

FORMATION SUR MAQUETTE DE DÉVELOPPEMENT PIC16FXXX - LS-PIC1



Sommaire

I-	Prése	entation et caractéristiques techniques	3		
1	- Pi	résentation :	3		
2	2- Ca	aractéristiques techniques	3		
З	3- Le	es modules principaux	4		
	3-1-	Schéma électrique de carte développement (18 broches) :	4		
	3-2-	Schéma électrique de carte développement (28 broches : 16F876) :	4		
	3-3-	Schéma électrique de carte développement (40 broches : 16F877) :	5		
	3-4-	Interface des entrées TOR :	5		
	3-5-	Interface des sorties LED :	6		
	3-6-	Interface de puissance avec relais	6		
	3-7-	Interface d'isolation galvanique (opto-coupleur TLP521-4)	7		
	3-8-	Carte LCD 1602	8		
	3-9-	Carte afficheur 7 seg	8		
	3-10-	Clavier matriciel 4 x 4	8		
	3-11-	Carte d'interfaçage RS232-USB (TX et RX)	9		
II-	Envir	onnement de programmation :	10		
1	- E	TAPES DE PROGRAMMATION :	10		
ł		TÉ N° 1: DÉCOUVERTE DE L'ENVIRONNEMENT PROGRAMMATION TEXTU	ELLE11		
ł		TÉ N° 2: COMPTEUR MODULO 10	15		
Activité N° 3: Gestion d'un afficheur LCD17					
ł	Activité n° 4: CAN - Voltmètre18				
ł		τέ Ν° 5: MLI – CONVERSIO Ν	19		
ł	Activité N° 6: Interface UART + LABVIEW				

I- Présentation et caractéristiques techniques

1- Présentation :

La valise **LS-PIC1** est une maquette de développement PIC 16Fxxx permettant l'étude sur des différentes type de microcontrôleur PIC (microchip) forme PDIP :

- a- PIC de 18 broches (16F84, 16F628, 16F88....)
- **b-** PIC de 28 broches (16F876..)
- **c-** PIC de 40 broches (16f877 ...)

Elle est principalement composée de :

- a- Trois cartes de développement (PIC 18, 28 and 40 broches)
- b- Programmateur Microchip PicKit 2
- c- Deux Interface entrées TOR (12 interrupteurs et 16 boutons poussoirs : 12 NO et 4 NC)
- **d-** Interface sorties LED
- e- Interface de puissance avec relais
- f- Interface opto-coupleurs
- g- Carte LCD 1602
- h- Carte afficheur 7 seg
- i- Carte d'entrée analogique (utilisation de module CAN)
- j- Clavier matriciel 4 x 4
- k- Carte d'interfaçage RS232-USB (TX et RX)
- I- Plaque d'essai large
- 2- Caractéristiques techniques
- > Tension d'alimentation : 220ACV±5% 50Hz
- Tension d'alimentation des circuits : +5VDC
- > Puissance de l'ensemble : 5W
- Environnement : température -10°C~+60°C Humidité<90%(25°C)</p>

3- Les modules principaux

3-1- Schéma électrique de carte développement (18 broches):



3-2- Schéma électrique de carte développement (28 broches : 16F876) :



3-3- Schéma électrique de carte développement (40 broches : 16F877) :



3-4- Interface des entrées TOR :



1010

sw?

sw

10K

SW7

R? 10K R? 10K

sw?

SW?

10K

sw

R? 10K

Gsw

GNDS



3-5- Interface des sorties LED :



3-6- Interface de puissance avec relais



Description: Relais 5v-220 10A

Un relais électronique est un interrupteur qui se commande avec une tension continue de faible puissance. La partie interrupteur sert à piloter des charges secteur de forte puissance (jusqu'à 10A couramment).

Fiche technique Model: SRD-5VDC-SL-C :

- 1. Coil voltage: DC 5V
- 2. Coil resistance: $70\Omega \sim 80\Omega$
- 3. Contact resistance: 100Ω Max
- 4. Operation time: 10msec Max
- 5. Release time: 5msec Max
- 6. Operating current: 43mA ~ 46mA
- 7. Release current: 15mA ~ 18mA

- 8. Pin: 5Pin
- 9. Insulation resistance: ≥100M (Ohm)
- 10. Between coil & contacts: AC 1500V 50HZ ~ 60HZ/min
- 11. Between contacts: AC 1000V 50HZ ~ 60HZ/min
- 12. Rated load:: 10A 250V AC / 10A 125V AC / 10A 30V DC / 10A 28V DC

Branchement du relais



ĴD?

