

Pràctiques de Sistemes Digitals i Microprocessadors
Curs 2022-2023

Pràctica 2B

LSControlTower

Alumnes	Login	Nom
	alex.cano	Alex Cano Gallego
	biel.carpi	Biel Carpi Closa

Entrega	Placa	Memòria	Nota

Data	19/05/2023
------	------------

Còpia per als alumnes

Pràctiques de Sistemes Digitals i Microprocessadors
Curs 2022-2023

Pràctica 2B

LSControlTower

Alumnes	Login	Nom
	alex.cano	Alex Cano Gallego
	biel.carpi	Biel Carpi Closa

Entrega	Placa	Memòria	Nota

Data	19/05/2023
------	------------

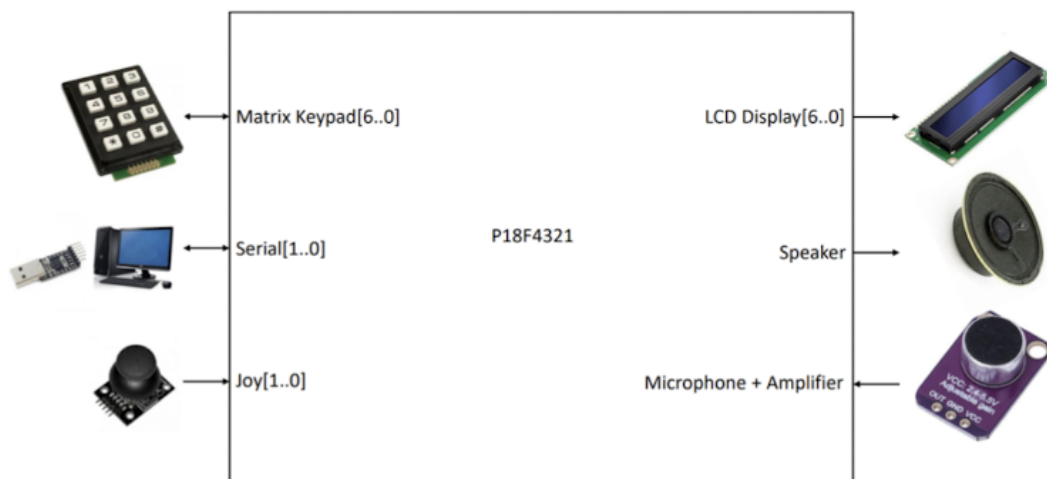
INDEX

1. RESUM DE L'ENUNCIAT	4
2. DISSENY	6
3. ESQUEMA ELÈCTRIC.....	7
4. DIAGRAMA DE TADS.....	8
5. DIAGRAMES DELS MOTORS.....	10
5.1. MOTOR MENU	10
5.2. MOTOR COUNTER.....	11
5.3. MOTOR JOYSTICK	12
5.4. MOTOR KEYPAD.....	13
5.5. MOTOR LCD	14
5.6. MOTOR MICRO	15
5.7. MOTOR MAQUESINA	16
5.8. MOTOR MENU RECORDINGS.....	17
5.9. MOTOR PORT	18
.....	18
5.10. MOTOR SAVE	19
5.11. MOTOR BUZZER.....	20
6. CONCLUSIONS	21
7. PLANIFICACIÓ	22

1. Resum de l'enunciat

Per aquesta pràctica de sistemes digitals i microprocessadors se'ns demana utilitzar el microcontrolador PIC18F4321 programat amb llenguatge C, fent servir la metodologia de sistemes cooperatius. Un cop haver construït el vaixell en la fase anterior, aquest cop se'ns demana construir la interfície entre el vaixell i la torre de control.

El sistema a implementar inclou components com un joystick, altaveu, micròfon amb amplificador, teclat matriu 3x4 i una pantalla LCD de 2 files i 16 columnes. A més, s'estableix una connexió sèrie amb un dispositiu extern que executa una aplicació de Java per proporcionar una interfície gràfica al sistema.



Quan s'encén el microcontrolador, es mostra al LCD un menú per seleccionar la torre de control amb què es vol iniciar la comunicació. L'usuari introdueix el codi del port amb el teclat matriu, i aquest accedeix després al menú principal. El codi del port consta de tres lletres, per exemple Barcelona (BCN), Madrid (MAD), Londres (LDN), etc.

Per seleccionar el port, es fa servir el teclat matriu en format SMS amb efectes sonors. Cada tecla genera un so de diferent freqüència al ser premuda. El format SMS implica prémer ràpidament la tecla corresponent diverses vegades per seleccionar la lletra desitjada. Si es prem una altra tecla o es passa més d'un segon sense prémer, es considera el caràcter seleccionat. En prémer la mateixa tecla diverses vegades en menys d'un segon, les lletres associades a aquesta tecla s'incrementen cíclicament.

S'utilitza el joystick per navegar pel menú principal que es mostra al LCD després de seleccionar la torre de control. L'usuari pot triar entre diferents opcions, com ara iniciar un enregistrament de veu, reproduir enregistraments, modificar l'hora actual, mostrar el temps transcorregut i finalitzar la comunicació.

En el mode Start Recording, l'usuari pot fer enregistraments de veu mitjançant el micròfon i enviar-los a la torre de control utilitzant una interfície Java. S'adquireixen mostres analògiques a una freqüència de mostreig de 4KHz i s'envien a través del canal sèrie cap a la interfície Java. Cada enregistrament té una durada de 8 segons (i, per tant, 32768 mostres). Després de finalitzar l'enregistrament, es genera una melodia fent servir l'altaveu.

En el mode Play Recordings, es mostra un llistat dels enregistraments realitzats i se'n selecciona un per reproduir-lo. La interfície Java respon a les ordres enviades des del microcontrolador i reproduceix l'enregistrament a través dels altaveus de l'ordinador.

Per modificar l'hora actual, l'usuari pot introduir el temps en format MM:SS utilitzant el teclat matriu. El temps es desa en prémer la tecla (#).

En la manera de mostrar el temps transcorregut, es mostra en temps real els minuts i els segons que han passat des de l'inici de la comunicació amb el port.

Finalment, en el mode End communication es mostra un missatge de comiat i es torna al menú d'accés per establir comunicació amb una altra torre de control.

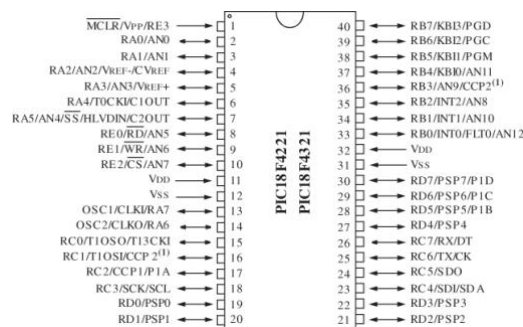
2. Disseny

A l'hora de plantejar el disseny hem de decidir quines seran les configuracions del nostre microcontrolador. Donat que disposem d'un cristall de 10MHz extern, hem optat per habilitar HSPLL i aconseguir una freqüència d'oscil·lació de 40MHz (temps d'instrucció de 100 nanosegons). Com que no tenim restriccions d'energia, d'aquesta manera ens assegurem que el programa corri el més fluid possible.

Un dels aspectes clau per al disseny del sistema és escollir quin serà el temps d'interrupció del perifèric TIMER0. És evident que la restricció més important a l'hora de decidir aquest valor és el número de mostres que hem d'enviar per segon (4KHz mostres). Per poder comptar 4000 vegades dins d'un segon, utilitzarem un temps d'interrupció del TIMER0 de 250us (i, així, cada cop que ens interrompi sabrem que ja podem enviar la pròxima mostra).

