Main LPC

Commande embarquée

Plan du logiciel

- Tempo précise
- Gestion des positions
- Gestion des mouvements sans collisions
- Gestions de moteurs
- Interpréteur de commandes sérielles

Caractéristiques du soft embarqué

Codage

- Code Arduino
- C++ et classes
- pas de String
- 3 fichiers .INO et 3 inclusions .H
  - Main_lpc.ino
  - Move_key.ino
  - Tempo.ino

Qualités

- Robuste
- Modulaire
- Rapide
- Simple
- Interfaçable
- Compatible «terminal»

Boucle principale - Tempo

- Pulsation de 100 ms
- Faire clignoter la LED13

```
void loop()
{
  unsigned long currentMillis = millis();  // t écoulé
  if (currentMillis - previousMillis > cycleTime)
  {
    previousMillis = currentMillis;
    pulse_01Sec();
    etat_LED13 = !etat_LED13;
    digitalWrite(LED13, etat_LED13);
  }

  interpreter();

}//loop
```

Position de servo

- Les servo sont piloté par angle
- Valeur 0 à180
- Souvent :
  - O : ouvert
  - F : fermé
- Bras: deux valeurs
  - A: angle
  - H: hauteur

Dépend évidemment de la l'orientation du moteur!

```
#define POUCE_O (0)
#define POUCE_F (160)  // max - 20%
#define POUCE_D (75)
#define IDX_O (160)
#define IDX_F (0)
...
// définitions des angles - bras
#define A_COU  15  // a
#define H_COU  180
```

Positions - classe «Etats»

```
class Etats
{
  public:
  // positions: mémorise la dernière valeur appliquée (0..180) aux servos
  int p_pouce, p_idx_v, p_index, p_majeur, p_annulaire, p_ecart, p_hauteur;

  void init();      // main en position de repos
  void read_pos();  // relire les positions appliquées aux servos
  void pose_clef_d(int _pouce, int _idx_v, int _index, int _majeur, int _annulaire);

};
```

Activité et servo

- Connectique simple: 5V, pulse, 0V
- La position est donnée par la largeur de pulse
- Ce n'est pas linéaire, ni précis
- On n'a pas de feed back de la position
- Le moteur a une électronique intégrée
- Si la pulse disparait, 2 comportements possibles:
  - Le moteur ne tient plus la position
  - Le moteur reste où il est et tient la position
- En cas d'échauffement, le moteur se coupe

  Ex. moteur forcé pour 2 doigts

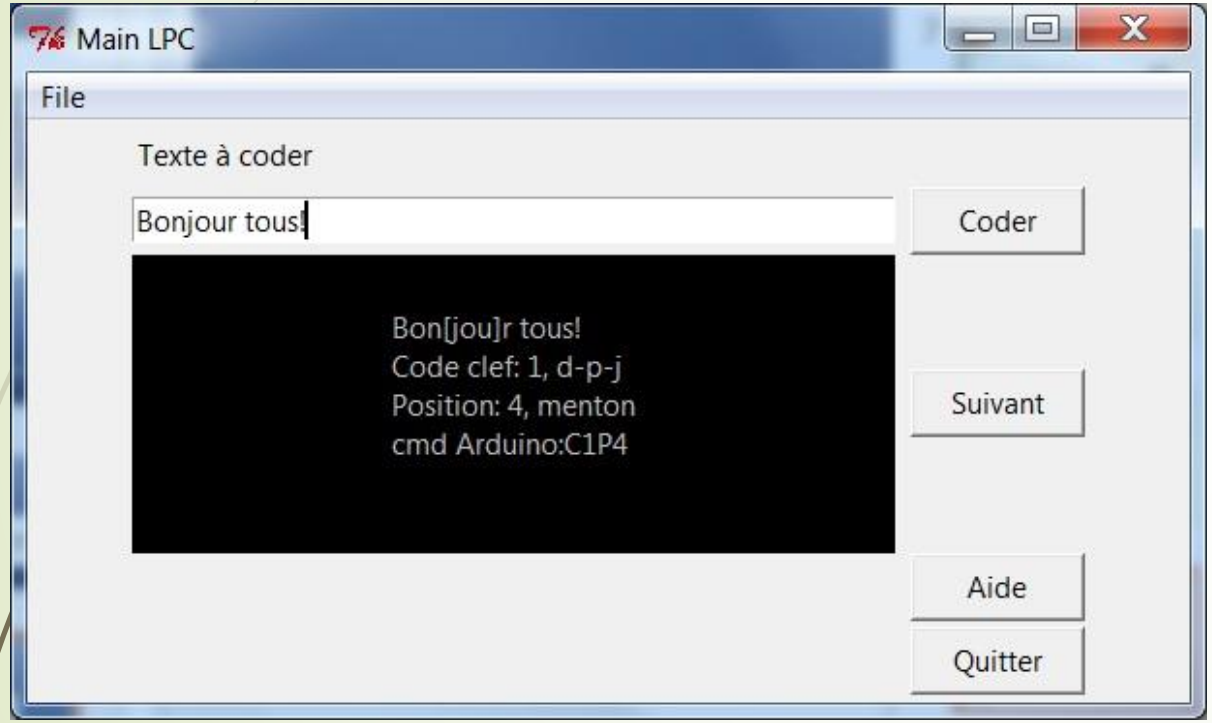
Gestion des mouvements

- Bouger seulement si un changement est détecté
  - Sinon, les moteurs sont activés sans mouvements
- D'abord les doigts qui ouvrent
- Ensuite le pouce si ouverture
- Si fermeture, ordre inverse: pouce, puis doigts
- La tempo assure la transition
- Après 2 s, on coupe l'impulsion (plus de bruit du maintien)

Interpréteur

- Commandes simples
  - Position -> P<n°>
  - Configuration -> C<n°>
- Peuvent être consécutives, dans n'importe quel ordre
- En cas d'erreur: pas d'action, affiche l'erreur
- Tempo des mvmts consécutifs modifiable: 100 – 2000 ms

Interpréteur Python



- Lit du texte
- Le simplifie
- En extrait les positions
- Les envoie à la main LPC

C’est fini. Questions?

Et démonstration, bien sûr.