IEEE
CAIRO UNIVERSITY STUDENT BRANCH

Embedded Systems Workshop

Session 5

IEEE

Hello!

Instructor: Mohamed Magdy

Cairo University, Electrical Engineering (EPE) Department.

Contact me: mohamed.naem04@eng-st.cu.edu.eg

LinkedIn: https://shorturl.at/hgslP

Attendance

place qr code

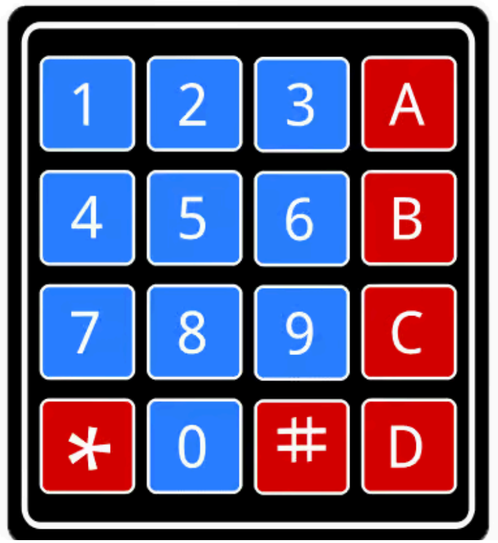
AGENDA :

Keypad scanning

LCD interfacing

Keypad

What is keypad ?

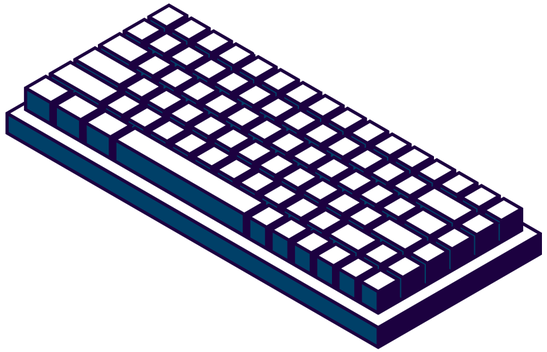


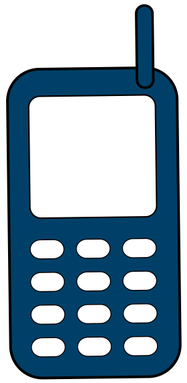
What is keypad ?

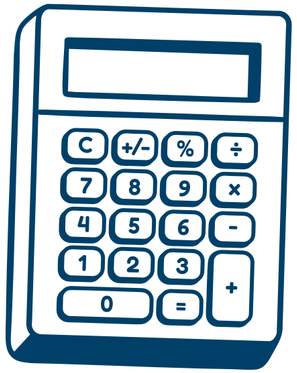
Keypad: is a set of a mechanical switches that are arranged in a matrix to minimize the number of the used pins and it is used as an input device to read the key pressed by the user and then process it by the MCU

