

---

# *Perifèrics*

# *Practica 3*

---

## INDEX

1.	Descripció de l'arquitectura del programa .....	2
2.	Diagrama de mòduls .....	3
3.	Descripció dels mòduls del controlador .....	5
4.	Freqüència del ADC .....	6
5.	Captures de pantalla i formes d'ona .....	9
6.	Conclusions .....	10

## 1. Descripció de l'arquitectura del programa

Inicialment les configuracions dels diferents mòduls emprats a la practica son cridats en el main del programa. Aquests mòduls son el **timer2**, la **DMA**, el **ADC** i el **LCD** els quals explicarem amb mes detall en els punts Diagrama de mòduls i Descripció dels mòduls del controlador.

El timer 2 l'hem emprat com a comptador free running per així poder comptar temps dins de la interrupció del timer2 i ser capaços de determinar el nombre de pulsacions fetes per l'usuari mitjançant el polsador. Aquest paràmetre ens modifica una variable que correspon a la velocitat a la que treballarem mes endavant amb el ADC.

Quan l'usuari ha configurat aquest paràmetre (si ho fa ja que no es necessari, sent el valor per defecte de 10us) i prem el polsador mes de dos segons i mig, iniciem la conversió del ADC.

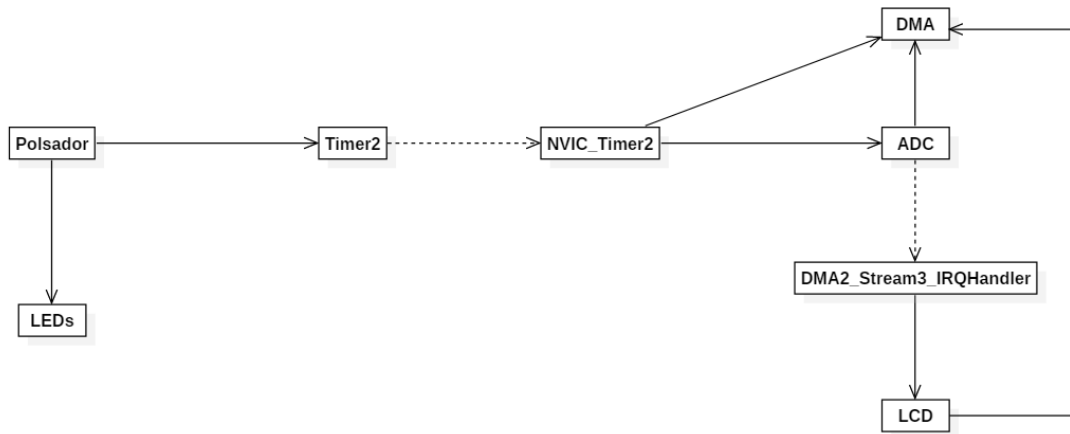
Aquesta crida es troba dins de la interrupció del timer 2 també. Al tenir el ADC configurat de forma que només faci una sola conversió, es fan les 300 conversions necessàries de les mostres dins d'un bucle.

Al finalitzar la conversió, els valors adquirits es mouen de la regió de memòria actual a una altra fent us de la DMA. Aquesta acció es realitza mitjançant la DMA ja que es aproximadament unes 3 vegades mes ràpides que fent-ho mitjançant la CPU. Tant la regió de memòria inicial com la nova regió de memòria a la que desplacem les dades es arbitraria, sent la placa mateixa la que decideix la ubicació exacta.

En aquest punt, agafem de nou les dades reubicades anteriorment i les ficarem a la secció de la SDRAM que consulta el LCD. Així doncs, al activar el LCD, es llegiran les dades inserides a la casella (0xD0000000 + 0x50000) i es printaràn correctament al realitzar una regla de tres.