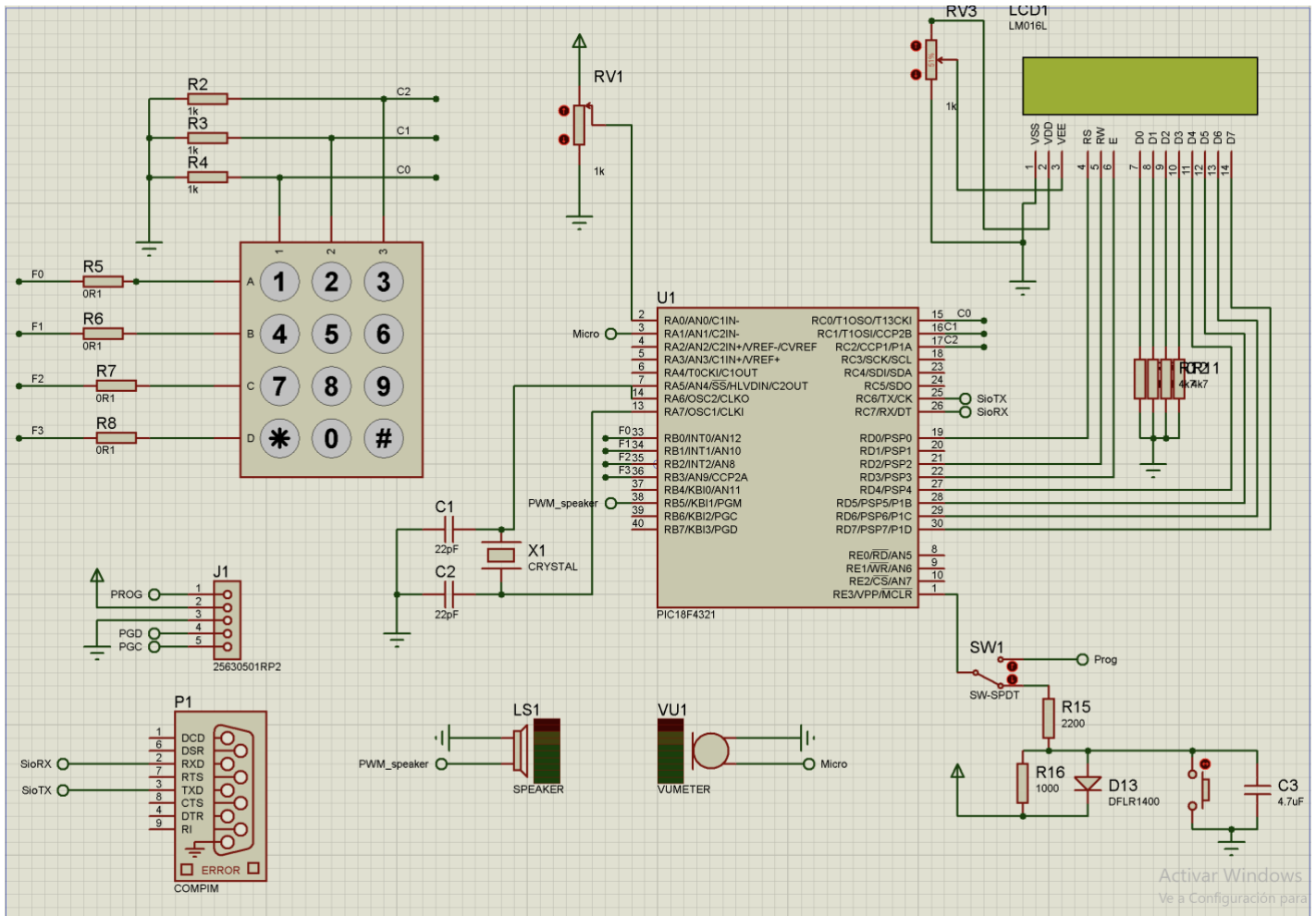
Un altre aspecte fonamental és el baud rate al que anirà la transmissió pel canal sèrie entre el microcontrolador i l'ordinador. Normalment utilitzem baud rates de 9600, suficients per enviar i rebre 9600 bits per segon. En el nostre cas, fent un petit càlcul, necessitem enviar 4K mostres per segon, on cada mostra ocupa 1 Byte (+2 bits extres per a la transmissió). Per tant, hem d'enviar 40K bits per segon. Així doncs, necessitem un baud rate més gran que 40K bits/s. Hem triat 57600 de baud rate, que és el rate estàndard més baix que compleix la nostra condició.

Per una altra part, hem hagut de decidir a on posàvem els pins de sortida per connectar els perifèrics necessaris. Podria semblar que els pins es poden triar a l'atzar, però és molt important fer-ho de la millor manera ja que ens podem evitar moltes línies de codi. Per connectar la pantalla LCD hem necessitat 7 pins de tal manera que he decidit connectar al PORTD de manera agrupada. En quant als perifèrics analògics (Joystick Eix Y i Micròfon), les hem connectat al RA0 i RA1 per tenir tots els altres ports A disponibles com a pins digitals.

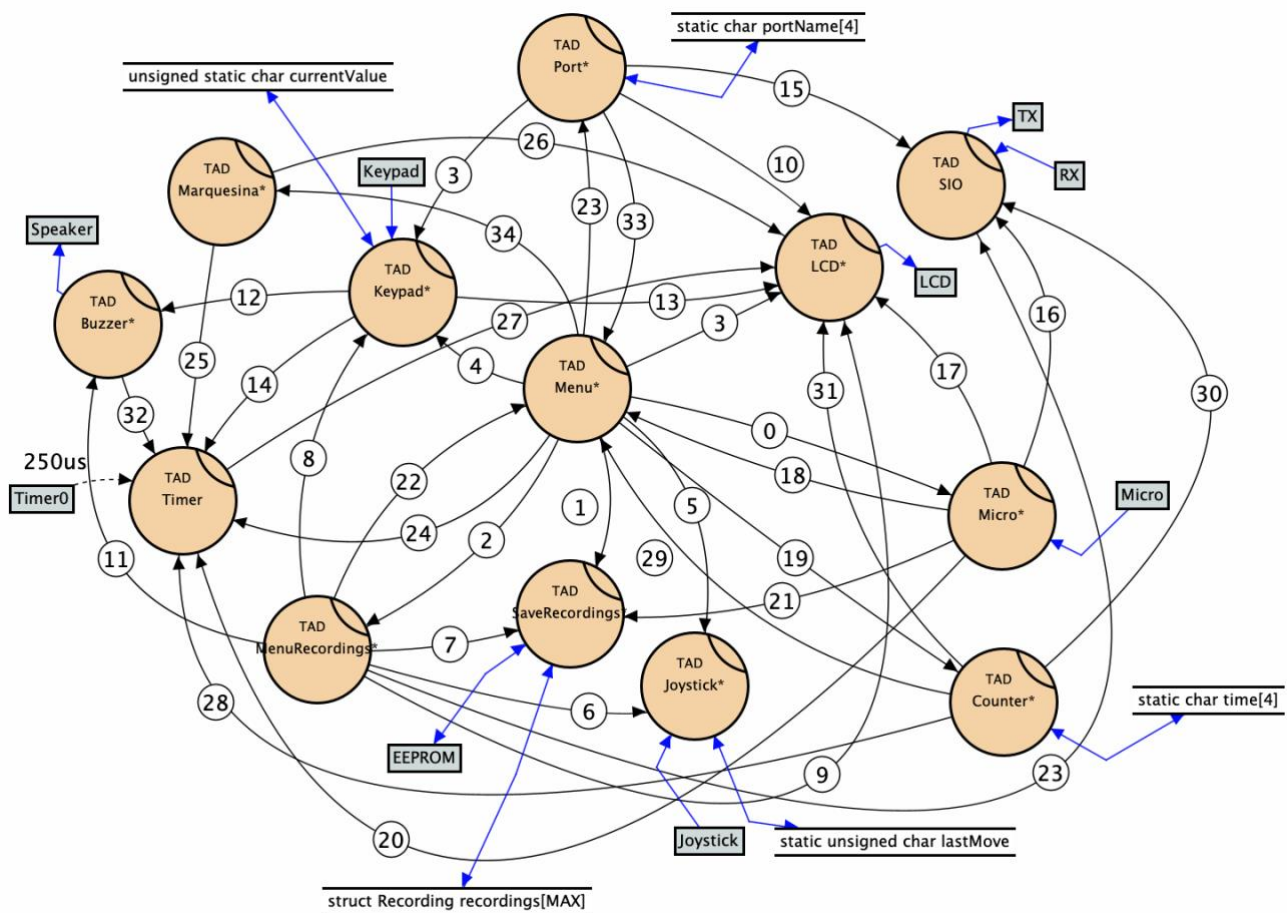


En quant a nivell software, s'ha utilitzat el paradigma cooperatiu per aprofitar al màxim el rendiment del microcontrolador. S'ha implementat mitjançant TADs (*Abstract Data Types*), que són mòduls de codi independents i interconnectats entre si (explicació en profunditat en l'apartat del diagram de TADs).