Many applications requires large number of keys connected to computing systesms

Examples : PC Keyboard ,Cell Phone Keypad and Calculators

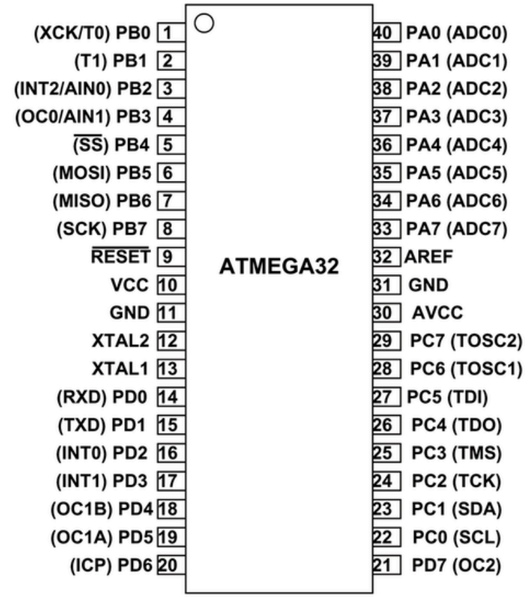






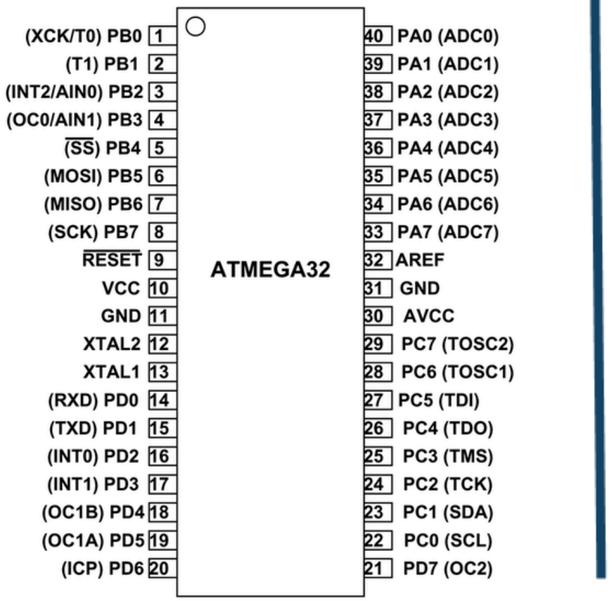
What is the Challenge ?!

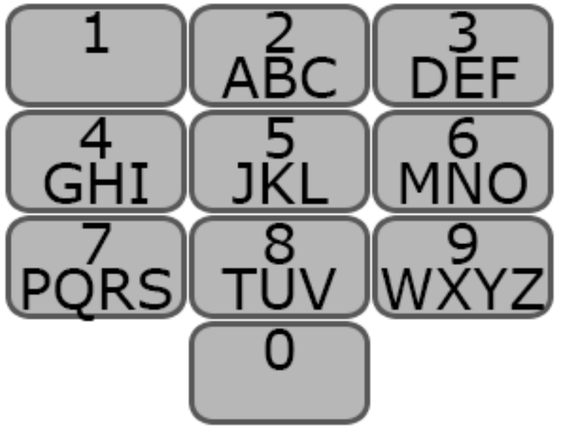
When we want to interface one key to the microcontroller then it needs one GPIO pin
But when we want to interface many keys like 9 , 12 ,16 or even more then it may acquire all the GPIO pins of the microcontroller



What is the Challenge ?!

Lets take the phone keypad as example if we are going to implement a system that contains 12 switches like this image in the traditional way each switch will need 1 input digital pin from the MCU this means that the switches in our system will consume 12 pins.





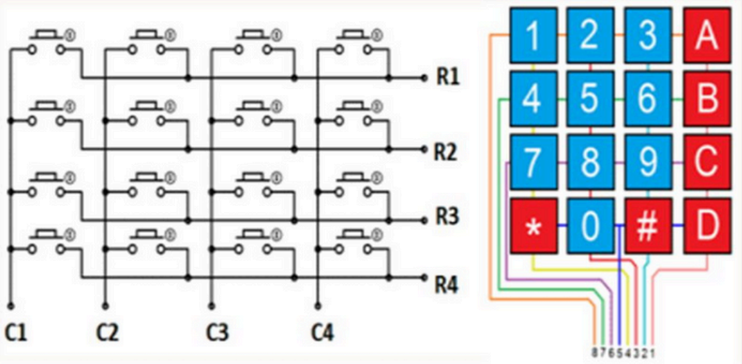
Solution :

To save some GPIO pins of microcontroller we can use Matrix keypad .

What is Matrix Keypad ?

Solution :

To save some GPIO pins of microcontroller we can use Matrix keypad .



