

**MICROCONTRÔLEURS  
MICROCONTRÔLEURS ET SYSTÈMES NUMÉRIQUES  
PROJET DE FIN DE SEMESTRE****1. MICROCONTROLEURS: PROJET DE FIN DE SEMESTRE 2024**

Un projet de fin de semestre est proposé dans les dernières semaines du semestre de printemps. Le but du travail proposé est de permettre d'approfondir et étendre les connaissances théorique acquises, ainsi que de développer des compétences de gestion de projet et de développement de projet en groupe.

**1.1 OBJECTIF DU PROJET 2024**

Le projet de fin de semestre s'attache à développer les compétences nécessaires en matière de la gestion et de l'organisation d'un projet à base de microcontrôleur et aborde certains sujets étudiés de façon théorique et pratique. Les objectifs de ce travail sont formulés comme suit

- planification d'un projet à base de microcontrôleur, et développement du projet dans le cadre d'un travail en groupe,
- étude d'un périphérique et son protocole de communication, ainsi que son interfacçage avec un microcontrôleur, et implémentation dans le projet,
- rédaction de rapport technique.

L'étude proposée doit permettre un développement efficace et rapide du prototype.

**2. ORGANISATION DU PROJET 2024**

Les étudiants travaillent dans leur groupe, tel qu'il a été défini pour les travaux pratiques [1], et remettent un projet par groupe. Le kit STK-300 est à disposition en prêt, les logiciels sont gratuits et peuvent être installés sur des PCs privés, des oscilloscopes USB peuvent être empruntés pour la durée du projet. Il est important de comprendre que le travail proposé est un projet, et ce sont les membres du groupe qui doivent sur la base des compétences acquises dans les TP développer et trouver les solutions nécessaires à présenter un projet fonctionnel, et dans ce sens, le projet n'est pas un TP guidé. Les salles de TP (DLL) seront ouvertes aux heures de TP et des assistants seront présents; ils ne pourront cependant pas se substituer au travail des groupes et effectuer du debug par exemple, et répondront à des questions précises.

Les règles sanitaires émises par la Confédération Suisse et leur application édictée par l'EPFL doivent être respectées à tout moment [2] et pourront dicter des modifications d'organisation.

**2.1 DÉFINITION DU PROJET**

Chaque groupe de deux étudiants définit un sujet de projet (application) faisant usage du MCU Atmega128L, et de plusieurs périphériques du kit. Un périphérique nouveau et obligatoire est présenté en Table 2.1 et Figure 2.1 [3]. Le développement du driver (pilote) de ce périphérique fait partie du projet; une méthode possible est proposée en Annexe 1.

No	Nom (*)	Description et exemple d'utilisation	Nombre
x	keypad 4x4 [3]	Pavé tactile 4x4 touches Exemple: interface utilisateur générique, entrée de données simples (mot de passe) ou complexes (texte, multiples keyboard virtuels, etc), contrôle de systèmes (valeurs en mémoire, actionneurs, capteurs, etc.)	1/kit

Table 2.1: Nouveau périphérique obligatoire

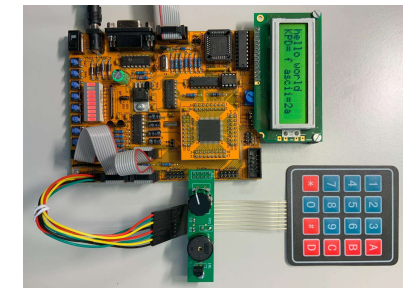
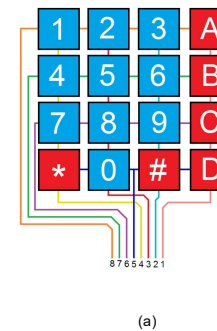


Figure 2.1: Carte périphérique 4x4 keypad, vue du dessus et connexions, tiré de [1]. (b) Vue du système connecté sur la carte STK-300.

La liste des périphériques supplémentaires est donnée en Table 2.2, et est basée sur les périphériques disponibles dans le kit utilisé au cours des travaux pratiques. Sur la base de ce kit, vous pouvez imaginer votre propre projet; il est aussi possible d'ajouter des périphériques supplémentaires. Donnez libre cours à votre créativité dans la définition d'une application et l'usage des périphériques. Les spécifications techniques des périphériques sont disponibles sous forme de datasheets sur le serveur du cours: \\stiitcgen0a\cours\Micro contr-oleur\Documentation et sur le site Moodle du cours.