3. Esquema elèctric



4. Diagrama de TADs



Per al bon desenvolupament i modulació de la pràctica, hem definit 13 TADs (dels quals 11 implementen un motor).

Per una banda, cada perifèric està relacionat amb un TAD que s'encarrega de gestionar-lo (enviar o rebre informació per aquest). També tenim TADs encarregats de gestionar funcions específiques (com la marquesina, o bé cada menú). Tots els TADs estan interconnectats (descriu al *diccionari.c*) i són mòduls amb funcionalitats independents.

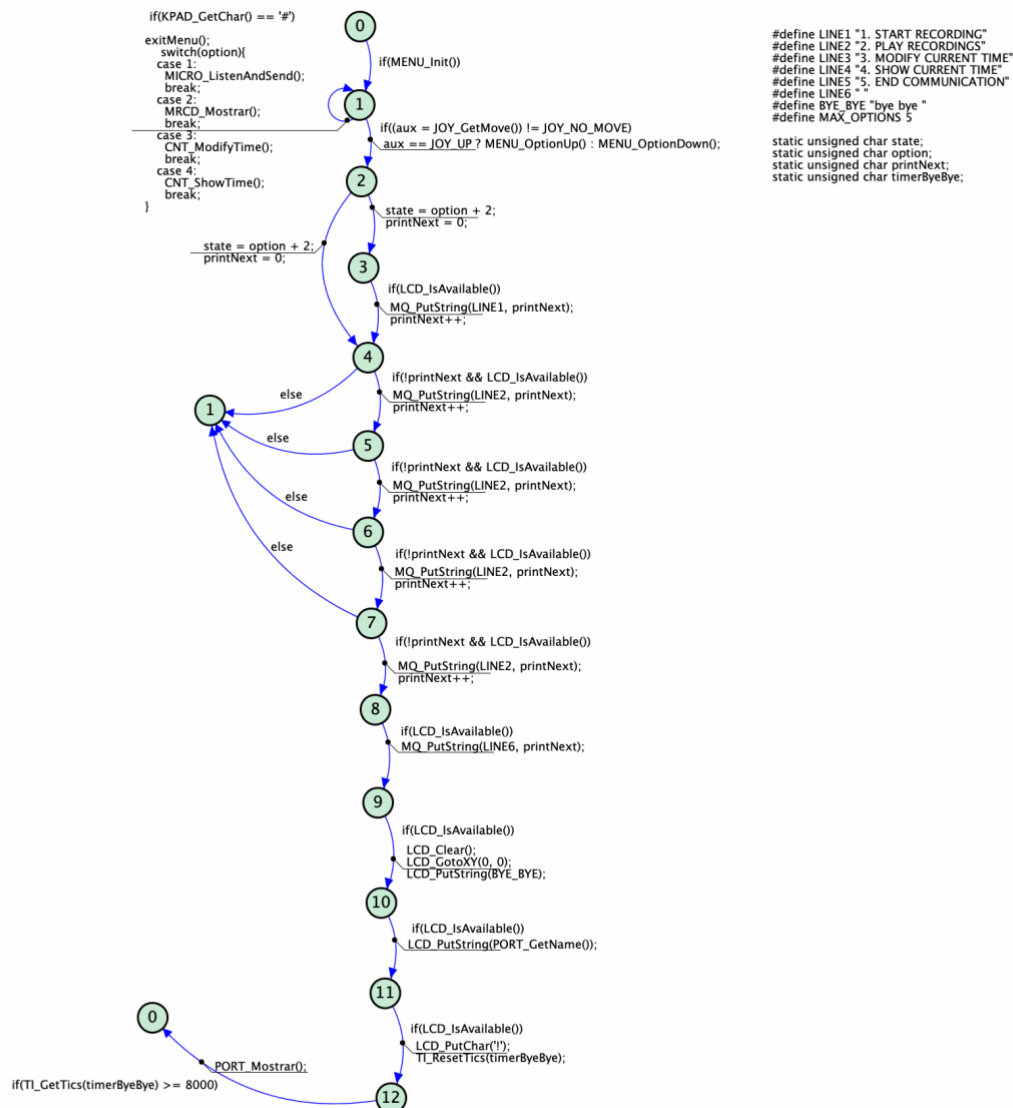
A continuació, fem una descripció de cada TAD i la seva funcionalitat. En els apartats a continuació, es pot trobar un diagrama detallat de cada motor i la seva corresponent explicació.

- **Timer:** Encarregat de gestionar i guardar timers virtuals que poden ser utilitzats per qualsevol mòdul amb les respectives funcions.
- **Buzzer*:** Encarregat de gestionar el perifèric del buzzer. Li enviarà un PWM de diferent període depenent de la nota que se li hagi demanat tocar.

- **Keypad*:** Gestiona el teclat matriu, guardant sempre l'última tecla pressionada i tenint-la en una variable per si qualsevol altre mòdul li consulta si s'ha pressionat alguna tecla.
- **LCD*:** Gestiona i abstrau la funcionalitat del LCD. S'ha modificat per tal d'afegir una funció `PrintString()` que sigui cooperativa (és a dir, que es faci amb motor).
- **Marquesina*:** TAD que s'intercomunica amb el LCD i gestiona el text en format marquesina sempre i quant aquest no hi càpiga per la LCD actual (de 16 columnes).
- **SIO:** Encarregat de proporcionar les funcions per enviar i rebre informació a través del canal sèrie del microcontrolador.
- **Joystick*:** TAD que escolta el perifèric del Joystick (només l'eix Y), mostrejant periòdicament el seu valor per tal d'averiguar si s'ha tirat amunt o avall (en cas afirmatiu, guardant un valor a una variable per tal que els altres mòduls puguin preguntar si hi ha hagut algun moviment de joystick).
- **SaveRecordings*:** Encarregat de descarregar tots els *recordings* de la EEPROM, i guardar-ne un cada vegada que se'n hagi enregistrat un de nou.
- **Port*:** TAD que mostra el menú del port, interactuant amb els altres respectius mòduls per a aconseguir la funcionalitat del menú.
- **Menu*:** Encarregat de mostrar el menú principal, interactuant amb els altres respectius mòduls per a aconseguir la funcionalitat del menú.
- **Counter*:** Incrementa i gestiona el temporitzador global de temps (MM:SS). També inclou el menú de modificar o ensenyar el counter, interactuant amb els altres respectius mòduls per a aconseguir les funcionalitat dels dos menus.
- **MenuRecordings*:** TAD que mostra el menú amb tots els recordings, interactuant amb els altres respectius mòduls per a aconseguir la funcionalitat del menú.
- **Micro*:** Gestiona l'enviament de totes les mostres del micròfon a través de la SIO cap a l'ordinador receptor, a una velocitat de 4KHz.