ĴD?

χΩ?

۶D?

Le TLP521-4 est un isolateur traversant couplé optiquement dans un boîtier DIP 16 broches L'isolateur se composent de diodes émettrices de lumière infrarouge et de photo transistor NPN silicium. Cet isolateur est largement utilisé pour les terminaux informatiques, les contrôleurs de système industriel, la transmission du signal entre les systèmes de différents potentiels et impédances.

D?

- > Un ratio élevé de transfert de courant de 50%
- Haute tension d'isolement de 7,5K
- BVCEO élevé de 55V(min)

ĴD4

ĴDS

ĵD3

ົກ

ĴD6

ĴD7

ĴD8

- Reconnu UL E91231
- > Sélections personnalisée électriques disponibles
- Courant If (Forward) de 50mA
- Quad canaux



D?

۶Œ

JD?

JD?

JD?

U3 TLP521-4

Applications

> Industrie, Test et Mesure, Traitement du Signal





3-9- Carte afficheur 7 seg





3-10- Clavier matriciel 4 x 4



3-11- Carte d'interfaçage RS232-USB (TX et RX)



- II- Environnement de programmation :
- **1- ETAPE DE PROGRAMMATION :**



ACTIVITÉ N° 1: DÉCOUVERTE DE L'ENVIRONNEMENT DE PROGRAMMATION

File Edit View Project Build Run Tools Help

🔒 <u>N</u>ew Project...

Open Project...

Open Project Group...

20

Shift+Ctrl+M

Shift+Ctrl+O

🔁 🔥 - 📴 🖻

🗧 Code Explore

TEXTUELLE

Lancer le logiciel mikropascal pro et créer un nouveau projet :

1- Dans le menu principal cliquer sur « *Project* » par la suite sur « *New project* »



Cancel

Next 🗭

4 Back

9- Cocher la case « **Include All** » cette option permet d'utiliser les fonctions prédéfinies de Mikropascal

10- Cliquer sur le bouton « Next »

New Project Wizard	11. Coober la cooo - Open Edit Preiest
Step 4: You have successfully created a new project. Click "Finish" to close a wizard.	window to set Configuration bits » 12- Cliquer sur le bouton « Finish »
Checking 'Open Edit Project' option will open 'Edit Project' window after closing this wizard. This enables you to easily setup your device and project. 12 <u> Back Finish Cancel</u>	
Edit Project Oscillator Selection M INTOSC oscillator: I/O function on RA6/OSC2/CLKOUT pin, I/O fur M Watchdog Timer Disabled Disabled • Power-up Timer Disabled Disabled • Brown-out Detect • Disabled •	CU and Oscillator MCU Name P16F628A Oscillator Frequency [MHz] 12.000000 13 Heap Size 2000 onfiguration Registers CONFIG : \$2007 : 0x2118 Load Scheme Save Scheme Default 14 OK
	General Output Settings Cancel

- 13- Assurer les réglages des bits de configurations comme indiqué dans la figure précédente
- 14- Cliquer sur le bouton « OK »

start Page 🔀 📄 activite 1.mpas 🔀								
•	<pre>. program activite1; 45</pre>							
-	Var 5							
-	E0 : sbit at RA0_bit;							
-	E1 : sbit at RA1_bit;							
-	E2 : sbit at RA2_bit;							
-	E3 : sbit at RA3_bit;							
-	S0 : sbit at RB0_bit;							
-	S1 : sbit at RB1_bit;							
-	S2 : sbit at RB2_bit;							
10	53 : sbit at RB3_bit;							
-	begin							
-	TRISA:-SFF; // port A entrées							
-	TRISB:=\$F0; // RB0 a RB3 sorties							
-	CMCON:=\$07; // désactivation du comparateur							
-	while true do							
-	Begin							
-	CITE EL NOR EL							
	SITE LI AUR LZ;							
20	52 E2 AOR E3,							
20	and:							
end								
	chu.							

- 15- Saisir le programme ci-contre
- 16- Compiler le projet et vérifiez que la compilation a réussi : message «*Compiled Successfuly* » dans l'onglet messages.

