



# Introduction aux Systèmes embarqués

## CHAP 4 : Gestion des claviers matriciels

1.	Introduction .....	2
2.	Installation de la bibliothèque keypad .....	2
3.	Liste des fonctions de la librairie Keypad .....	3
4.	Création d'un "objet clavier" .....	3
4.1.	Le constructeur de création .....	3
4.2.	Paramètres .....	3
4.3.	Exemple de création .....	3
5.	Description des fonctions .....	3
5.1.	void begin() .....	3
5.2.	char getKey() .....	3
5.3.	KeypadState getState() .....	4
5.4.	setHoldTime(unsigned int time) .....	4
5.5.	setDebounceTime(unsigned int time) .....	4
5.6.	addEventListener(keypadEvent) .....	4
6.	Principe d'utilisation .....	4
7.	Exemple de programme .....	4

## 1. Introduction

Quand les boutons et les interrupteurs ne sont pas suffisants pour les exigences d'une application, un clavier matriciel est une bonne alternative. Il consiste en une matrice de LxC boutons poussoirs.

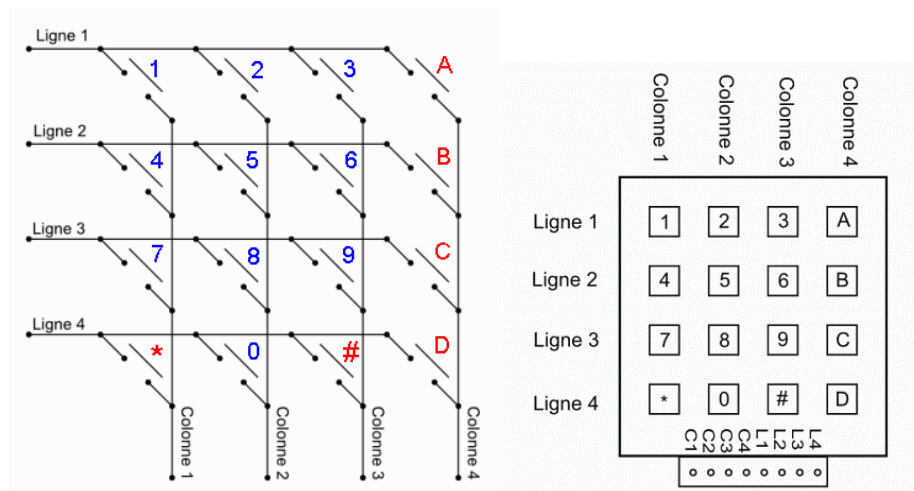
Le clavier étudié dans la cadre de ce cours comporte 16 touches, dont 10 numériques (0-9) et 6 touches marquées \*, #, A, B, C et D présentées sous la forme d'une matrice.



Donc au lieu d'avoir 16 fils (1 par touche) et une masse, le multiplexage n'utilise que 8 sorties : 4 lignes et 4 colonnes.

Il est généralement utilisé en téléphonie, saisie de données numériques dans un programme Arduino (calibrage...) ou pour le choix de menu...

Typiquement, le brochage d'un clavier matriciel est le suivant :



## 2. Installation de la bibliothèque keypad

La librairie Keypad est une librairie pour le système Arduino. Cette librairie est basée sur le tutoriel consacré aux claviers :

Elle a été créée pour aider à l'abstraction matérielle, et à l'amélioration de la lisibilité du code. Elle "cache" (et automatise) les instructions pinMode et digitalRead appelées par l'utilisateur. La librairie Keypad s'intègre les librairies d'abstraction matérielle :

Dézipper et mettre le répertoire Keypad après téléchargement depuis le site ARDUINO dans votre répertoire Arduino dans le sous-répertoire "\libraries. Dans le logiciel Arduino, créer un nouveau programme (ou en ouvrir un) et sélectionner depuis la barre de menu "Sketch->Import Library-

>Keypad" (Programme > Importer Librairie > Keypad ). Une fois que votre librairie est importée, la ligne "#include <Keypad.h>" doit apparaître au début de votre programme.