No	Nom (*)	Description et exemple d'utilisation	Nombre
LCD	Affichage LCD 2x16 (Hitachi44780U 2x16 LCD)	Affichage 2x16 caractères Exemple: affichages texte, graphiques, affichages changeant dynamiquement, ...	1/kit
UART	Cable d'interface USART/USB	Permettant la communication avec un PC par son port USB	1/kit
M1	Moteur pas à pas (Stepper motor x27)	Moteur pas-à-pas, peut amener de petites charges (aiguille, typiquement) sur des rotations permanentes ou à une position angulaire précise. Exemple: aiguille analogique d'affichage (montre, cadran d'affichage de valeur analogique (vitesse), ...	1/kit
M2	Interface IR NEC (IR, IR Receiver)	Télécommande et récepteur IR. Exemple: contrôle d'un actionneur (moteur) ou d'une valeur numérique (LCD), ...	1/kit
M2	Buzzer (Buzzer)	Buzzer piezo-électrique. Exemple : Alarme, musique simple, ...	1/kit

Table 2.2: Liste des périphériques du kit, et exemples de leur utilisation dans des projets

No	Nom (*)	Description et exemple d'utilisation	Nombre
M2	Encodeur angulaire (Rotary encoder)	Interface utilisateur rotative par pas, avec un bouton-poussoir. Exemple: contrôle de volume, contrôle d'affichage LCD, ...	1/kit
M3	Potentiomètre	Réalise une valeur de tension analogique. Exemple: contrôle de valeur analogique, volume, serrure électronique analogique, ...	1/kit
M3	BNC	Connecteur BNC permettant l'entrée de valeur analogique. Exemple: détecteur de fréquence, accordeur électronique, ...	1/kit
M4	Clavier PS/2	Connecteur pour clavier PS/2. Exemple: Interface utilisateur vers LCD, USART, autre MCU, ...	1/kit
M4	Détecteur de distance (Sharp GP2Y0A21)	Détecteur de distance analogique; délivre une valeur en tension qui est proportionnelle à une distance mesurée. Exemple: mesure de distance, prédiction de collision, interface utilisateur pour le contrôle ou le jeu, ...	1/kit
M4	Moteur servo (Servo motors)	Deux moteurs servo. Le servo Futaba S3003 est contrôlé en valeur angulaire (pas de rotation complète); le servo SG90 permet une rotation complète et est contrôlé en vitesse. Les moteurs servo tolèrent des charges plus élevées que les moteurs pas-à-pas. Exemple: Serrure, ouverture-fermeture de clapets, vis, mouvement d'objets, ...	1/kit 1/kit
M5	EEPROM (I2C)	Mémoire externe EEPROM communiquant en I2C. Mémoire non-volatile. Exemple: stockage de valeurs de mesures, paramètres de l'application, codes d'utilisateurs, ...	1/kit
M5	Capteur de température (1-wire)	Capteur de température communiquant en 1-wire. Exemple: Station météo, température instantanée, ...	1/kit

Table 2.2: Liste des périphériques du kit, et exemples de leur utilisation dans des projets

(\*) le nom du projet correspond à la dénomination du répertoire dans lequel se trouve la documentation technique du périphérique (datasheets) et disponible sur le serveur \\stiitcgen0a\cours\Micro contrôleur\Documentation et sur le site Moodle du cours. Le schéma de la carte STK300 est disponible dans le sous-répertoire \STK300, et le plan du PCB comprenant le périphérique dans le sous-répertoire \MICRO210-EE208 modules schematics

2.2 EXIGENCES ET CARACTÉRISTIQUES DE L'APPLICATION

L'étude doit être effectuée en considérant que l'application complète est développée en assembleur AVR. L'utilisation du périphérique nouveau obligatoire (keypad4x4) et de deux périphériques supplémentaires (Table 2.2) au minimum est obligatoire, par exemple: keypad4x4, affichage LCD et moteur servo, au minimum (les boutons-poussoirs, LEDs et le buzzer peuvent être utilisés mais ne comptent pas comme un module périphérique supplémentaire).

Les points suivants seront respectés:

- Multitâche
- Modularité (sous-routines intelligentes et réutilisables, bien documentées)
- Responsivité (le programme réagit toujours en temps réel)

Le programme:

- utilise une ou plusieurs interruptions,
- mesure et traite une quantité ou valeur, ou traite une information externe, et communique un résultat ou contrôle un dispositif,
- possède une interface utilisateur permettant d'entrer des paramètres,
- affiche un résultat sur LCD ou terminal par UART (RealTerm) ou autre,
- utilise les macros et bibliothèques présentées au cours.