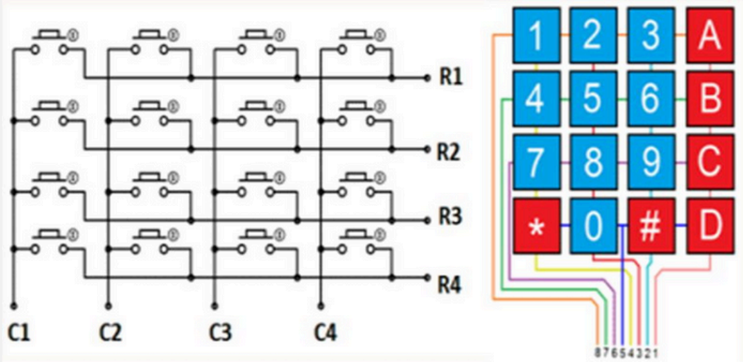
What is Matrix Keypad ?

Nothing is different from normal push buttons; the only difference is in the way the switches are connected.

Solution :

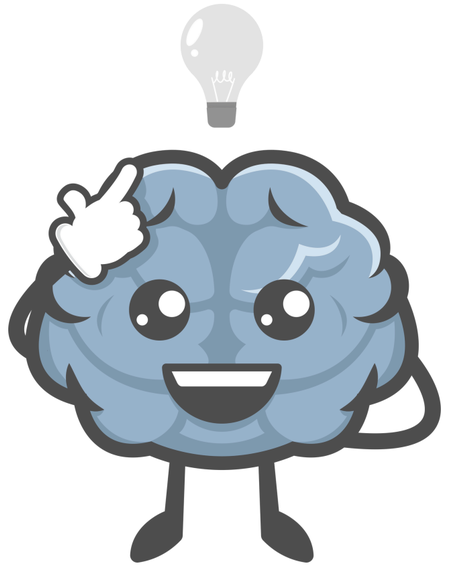
To save some GPIO pins of microcontroller we can use Matrix keypad .

Matrix Connection



What is Matrix Keypad ?

Nothing is different from normal push buttons; the only difference is in the way the switches are connected.

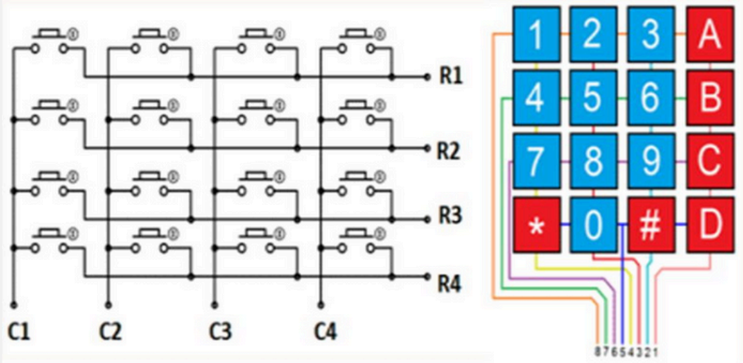


How many GPIO pins required to interface with 16 keys ?!

Conclusion :

Keypad is most widely used input device to provide input from the outside world to the microcontroller The Keypad makes and application more users interactive.