## 2. Diagrama de mòduls

En la Il·lustració 1 - Diagrama general dels mòduls emprats en la pràctica següent, es pot observar les relacions existents entre els diferents mòduls. Addicionalment, es mostra a continuació de la imatge quina es la configuració de cada mòdul per a esclarir el seu funcionament:



Il·lustració 1 - Diagrama general dels mòduls emprats en la pràctica

- Timer 2: Free running timer per controlar el temps que l'usuari prem el boto i modificat a mitja execució per a generar interrupcions i generar peticions de ADC.
  - Configuració:

```
TIM_TimeBaseInitTypeDef SetupTimer;
/* Enable timer 2, using the Reset and Clock Control register */
RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM2, ENABLE);
SetupTimer.TIM_Prescaler = 0x0000;
SetupTimer.TIM_CounterMode = TIM_CounterMode_Up;
SetupTimer.TIM_Period = 0xFFFFFFFF;
SetupTimer.TIM_ClockDivision = TIM_CKD_DIV1;
TIM_TimeBaseInit(TIM2, &SetupTimer);
TIM_Cmd(TIM2, ENABLE); /* start counting by enabling CEN in CR1 */
```

- ADC: Conversió de dades de Analògic a Digital.
  - Configuració:

```
/* ADC Common Init
***** */
ADC_CommonInitStructure.ADC_Mode = ADC_Mode_Independent;
ADC_CommonInitStructure.ADC_Prescaler = ADC_Prescaler_Div2;
ADC_CommonInitStructure.ADC_DMAAccessMode = ADC_DMAAccessMode_Disabled;
ADC_CommonInitStructure.ADC_TwoSamplingDelay =
ADC_TwoSamplingDelay_5Cycles;
ADC_CommonInit(&ADC_CommonInitStructure);

/* ADC3 Init
***** */
ADC_InitStructure.ADC_Resolution = ADC_Resolution_12b;
ADC_InitStructure.ADC_ScanConvMode = DISABLE;
ADC_InitStructure.ADC_ContinuousConvMode = ENABLE;
ADC_InitStructure.ADC_ExternalTrigConvEdge =
ADC_ExternalTrigConvEdge_None;
ADC_InitStructure.ADC_ExternalTrigConv = ADC_ExternalTrigConv_T1_CC1;
```

```
ADC_InitStructure.ADC_DataAlign = ADC_DataAlign_Right;
ADC_InitStructure.ADC_NbrOfConversion = 1;
ADC_Init(ADC3, &ADC_InitStructure);
```

- LCD: Liquid Cristal Display
  - Configuració:

```
SystemInit();
/* LCD initialization */
LCD_Init();
/* LCD Layer initialization */
LCD_LayerInit();
// Enable Layer1
LTDC_LayerCmd(LTDC_Layer1, ENABLE);
/* Enable the LTDC */
LTDC_Cmd(ENABLE);
/* Set LCD foreground layer */
LCD_SetLayer(LCD_FOREGROUND_LAYER);
LCD_Clear(LCD_COLOR_WHITE);
//LCD_SetBackColor(LCD_COLOR_BLACK);
SDRAM_Init();
```

- DMA: Direct Memory Access, utilitzat per escriure i llegir de la SDRAM.
  - Configuració DMA per ADC:

```
/* DMA2 Stream0 channel2 configuration */
DMA_InitStructure.DMA_Channel = DMA_Channel_2;
DMA_InitStructure.DMA_PeripheralBaseAddr = (uint32_t)ADC3_DR_ADDRESS;
DMA_InitStructure.DMA_Memory0BaseAddr = (uint32_t)&ADC3ConvertedValue;
DMA_InitStructure.DMA_DIR = DMA_DIR_PeripheralToMemory;
DMA_InitStructure.DMA_BufferSize = 1;
DMA_InitStructure.DMA_PeripheralInc = DMA_PeripheralInc_Disable;
DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Disable;
DMA_InitStructure.DMA_PeripheralDataSize =
DMA_PeripheralDataSize_HalfWord;
DMA_InitStructure.DMA_MemoryDataSize = DMA_MemoryDataSize_HalfWord;
DMA_InitStructure.DMA_Mode = DMA_Mode_Circular;
DMA_InitStructure.DMA_Priority = DMA_Priority_Medium;
DMA_InitStructure.DMA_FIFOMode = DMA_FIFOMode_Disable;
DMA_InitStructure.DMA_FIFOThreshold = DMA_FIFOThreshold_HalfFull;
DMA_InitStructure.DMA_MemoryBurst = DMA_MemoryBurst_Single;
DMA_InitStructure.DMA_PeripheralBurst = DMA_PeripheralBurst_Single;
DMA_Init(DMA2_Stream0, &DMA_InitStructure);
DMA_Cmd(DMA2_Stream0, ENABLE);
```

