 CONTINEA <small>Microprocesamiento modular + Conectividad</small>	RCM4300 + Display gráfico Powertip 128x64	Nota de Aplicación
		CoAN-005
	Micro de 3.3V ↔ LVX3245 ↔ display de 5V	Publicado: 00/00/0000
		Página 1 de 8

Revisión	Fecha	Comentario	Autor
0	09/12/2008	Trabajo realizado sobre la base de las notas de aplicación CAN-003 y CAN-008 de Cika Electrónica	Ulises Bigliati

ÍNDICE

Introducción	2
Objetivos.....	2
Implementación	2
Descripción del hardware	3
Selección del pin-out del microprocesador	3
Circuito de interfaz de interconexión entre micro y display	3
Ejemplo de un cable de conexión para un display Powertip PG12864.....	4
Descripción del software involucrado	4
Estructura de archivos del proyecto	5
Archivos del proyecto.....	5
Archivos fuente del proyecto.....	5
Módulos de software del proyecto	6
Módulo lcd_driver.lib	6
Módulo lcd_graphic.lib.....	6
Módulo fonts.lib	7
Módulo principal	7
Referencias	8

 CONTINEA Microprocesamiento modular + Conectividad	Micro de 3.3V ↔ display de 5V	Nota de Aplicación
		CoAN-005
	RCM4300 + Display gráfico Powertip 128x64	Publicado: 00/00/0000
		Página 2 de 8

Introducción

Ya no es novedad para nadie que los dispositivos electrónicos digitales presentan cada vez menores consumos de potencia a la vez que son cada vez más veloces, y consecuentemente funcionan con tensiones de trabajo cada vez más bajas.

En contraste con esta realidad y por lo pronto, en nuestro mercado, la electrónica de 5 Volts continúa siendo mucho más común que la de 3.3 Volts.

Es por esto que se nos puede presentar la ocasión de interfacear dispositivos tales como displays gráficos con buses de datos bidireccionales y se haga indispensable la utilización de circuitos integrados especialmente diseñados para adaptar los niveles de señal en ambos sentidos: desde y hacia el microprocesador o microcontrolador.

Objetivos

Nos proponemos realizar entonces, una implementación práctica que ejemplifique la forma de utilizar microprocesadores que presenten electrónica de 3.3V junto con periféricos que funcionan a 5 Volts en esquemas de funcionamiento que requieran una interfaz de comunicación bidireccional entre ellos.

En esta nota se presenta un esquema de interconexión entre un microprocesador con alimentación y niveles lógicos de 3.3 Volts y un periférico de 5 Volts que consiste en un display gráfico que posee un bus de datos bidireccional. Para lograr la interfaz entre ellos, nos valdremos de la utilización de un C.I. que es un transceiver adaptador bidireccional de niveles de tensión que utiliza alimentación dual (3.3V y 5V) para su funcionamiento.

Implementación

En esta nota de aplicación utilizamos un módulo Rabbit RCM4300 en conjunto con un display gráfico Powertip PG12864¹. El esquema de interconexión entre ambos se establece mediante el circuito integrado 74LVX3245² mas un circuito integrado adicional: 74HCT245, que es necesario para elevar la tensión de 3.3 a 5Volts en las líneas de control que van desde el microprocesador hacia el display en ese único sentido.

La implementación de este ejemplo se llevó a cabo para mayor simplicidad, aprovechando la placa de prototipos que está presente en cualquier Kit de desarrollo de Rabbit.

Como en este caso utilizamos un RCM4300, la placa de prototipos es la correspondiente a este tipo de módulos, y entre otras cosas que no son de interés para este trabajo, nos provee los 3.3Volts para la alimentación del módulo Rabbit y nos permite también obtener los 5 Volts necesarios para alimentar el display y la electrónica de interfaz entre el microprocesador y el display, que como ya veremos posee una etapa que requiere alimentación dual (3.3 y 5 Volts).

Obviamente, si se desea alimentar el backlight del display esto debe realizarse mediante una fuente aparte dado el alto consumo que esto representa.

No se proporcionan en esta nota mayores detalles en cuanto al funcionamiento en sí mismo del display LCD, ya que de ser necesarios, estos pueden ser obtenidos consultando la nota de aplicación CAN-003³.