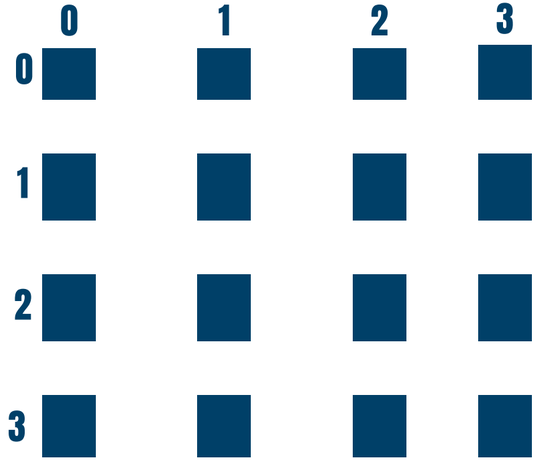
Matrix Connection



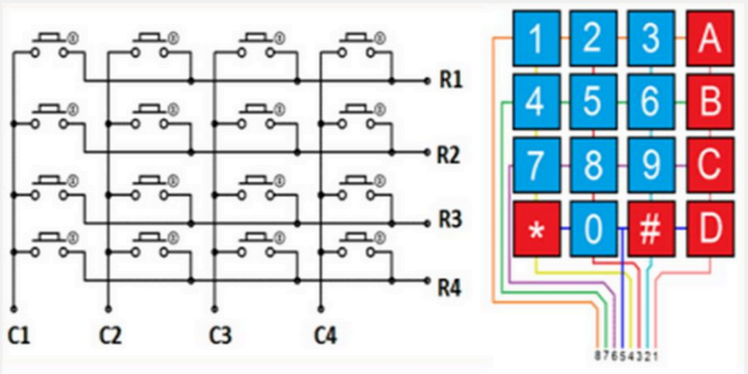
Matrix Connection Method

Multiplexed matrix keypad in this technique keys are connected in a matrix

