

Rapport individuel d'élève ingénieur
de 2^e année

Filière : Informatique des systèmes interactifs pour l'embarqué,
la robotique et le virtuel

Linux Embarqué : Mise en place d'un afficheur GLCD

Présenté par : **PEREZ KELIAN**

Enseignant : **Jacques LAFFONT**

Mars 2023

Campus des Cézeaux. 1 rue de la Chébarde. TSA 60125. 63178 Aubière CEDEX

Table des matières

Introduction	3
1 Principe de fonctionnement du GLCD	3
2 Branchement du GLCD à la Raspberry PI 3	4
3 Ecriture d'un driver pour le GLCD	6
4 Notes sur le fonctionnement	6
Références bibliographiques	7

Introduction

Ce rapport individuel s'inscrit dans un travail en collaboration avec Duneshwar BADEL. L'objectif est de mettre en place un afficheur GLCD contrôlé via Raspberry Pi 3. Le travail effectué ici se base sur diverses sources trouvées sur internet concernant la mise en place d'un afficheur LCD sur Raspberry Pi ainsi que sur le travail effectué sur une carte EasyPIC 6 avec un GLCD au premier semestre lors du cours « Architecture Avancée ». Le GLCD correspond donc à celui utilisé pendant ce cours, un GLCD Xiamen Ocular GDM12864 BS0108 18x164.



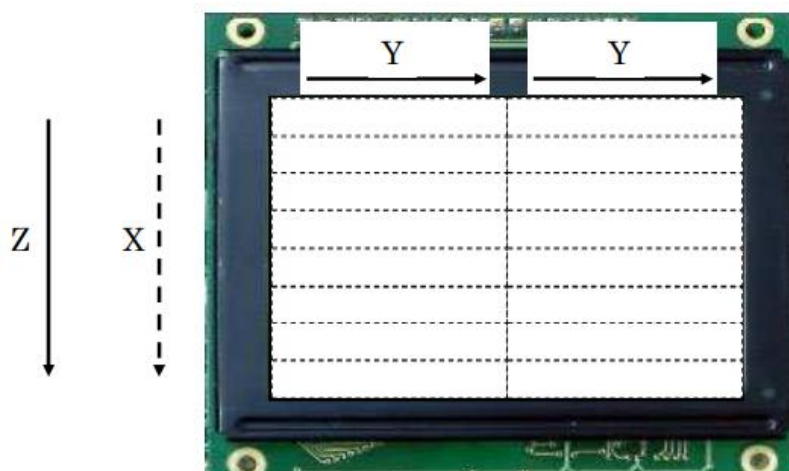
1 Principe de fonctionnement du GLCD

Le GLCD Xiamen Ocular est un afficheur pixel par pixel de 128x64 pixels. Il est possible d'y afficher des images comme du texte. En pratique l'écran est considéré comme deux écrans de 64x64 mis côte à côte et il est possible de contrôler l'activation de ces écrans « virtuel » de manière indépendante ce qui résultera en l'activation d'un côté unique ou de l'ensemble de l'écran. De même la mémoire considère l'écran comme ces 2 zones de 64x64. Un pixel possède 3 axes de coordonnées X,Y et Z lorsqu'il est en mémoire.

X correspond à la page (PAGE) : X(2 :0) de 0 à 7 correspond au numéro de ligne du caractère.

Y correspond à la colonne (COLUMN) : Y(5 :0) de 0 à 63 (et non 128 car on a 2 contrôleurs) correspond à l'abscisse du pixel

Z correspond à la ligne de départ (START LINE) : Z(5 :0) de 0 à 63 correspond à l'ordonnée du pixel.