En cuanto al software de demostración desarrollado para realizar esta aplicación práctica se utilizó el entorno de desarrollo Dynamic C en su versión 10.40. El software utilizado se describirá en la sección correspondiente en esta misma nota, pero cabe destacar que fue construido sobre la base provista por la nota de aplicación CAN-008⁴

¹ Comercializado por Cika Electrónica

² Comercializado por Cika Electrónica

³ http://www.cika.com.ar/soporte/AppNotes/CAN-003_LCDgraphHD61202-Rabbit.pdf by Sergio Caprile

⁴ http://www.cika.com.ar/soporte/AppNotes/CAN-008_LCDgraphHD61202TEXT-Rabbit.pdf by Sergio Caprile

	Micro de 3.3V ↔ display de 5V	Nota de Aplicación
	RCM4300 + Display gráfico Powertip 128x64	CoAN-005
		Publicado: 00/00/0000
		Página 3 de 8

Descripción del hardware

Selección del pin-out del microprocesador

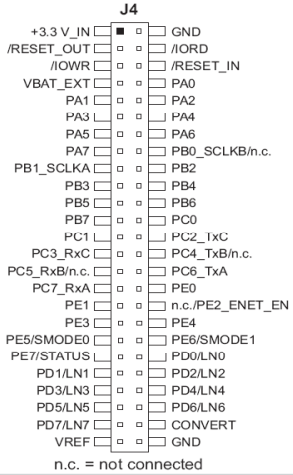
A continuación comenzaremos a definir las entradas y salidas que en nuestro microprocesador cumplirán con las funciones necesarias para controlar el display.

En la tabla de abajo, puede notarse una diferencia si se compara con una interfaz de control de display que no requiera la adaptación de niveles de señal en forma bidireccional. Esto se manifiesta por la necesidad de utilizar una patita más del microprocesador para controlar el sentido del bus de datos (PA[0:7]) estas son las líneas para las cuales es necesario adaptar los niveles de tensión de 5 a 3Volts y viceversa.

El renglón resaltado corresponde a la asignación del pin de control referido. Este control actúa en forma directa sobre el circuito integrado 74LVX3245 que cumple funciones de interfaz y como se ha adelantado, consiste en un transceiver que se encarga de la traducción de los niveles de tensión.

Pin-out				
Pin	Nombre	Sentido	Uso (display)	Notas
8-15	PA [0:7]	Input/Output	Datos[0:7]	Bus de datos bidireccional
18	PB2	Output	CS1	Chip selects
19	PB3	Output	CS2	
36	PE4	Output	74LVX3245 (T/R)	Sentido del bus
37	PE5	Output	D/I	Data/Instruction
38	PE6	Output	R/W	Read/Write
39	PE7	Output	E	Chip enable

En la tabla de arriba podemos apreciar la designación de pines que hemos decretado para nuestra implementación, por supuesto que estos pines podrán ser otros de acuerdo a las necesidades del proyecto real que pudiera implementar el presente ejemplo. Para mayor comodidad, a la derecha se presenta un vista desde abajo del pin-out del conector del módulo Rabbit RCM4300 que fuera utilizado para el ejemplo



Note: These pinouts are as seen on the Bottom Side of the module.


Circuito de interfaz de interconexión entre micro y display

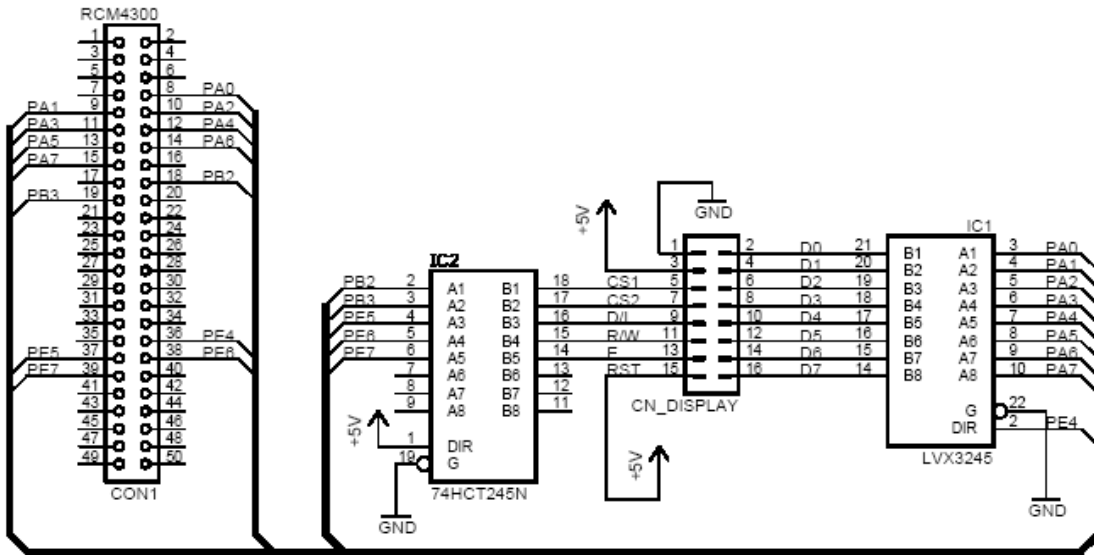
Una vez definidos los pines que serán destinados para controlar el display nos dedicaremos al diseño del circuito de interfaz que en sí mismo representa el objeto central de esta nota.

