur port série
- Le programmeur permet également de communiquer par UART



# Programmation - Software

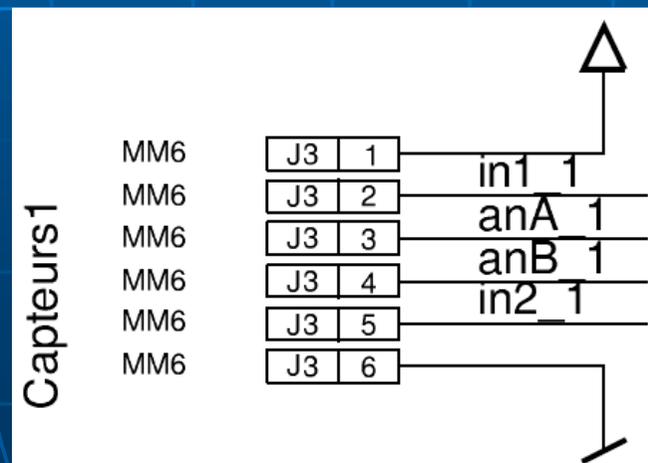
Set complet de routines pour:

- Initialiser la communication UART
- Envoyer des chaînes de caractères par UART
- Envoyer des nombres par UART
- Recevoir des caractères par UART

La programmation se fait avec un logiciel tiers

# Capteurs - Hardware

- Possibilité de brancher deux capteurs ou modules
- Ports multifonctions avec alimentation et 4 entrées sorties digitales dont 2 entrées analogiques possibles
- Chaque pin est configurable



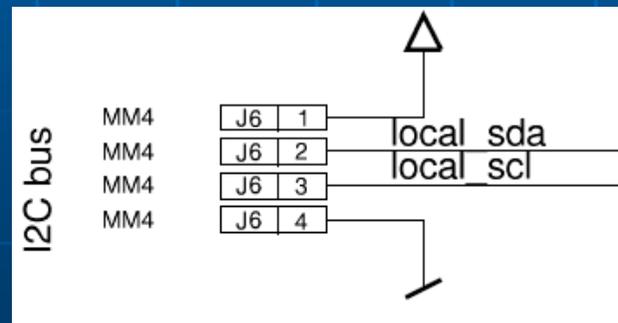
# Capteurs - Software

Set complet de routines pour:

- Initialiser les fonctions
- Lire la valeur d'un pin
- Changer l'état d'un pin
- Acquérir une tension analogique

# I2C – Hardware

- Communication possible entre le gestionnaire de batterie, le PIC, le ARM et des périphériques externes
- Le ARM est maître. Les autres périphériques sont esclaves



# I2C - Software

Set complet de routines pour:

- Initialiser l'I2C
- Écrire une valeur dans un registre d'un module
- Lire un registre d'un module

# CAN – Hardware

- Le bus CAN permet une communication entre les microcontrôleurs ARM de chaque modules
- Le bus n'est pas maître-esclave

# CAN - Software

Set complet de routines pour:

- Initialiser le bus CAN
- Envoyer des données avec un identifiant à choix
- Lire des données reçues
- Filtrer les messages reçus selon les identifiants

# Améliorations

- Quelques bogues hardware mineurs
- Problèmes software avec le bus I2C
- Problèmes software avec le bus CAN

# Conclusion

- Le microcontrôleur ARM dispose d'une grande puissance de calcul
- Le bus CAN permet une communication rapide entre les modules
- L'architecture est extensible grâce aux ports pour capteurs et à l'I2C
- Les fonctions de base pour manipuler les I/O sont implantées
- Possibilité de brancher des périphériques haut débit (camera,...)
- Amphibot II peut maintenant être totalement autonome
- Quelques problèmes doivent encore être régler...