Row/Column sytle



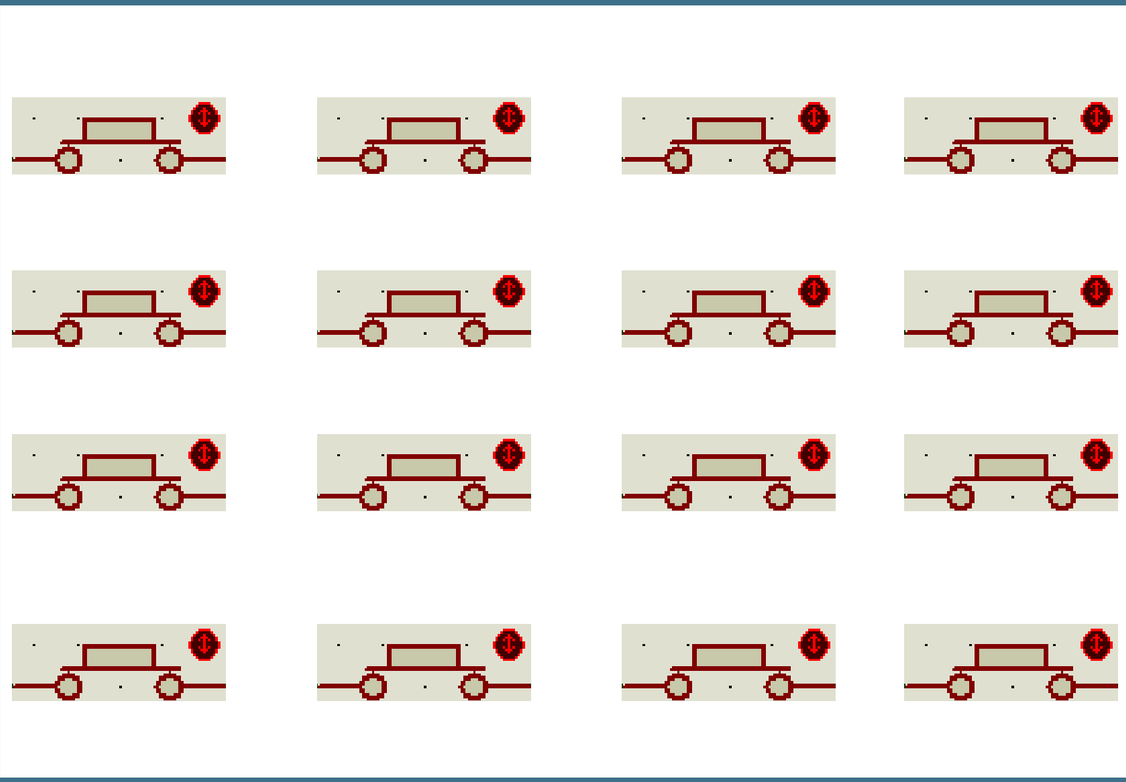
Matrix Connection



matrix

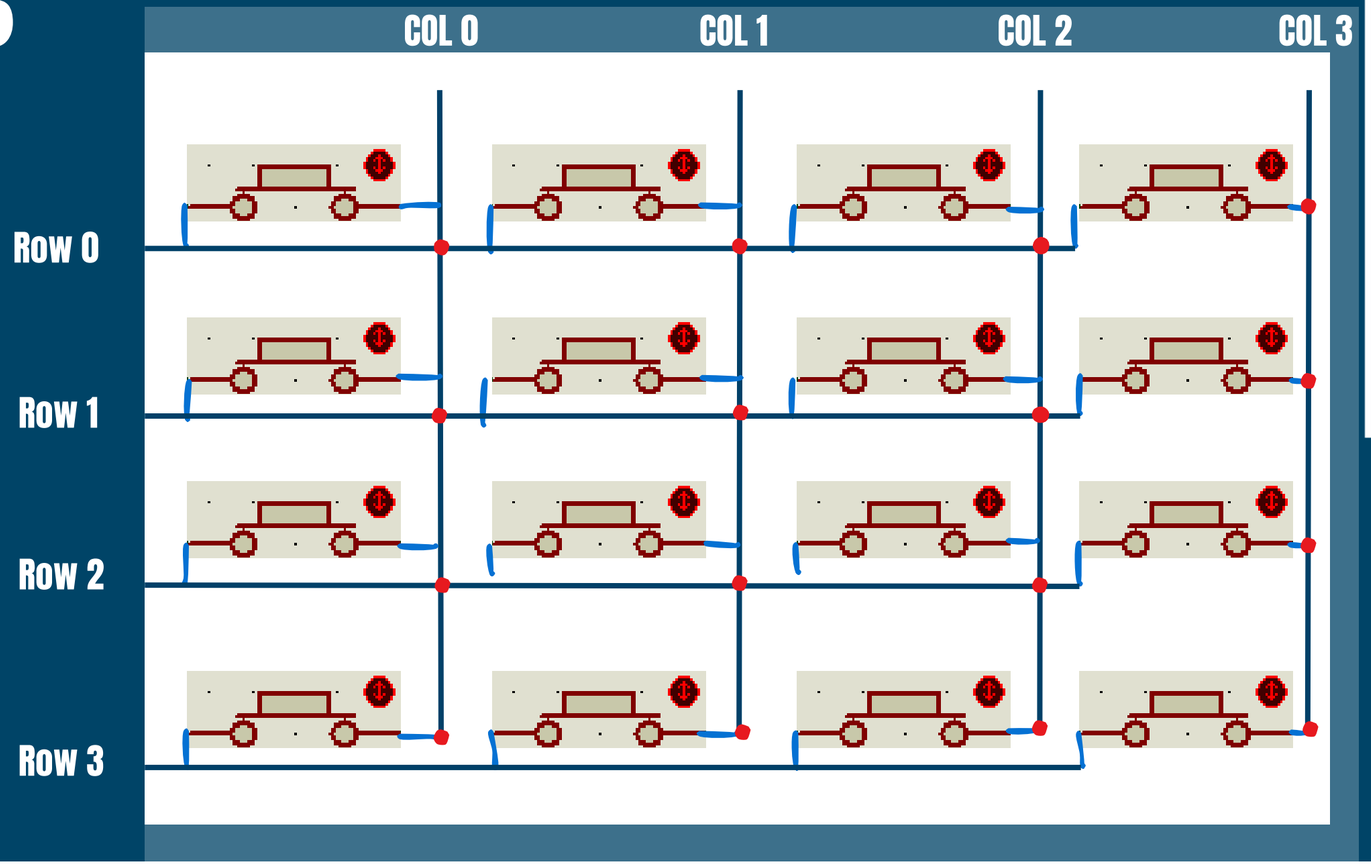
Understand Hardware

How to make this connection ?

