

Le contrôleur de LCD , RW1063

[1] *RW1063.pdf* (contrôleur d'écran LCD)

[2] *LCD_SPI_Midas.pdf* (ensemble LCD incluant le RW1063)

1. Introduction

Le RW1063 est un contrôleur de LCD. Il est monté sur le module LCD SPI que l'on utilise en TP. Il existe par ailleurs d'autres modules LCD qui embarque ce contrôleur mais en interface parallèle 8 bits, 4 bits ou encore I2C. Le présent document s'appuie sur la ressource *RW1063.pdf*.

NB : le présent document est une synthèse incomplète, voir [1] pour tous les détails.

2. L'interface SPI

Les lignes utilisées pour s'interfacer en SPI sont repérées sur le schéma ci-dessous :

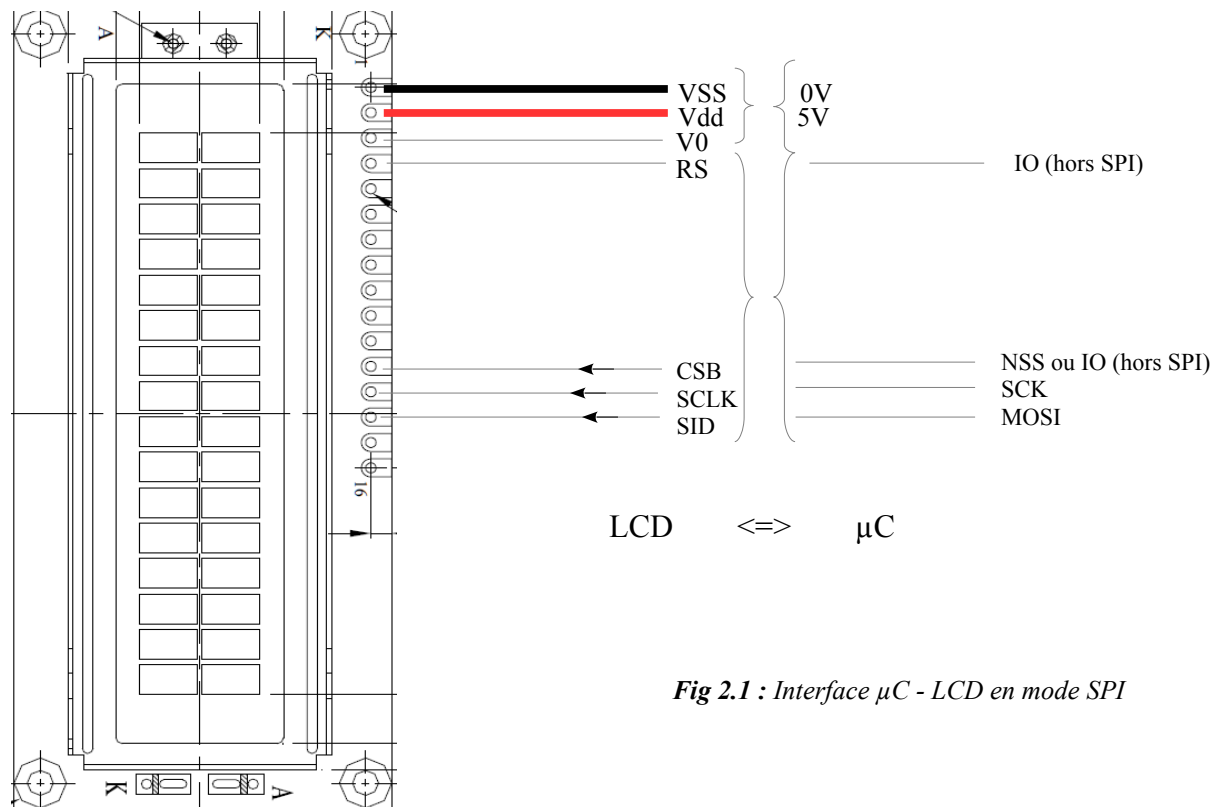


Fig 2.1 : Interface µC - LCD en mode SPI

➤ **Example of timing sequence**

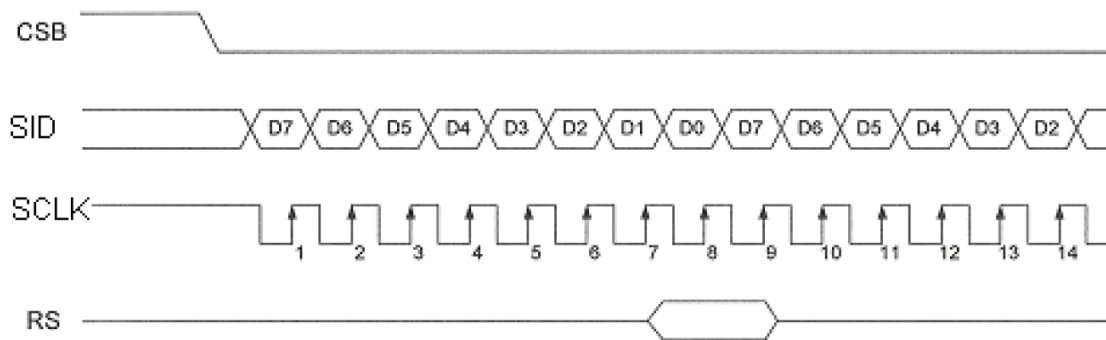


Fig 2.2 : exemple de chronogramme SPI

Note: Following is the master SPI clock mode of MPU.