- Configuració DMA per Memory2Memory:

```
DMA_InitStructure.DMA_Channel = DMA_Channel_2;
DMA_InitStructure.DMA_PeripheralBaseAddr = (uint32_t)&totalMostres;
DMA_InitStructure.DMA_Memory0BaseAddr = (uint32_t)&destination;
DMA_InitStructure.DMA_DIR = DMA_DIR_MemoryToMemory;
DMA_InitStructure.DMA_BufferSize = ARRAYSIZE;
DMA_InitStructure.DMA_PeripheralInc = DMA_PeripheralInc_Enable;
DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Enable;
DMA_InitStructure.DMA_PeripheralDataSize =
DMA_PeripheralDataSize_HalfWord;
DMA_InitStructure.DMA_MemoryDataSize = DMA_MemoryDataSize_HalfWord;
DMA_InitStructure.DMA_Mode = DMA_Mode_Normal;
```

```

DMA_InitStructure.DMA_Priority = DMA_Priority_Medium;
DMA_InitStructure.DMA_FIFOMode = DMA_FIFOMode_Enable;
DMA_InitStructure.DMA_FIFOThreshold = DMA_FIFOThreshold_Full;
DMA_InitStructure.DMA_MemoryBurst = DMA_MemoryBurst_INC8;
DMA_InitStructure.DMA_PeripheralBurst = DMA_PeripheralBurst_INC8;
DMA_Init(DMA2_Stream3, &DMA_InitStructure);

```

### 3. Descripció dels mòduls del controlador

Tal i com hem mostrat en la figura Il·lustració 1 - Diagrama general dels mòduls emprats en la pràctica, el funcionament de la practica es el següent:

- Timer 2:**

Timer configurat com a free running timer sense interrupció. En el main, dins del bucle infinit, es consulta si el polsador esta premut consultant el pin adient. Si esta premut, s'agafa el temps de referencia actual del timer 2 i ens enbuclem de nou fins que el usuari deixa de prémer el boto.

Al deixar de prémer el boto, agafem de nou la referencia actual del timer 2 i es fa la diferencia amb el moment en que l'usuari ha premut el boto inicialment.

Amb això podem saber el temps que l'usuari ha tingut premut el boto i actuar de forma corresponent.

El timer 2 es reconfigura al prémer el boto durant mes de dos segons i mig per a que generi interrupcions. D'aquesta forma a cada interrupció generada pel timer, es fa una crida al ADC per a que realitzi una conversió. La interrupció assegura que el ADC tingui temps suficient per a realitzar la conversió.
- ADC:**

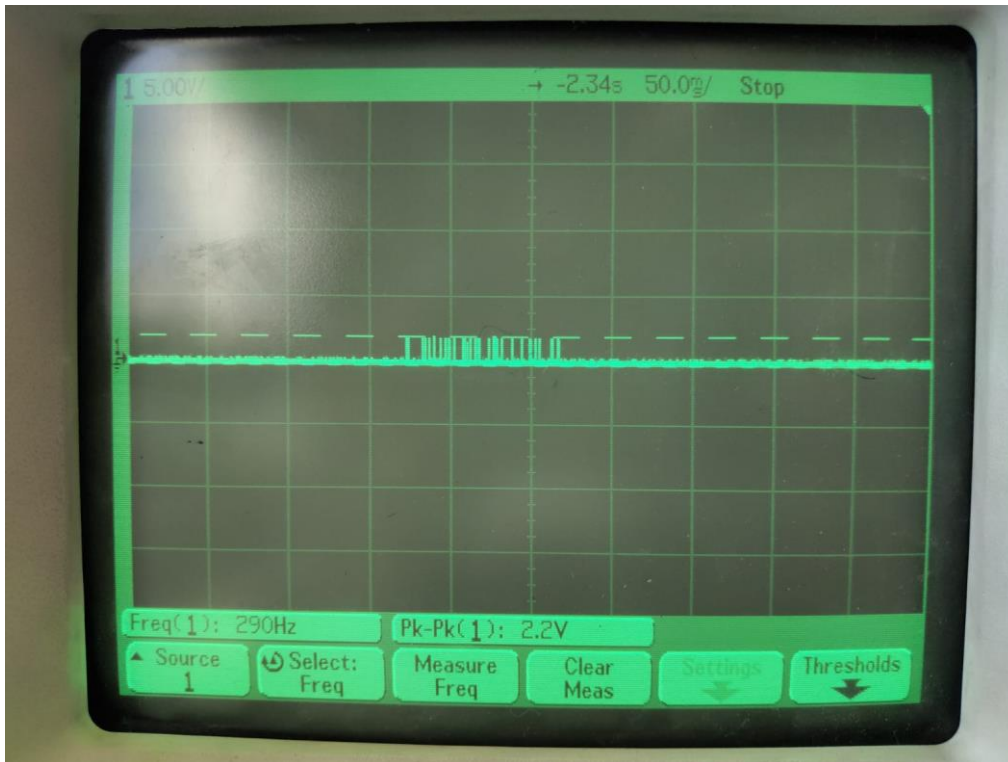
El ADC (Analog to Digital Converter) esta configurat de forma que realitzi una sola conversió. Com a bona praxis, les crides al ADC estan dins de la interrupció del timer així garantim una conversió espaïda i sense solapaments.