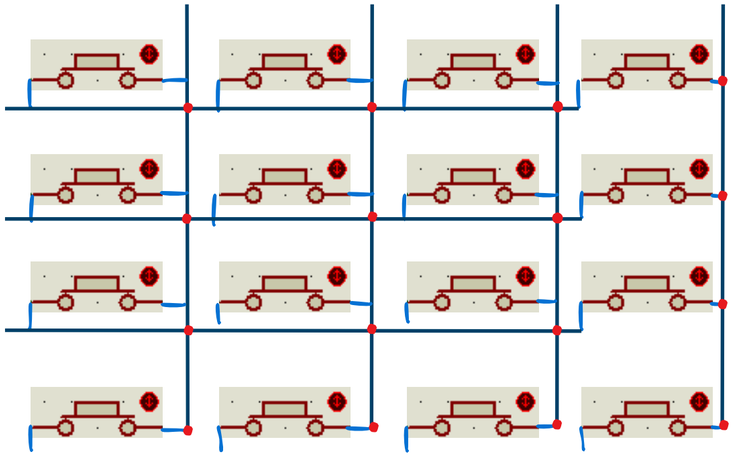


Understand Hardware

How to make this connection ?



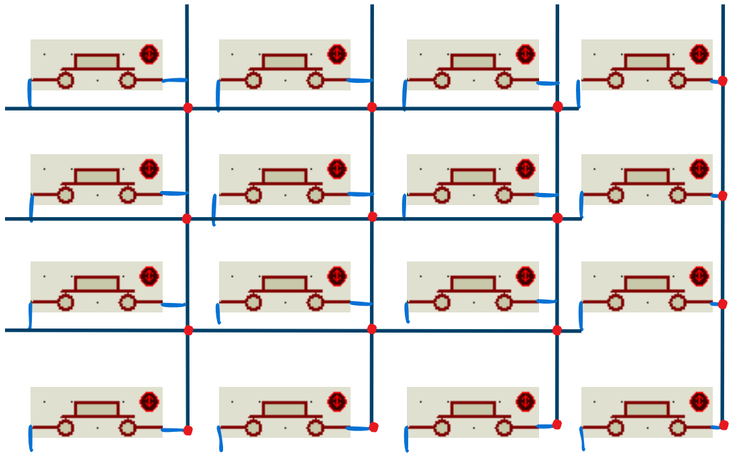
interfacing with Keypad



A Keypress establishes a connection between the corresponding row and column between which the switch is placed otherwise ; there is no connection between rows and columns

In order to read the Keypress, we need to configure the rows as outputs and columns as inputs

interfacing with Keypad

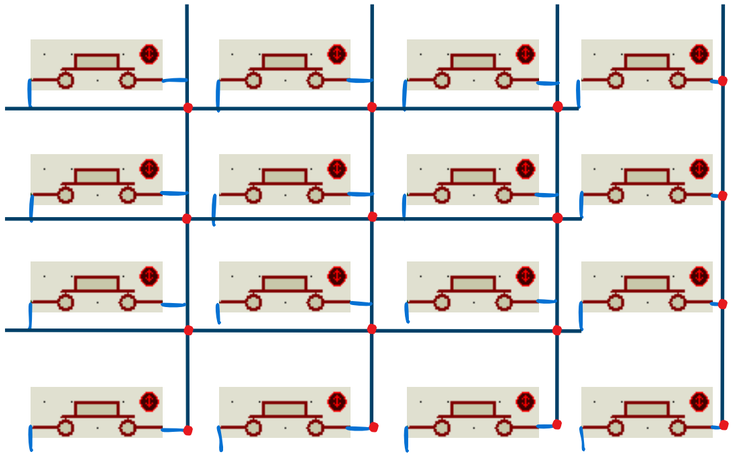


Columns are read after applying signals to the rows in order to determine whether or not a key is pressed and if pressed ,which key is pressed

questions ?

- **rows and col inputs ?** — floating as we need at least a signal to determine if key is pressed or not so we make one of them output and other is input 1 send and other receive (Ping → Pong)
- **rows output ?** — as output rows send signals high or low so we cam make one of them low and other is high or vice versa according to the connection (pull up) and this is called scanning
- **Col input ?** — To see if there is any signals reach the button (Pull up) (High) if key pressed it goes down act as a sensor
- **rows and col outputs ?** — Short Circuit maybe cause damage to the MCU means all are (HIGH)

Scanning



- if the button is pressed we need to get the location of the button , means the corresponding ROW AND COLUMNS NUMBER .
- Scan process starts by setting all the rows and columns as input pins .
- columns pins are connected to external pull up resistors to pull these pins to logic high values .
- to scan switches in a row, MC configures it as output and sets it to low , then checks one at a time .
- If a column line goes low , MC detects a pressed switch , it will be the switch connected between this row and column pins otherwise , no pressed switch , no pressed switch and goes to scan next row and so on .