III Messages 🔟 Quick Converter					
V Errors	Warnings	🛛 Hints			
Line	Message No.		Message Text		
1	1015	10	Hint: Compiling unit "C:\BacST\activite1.mpas"		
21	1010	16	Hint: Unit "activite 1.mpas" has been recompiled		
0	134		Compiled Successfully		
0	139		All files Compiled in 47 ms		

17- Lancez le logiciel « ISIS » et saisissez le schéma de simulation.



20-Bancher le programmateur PICKIT 2 au PC via le câble USB et connecteur de programmation de la valise



- 21- Lancer le logiciel PICkit 2.
- 22-Cliquer sur l'icône de Read pour détecter le type du microcontrôleur.
- 23- Dans le menu principal cliquer sur : *File... import ... fichier.hex*
- 24- Programmer le microcontrôleur PIC16F876A

	🍟 PICkit 2 Pre	ogrammer							_	
	File Device	Family I	Programmer	- Tools	View	Help				
	Midrange/SI	andard Con	figuration –							
	Device:	PIC16F8	76A		Configu	uration: 2F	FCF			
23	User IDs:	FF FF FF	FF							
	Checksum:	OFCF			OSICC4	AL:	[BandGap:		
	PICkit 2 fo	und and (connecte	:d.				MIC	ROCH	11P
	PIC Device	e Found.								
								D Target —		
	Read	Write	Verify	Erase	Bla	ank Check		Uneck /MCLR	5,0	Ť
	Program M									_
22	Enabled	Hex Only		Source:	None (Em	ipty/Erased)			
	0000	3FFF	3FFF	3F	FFF	3FFF	3FFF	3 F FF	ЗFFF	
	0008	3FFF	3 FFF	зн 24	FF F	3FFF	3FFF	3 F FF	ЗFFF	
	0010	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	
	0018	3 FFF	3FFF	3FFF	3FFF	3FFF	3 FFF	3 FFF	ЗFFF	
	0020	3FFF	3FFF	ЗFFF	3FFF	3FFF	3FFF	3 F FF	ЗFFF	

Remarque : le câble ICSP doit être connecté au PICkit2 et connecteur de programmation de microcontrôleur de cette manière :

Le fil noir au indicateur de coté PICkit2 de l'autre de partie de connecteur de programmation prés de la diode rouge



Attention il ne faut surtout pas avoir à la fois une alimentation par PICkit 2 et une alimentation externe : RISQUE DE COURT CIRCUIT ! Et DESTRUCTION de PICKIT !

ACTIVITÉ N° 2: COMPTEUR MODULO 10

On souhaite réaliser un compteur modulo 10 en utilisant la maquette de développement PIC16FXXX conformément au montage suivant :



- 1. Reliez C1 du bloc afficheur 7 segments à OV du bloc alimentation
- 2. Reliez VDD du bloc afficheur 7 segments à 5V du bloc alimentation
- 3. Reliez les entrées des segments [Da Db Dc Dd De Df Dg] respectivement aux broches RB0, RB1, RB2, RB3, RB4, RB5 et RB6 du microcontrôleur PIC16F876A.
- 4. Essayer d'utiliser le gestionnaire d'afficheur 7 segment







Programme N°1	Programme N°2
program compteur10;	program COMPTEUR2;
begin	var i : integer;
Trisb:=0;	const chiffre : array[10] of byte =(192, 249, 164,176,
portb:=0;	153, 146, 130, 248, 128,144);
while (1=1) do	begin
begin //segments [gfedcba]	Trisb:=0;
portb:=192 ; // chiffre 0	portb:=0;
delay_ms(1000); // attente d'une seconde	while (1=1) do
portb:=249; // chiffre 1	begin for i := 0 to 9 do // compteur
delay_ms(1000);	begin
portb:=164; // chiffre 2	portb:=chiffre[i]; // affichage du chiffre
delay_ms(1000);	delay_ms(1000); // attente d'une seconde
portb:=176; // chiffre 3	end;
delay_ms(1000);	end;
portb:=153; // chiffre 4	end.
delay_ms(1000);	
portb:=146; // chiffre 5	
delay_ms(1000);	
portb:=130; // chiffre 6	
delay_ms(1000);	
portb:=248; // chiffre 7	
delay_ms(1000);	
portb:=128; // chiffre 8	
delay_ms(1000);	
portb:=144; // chiffre 9	
delay_ms(1000);	
end;	
end.	