2.3 MÉTHODE DE TRAVAIL

Dans le cadre d'un projet complexe, une analyse système sur papier, dite "top-down" est généralement favorable à un travail efficace, au cours de laquelle les spécifications du système sont préparées; dans une deuxième phase, les différents blocs composant le système sont extraits des spécifications, et leurs canaux de communication définis. Ces blocs fonctionnels sont à leur tour définis au moyen de sous-blocs, de façon itérative jusqu'à obtenir une liste de sous-blocs de tailles raisonnables, et qui peuvent être codés ou développés facilement. Cette technique s'applique généralement aussi bien au développement matériel qu'au logiciel. Dans le cadre de ce projet, elle sera appliquée principalement au logiciel; la partie matérielle étant déjà développée, il suffira de choisir de façon appropriée sur quel port le périphérique peut ou doit être connecté. Cette partie du travail (étude préliminaire et spécifications du projet) s'effectue sur papier, et avant de ne coder quoi que ce soit.

Une fois cette analyse effectuée, une option intéressante consiste à développer un driver simpliste pour le périphérique, permettant de garantir rapidement qu'une communication bidirectionnelle soit possible. Pour cela il est nécessaire de:

- étudier les datasheets du périphérique de façon détaillée, puis
- définir les méthodes et modes d'accès au périphérique
  - comment configurer le MCU et le périphérique,
  - comment transmettre des commandes au périphérique,
  - comment recevoir des données du périphérique;
- coder un driver simple qui permette de savoir qu'une donnée basique soit acceptée par le périphérique, qui à son tour y réponde simplement (par exemple, allumer un segment d'un périphérique);
- si cela ne fonctionne pas, il faut étudier le code du driver et s'assurer qu'il corresponde bien à ce qui est spécifié dans les datasheets, puis corriger;
- il faut vérifier les signaux au moyen de l'oscilloscope; si le driver est simple, il devrait être possible rapidement de détecter les erreurs, le nombre de trames ou paquets étant limité à un ou deux;
- si cela ne fonctionne pas, alors seulement faites appel au support par le forum disponible sur le site Moodle.

Cette analyse a pour but de permettre de coder un driver simple du périphérique. Le driver doit ensuite être amélioré afin de traiter toutes les fonctions du périphérique, et connecté à divers autres modules, l'un après l'autre, de manière ascendante ("bottom-up"). Cette méthode permettra de faire fonctionner des parties de plus en plus importantes du projet dans un même bloc fonctionnel. D'autres blocs fonctionnels peuvent être développés en parallèle.

2.4 DÉLIVRABLES, DÉLAIS ET NOTATION DU PROJET

Le projet peut être débuté dès réception de ce document. Les questions relatives au projet peuvent être présentées sur le forum du cours (Moodle), ou à l'occasion de sessions pendant les heures normalement allouées aux TP.

Le travail de fin de semestre est noté sur la base des livrables remis sur le site Moodle du cours, et compte pour 45% de la note finale du cours. Une seule remise par groupe est nécessaire.

1. rapport technique (Section 2.4.2),
2. démonstration du projet sous forme de vidéo simple (Section 2.4.3),
3. code source complet (Section 2.4.4).

Les livrables doivent être remis sur le site Moodle du cours au plus tard le Mardi 28.05.2024 à 08:00.

Le kits STK-300 et autre matériel en prêt seront rendus par les groupes selon le plan suivant:

- Mercredi 29.05.2024, 08:00-10:00 en MED 2-2419, ou
- Jeudi 30.05.2024, 15:00-17:00 en MED 2-2519.

La totalité du matériel en prêt doit être rendu à cette occasion, et les pannes doivent être annoncées par une fiche de panne insérée dans le kit. Aucune note ne sera donnée au cours sans rendu des kits complets. Une alternative sera préparée et placée sur le site Moodle si les conditions sanitaires [2] ne permettent pas la remise aux dates ci-dessus.

### 2.4.1 CRITÈRES DE NOTATION

Le travail sera évalué sur la conformité aux exigences et l'opération correcte du projet, la clarté et le détail des explications du système général ainsi que des modes d'accès au périphérique, l'originalité et l'investissement. Le schéma suivant sera appliqué:

- Rapport
  - 10% structure
  - 10% contenu utilisateur (manuel d'utilisation)
  - 20% contenu technique
- Projet et démonstration
  - 10% originalité/créativité
  - 10% travail investi
  - 10% qualité et réussite du programme
- Code/programme
  - 10% modularité
  - 10% documentation
  - 10% concepts avancés (pointeurs et opération complexe sur la mémoire, opérations mathématiques avancées, périphériques complexes, etc.)