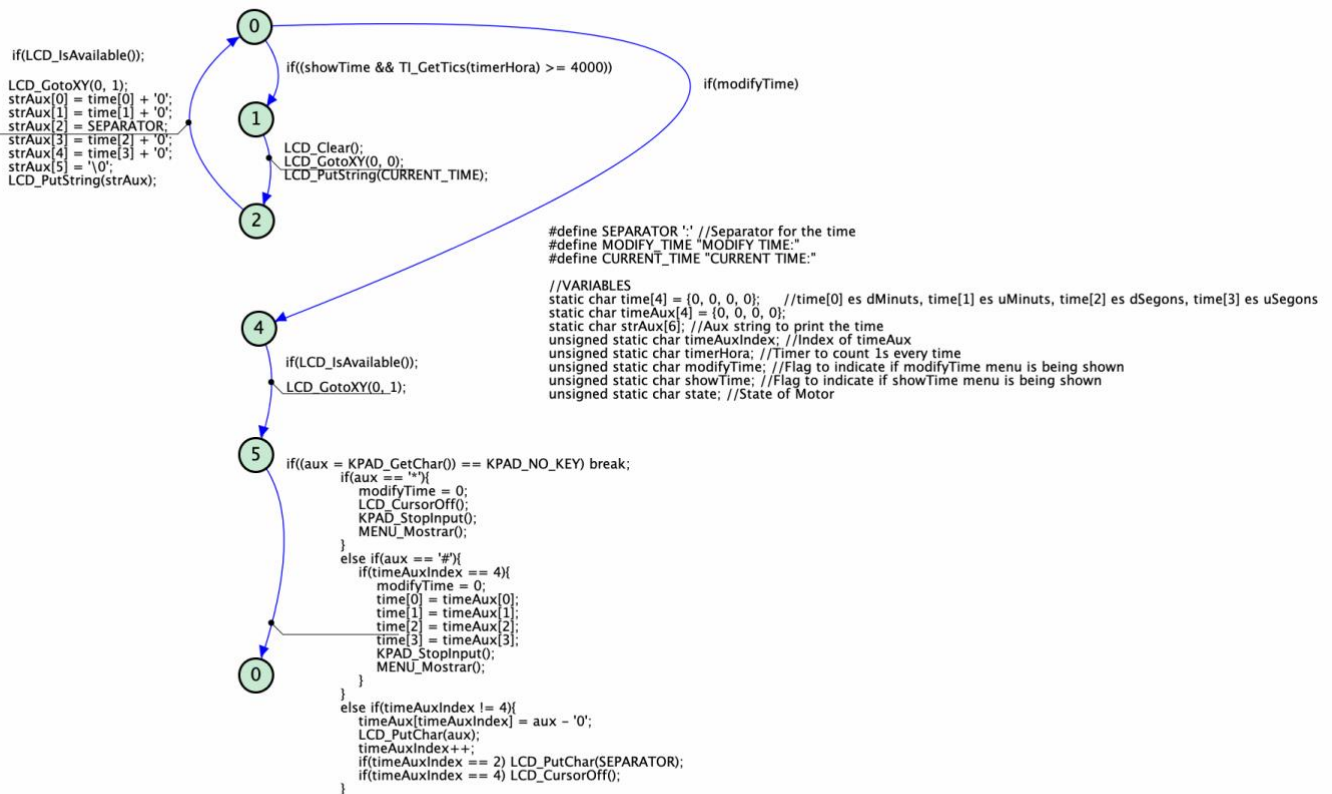
5. Diagrames dels motors

5.1. Motor Menu



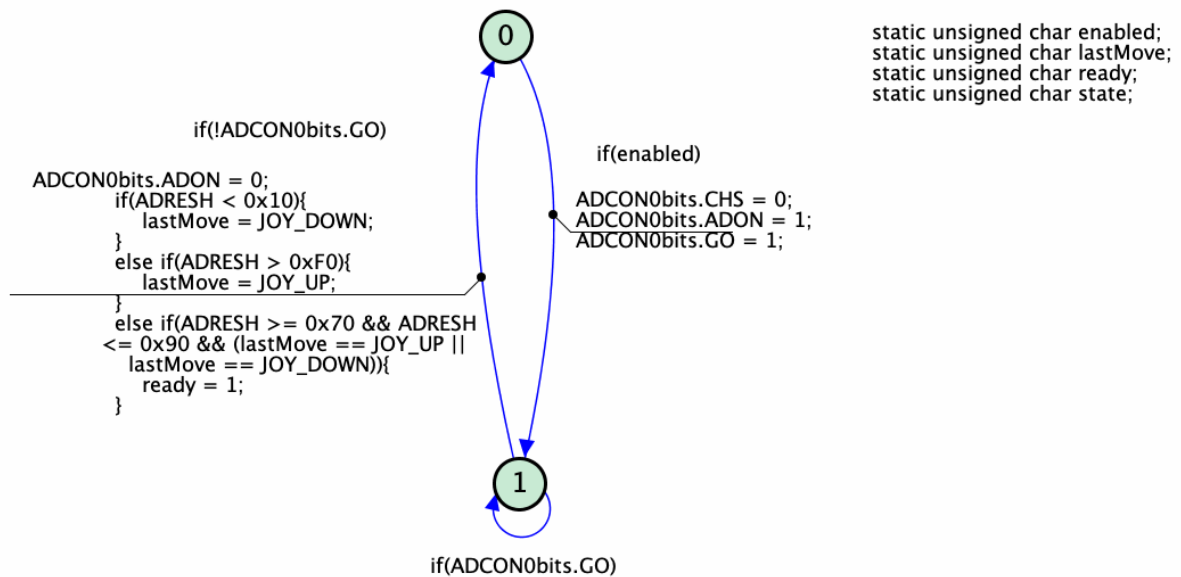
El motor del menu s'encarrega d'escoltar si es prem el # en alguna opció (i entrar al següent menu), si el joystick s'ha mogut amunt o avall, i actualitza el menú (parlant amb el TAD marquesina) en cas que així sigui. Finalment, també et diu Bye Bye i s'espera dos segons abans de tornar al menu del port.

