# Memòria de pràctiques de DSBM

David Álvarez i Jordi Altayó

Desembre de 2016

# Índex

**1 Introducció** **2**

**2 Pràctiques bàsiques** **3**
2.1 P1 - Introducció a *Eclipse* . . . . . 3
2.2 P2 - Accés al LCD . . . . . 3
2.3 P3 - Acceleròmetre 3D . . . . . 3
2.4 P4 - Interrupcions i mesures de temporització . . . . . 3

**3 Pràctiques complementàries** **4**
3.1 C1 - Lectura del teclat . . . . . 4
3.2 C2 - Convertidor A/D . . . . . 6
3.3 C3 - Encoder . . . . . 6

**4 Pràctiques amb RTOS** **7**
4.1 R1 - Introducció a ChibitOS/RT . . . . . 7
4.2 R2 - Compartició d'elements comuns . . . . . 7

# Capítol 1

# Introducció

L'objectiu d'aquesta memòria de pràctiques no és només sintetitzar tot el treball realitzat durant aquest quadrimestre sinó també una eina autoexplicativa d'ell mateix.
És per això que en les explicacions realitzades en aquest document es dóna per suposat que el lector té un coneixement ferm sobre la matèria en qüestió però no disposa d'accés als manuals de practiques.

# Capítol 2

# Pràctiques bàsiques

## 2.1 P1 - Introducció a *Eclipse*

## 2.2 P2 - Accés al LCD

## 2.3 P3 - Acceleròmetre 3D

## 2.4 P4 - Interrupcions i mesures de temporització

# Capítol 3

# Pràctiques complementàries

## 3.1 C1 - Lectura del teclat

Configurar el teclat

```
void initKeyboard() {
    // Configura com a sortida en drenador obert els ports PD0...PD3
    KEY_PORT->MODER &= ~0b11111111;
    KEY_PORT->MODER |= 0b01010101;
    KEY_PORT->OTYPER |= 0b1111;

    // Configura com a entrada amb pull up els ports PD6...PD9
    KEY_PORT->MODER &= ~(0b11111111 << 2 * KEY_COL1_PAD);
    KEY_PORT->PUPDR &= ~(0b11111111 << 2 * KEY_COL1_PAD);
    KEY_PORT->PUPDR |= 0b01010101 << 2 * KEY_COL1_PAD;

    (KEY_PORT->BSRR.H.clear) = KEY_ROW1_BIT;
    (KEY_PORT->BSRR.H.clear) = KEY_ROW2_BIT;
    (KEY_PORT->BSRR.H.clear) = KEY_ROW3_BIT;
    (KEY_PORT->BSRR.H.clear) = KEY_ROW4_BIT;
}
```

Llegir de teclat:

```
int32_t readKeyboard() {
    // Posa PD0 a '0' i PD1, PD2, PD3 a '1' (flotant) per explorar la primera
    // filera

    (KEY_PORT->BSRR.H.clear) = KEY_ROW1_BIT;
    (KEY_PORT->BSRR.H.set) = KEY_ROW2_BIT;
    (KEY_PORT->BSRR.H.set) = KEY_ROW3_BIT;
    (KEY_PORT->BSRR.H.set) = KEY_ROW4_BIT;
    DELAY_US(10);

    if((~(KEY_PORT->IDR) & KEY_COL1_BIT) == KEY_COL1_BIT) {
        return (int32_t)0; // '1' detectat
    }else if((~(KEY_PORT->IDR) & KEY_COL2_BIT) == KEY_COL2_BIT) {
        return (int32_t)1; // '2' detectat
    }else if((~(KEY_PORT->IDR) & KEY_COL3_BIT) == KEY_COL3_BIT) {
        return (int32_t)2; // '3' detectat
    }else if((~(KEY_PORT->IDR) & KEY_COL4_BIT) == KEY_COL4_BIT) {
        return (int32_t)3; // 'A' detectat
    }

    // Posa PD1 a '0' i PD0, PD2, PD3 a '1' (flotant) per explorar la segona
    // filera
```

```
    (KEY_PORT->BSRR.H.set) = KEY_ROW1_BIT;
    (KEY_PORT->BSRR.H.clear) = KEY_ROW2_BIT;
    (KEY_PORT->BSRR.H.set) = KEY_ROW3_BIT;
    (KEY_PORT->BSRR.H.set) = KEY_ROW4_BIT;
    DELAY_US(10);

    if((~(KEY_PORT->IDR) & KEY_COL1_BIT) == KEY_COL1_BIT) {
        return (int32_t)4; // '4' detectat
    }else if((~(KEY_PORT->IDR) & KEY_COL2_BIT) == KEY_COL2_BIT) {
        return (int32_t)5; // '5' detectat
    }else if((~(KEY_PORT->IDR) & KEY_COL3_BIT) == KEY_COL3_BIT) {
        return (int32_t)6; // '6' detectat
    }else if((~(KEY_PORT->IDR) & KEY_COL4_BIT) == KEY_COL4_BIT) {
        return (int32_t)7; // 'B' detectat
    }

    // Posa PD2 a '0' i PD0, PD1, PD3 a '1' (flotant) per explorar la tercera
    // filera
    (KEY_PORT->BSRR.H.set) = KEY_ROW1_BIT;
    (KEY_PORT->BSRR.H.set) = KEY_ROW2_BIT;
    (KEY_PORT->BSRR.H.clear) = KEY_ROW3_BIT;
    (KEY_PORT->BSRR.H.set) = KEY_ROW4_BIT;
    DELAY_US(10);

    if((~(KEY_PORT->IDR) & KEY_COL1_BIT) == KEY_COL1_BIT) {
        return (int32_t)8; // '7' detectat
    }else if((~(KEY_PORT->IDR) & KEY_COL2_BIT) == KEY_COL2_BIT) {
        return (int32_t)9; // '8' detectat
    }else if((~(KEY_PORT->IDR) & KEY_COL3_BIT) == KEY_COL3_BIT) {
        return (int32_t)10; // '9' detectat
    }else if((~(KEY_PORT->IDR) & KEY_COL4_BIT) == KEY_COL4_BIT) {
        return (int32_t)11; // 'C' detectat
    }

    // Posa PD3 a '0' i PD0, PD1, PD2 a '1' (flotant) per explorar la quarta
    // filera
    (KEY_PORT->BSRR.H.set) = KEY_ROW1_BIT;
    (KEY_PORT->BSRR.H.set) = KEY_ROW2_BIT;
    (KEY_PORT->BSRR.H.set) = KEY_ROW3_BIT;
    (KEY_PORT->BSRR.H.clear) = KEY_ROW4_BIT;
    DELAY_US(10);

    if((~(KEY_PORT->IDR) & KEY_COL1_BIT) == KEY_COL1_BIT) {
        return (int32_t)12; // '*' detectat
    }else if((~(KEY_PORT->IDR) & KEY_COL2_BIT) == KEY_COL2_BIT) {
        return (int32_t)13; // '0' detectat
    }else if((~(KEY_PORT->IDR) & KEY_COL3_BIT) == KEY_COL3_BIT) {
        return (int32_t)14; // '#' detectat
    }else if((~(KEY_PORT->IDR) & KEY_COL4_BIT) == 0) {
        return (int32_t)15; // 'D' detectat
    }

    // Si no s'ha polsat cap tecla retorna 32
    return (int32_t)32;
}
```

### 3.2 C2 - Convertidor A/D

### 3.3 C3 - Encoder

# Capítol 4

# Pràctiques amb RTOS

## 4.1 R1 - Introducció a ChibitOS/RT

## 4.2 R2 - Compartició d'elements comuns