Les manquements aux exigences mentionnées plus haut (par exemple: pas d'interruptions/temps réel, fichiers non disponibles sur le serveur, retards) seront traduits par une réduction de la note.

### 2.4.2 RAPPORT TECHNIQUE

Le format du rapport est pdf, maximum 7 pages au format standard (police d'écriture de 11 points minimum, marges de 2 cm). Veuillez nommer le fichier en respectant le code suivant: "MCU2024-GXXX.pdf," où "GXXX" désigne votre code de groupe (No. du kit), par exemple "MCU2024-G212.pdf"

Les divers Sections consécutives du rapport seront adaptées aux besoins du rapport; par exemple, une liste de sections pourrait être:

1. Page 1: Information administratives (auteurs, dates), titre, introduction et description générale de l'application sans schémas
2. Pages 2 à 4: Mode d'emploi (mise en opération système et utilisateur), puis description technique de l'application et du matériel (utilisation des ports, interfaces d'acquisition et d'affichage, périphériques, interruptions), fonctionnement du programme (top-down), présentation des modules (parties indépendantes du programme)
3. Pages 5 à 7: Description de détail de l'accès aux périphériques, références (conclusion non nécessaire).
4. Pages suivantes: Annexes. Divers annexes dont une obligatoire consistant en la totalité du code source de l'application.

### Remarques

Il est bien de définir déjà à ce stade de l'étude les fonctions qui sont répétitives et qui pourront être placées dans une librairie (macro, ou sous-routines).

Les méthodes d'interruptions doivent être décrites.

Les allocations de mémoire programme et données doivent être décrites.

### 2.4.3 DÉMONSTRATION

La démonstration est effectuée sous forme d'une vidéo brève de trois minutes au maximum présentant le projet, les diverses fonctionnalités réalisées et les solutions techniques particulières apportées. La vidéo peut être réalisée par des moyens légers (téléphone cellulaire, par exemple) et aucun montage n'est nécessaire. Il est important par contre que les fonctionnalités soient démontrées et expliquées et il est donc important que la présentation soit très bien préparée et organisée.

### 2.4.4 CODE SOURCE

Le code source de l'application sera remis en version électronique (fichiers) et sous forme pdf en annexe au rapport technique.

### 2.5 RÉFÉRENCES

- [1] listes des groupes de TP: site Moodle, Mars 2024.
- [2] <https://www.epfl.ch/campus/security-safety/en/health/coronavirus-covid19/>, December 5, 2022, et suivant.
- [3] Parallax Inc., 4x4 Matrix Membrane Keypad (#27899), v1.2, 2011.
- [4] Atmel, 8-bit Atmel Microcontroller with 128KBytes in-System Programmable Flash, ATmega128, ATmega128L
- [5] Fairchild Semiconductor, MM74C922 • MM74C923 16-Key Encoder • 20-Key Encoder, revised 1999

## 3. ANNEXES

### 3.1 ANNEXE I: DÉVELOPPEMENT DU SYSTÈME DE CONTRÔLE DU KEYPAD 4x4

La carte 4x4 keypad en une matrice d'interconnexion dont il faut gérer le contrôle afin de détecter laquelle parmi les seize touches a été activée, Figure 3.1. La matrice est accédée en mode d'entrée (du point de vue de la matrice) par ses colonnes nommées X1 à X4; la matrice est accédée en mode de sortie par ses lignes nommées Y1 à Y4 (la ligne Y5 n'est pas relevante pour une matrice 4x4). Le plan de connectique des colonnes et lignes est présenté en Figure 2.1(a).

Conceptuellement, la détection de touche est effectuée en trois phases consécutives

Dans une première phase, une détection de touche et une détection de la ligne dans laquelle cette touche se trouve sont effectuées, Figure 3.1(a).

- Toutes les colonnes X1 à X4 sont mises à GND et toutes les lignes Y1 à Y4 sont tirées à VDD. Pour ce faire, il est nécessaire de configurer les quatre pins du port choisi en PULL-UP [4] (noté "internal" sur la Figure 3.1(a)).

- Si un bouton est appuyé (bouton No. 6 dans l'exemple), un chemin conducteur est créé entre la colonne et la ligne qui décharge la ligne à GND. Dans l'exemple, Y2 prend la valeur GND alors que les autres lignes restent inchangées.
- Y2 est mémorisé.

