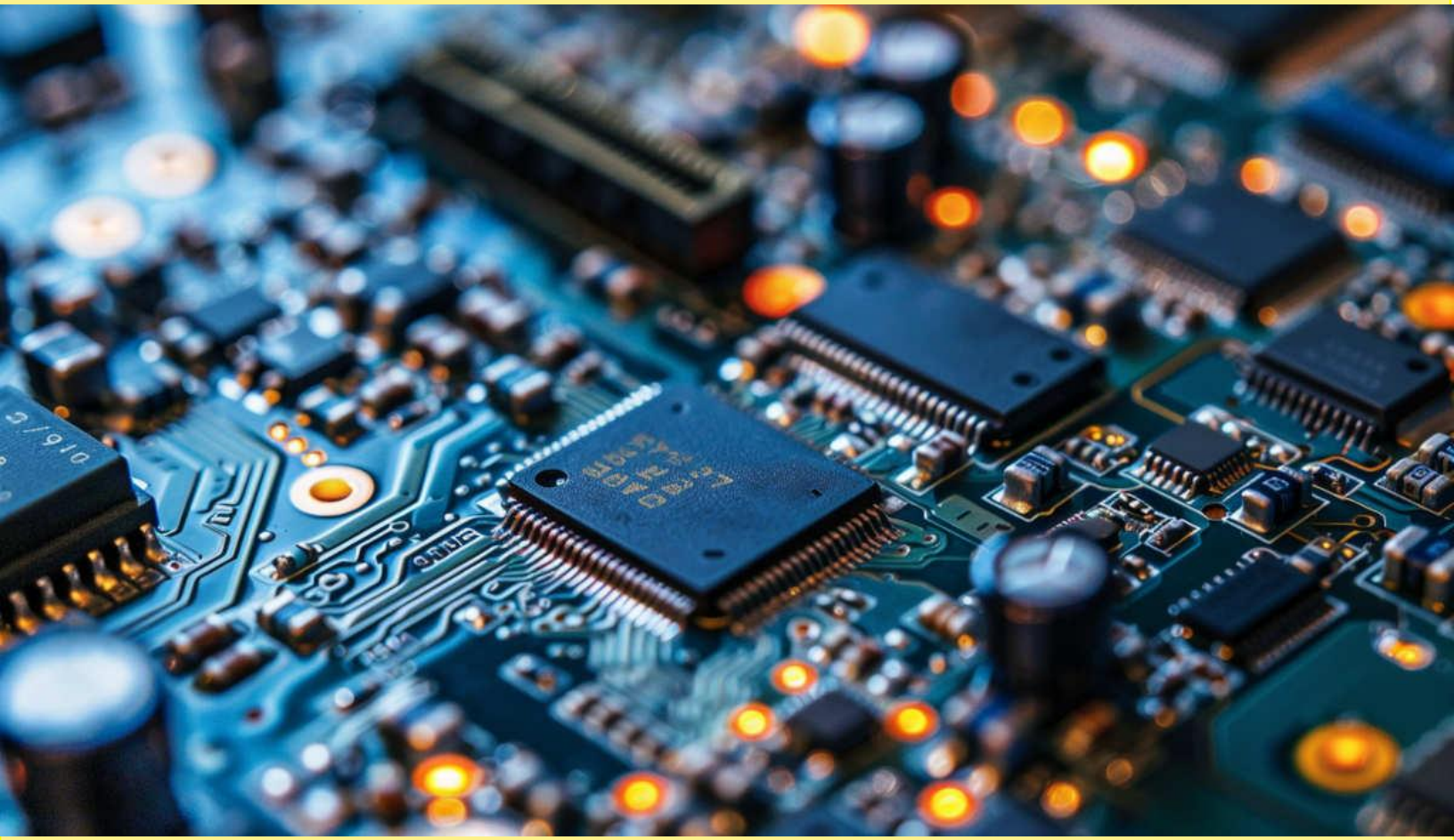
CPI A2 – PROJET SYSTEMES EMBARQUÉS

PROSIT 3 – AUTO COMPILATION



CESI
ÉCOLE D'INGÉNIEURS

Contexte :

Notre société produisant la fameuse station météo destinée à être embarquée sur des navires commence à réfléchir à l'automatisation de la prise d'information des capteurs vers une interface utilisateur.