Al estar el pin d'entrada del ADC al aire, això genera uns valors de conversió compresos entre 800 i 1000.
- DMA:**

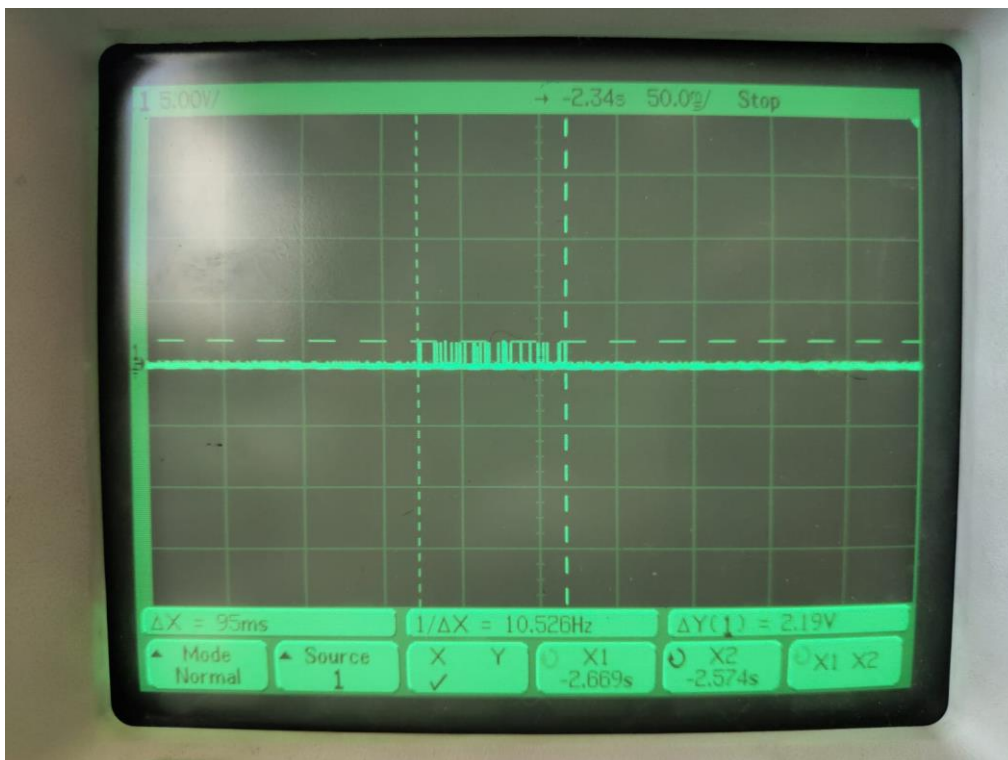
La DMA (Direct Memory Access) es configura de forma anàloga en tots els casos. L'única diferencia que hi ha es en el camp `DMA_InitStructure.DMA_DIR`, per indicar la direcció de la conversió (MemoryToMemory o PeripherailToMemory) i en el camp `DMA_InitStructure.DMA_BufferSize` el qual indica la mida de dades a convertir. Es podria haver mantingut aquest segon camp constant a un valor de 300, però la conversió del ADC s'ha realitzat de dada a dada i s'ha mantingut aquest format.
- LCD:** Per a treballar amb el LCD s'ha fet us de la memòria la qual el LCD consulta, on anirà llegint els punts que ha de anar mostrant. Aquests punts són el resultat de realitzar les 300 conversions amb el ADC. Després d'agafar una mostra de la conversió, se li apliquen un càlcul senzill per a mostrar-lo a la pantalla LCD en la posició correcta dels eixos centrals d'aquesta. També se li afegixen uns marges al seu voltant d'una amplada de 10 píxels.