5.2. Motor Counter



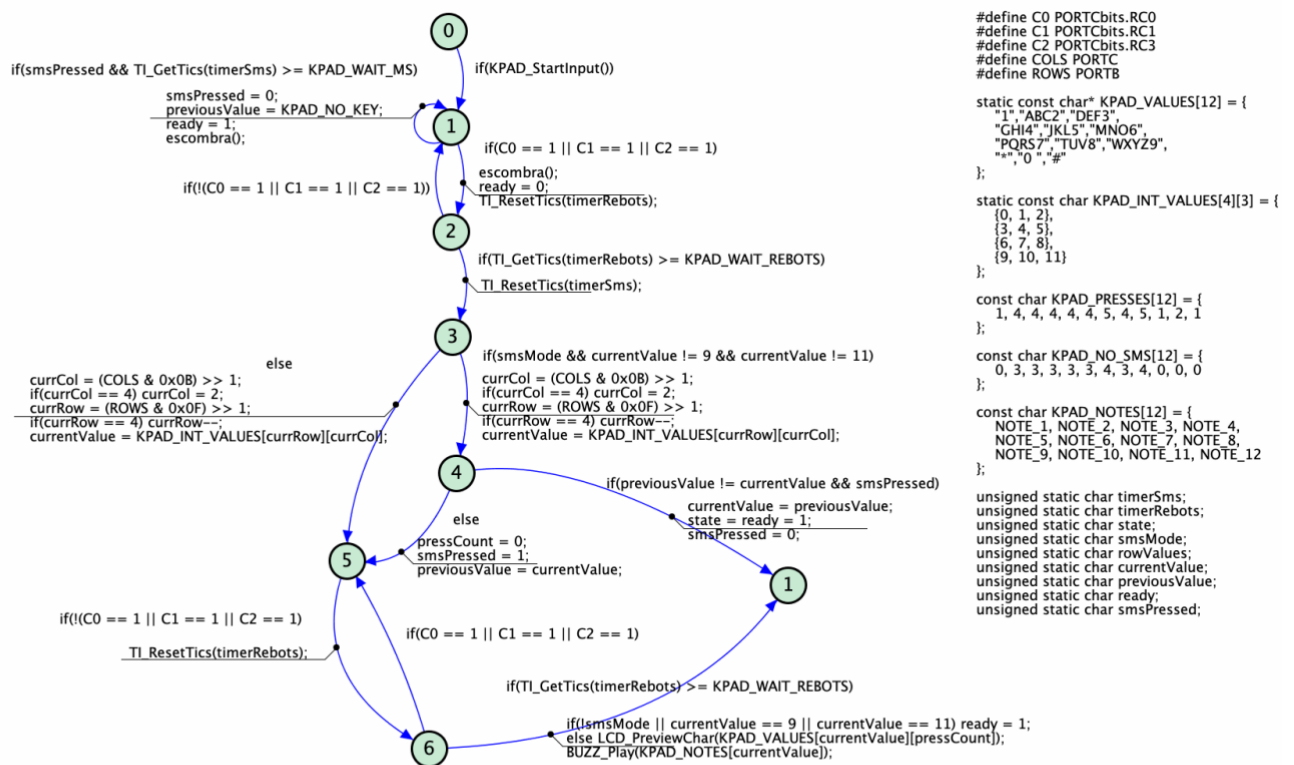
El motor del counter s'encarrega de comptar cada segon i incrementar la variable de temps quan això passi. A part del comptatge de temps, també mostra el menú de modificar el temps i el menú d'ensenyar el temps actual. En aquests dos menús, esta escoltant sempre el teclat matriu per si es prem algun número, * o # i fer la operació corresponent.

5.3. Motor Joystick



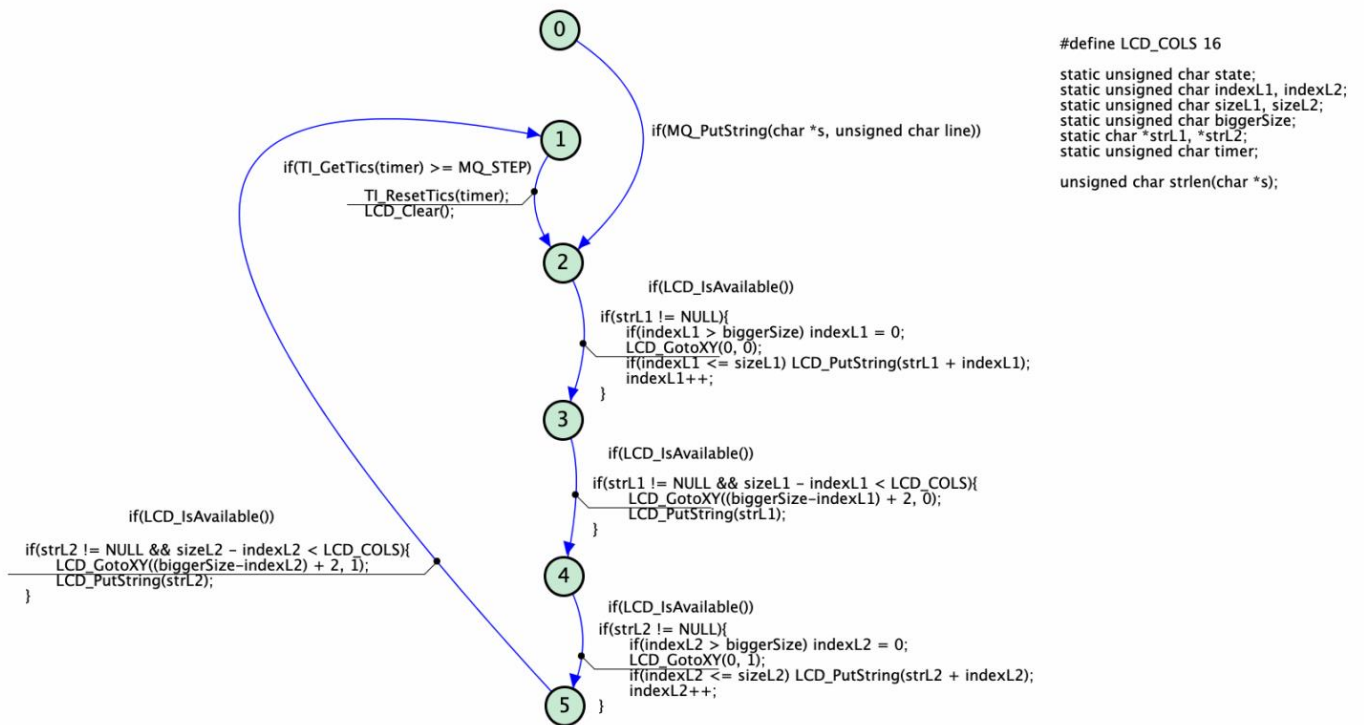
El motor del joystick és molt senzill i simplement s'encarrega d'actualitzar la variable *lastMove* en el cas que el joystick s'hagi mogut (es van prenent mostres del ADC, i si es detecta que va cap a una direcció i després torna al centre es considera un moviment vàlid).

5.4. Motor Keypad



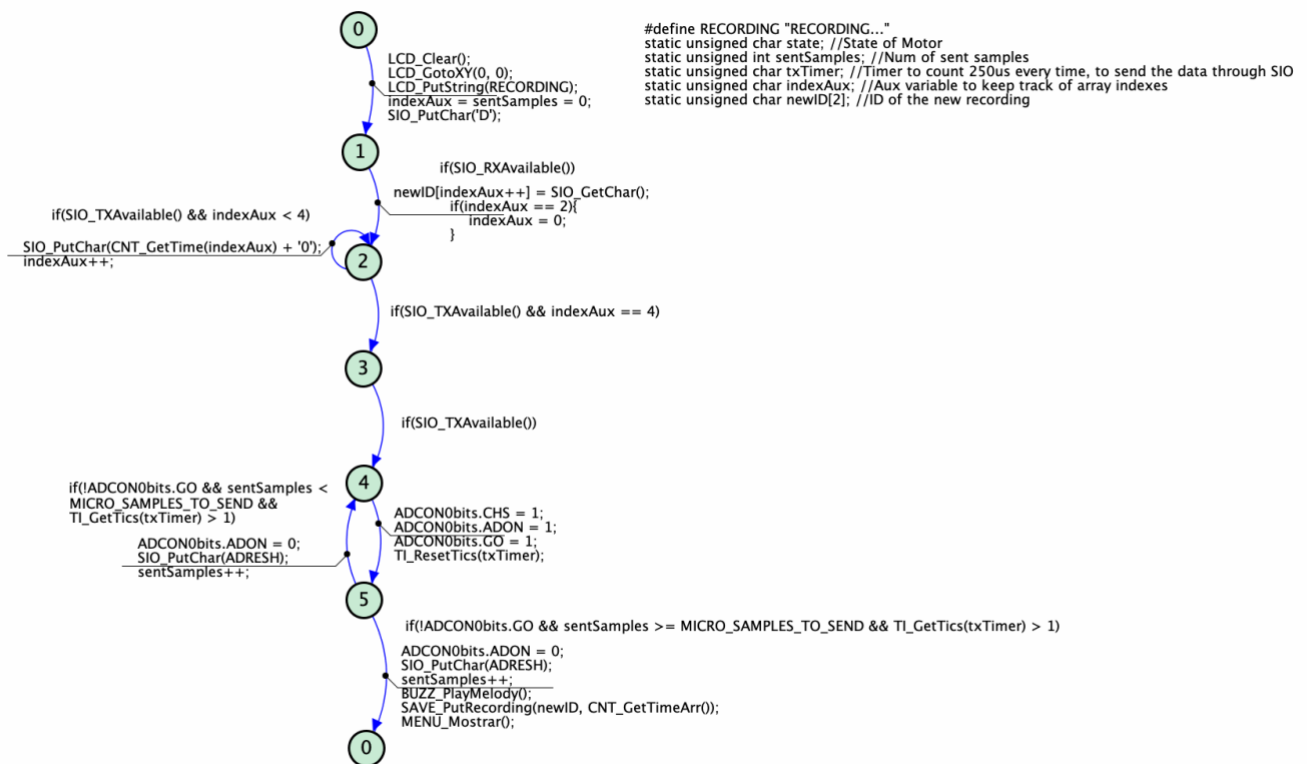
El motor del teclat matriu s'encarrega de fer l'escombrat d'aquest i detectar les pulsacions de les tecles (deixant passar un temps per als rebots un cop es prem i un cop es deixa de prémer). Si es detecta alguna pulsació, s'actualitza la variable *currentValue*, que conté la última tecla premuda. També gestiona el mode SMS, en el que conta si la nova tecla premuda és igual a l'última que s'ha premut abans d'un segon.