Mots inconnus :

Alphabet : ensemble fini d'objets appelés caractères
Mot (sur un alphabet) : suite finie de caractères. Le mot vide sera noté ; l'opération de concaténation de deux mots m1 et m2 est notée par simple juxtaposition m1m2.
Préprocesseur : actions du préprocesseur visant à "nettoyer" le fichier source en appliquant des macros, supprimant les textes inutiles ou ajoutant les bibliothèques identifiées
Compilation : Programme informatique permettant de transformer un programme écrit dans un langage en un code machine exécutable par le processeur ou une machine virtuelle.
Assemblage : Phase de la compilation écrivant le code source transformé en assembleur en binaire exécutable.
Édition de liens : Permet de rassembler chaque fichiers objets, en remplacement des variables/pointeurs par leurs adresses réelles en mémoire.
Automate : Mécanisme cyclique permettant à l'aide de transitions de partir d'un état initial vers un état final en passant par des états intermédiaires.
AVR-GCC : Compilateur basé sur GCC permettant d'interagir sur le microcontrôleur de type Atmel AVR.
GCC : GNU Compiler Collection, compilateurs open source capables de compiler dans divers langages de programmation.
Analyse lexicale : vérifier que les symboles utilisés font bien partie de ce que le langage reconnaît.
Analyse syntaxique : vérifier que les symboles sont bien écrits dans un ordre autorisé par la grammaire du langage.
Analyse sémantique : vérifier que les correspondances entre les symboles ou le sens donné à une expression est bien cohérent avec les règles déterminées dans le langage.

Problématique :

Comment réaliser un script complet d'automatisation de la compilation au téléversement ?

Plan d’action :

- Comprendre le fonctionnement du compilateur GCC [90min].
- Comment assurer la communication entre le PC et l’Arduino [90min].
- Expliquer les caractéristiques d'un automate fini déterministe [45min].
-Simuler le comportement d'un automate fini simple [60min].
-Développer un script bash d’automatisation [90min].

Réalisation :

Analyse et fonctionnement du compilateur GCC.

1 le compilateur

Il y'a plusieurs phase lors de la compilation

1 L'analyse lexical --> liste tout les token" if , nom de variable ,+"

2 L'analyse syntaxique détecte si la syntaxe est bonne

3 L'analyse semantique verifie que les opérations sont logiques

exemple si on a 1+Variable mais que variable est du texte = ça marche pas

Puis on a une phase d'optimisation du code

Enfin on peut générer le code en ASM

ou code machine

puis les automate et après on doit créer un code bash pour automatisé la compilation et le téléversment

Option	Signification
--help	Affiche l'aide de GCC.
--version	Donne la version de GCC.
-E	Appelle le préprocesseur. N'effectue pas la compilation.
-S	Appelle le préprocesseur et effectue la compilation. N'effectue pas l'assemblage ni l'édition de lien. Seuls les fichiers assembleur (« .S ») sont générés.
-c	Appelle le préprocesseur, effectue la compilation et l'assemblage, mais ne fait pas l'édition de lien. Seuls les fichiers objet (« .o ») sont générés.
-o nom	Fixe le nom du fichier objet généré lors de la compilation d'un fichier source.
-g	Génère les informations symboliques de débogage.
-fexceptions	Active la gestion des exceptions C++.
-fpic	Génère du code relogeable. Cette option est nécessaire pour la compilation des fichiers utilisés dans une DLL ou un fichier chargeable dynamiquement.
-On	Indique le niveau d'optimisation (n peut prendre les valeurs allant de 0 à 3, ou « s » pour optimiser la taille des binaires).
-mcpu=cpu	Indique le type de processeur pour lequel le code doit être optimisé. Le code fonctionnera sur tous les processeurs de la famille de ce processeur.
-march=cpu	Indique le type de processeur pour lequel le code doit être généré. Le code généré sera spécifique à ce processeur, et ne fonctionnera peut-être pas sur un autre modèle de la même famille. Cette option active automatiquement l'option -mcpu avec le même processeur.
-pipe	Utilise les pipes systèmes au lieu des fichiers temporaires pour les communications entre le préprocesseur, le compilateur et l'assembleur.
-w	Supprime tous les warnings.
-W	Active les warnings supplémentaires.
-Wall	Active tous les warnings possibles.
-mwindows	Crée un exécutable GUI Windows.
-mdll	Crée une DLL Windows.
-fvtable-thunks	Utilise le mécanisme des tables de fonctions virtuelles. Cette option est nécessaire pour utiliser les interfaces COM sous Windows.

https://c.developpez.com/cours/mode-emploi-gcc/