2 Branchement du GLCD à la Raspberry Pi 3

Le GLCD Xiamen Ocular dispose de vingt broche qu'il est nécessaire de relier à la raspberry :

Vss : Masse

Vdd : Alimentation +5V

Vo : Tension de contraste (0 à +5V)

RST : Reset

CS1,CS2 : Chip Select (sélection de la moitié de l'écran à activer)

CS1 = 1, CS2=0 : Activation du contrôleur gauche

CS1 = 0, CS2=1 : Activation du contrôleur droit

CS1 = 1, CS2=1 : Activation des deux contrôleurs simultanément

RS : Sélection du registre (Register Select)

RS = 0 : envoi d'une commande (instruction)

RS = 1 : envoi d'une donnée

R/W : Lecture ou écriture (Read/Write)

R/W = 0 : Ecriture

R/W = 1 : Lecture

E : Validation (Enable)

D(7:0) : Bus de données

En utilisant le site pinout.xyz qui propose un guide détaillé de brochage de Raspberry Pi 3 (voir image ci-après) et la documentation de la carte EasyPIC 6 on peut attribuer les pins du GLCD à la Raspberry comme suit :

Vss : Masse qui peut correspondre aux **pins indiqués 6, 9, 14, 20, 25, 30, 34 ou 39**

Vdd : Alimentation +5V peut correspondre aux **pins 2 ou 4**

Vo : Tension de contraste relié à **un potentiomètre relié à la masse.**

R/W : Lecture ou écriture (Read/Write) relié à la **masse** car on reste toujours en mode écriture

Les ports suivants sont reliés à l'un des ports GPIO de la raspberry

RST : Reset on a choisi **GPIO 4**


























CS1,CS2 : Chip Select on a choisi **GPIO 17 et 18**

RS : Sélection du registre (Register Select) on a choisi **GPIO 27**

E : Validation (Enable) on a choisi **GPIO 22**








D(7:0) : Bus de données on a choisi (dans l'ordre croissant) **GPIO 23, 24, 25, 5, 6,12, 13, 16**

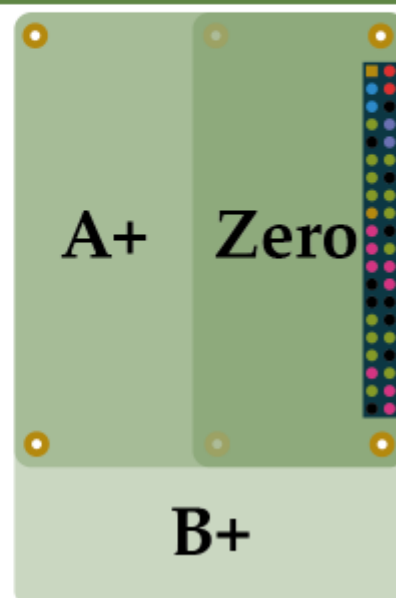
Raspberry Pi Pinout

3v3 Power	1			2	5v Power
GPIO 2 (I2C1 SDA)	3			4	5v Power
GPIO 3 (I2C1 SCL)	5			6	Ground
GPIO 4 (GPCLK0)	7			8	GPIO 14 (UART TX)
Ground	9			10	GPIO 15 (UART RX)
GPIO 17	11			12	GPIO 18 (PCM CLK)
GPIO 27	13			14	Ground
GPIO 22	15			16	GPIO 23
3v3 Power	17			18	GPIO 24
GPIO 10 (SPI0 MOSI)	19			20	Ground
GPIO 9 (SPI0 MISO)	21			22	GPIO 25
GPIO 11 (SPI0 SCLK)	23			24	GPIO 8 (SPI0 CE0)
Ground	25			26	GPIO 7 (SPI0 CE1)
GPIO 0 (EEPROM SDA)	27			28	GPIO 1 (EEPROM SCL)
GPIO 5	29			30	Ground
GPIO 6	31			32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33			34	Ground
GPIO 19 (PCM FS)	35			36	GPIO 16
GPIO 26	37			38	GPIO 20 (PCM DIN)
Ground	39			40	GPIO 21 (PCM DOUT)

Legend

Orientate your Pi with the GPIO on the right and the HDMI port(s) on the left.