Para ello necesitamos un bus de datos bidireccional de 8 bits con niveles de tensión de 3.3 Volts del lado del microprocesador y niveles de 5 Volts del lado del Display, mediante este bus de datos vincularemos el puerto PA[0:7] del Rabbit4000 con las líneas de datos del display [DB0-DB7]. A estos fines se incorpora por supuesto el ya varias veces referenciado 74LVX3245.

Por otra parte, será necesario elevar la tensión de las líneas de control (CS1, CS2, D/I, R/W y E) desde los 3.3Volts que provee el micro hasta los 5 Volts que requiere el display. Para esto, elegimos utilizar el circuito integrado 74HCT245 en una configuración fija que permita la comunicación permanentemente en el sentido del micro hacia el display. Gracias a este C.I. obtenemos fácilmente el nivel lógico requerido por el display a partir los niveles lógicos del micro.

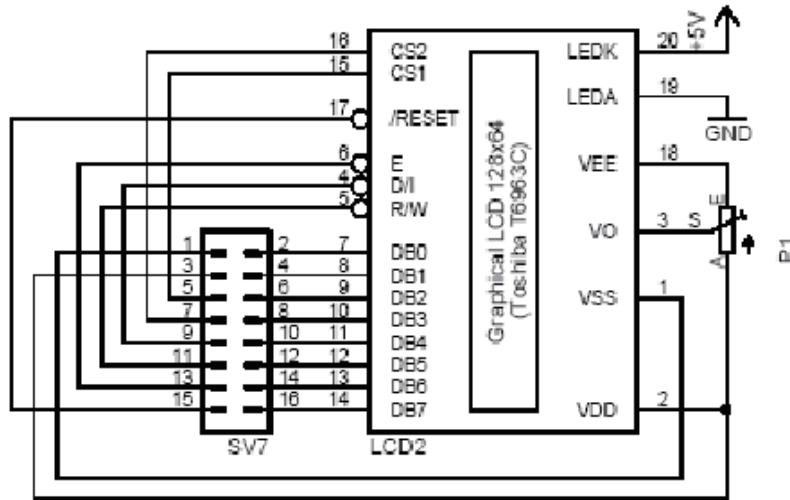
Finalmente, a la salida de ambos circuitos integrados se encuentra el conector IDC de 2x16 que reúne los contactos de conexión física con el display, según el diseño de un cable que podrá apreciarse bajo el título próximo.

 <p>Microprocesamiento modular + Conectividad</p>	Micro de 3.3V ↔ display de 5V	Nota de Aplicación
	RCM4300 + Display gráfico Powertip 128x64	CoAN-005
		Publicado: 00/00/0000
		Página 4 de 8




Ejemplo de un cable de conexión para un display Powertip PG12864

La figura muestra un esquema de ejemplo que podría utilizarse para construir un cable de conexión entre el módulo del display y el PCB donde esté conectado el módulo Rabbit. En este ejemplo se utilizaría un conector hembra IDC de 2x16. Nótese que en este ejemplo el potenciómetro para el contraste está conectado en forma directa sobre los contactos del módulo del display.



Descripción del software involucrado

Para comprobar el funcionamiento de los circuitos que acabamos de construir escribimos un sencillo programa de demostración en DC10.40 que despliega una serie de textos de muestra. El software está basado prácticamente en su totalidad en la nota de aplicación CAN-008 escrita por Sergio Caprile de Cika Electrónica, sin embargo se realizaron varias modificaciones que agregan valor al trabajo original, en el sentido de la modularidad y de la generalización, ya que se reestructuró el código generando una serie de libraries que proporcionan un grado de portabilidad mayor, para facilitar la tarea del desarrollador que desee implementar estas libraries en sus propios proyectos con mínimas o ninguna modificación.