#### 4. Freqüència del ADC

A continuació es mostra en captures de pantalla del oscil·loscopi, les durades i freqüències que necessita el ADC al treballar a 10 $\mu$ s, 100 $\mu$ s o 1ms.

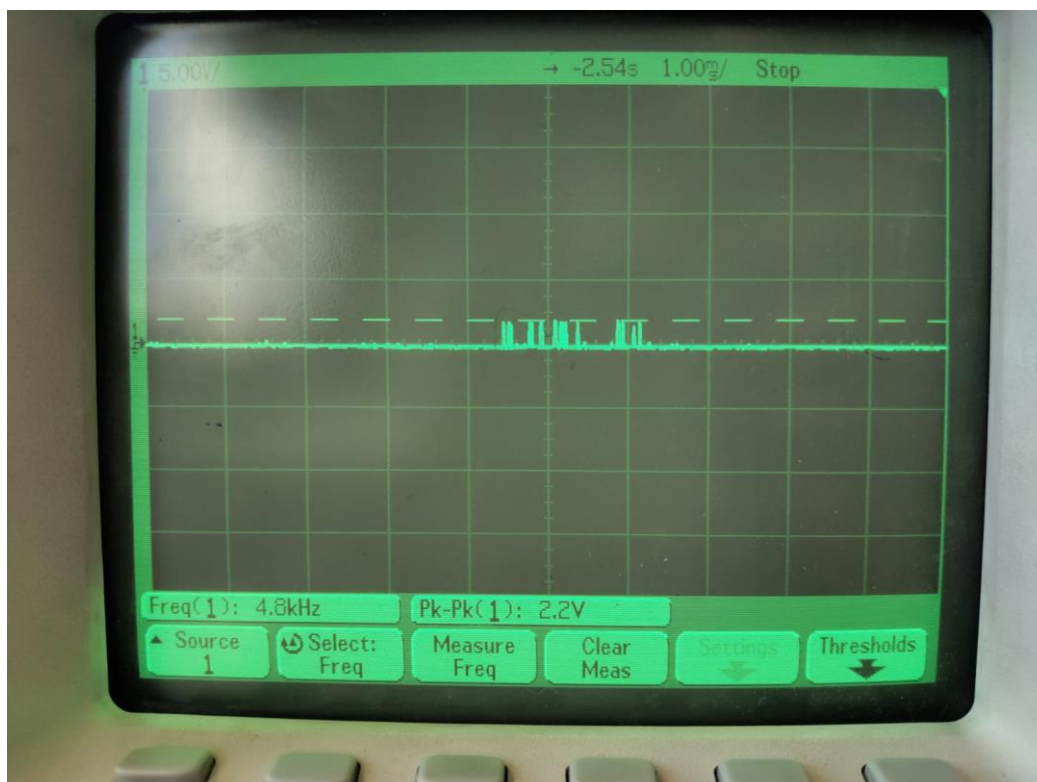