-  GPIO (General Purpose IO)
-  SPI (Serial Peripheral Interface)
-  I²C (Inter-integrated Circuit)
-  UART (Universal Asynchronous Receiver/Transmitter)
-  PCM (Pulse Code Modulation)
-  Ground
-  5V (Power)
-  3.3V (Power)



3 Ecriture d'un driver pour le GLCD

Nous avons ensuite écrit un driver pour que notre GLCD puisse être commandé par notre Raspberry PI. Pour cela nous avons utilisé 3 programmes précis. Le premier correspond au driver GLCD écrit en classe d'Architecture Avancée afin de faire fonctionner notre écran sur la carte EasyPIC 6, ce programme est rédigé en C et devra donc être réadapté. Le second programme correspond à un driver (open source) écrit en Python par la société Adafruit qui permet de contrôler un écran LCD avec une Raspberry. Enfin le troisième programme correspondait à la propre tentative de la personne qui a écrit le tutoriel pour connecter un GLCD à une raspberry sur lequel nous nous sommes basé afin de travailler sur notre GLCD. Nous avons écrit notre driver en python et avons ajouté un ensembles de tests. Ce programme intègre la bibliothèque python RPi.GPIO qui permet de gérer les port GPIO d'une Raspberry. Pour essayer notre programme il suffit de se connecter à la raspberry et de le placer dans les fichiers la carte puis de le lancer avec la commande : `sudo python programme.py`

Attention il est à noter que nous avons simplement écrit le driver. Par manque de temps et de moyens nous n'avons pas pu le tester physiquement. Ainsi même si l'écriture du driver était une réécriture du drivers EasyPIC 6 du GLCD pour qu'il fonctionne sur une RaspberryPi, il est évident que cette partie du travail doit être vérifiée et corrigée. Notre driver représente donc une ébauche de ce qui pourrait être écrit pour pouvoir permettre à l'utilisateur de tracer des droites et des carrés ainsi que d'allumer/éteindre des pixels sur notre GLCD.

4 Notes sur le fonctionnement

La pagination du GLCD:

- $X = Z/8$
- Il est possible d'écrire une image bitmap en coordonnées X/Y par envois d'octet :
8 lignes de caractères ($X = 0$, $Y = 0$ correspond à « en haut à gauche »)

On peu aussi lancer les instructions suivantes (issues de la documentation fourni en Architecture avancée)

Instructions	RS R/W D7 D6 D5 D4 D3 D2 D1 D0	Description
Display on/off	0 0 0 0 1 1 1 1 1 D	Affichage en fonction (D=1) Ou hors fonction (D=0) DD RAM visible ou non
Set Y address	0 0 0 1 Y5 Y4 Y3 Y2 Y1 Y0	Coordonnée Y (Colonne)
Set X address	0 0 1 0 1 1 1 X2 X1 X0	Coordonnée X (Page)
Set Z address	0 0 1 1 Z5 Z4 Z3 Z2 Z1 Z0	Coordonnée Z (start line)
Status read	0 1 B 0 F R 0 0 0 0	Lecture de l'état B : Busy, F : On/Off, R : in Reset
Write display data	1 0 D7 D6 D5 D4 D3 D2 D1 D0	Ecriture en DD RAM Y post-incrémenté de 1
Read display data	1 1 D7 D6 D5 D4 D3 D2 D1 D0	Lecture du data bus.

Références bibliographiques

Branchement de la Raspberry PI 3 :

https://fr.pinout.xyz/pinout/pin5_gpio3#

Datasheet du GLCD utilisé :

<https://www.sparkfun.com/datasheets/LCD/GDM12864H.pdf>

Datasheet de la carte EasyPIC 6 :

https://ent.uca.fr/moodle/pluginfile.php/1252128/mod_folder/content/0/Datasheet%20Carte%20EasyPIC6%20manual%20v100.pdf?forcedownload=1

Cours sur les LCD/GLCD de l'ISIMA :

https://ent.uca.fr/moodle/pluginfile.php/1665712/mod_resource/content/1/Syst%C3%A8mes%20Embarqu%C3%A9s_20_21%20Ch%204%20%20Partie%207_8%20%20Les%20afficheurs%20LCD%20et%20GLCD.pdf

Tutoriel d'utilisation d'un LCD sur une Raspberry :

<https://www.glennklockwood.com/electronics/hd44780-lcd-display.html>

Recherches du site MP Electronic Devices sur l'utilisation d'un LCD avec un Raspberry PI :

<https://www.pinteric.com/displays.html>