Idle state for clock is a high level · data transmitted on rising edge of SCLK, and data is hold during low level.

3. Le contrôle du LCD

3.1. Rôle des pin utilisées

3.1.1. Alimentation et contraste

Le circuit s'alimente via les bornes V_{dd} et V_{ss} (5V, 0V). L'entrée V0 permet de régler le contraste. La valeur 0V permet d'avoir le contraste maximum.

3.1.2. SPI

Au niveau du SPI, comme le montre la *fig 1.1*, la liaison utilise :

- **SCLK** : horloge du SPI : repos à l'état '1'. Le verrouillage de chaque donnée se fait sur front montant (cf *fig 2.2*). Période minimale de 640ns, soit une fréquence maximale de 1,56MHz,
- **SID** : Serial Input Data. En mode maître, cette ligne unidirectionnelle (μC vers LCD) est la ligne **MOSI** (**Master Out, Slave In**),
- **CSB** : c'est le *chip select*, actif bas (doit être à '0' pour que le LCD soit validé). Il peut être relié à **NSS** (ligne faisant partie de l'interface SPI) ou piloté par une IO indépendante, ou même mise à la masse si le SPI en question est uniquement relié au LCD.

3.1.3. Autre

- **RS** (*register Selection*) : à relier à une IO indépendante, permet de différencier Data et Instruction.
 - **RS = 1** : la donnée est stockée dans **DR** (donc destiné à l'affichage)
 - **RS = 0** : la donnée est stockée dans l'Instruction Register (c'est donc une instruction : clear, Shift...)

3.2. Fonctionnement, constitution

La *RAM* interne appelée *DDRAM* (*Display Data RAM*) contient les codes *ASCII* à afficher sur le *LCD*. Elle s'organise de la manière suivante :

- de l'adresse 0x0 à 0x27 on trouve 40 caractères dont 16 seront affichés sur la première ligne du *LCD*,
- de l'adresse 0x40 à 0x67 on trouve 40 caractères dont 16 seront affichés sur la seconde ligne du *LCD*.

L'écriture / lecture dans la *DDRAM* se fait via un registre, *DR* et grâce à un pointeur d'adresse (*Adress Counter*, *AC*). Ce pointeur est automatiquement incrémenté lors d'une écriture, décrémenté lors d'une lecture.

Il existe une autre *RAM*, la *CGRAM* (*Character Generation RAM*) qui permet à l'utilisateur de créer son propre symbole correspondant à un code *ASCII* donné. On n'utilisera pas cette fonctionnalité dans les TP.

Les données transmises (uniquement en écriture puisqu'il n'y a pas de ligne de donnée en sortie), sont soit des données à afficher, soit des instructions.

3.3. Les instructions

Dans la documentation [1], on trouvera la liste exhaustive des instructions. Ci-dessous, on donne celles qui seront le plus utiles :

Clear Display : le curseur est positionné à l'origine et le pointeur d'adresses est mis à 0 (*Adress Counter* =0). Toutes les données sont initialisées pour afficher aucun caractère,

Return Home : le curseur est positionné à l'origine et le pointeur d'adresses est mis à 0 (*Adress Counter* =0),

Display ON/OFF : Active ou non l'affichage. Le mode d'affichage du curseur est également ajustable,

SetDDRAM Adress : positionne le pointeur d'adresse : permet de modifier l'affichage en cours, à une localisation donnée,

Function Set : indispensable à l'initialisation (voir table page suivante) :

$DL = 1$ (*SPI* mode)

$N = 1$ (deux lignes exploitées)

$F = 0$ affichage 5x8 pts par caractère.

Sur la page suivante, on trouve le tableau complet qui détaille toutes les instructions.

Instruction	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Description	Time (540KHz)
Read display data	1	1	Read data								Read data from DDRAM/CGRAM	18.5us
Write display data	1	0	Write data								Write data into DDRAM/CGRAM	18.5us
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM, and set DDRAM address to "00H" from AC	0.76ms
Return Home	0	0	0	0	0	0	0	0	1	X	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	0.76ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Assign cursor moving direction and specify display shift. These operations are performed during data read and write. I/D="1": increment I/D="0": decrement	18.5us
Display ON/OFF	0	0	0	0	0	0	1	D	C	B	Set Display /Cursor/Blink On/OFF D="1": display on D="0": display off C="1": cursor on C="0": cursor off B="1": blink on B="0": blink off	18.5us
Cursor or Display shift	0	0	0	0	0	1	S/C	R/L	X	X	Cursor or display shift S/C="1": display shift S/C="0": cursor shift R/L="1": shift to right R/L="0": shift to left	18.5us
Function Set	0	0	0	0	1	DL	N	F	X	X	Set Interface Data Length DL= 8-bit interface/ 4-bit interface N = 2-line/1-line display F= 5x8 Font Size / 5x11Font Size	18.5us
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter	18.5us
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter	18.5us
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Can know internal operation is ready or not by reading BF. The contents of address counter can also be read. BF="1": busy state BF="0": ready state	0us

4. Initialisation du LCD

Le tableau page suivante montre la séquence d'initialisation à suivre lorsqu'on utilise le LCD en SPI.

- Serial Interface code (Fosc=540KHz)