5.5. Motor LCD



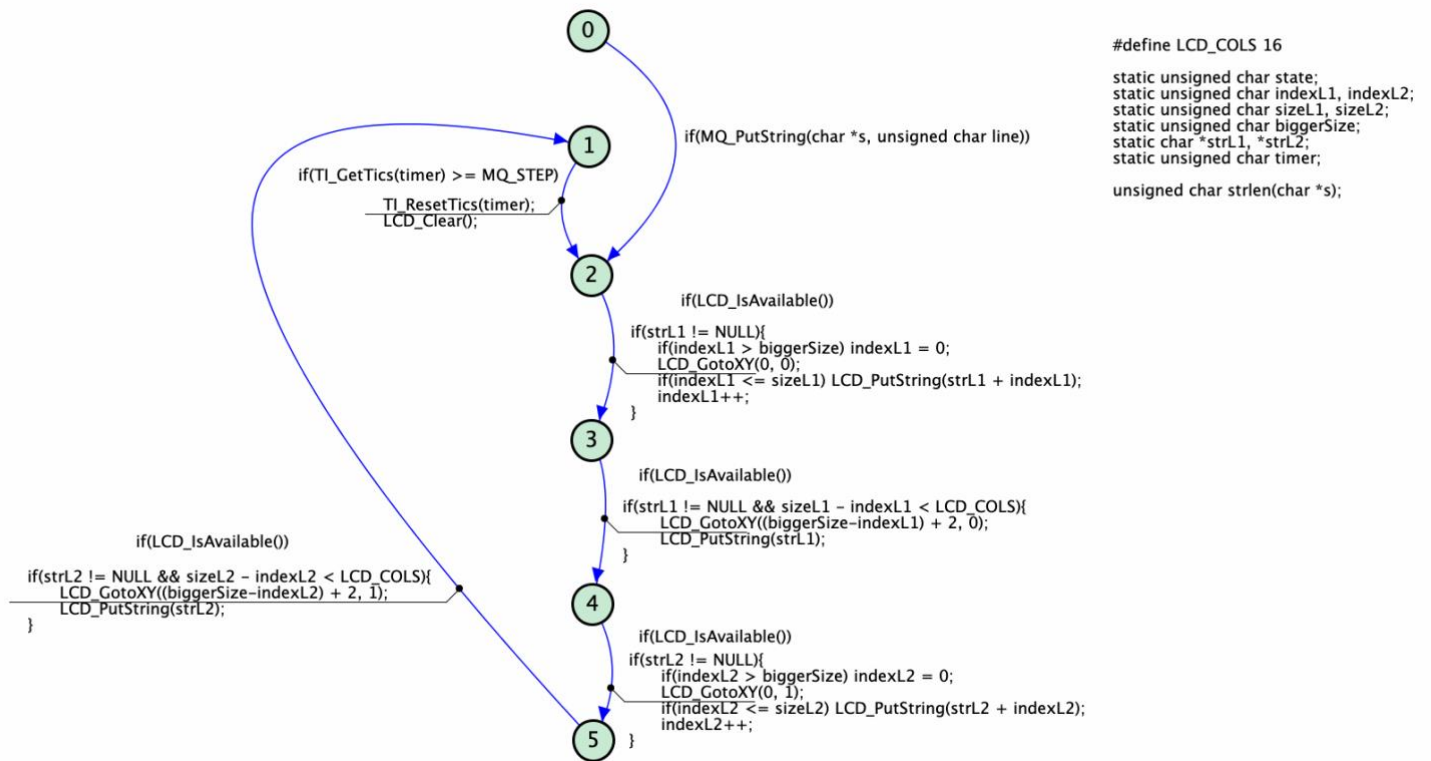
El motor del LCD és molt senzill i responsable únicament d'acabar una crida a la funció LCD_PutString(char *str). Un cop se'ns passa un string amb aquesta funció, el motor va fent PutChars fins que s'ha acabat de mostrar tot el string.

5.6. Motor Micro



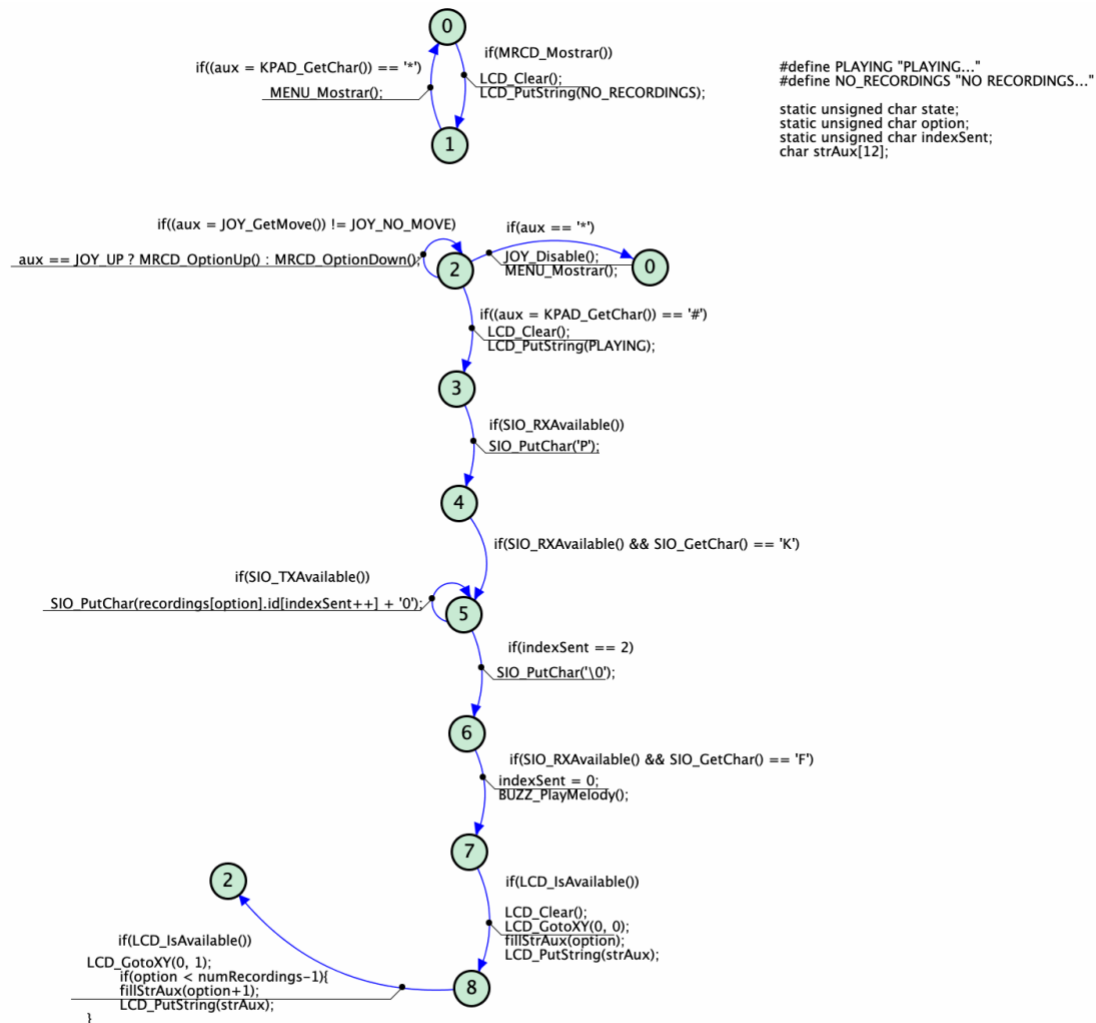
El motor del micròfon és el responsable de gestionar quan comença una gravació. S'encarrega de rebre l'ID de la nova gravació que li envia l'ordinador, i llavors procedeix a enviar el temps actual de la gravació i les 32768 mostres a 4KHz (8 segons en total). Finalment, torna al menú principal.