	Micro de 3.3V ↔ display de 5V	Nota de Aplicación
	RCM4300 + Display gráfico Powertip 128x64	CoAN-005
		Publicado: 00/00/0000
		Página 5 de 8

Estructura de archivos del proyecto

Físicamente este mini-proyecto está organizado según puede apreciarse en el siguiente esquema de árbol. Esta es una organización típica para cualquier proyecto en DC, sin embargo no deja de ser una cuestión de preferencia personal en cuanto a la forma de organización de los archivos de un proyecto.

```

myProyects__CoAN-005
|__CoAN-005.c
|__CoAN-005.dcp
|__lib.dir
|__myLibs
|   |__lcd_driver.lib
|   |__lcd_graphic.lib
|   |__fonts.lib
|   |__utilities.lib

```


A continuación se explicará la naturaleza y funciones específicas de cada uno de los elementos del proyecto, para una mejor comprensión se separarán estos elementos en dos categorías:

Archivos del proyecto

- **myProyects:**
Directorio de proyectos dentro del directorio de instalación del DC. Este es un directorio creado por el usuario (por el autor de esta nota) y podrá tomar cualquier nombre, sin embargo sería conveniente mantener el nombre provisto a fines de agilizar las pruebas de quien desee reproducir este proyecto en su computadora.
- **CoAN-005:**
Directorio particular del proyecto actual.
- **myLibs:**
Directorio conteniendo las librerías particulares creadas para este proyecto, las cuales se listan bajo el próximo título.
- **lib.dir:**
Archivo de referencia de librerías para este proyecto.
- **CoAN-005.dcp:**
Archivo de proyecto del DC. Este archivo mantiene los datos de configuración particulares para este proyecto (archivo principal, lib.dir, puerto de comunicación, archivo lib.dir, etc, etc).

Archivos fuente del proyecto

- **lcd_driver.lib:**
Contiene las funciones de mas bajo nivel de acceso al display LCD.
- **lcd_graphic.lib:**
Contiene las funciones gráficas y de texto de mas alto nivel de acceso al display LCD.
- **fonts.lib:**
Contiene las tipografías utilizadas por la library **lcd_graphic.lib**
- **utilities.lib:**
Solo contiene una rutina de retardos: msDelay(int).
- **CoAN-005.c:**
Archivo de programa de la aplicación.

 CONTINEA Microprocesamiento modular + Conectividad	Micro de 3.3V ↔ display de 5V	Nota de Aplicación
	RCM4300 + Display gráfico Powertip 128x64	CoAN-005
		Publicado: 00/00/0000
		Página 6 de 8

Módulos de software del proyecto

Módulo lcd_driver.lib

Como ya se ha mencionado, esta library es la que accede al nivel del hardware para controlar el funcionamiento del display LCD. La interfaz que presenta esta library para el desarrollador se puede dividir en tres secciones:

A) Sección de definiciones de puertos:

Se encuentran en forma de directivas #define bajo las líneas de comentario:

```
// LCD control port
// LCD chip select port
// LCD control signals
```

Por supuesto, si nada se modifica en esta sección, el código original estará actuando de acuerdo a las configuraciones de hardware que describiéramos en el apartado correspondiente (Selección del pin-out...).

B) Funciones de inicialización para el programa principal:

Desde la perspectiva del programa principal, solo se debería llamar a esta función en el inicio, esta función representa a la interfaz de este módulo para una capa de alto nivel

```
void LCD_init ();
```

C) Funciones de acceso de bajo nivel para la library gráfica de alto nivel:

Las siguientes funciones constituyen la interfaz del módulo para vincularse con el módulo de más alto nivel que contiene las primitivas gráficas y de impresión de texto, si se desea mantener cierto grado de modularidad, estas funciones no deberían llamarse desde el programa principal, sino solamente desde la library gráfica de mayor nivel:

```
void LCD_PanelsOn();
void LCD_WriteData(int data);
void LCD_WriteCmd(int cmd);
int LCD_ReadData();
int LCD_Status();
int LCD_Read();
void LCD_Write(int data);
void LCD_CalcPage(unsigned int y, unsigned int *page, unsigned
int *row);
void LCD_SelSide(int side);
```

Módulo lcd_graphic.lib

Esta library, no requiere sección de definiciones, en cambio solo presenta la inclusión de las libraries que utiliza en forma subordinada mediante las directivas de compilador:


```
#use lcd_driver.lib //Low level functions
#use fonts.lib //graphics fonts
```

A estas alturas se empiezan a hacer evidentes algunas ventajas de la modularidad del código, por ejemplo, es claro que si sustituimos lcd_driver.lib por su equivalente para otro tipo de display, siempre que mantengamos la interfaz intacta, es muy probable que haya que realizar muy poco trabajo de mantenimiento adicional en el código. Lo mismo con las tipografías.