Il·lustració 2 – Frecuència de les conversions a 1ms

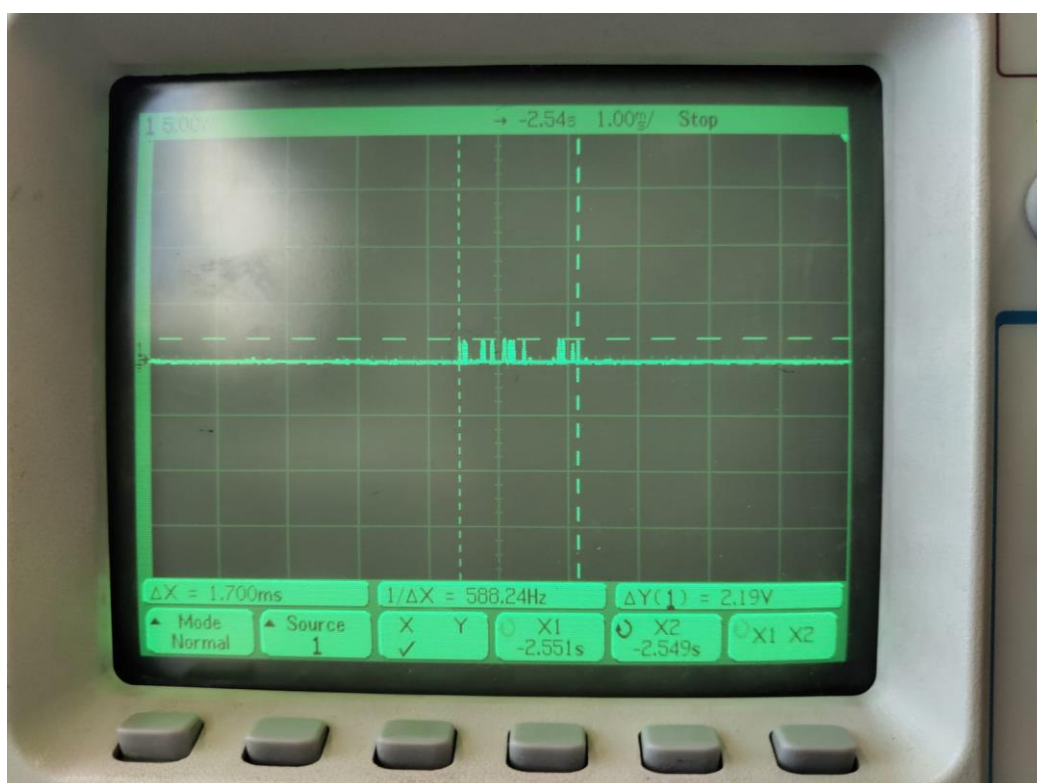


Il·lustració 3 – Durada de les conversions a 1ms



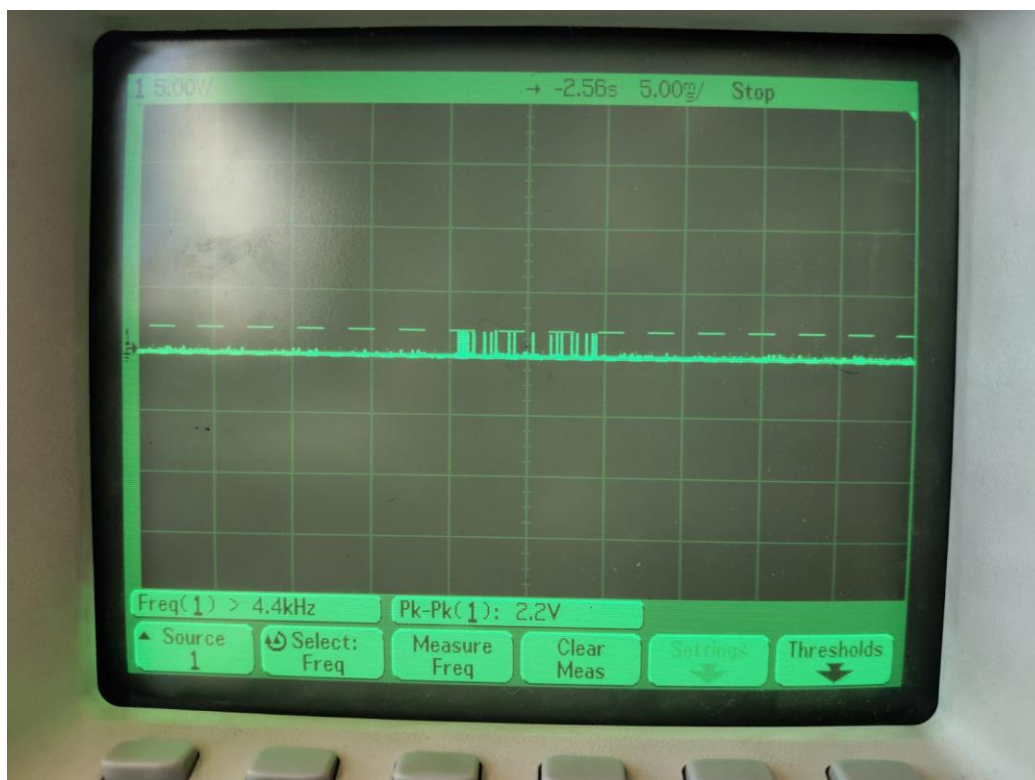