5.7. Motor Marquesina



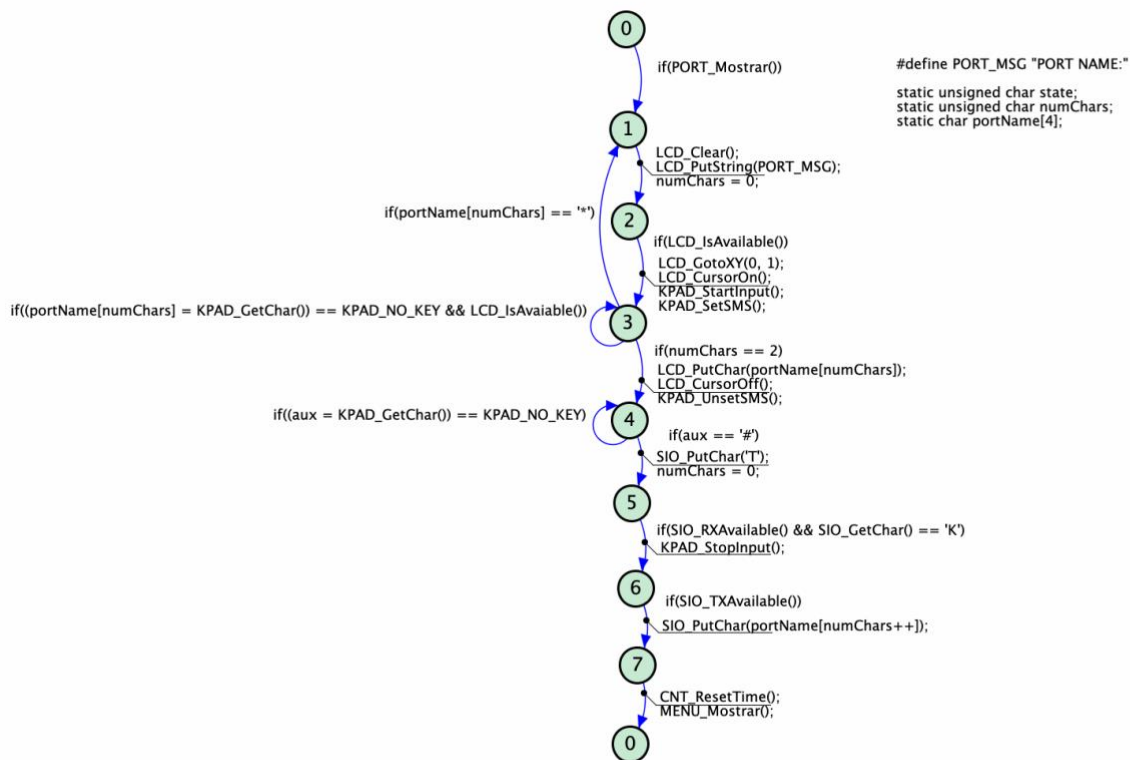
El motor de la marquesina és l'encarregat de gestionar les diferents crides que es fan a `MQ_PutString(char *str)`. Cada 1 segon s'activa aquest motor i mou els dos (o un) strings que se li han demanat de mostrar cap a l'esquerra. Un cop s'ha acabat de mostrar el string més llarg dels dos, els dos tornen a començar a mostrar-se.

5.8. Motor MenuRecordings



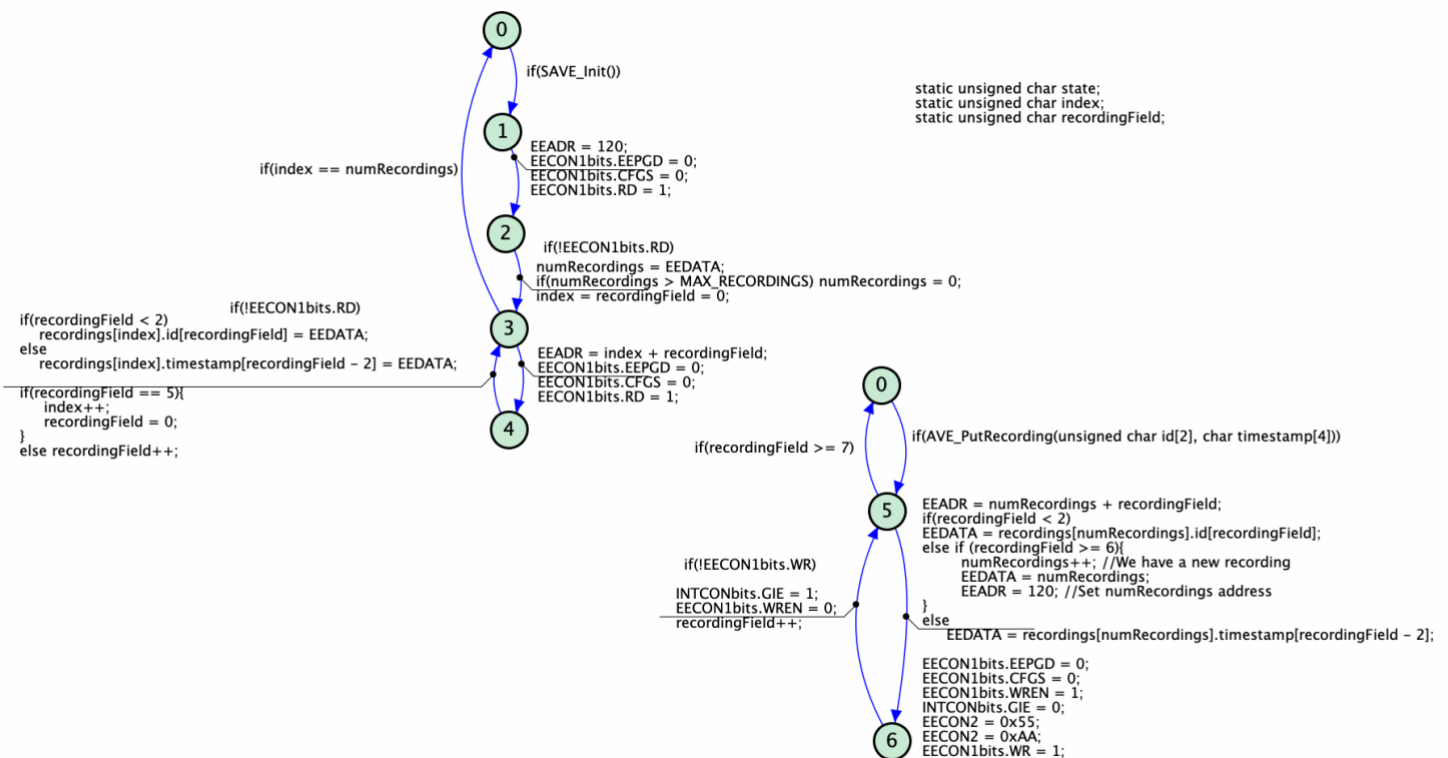
El motor del MenuRecordings gestiona el menú on es mostren tots els recordings. Molt similar al menú principal, s'encarrega d'escoltar si es prem el # en alguna opció (i enviar a l'ordinador l'ID per reproduir aquella gravació), si el joystick s'ha mogut amunt o avall, i actualitza el menú en cas que així sigui.

