

Nom i Cognoms: _____ Una possible solució _____

- 1) (1 p.) Si tenim un conversor A/D de 10 bits, amb $V_{ref-} = 1V$ i $V_{ref+} = 4,2V$, calculeu la tensió d'entrada corresponent a un resultat de conversió igual a 295, sabent que la freqüència de mostreig és de 44KHz, i la del micro 8MHz. JUSTIFICA EL RESULTAT

$$(4,2V - 1V) / 1024 \text{ nivells} = 3,125mV/\text{nivell}$$

$$V_i = 1V + 3,125 * 295 = 1,922V$$

$$V_i = \underline{\quad 1,922V \quad}$$

- 2) (1 p.) Un micro amb un oscil·lador molt poc precís fa de màster en una comunicació SPI. Degut a aquesta imprecisió, la freqüència del senyal de rellotge SCLK té una certa indeterminació. Concretament $f_{SCLK} = 100KHz \pm 10\%$.

Si enviem trames de 256KB, quantifiquem quin percentatge de bits rebrà incorrectament el dispositiu slave per causa d'aquesta indeterminació.

0%

Al tractar-se d'una comunicació síncrona, el slave rep el senyal SCLK del master. Per tant llegirà les dades de forma correcta encara que no sigui a una freqüència constant.

$$\varepsilon = \underline{\quad 0 \quad} \%$$

- 3) (1 p.) Justifica quant de temps es trigaria com a mínim per transmetre 100KB de dades utilitzant una UART configurada a 9600 bps? Considereu: 8 bits de dades, sense paritat, 1 bit de Stop, i parell diferencial.

$$100.000 \text{ bytes} * (8 \text{ bits dades} + 1 \text{ start bit} + 1 \text{ stop bit}) = 1 \text{ Mbit a transmetre}$$

$$1000000 \text{ bits} / 9600 \text{ bps} = 104,16 \text{ segs}$$

$$T = \underline{\quad 104,16 \text{ seg.} \quad}$$

- 4) (2 p.) Quants cicles T_{AD} necessita el ADC del PIC18F4550 per a fer una conversió de 10 bits? i quants comparadors necessita la seva arquitectura? i si es tractés d'un ADC flash? Respon en la taula següent, i justifica els valors.

	# cicles	# comparadors
ADC PIC	12	1
ADC flash	1	1024

Nom i Cognoms: _____

5) Es vol utilitzar un conjunt GLCD – Panell tàctil similar al que heu utilitzat al laboratori per implementar un teclat tàctil hexadecimal. Amb aquesta finalitat es visualitzen els caràcters de '0' a 'F' disposats de forma matricial al GLCD (figura 1).

Recordeu que les funcions (entre d'altres) que permeten interaccionar amb GLCD + panell tàctil són:

```
byte tpPressDetect(void);
byte tpGetX(void);
byte tpGetY(void);
void putc(byte page, byte column, char c);
```

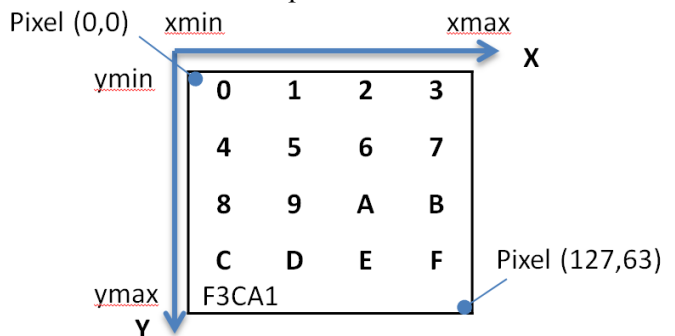


Fig.1. Teclat tàctil Hexadecimal

a) (2,5 pts) Escriure el codi de la rutina **byte tpread(void)** que donades les lectures del panell tàctil retorni si hi ha (o no hi ha) activació del panell tàctil; i en dues variables globals –PixelX, PixelY- indiqui sobre quin píxel del GLCD es realitza la pressió. Els valors màxims i mínims obtinguts en la calibració del panell tàctil estan disponibles a les variables globals xmax, xmin, ymax, ymin.

Utilitzeu la definició dels eixos segons s'indica a la figura 1.

b) (2,5 pts) Escriure el codi de la rutina **byte writehex(byte PixelX, byte PixelY)** que a partir de les pulsacions sobre el panell tàctil –valors PixelX, PixelY- mostri, de forma consecutiva a la pàgina 7 del GLCD, els caràcters ASCII premuts. Retorna el valor entre 0 i 15 corresponent al caràcter premut.

```
// Solution for Section a
// PixelX and PixelY are global variables
byte tpread(void) {
    byte adcr, tppress;
    int temp;
    if (tpPressDetect()) { // Touch panel is PRESSED
        tppress=1;
        adcr= tpGetX(); // Read X value from ADC
        temp=127*(adcr-xmin); // Convert to pixel position
        PixelX= (byte) (temp/(xmax-xmin));
        adcr= tpGetY(); // Read Y value from ADC
        temp=63*(adcr-ymin); // Convert to pixel position
        PixelY= (byte) (temp/(ymax-ymin));
    }
    else {
        tppress=0;
        PixelX= 255; PixelY=255;
    };
    return(tppress);
}
```

```
// Solution for Section b
// Characters -ASCII codes- for Hexadecimal values
char Hextable[16]= {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9',
                    'A', 'B', 'C', 'D', 'E', 'F'};
byte column= 0; // Column position for writing char

// Hextable[] and column are global variables
byte writehex(byte x, byte y) {
    byte cell;
    char hexval;
    if ( (x>127) || (y>63) ) return(0xFF);
    cell= (x/32)+4*(y/16); // Index of the tactil keyboard matrix
    hexval= Hextable[cell];
    putc(7, column, hexval); // Write HEX characer at page 7
    column= column+6; // Move column for writing next value
    if (column>121) column=0; // 0 when reaching last position
    return(cell);
}
```