Il·lustració 4 – Frecuencia de les conversions a 100us

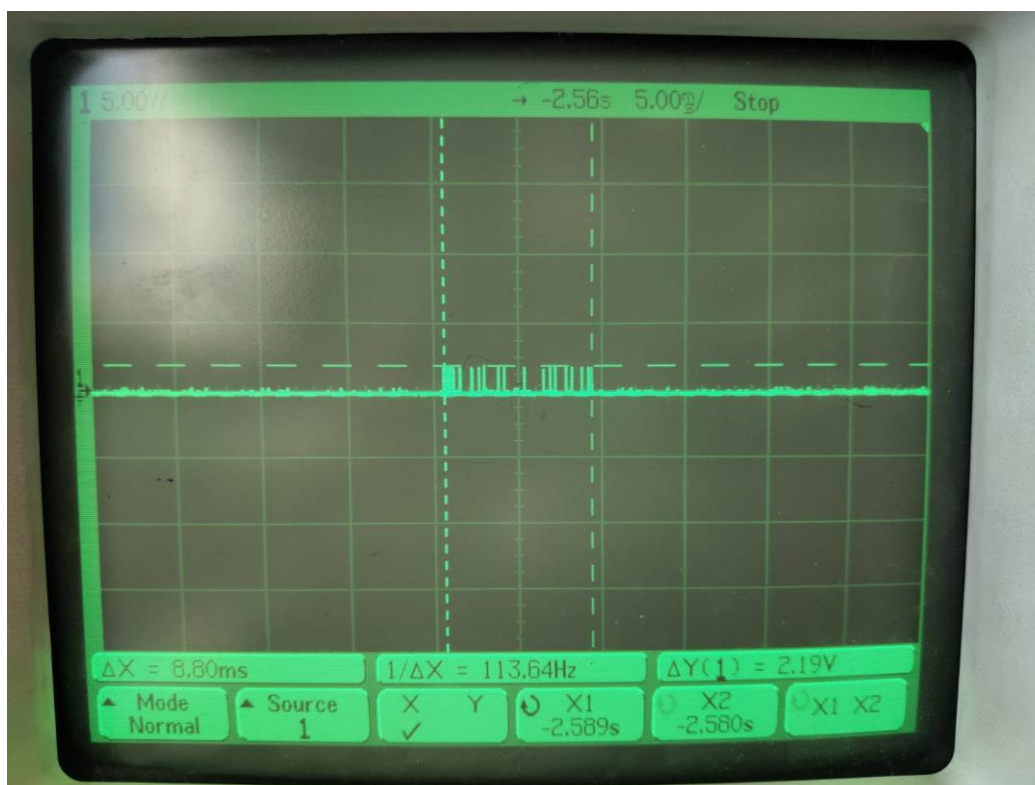


Il·lustració 5 – Durada de les conversions a 100us





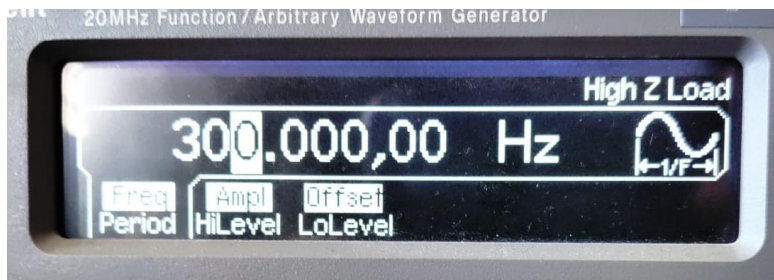
Il·lustració 6 – Frecuencia de les conversions a 10us