Coding time

File.h

```c
/******************************************************************************
 *
 * Module: KEYPAD
 *
 * File Name: keypad.h
 *
 * Description: Header file for the Keypad driver
 *
 * Author: Mohamed Magdy
 *
 *******************************************************************************/

#ifndef KEYPAD_H_
#define KEYPAD_H_

#include "../common/std_types.h"

/*******************************************************************************
 *                                Definitions                                  *
 *******************************************************************************/

/* Keypad configurations for number of rows and columns */
#define KEYPAD_NUM_COLS                 4
#define KEYPAD_NUM_ROWS                 4

/* Keypad Port Configurations */
#define KEYPAD_ROW_PORT_ID              PORTB_ID
#define KEYPAD_FIRST_ROW_PIN_ID         PIN0_ID

#define KEYPAD_COL_PORT_ID              PORTB_ID
#define KEYPAD_FIRST_COL_PIN_ID         PIN4_ID

/* Keypad button logic configurations */
#define KEYPAD_BUTTON_PRESSED           LOGIC_LOW
#define KEYPAD_BUTTON_RELEASED          LOGIC_HIGH

/*******************************************************************************
 *                      Functions Prototypes                                   *
 *******************************************************************************/

/*
 * Description :
 * Get the Keypad pressed button
 */
uint8 KEYPAD_getPressedKey(void);

#endif /* KEYPAD_H_ */
```