La colonne est détectée dans une deuxième phase consécutive et indépendante de la première phase.

- Toutes les colonnes sont mises à VDD. Puis, l'une après l'autre, chaque colonne est placée à GND alors que les autres restent inchangées à VDD.
- Dans l'exemple présenté en Figure 3.1(b), la colonne X1 est à GND ce qui ne cause aucune modification des lignes de sorties Y1 à Y4.
- Puis, dans l'exemple présenté en Figure 3.1(c), la colonne X2 est à GND ce qui ne cause aucune modification des lignes de sorties Y1 à Y4.
- Finalement, Dans l'exemple présenté en Figure 3.1(d), la colonne X3 est à GND ce qui force la ligne Y2 à GND. La colonne X3 est mémorisée.

Le debouncing du signal lu aux touches nécessite un délai d'environ 10-30 ms effectué par un compteur.

Le décodage du caractère est effectué en troisième phase. Dans l'exemple suivi, l'appui de la touche a été détecté à l'intersection de la colonne X3 et de la ligne Y2. Cette information est stockée sous forme de deux valeurs (colonne=3, ligne=2) et doit être transformée en la touche correspondante, No 6 dans l'exemple. Une LUT est utilisée à cet effet qui est implémentée comme une table stockée en mémoire-programme; un pointeur est initialisé au début de la table; ce pointeur reçoit un offset (décalage) à chaque touche appuyée qui permet d'extraire la valeur correcte.

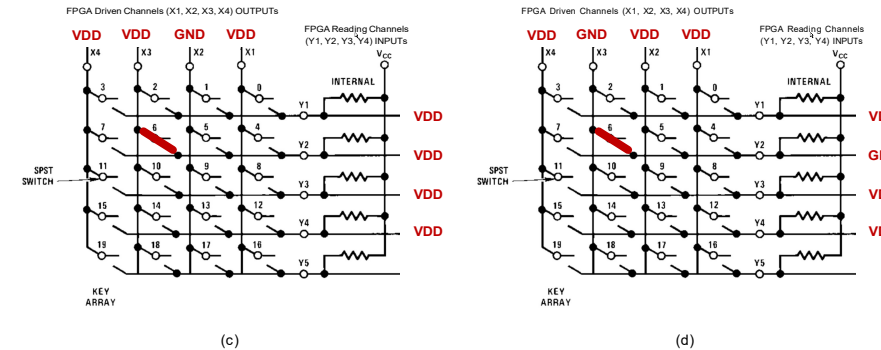


Figure 3.1: Opération du décodeur 4x4 keypad, tiré de [5]; (a) première phase, détection de ligne. (b), (c), (d) phase deux, états consécutifs jusqu'à la détection de colonne en (d).

Un fichier présentant une partie de l'implémentation des phase 1 et phase 2 est proposé sous le nom de kpd4x4.asm. Certaines parties systématiques doivent être complétées et nécessitent une compréhension complète des deux phases; la phase trois n'est pas implémentée. Finalement, kpd4x4.asm est une application individuelle de lecture du keypad, et doit être adapté et intégré à une application complexe.

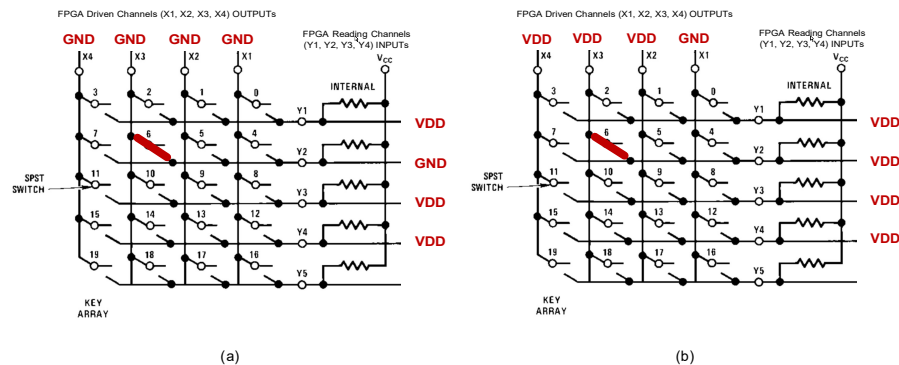


Figure 3.1: Opération du décodeur 4x4 keypad, tiré de [5]; (a) première phase, détection de ligne. (b), (c), (d) phase deux, états consécutifs jusqu'à la détection de colonne en (d).