La interfaz para el desarrollador de esta library esta compuesta por las primitivas siguientes que podríamos agrupar en dos categorías:

Primitivas gráficas:

```
void LCD_home ( void );
void LCD_fill(unsigned char pattern);
void LCD_plot (unsigned int x, unsigned int y);
void LCD_dump (unsigned char *imgdata);
void LCD_line(unsigned char x1,unsigned char y1,unsigned char x2,unsigned char y2);
```

 CONTINEA <small>Microprocesamiento modular + Conectividad</small>	Micro de 3.3V ↔ display de 5V	Nota de Aplicación
	RCM4300 + Display gráfico Powertip 128x64	CoAN-005
		Publicado: 00/00/0000
		Página 7 de 8

```
void LCD_rectangle(unsigned char x1,unsigned char y1,unsigned char x2,unsigned char y2);
void LCD_print_vertical_line(unsigned int x, unsigned int startRow, unsigned int endRow);
#define LCD_clear() LCD_fill(0)
```

Funciones de texto:

```
void LCD_putchr6 ( char chr );
void LCD_putchr ( char chr );
void LCD_putchrB ( char chr , int row, int col );
void LCD_print6at ( unsigned int row, unsigned int col, char *ptr);
void LCD_printat ( unsigned int row, unsigned int col, char format, char *ptr);
```

Módulo fonts.lib

En este caso, esta library no contiene funciones, sino solamente la declaración y definición de los arrays que se almacenan en flash como constantes, y que conforman los diferentes sets de caracteres que tenemos disponibles, de esta forma podemos añadir tipografías fácilmente agregándolas en este módulo de software. Por el momento, esta library incluye los sets de caracteres siguientes:

```
font8x15[], font8x8[] y font5x7[]
```

Módulo principal


Mostraremos aquí solo el esqueleto del programa, y unas pocas líneas que ejemplifican el uso de las libraries, ya que el bloque de código que realiza las demostraciones de las funciones gráficas y/o de texto se explica por sí mismo, y por otra parte puede encontrarse una descripción mas detallada en la nota CAN-008.

```
#class auto
#use RCM43xx.LIB
//Esta es la única línea necesaria para invocar a los módulos que implementan
las //funciones gráficas y de control del display.
#use lcd_graphic.lib

//Estas definiciones son útiles para forzar el refresco periódico del estado de
//encendido de los paneles del LCD
#define LCDTRYRESTART //comment for not try restart the LCD panels
#define LCDTRYRESTARTINTERVAL 2 //refresh interval in seconds
main()
{
int i;
char numero[6];
float num,prom,vcc,sum;

LCD_init();

while(1)
{
costate
{//Finalente aquí vemos los llamado a las funciones gráficas de
//librería
LCD_clear();
LCD_print6at(0,0,"Texto en 6x8, 20 cpl");
LCD_printat(2,0,0,"Texto 8x8, 16cpl");
LCD_printat(4,0,'B',"Texto 8x16,16cpl");
// ....
//.....
//.....etc, etc....
waitfor(DelayMs(500));
}
}
```

	Micro de 3.3V ↔ display de 5V	Nota de Aplicación
	RCM4300 + Display gráfico Powertip 128x64	CoAN-005
		Publicado: 00/00/0000
		Página 8 de 8

```

//Aquí se ejecuta el modo de "refresco" periódico de los paneles de LCD
#ifdef LCDTRYRESTART
    //Patch for restart the lcd panels if they fail by noise problems
    costate
    {
        LCD_PanelsOn();
        waitfor(DelaySec(LCDTRYRESTARTINTERVAL));
    }
#endif
}

```

Referencias

Las siguientes referencias corresponden a hojas de datos consultadas para la realización de la nota de aplicación y pueden encontrarse como material adjunto de esta nota.

- **74HC245; 74HCT245**
Octal bus transceiver; 3-state
- **74LVX3245**
8-Bit Dual Supply Translating Transceiver with
3-state outputs
- **PG12864LRS-ANN-H-S0**
LCD module
- **RCM4300 User Manual**