IEEE CUSB

File.c

```c
/******************************************************************************
 *
 * Module: KEYPAD
 *
 * File Name: keypad.c
 *
 * Description: Source file for the Keypad driver
 *
 * Author: Mohamed Magdy
 *
 *******************************************************************************/
#include "keypad.h"
#include "../MCAL/gpio.h"
#include <util/delay.h>

/*******************************************************************************
 *                      Functions Prototypes(Private)                          *
 *******************************************************************************/

#if (KEYPAD_NUM_COLS == 3)
/*
 * Function responsible for mapping the switch number in the keypad to
 * its corresponding functional number in the proteus for 4x3 keypad
 */
static uint8 KEYPAD_4x3_adjustKeyNumber(uint8 button_number);
#elif (KEYPAD_NUM_COLS == 4)
/*
 * Function responsible for mapping the switch number in the keypad to
 * its corresponding functional number in the proteus for 4x4 keypad
 */
static uint8 KEYPAD_4x4_adjustKeyNumber(uint8 button_number);
#endif

/*******************************************************************************
 *                      Functions Definitions                                  *
 *******************************************************************************/

uint8 KEYPAD_getPressedKey(void)
{
    uint8 col,row;
    GPIO_setupPinDirection(KEYPAD_ROW_PORT_ID, KEYPAD_FIRST_ROW_PIN_ID, PIN_INPUT);
    GPIO_setupPinDirection(KEYPAD_ROW_PORT_ID, KEYPAD_FIRST_ROW_PIN_ID+1, PIN_INPUT);
    GPIO_setupPinDirection(KEYPAD_ROW_PORT_ID, KEYPAD_FIRST_ROW_PIN_ID+2, PIN_INPUT);
    GPIO_setupPinDirection(KEYPAD_ROW_PORT_ID, KEYPAD_FIRST_ROW_PIN_ID+3, PIN_INPUT);

    GPIO_setupPinDirection(KEYPAD_COL_PORT_ID, KEYPAD_FIRST_COL_PIN_ID, PIN_INPUT);
    GPIO_setupPinDirection(KEYPAD_COL_PORT_ID, KEYPAD_FIRST_COL_PIN_ID+1, PIN_INPUT);
    GPIO_setupPinDirection(KEYPAD_COL_PORT_ID, KEYPAD_FIRST_COL_PIN_ID+2, PIN_INPUT);
#if(KEYPAD_NUM_COLS == 4)
    GPIO_setupPinDirection(KEYPAD_COL_PORT_ID, KEYPAD_FIRST_COL_PIN_ID+3, PIN_INPUT);
#endif
    while(1)
    {
        for(row=0 ; row<KEYPAD_NUM_ROWS ; row++) /* loop for rows */
        {
            /*
             * Each time setup the direction for all keypad port as input pins,
             * except this row will be output pin
             */
            GPIO_setupPinDirection(KEYPAD_ROW_PORT_ID,KEYPAD_FIRST_ROW_PIN_ID+row,PIN_OUTPUT);

            /* Set/Clear the row output pin */
            GPIO_writePin(KEYPAD_ROW_PORT_ID, KEYPAD_FIRST_ROW_PIN_ID+row, KEYPAD_BUTTON_PRESSED);

            for(col=0 ; col<KEYPAD_NUM_COLS ; col++) /* loop for columns */
            {
                /* Check if the switch is pressed in this column */
                if(GPIO_readPin(KEYPAD_COL_PORT_ID,KEYPAD_FIRST_COL_PIN_ID+col) == KEYPAD_BUTTON_PRESSED)
                {
                    #if (KEYPAD_NUM_COLS == 3)
                        return KEYPAD_4x3_adjustKeyNumber((row*KEYPAD_NUM_COLS)+col+1);
                    #elif (KEYPAD_NUM_COLS == 4)
                        return KEYPAD_4x4_adjustKeyNumber((row*KEYPAD_NUM_COLS)+col+1);
                    #endif
                }
            }
            GPIO_setupPinDirection(KEYPAD_ROW_PORT_ID,KEYPAD_FIRST_ROW_PIN_ID+row,PIN_INPUT);
            _delay_ms(10); /* Add small delay to fix CPU load issue in proteus */
        }
    }
}
```

```c
#if (KEYPAD_NUM_COLS == 3)
/*
 * Description :
 * Update the keypad pressed button value with the correct one in keypad 4x3 shape
 */
static uint8 KEYPAD_4x3_adjustKeyNumber(uint8 button_number)
{
    uint8 keypad_button = 0;
    switch(button_number)
    {
        case 10: keypad_button = '*'; // ASCII Code of *
                 break;
        case 11: keypad_button = 0;
                 break;
        case 12: keypad_button = '#'; // ASCII Code of #
                 break;
        default: keypad_button = button_number;
                 break;
    }
    return keypad_button;
}
```

4*3

IEEE CUSB

File.c

```c
#elif (KEYPAD_NUM_COLS == 4)

/*
 * Description :
 * Update the keypad pressed button value with the correct one in keypad 4x4 shape
 */
static uint8 KEYPAD_4x4_adjustKeyNumber(uint8 button_number)
{
    uint8 keypad_button = 0;
    switch(button_number)
    {
        case 1: keypad_button = 7;
                break;
        case 2: keypad_button = 8;
                break;
        case 3: keypad_button = 9;
                break;
        case 4: keypad_button = '%'; // ASCII Code of %
                break;
        case 5: keypad_button = 4;
                break;
        case 6: keypad_button = 5;
                break;
        case 7: keypad_button = 6;
                break;
        case 8: keypad_button = '*'; /* ASCII Code of '*' */
                break;
        case 9: keypad_button = 1;
                break;
        case 10: keypad_button = 2;
                break;
        case 11: keypad_button = 3;
                break;
        case 12: keypad_button = '-'; /* ASCII Code of '-' */
                break;
        case 13: keypad_button = 13;  /* ASCII of Enter */
                break;
        case 14: keypad_button = 0;
                break;
        case 15: keypad_button = '='; /* ASCII Code of '=' */
                break;
        case 16: keypad_button = '+'; /* ASCII Code of '+' */
                break;
        default: keypad_button = button_number;
                break;
    }
    return keypad_button;
}

#endif
```

4*4

Questions

Break

LCD