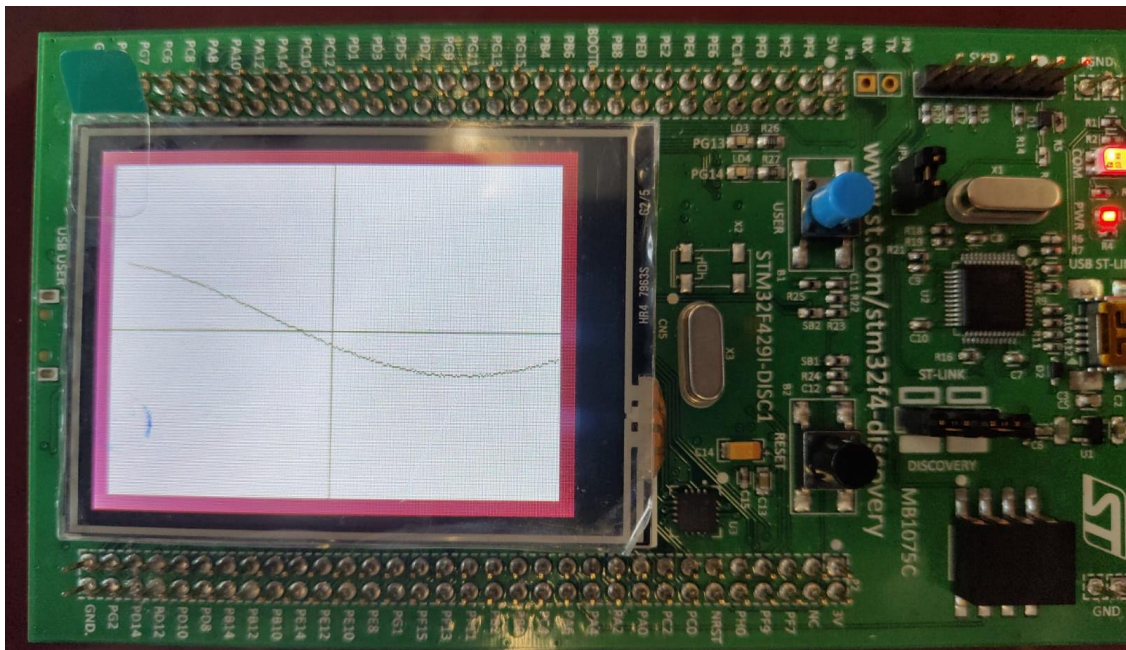
Il·lustració 7 – Durada de les conversions a 10us

## 5. Captures de pantalla i formes d'ona

A continuació es mostren captures del senyal introduït a la placa i mostrat per la pantalla LCD després de fer el procés d'adquisició de les mostres amb el ADC.



Il·lustració 8 – Senyal introduïda a la placa



Il·lustració 9 – Senyal de sortida mostrat pel display LCD

## 6. Conclusions

El disseny intern del LCD, novetat principal d'aquesta practica, es un element força curiós. Es pot treballar de diverses formes amb ell. Independentment de si es fa un de llibreries o es realitzen escriptures directament a memòria, la forma en la que es poden dur a terme aquestes tasques es bastant versàtil. De fet es tant versàtil que se li pot indicar a la pantalla que llegeixi valors a partir d'una adreça de memòria específica i no de la per defecte de la SDRAM. Amb això es pot fer multitud d'operacions, fins i tot mostrar gifs si es fiquessin diverses imatges a memòria i es refresquessin de forma suficientment rapida com per a que semblessin un vídeo curt (gif).

També es podrien combinar aquestes característiques amb altres, com serien la lectura d'imatges a traves d'una comunicació EUSART amb un PC per exemple. El LCD ofereix moltes possibilitats, i tot i que només hem treballat amb ell de forma bàsica, hem pogut apreciar el seu valor.