### 3. Liste des fonctions de la librairie Keypad

- Keypad(makeKeymap(userKeymap), row[], col[], rows, cols)
- begin()
- getKey()
- getState()
- setHoldTime(unsigned int time)
- setDebounceTime(unsigned int time)
- addEventListener(keypadEvent)

### 4. Création d'un "objet clavier"

#### 4.1. Le constructeur de création

Pour créer un "objet clavier", il faut utiliser le constructeur suivant :

```
Keypad(makeKeymap(userKeymap), row[], col[], rows, cols)
```

#### 4.2. Paramètres

- userKeymap : tableau à 2 dimensions définissant les symboles des touches
- row[] : tableau correspondant aux numéros des broches utilisées pour les lignes
- col[] : tableau correspondant aux numéros des broches utilisées pour les colonnes
- rows : nombre de lignes
- cols : nombre de colonnes

#### 4.3. Exemple de création

```
const byte rows = 4; //four rows
const byte cols = 3; //three columns

char keys[rows][cols] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'#','0','*'}
};
byte rowPins[rows] = {5, 4, 3, 2}; //connect to the row pinouts of the
keypad
byte colPins[cols] = {8, 7, 6}; //connect to the column pinouts of the
keypad

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, rows, cols );
```

Ce code crée un objet de type Keypad, appelé clavier, qui utilise les broches 5,4,3,2 pour les lignes et les broches 8,7,6 pour les colonnes. Le clavier à 4 lignes et 3 colonnes, et a donc 12 touches.

### 5. Description des fonctions

#### 5.1. void begin()

Initialise toutes les variables. En fait le constructeur le fait déjà.

```
void begin(makeKeymap(userKeymap))
```

#### 5.2. char getKey()

Renvoie la touche (sous forme caractère ASCII) qui est appuyée, si une l'a été.

### 5.3. KeypadState getState()

Renvoie l'état courant du clavier.

Les 4 états possibles sont IDLE, PRESSED, RELEASED et HOLD.

### 5.4. setHoldTime(unsigned int time)

Définit le nombre de millisecondes que l'utilisateur doit appuyer sur un bouton pour activer l'état HOLD.

### 5.5. setDebounceTime(unsigned int time)

Définit le nombre de millisecondes entre 2 appuis de touches. C'est une pause anti-rebond.

### 5.6. addEventListener(keypadEvent)

Crée un évènement si le clavier est utilisé. Vous pouvez regarder un exemple dans le logiciel Arduino dans "File -> Sketchbook -> Examples -> Library-Keypad -> EventSerialKeypad"

## 6. Principe d'utilisation

Pour inclure la librairie Keypad dans un programme, on ajoutera au début du programme la ligne suivante :

```
#include <Keypad.h>
```

Pour ce faire, le plus simple est d'utiliser le menu du logiciel Arduino Programme > Importer Librairie > Nom\_librairie

## 7. Exemple de programme

Chaque fois qu'on appuie sur une touche, on écrit son caractère dans le moniteur série.

```
#include <Keypad.h> // inclusion de la librairie pour clavier matriciel
//--- Constantes utilisées avec le clavier 4x4
const byte LIGNES = 4;           // 4 lignes
const byte COLONNES = 4;         // 4 colonnes

// déclaration des broches ---
const int L1=38;
const int L2=40;
const int L3=42;
const int L4=44;

const int C1=46;
const int C2=48;
const int C3=50;
const int C4=52;

//--- Définition des touches
char touches[LIGNES][COLONNES] = {
    {'1','2','3','A'},
    {'4','5','6','B'},
    {'7','8','9','C'},
    {'*','0','#','D'}
};

// tableaux de lignes et colonnes
byte BrochesLignes[LIGNES] = {L1, L2, L3, L4};
byte BrochesColonnes[COLONNES] = {C1, C2, C3, C4};
char touche; // variable de stockage de la valeur de la touche appuyée
// création d'un objet keypad = initialisation clavier
```

```
Keypad clavier = Keypad( makeKeymap(touches), BrochesLignes, BrochesColonnes,
LIGNES, COLONNES );
// les broches de lignes sont automatiquement configurées en ENTREE avec
pullup interne activé
// les broches de colonnes sont automatiquement configurées en SORTIE
void setup()
{
    Serial.begin(9600);
    delay(10); // pause rapide pour laisser un temps d'initialisation

    Serial.println("init OK") ; // affiche la chaîne texte - message de
test
    delay(2000); // pause de 2 secondes
} // fin de la fonction setup()

void loop()
{
    touche = clavier.getKey(); // lecture de la touche appuyée
    if (touche != NO_KEY){ // si une touche a été frappée -- gestion de la
touche appuyée
        Serial.print(touche);
        Serial.print(" ");
        delay(300); // pause entre 2 appuis
    } // fin gestion de la touche appuyée
}
```