5.9. Motor Port



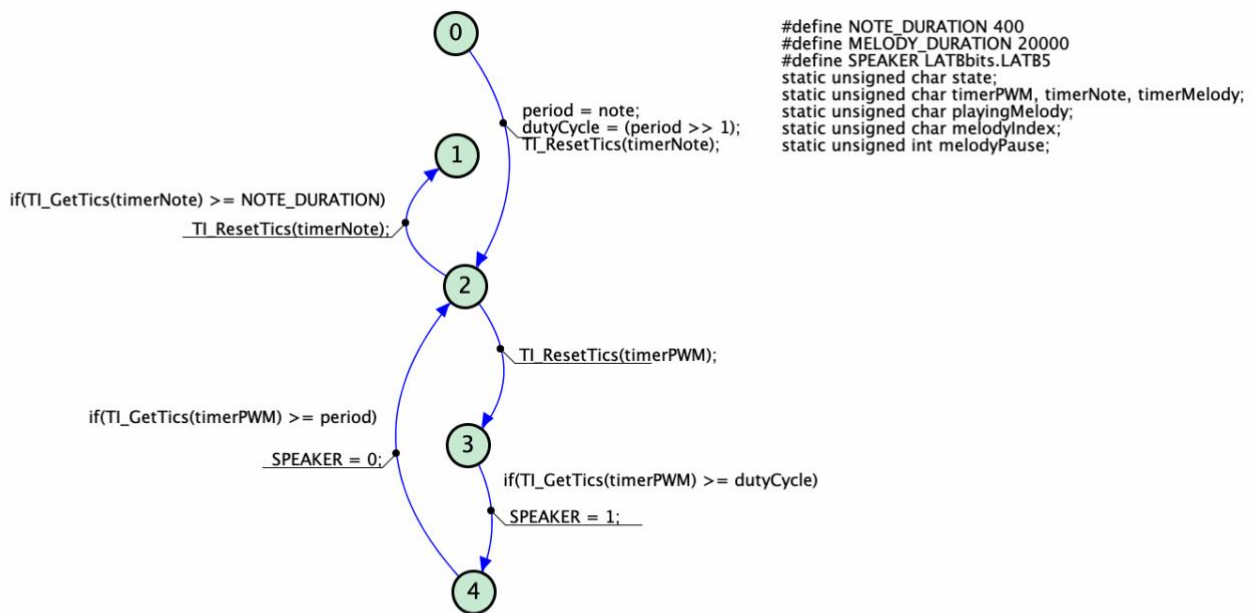
El motor del port gestiona el menú on s'ha d'introduir el port on es vol anar. Simplement escolta el teclat matriu en mode SMS i un cop s'han introduït 3 lletres es bloqueja i escolta * (per esborrar les 3 lletres) o # (per anar al menú principal, enviant i rebent la seqüència corresponent pel port SIO).

5.10. Motor Save



El motor del TAD SaveRecording és l'encarregat de gestionar el guardar i carregar els recordings de la EEPROM. Gestiona les corresponents accions per carregar i descarregar bytes a la EEPROM. És bastant senzill, únicament cal jugar amb les adreces de la EEPROM.

5.11. Motor Buzzer



El motor del buzzer s'encarrega d'escoltar les crides a BUZZ_Play(note) i BUZZ_PlayMelody() i simplement fa un PWM (segons la nota, un període de pwm diferent) pels ports del buzzer durant 0.1s (duració d'una nota).

6. Conclusions

Durant el desenvolupament de la segona i darrera pràctica del projecte, ens hem enfrontat a diversos desafiaments que ens han permès reforçar els nostres coneixements. Inicialment, ens trobem amb dificultats en començar a plantejar la pràctica. No obstant això, aconseguim superar aquesta etapa i dissenyar una estructura adequada per al desenvolupament del codi.

A mesura que vam implementar les funcionalitats bàsiques, ens vam adonar de la importància d'optimitzar el codi degut a les limitacions de memòria del microcontrolador PIC18F4321. Ens dediquem a realitzar millores per reduir el consum de memòria, com simplificar la manera de persistir les dades a l'EEPROM i reorganitzar les condicions i tipus de variables utilitzades. També es van optimitzar motors i es va reduir la quantitat d'estats necessaris per a determinades funcions.

Durant la realització de la practica, no vam tenir problemes greus que ens perjudicaven i ens impediéssin continuar, però sí que ens anaven sortint petits problemes com per exemple en el micròfon, ja que aquest ens va aturar en el procés ja que vam haver de canviar algunes configuracions per a que ens funcionés bé.

En conclusió, aquesta pràctica ens ha donat l'oportunitat d'aprendre a implementar un sistema cooperatiu capaç de fer tasques concurrents. A més, hem comprès la importància de realitzar un disseny eficient i una implementació optimitzada per reduir el consum de memòria al programa, especialment tenint en compte les limitacions de memòria del microcontrolador PIC18F4321, que disposa únicament de 8 KB de memòria flash i 0.5 KB de RAM. Aquests desafiaments ens han permès enfortir els nostres coneixements i habilitats en el desenvolupament de sistemes cooperatius i optimitzar recursos en entorns amb restriccions de memòria.

7. Planificació

Planificació Estimada

Tasca	Inici	Final	14-Apr	15-Apr	16-Apr	17-Apr	18-Apr	19-Apr	20-Apr	21-Apr	22-Apr	23-Apr	24-Apr	25-Apr	26-Apr	27-Apr	28-Apr	29-Apr	30-Apr	1-May	2-May	3-May	4-May	5-May	6-May	7-May	8-May	9-May	10-May	11-May	12-May	13-May	14-May	15-May	16-May	17-May	18-May	19-May
Plantejament	14-Apr	18-Apr																																				
Soldar	19-Apr	21-Apr																																				
Programar	22-Apr	10-May																																				
Debugar	11-May	15-May																																				
Memòria	16-May	19-May																																				

Planificació Real

Tasca	Inici	Final	14-Apr	15-Apr	16-Apr	17-Apr	18-Apr	19-Apr	20-Apr	21-Apr	22-Apr	23-Apr	24-Apr	25-Apr	26-Apr	27-Apr	28-Apr	29-Apr	30-Apr	1-May	2-May	3-May	4-May	5-May	6-May	7-May	8-May	9-May	10-May	11-May	12-May	13-May	14-May	15-May	16-May	17-May	18-May	19-May
Plantejament	14-Apr	17-Apr																																				
Soldar	17-Apr	21-Apr																																				
Programar	22-Apr	9-May																																				
Debugar	10-May	13-May																																				
Memòria	14-May	19-May																																				

En la planificació d'aquesta practica, com podem veure, no ens hem desviat molt en comparació del temps real, al temps estimat. Les parts on més temps hem dedicat, han sigut en la part de programar i en la de dissenyar. Tot i haver començat la pràctica poc a poc per culpa de la càrrega de treball d'altres assignatures, no hem deixat de fer-la i hem anat avançant cada dia. I finalment, quan hem tingut una mica més de temps, l'hem aprofitat per avançar-la al màxim i poder acabar-la a temps per a la entrega final.