ACTIVITÉ N° 3: GESTION D'UN AFFICHEUR LCD

Soit le montage suivant :



Programme N°1
Program affichage ;
// Connections du module Lcd
var LCD_RS : sbit at PORTB.0;
var LCD_EN : sbit at PORTB.1;
var LCD_D4 : sbit at PORTB.2;
var LCD_D5 : sbit at PORTB.3;
var LCD_D6 : sbit at PORTB.4;
var LCD_D7 : sbit at PORTB.5;
var LCD_RS_Direction : sbit at TRISB.0; var LCD_EN_Direction : sbit at TRISB.1; var LCD_D4_Direction : sbit at TRISB.2; var LCD_D5_Direction : sbit at TRISB.3; var LCD_D6_Direction : sbit at TRISB.4;
begin LCD_init(); LCD_CMD(_LCD_CURSOR_OFF); while true do begin LCD_out(1,1,'FORMATION PIC'); end; end.



ACTIVITÉ N° 4: CAN - VOLTMÈTRE

On désire de réaliser un voltmètre en utilisant CAN et afficheur LCD

Programme				
program voltmetre;				
var				
N : word ;				
Cal : real ;				
T: word;				
valeur_affichage : string[3];				
LCD_RS : sbit at portc.0;				
LCD_EN : sbit at portc.1;				
LCD_D4 : sbit at portc.2;				
LCD_D5 : sbit at portc.3;				
LCD_D6 : sbit at portc.4;				
LCD_D7 : sbit at portc.5;				
LCD_RS_Direction : sbit at TRISC.0;				
LCD_EN_Direction : sbit at TRISC.1;				
LCD_D4_Direction : sbit at TRISC.2;				
LCD_D5_Direction : sbit at TRISC.3;				
LCD_D6_Direction : sbit at TRISC.4;				
LCD_D7_Direction : sbit at TRISC.5;				
begin				
adcon1:=%1000000;				
lcd_init();				
Icd_cmd(_LCD_CURSOR_OFF);				
lcd_out(1,1,'V=');				
adc_init();				
while true do				
begin				
N :=adc_read(0);				
Cal := $(N^{5000})/1023;$				
I := Word(Cal);				
word I oStr(I, valeur_affichage);				
Ica_out(1,3,valeur_atticnage);				
ica_ou((1,10, mv);				
delay_ms(100);				
ena; ena.				



ACTIVITÉ N° 5: MLI – CONVERSIO N



Programme					
program MLI_conversion;					
var					
K : byte;					
N : word ;					
begin					
PWM1_init(250); // Initialiser le module PWM1 et choix de la fréquence de PWM1= 250 Hz					
ADCON1:=\$80; // \$8E Configuration des entrées du porta comme entrées analogiques y compris					
RA0					
PWM1_start; // démarrage du module PWM1					
while true do // boucle infinite					
begin					
N:= adc_read(0); // lecture de la conversion					
K:=N/4; // calcul					
PWM1_set_duty (K); // changement du rapport cyclique : le rapport cyclique est K/ 255					
end;					
end.					

ACTIVITÉ N° 6: INTERFACE UART + LABVIEW

On désire de créer une interface en LABVIEW pour commander un afficher LCD :

VISA resource name	String to lcd			
• СОМ1	formation			
baud rate (9600)	Write Clear Stop			



Programme

program testlcd; var i : char; var LCD RS : sbit at RB0 bit; var LCD EN : sbit at RB1 bit; var LCD D4 : sbit at RB2 bit; var LCD_D5 : sbit at RB3_bit; var LCD_D6 : sbit at RB4_bit; var LCD D7 : sbit at RB5 bit; var LCD RS Direction : sbit at TRISB0 bit; var LCD EN Direction : sbit at TRISB1 bit; var LCD D4 Direction : sbit at TRISB2 bit; var LCD D5 Direction : sbit at TRISB3 bit; var LCD D6 Direction : sbit at TRISB4 bit; var LCD_D7_Direction : sbit at TRISB5_bit; begin Lcd Init(); // Initialize LCD Lcd_Cmd(_LCD_CLEAR); // Clear display Lcd_Cmd(_LCD_CURSOR_OFF); // Cursor off uart1_init (9600); // init uart delay_ms (100); while true do begin if (uart1_data_ready() <> 0) then begin i:= uart1_read(); if (i = 13) then lcd_cmd (_lcd_clear); LCD chr cp(i); end: end: end.

Formation LS-PIC1 -

