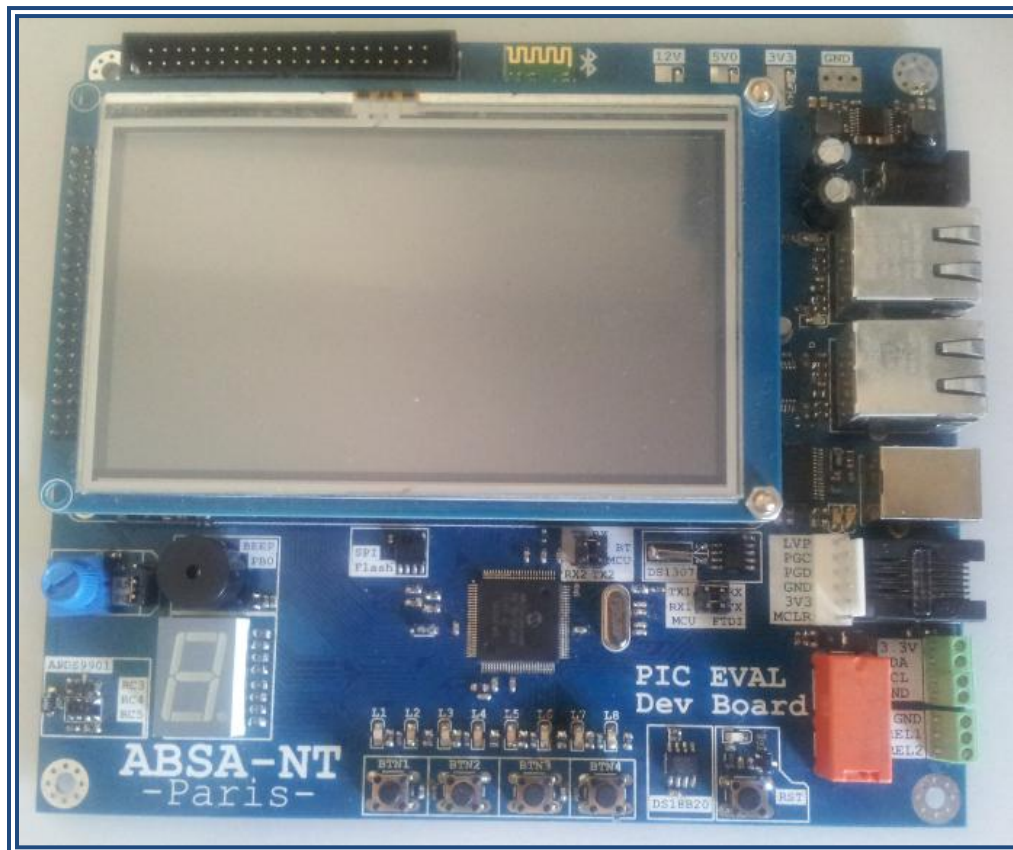




PIC EVAL Dev Board

PIC18F97J60





TP 3 :

Création d'un serveur web par plateforme PIC EVAL-ANFA

Pour répondre aux questions et justifier vos réponses, vous pouvez faire des copies d'écran ou des schémas.

I. Objectifs du TP

Le but de ce TP est de pouvoir communiquer son PC avec la carte Pic Eval via Ethernet en passant par un Serveur Web. **Ethernet** est un protocole de réseau local à commutation de paquets. C'est une norme internationale : *ISO/IEC 8802-3*.

On peut par exemple allumer les Leds en passant par un site web créé par le serveur web de la carte Pic Eval.

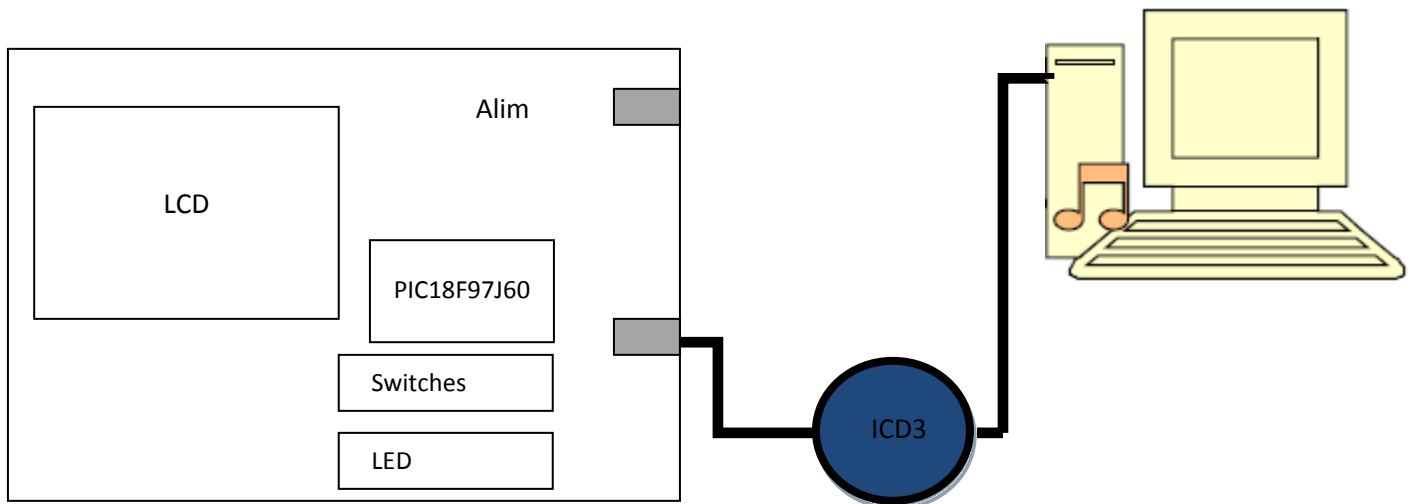
1. Matériel à votre disposition

Certains des éléments suivants sont présents sur votre plan de travail.

- Une carte Pic Eval
- Un ICD3
- Un câble USB
- Un câble d'alimentation pour la carte

2. Montage du TP

Il faut impérativement suivre les instructions du fichier 'MPLAB_Manuel.pdf' pour éviter tout dysfonctionnement.



2. Présentation de la carte PIC EVAL

La carte de développement «PIC EVAL» est un outil de développement adapté pour la programmation et l'expérimentation avec le Micro-contrôleur PIC18F97J60 de chez Microchip.

PIC EVAL intègre :

- 1 connecteur de programmation mémoire Pic.
- 1 connecteur USB pour la communication PC.
- 1 afficheur 7 segments et 8 leds.
- 1 Capteur de proximité et lumière.
- 1 module Bluetooth / Android.
- 1 microcontrôleur PIC 18F67J60.
- 1 mémoire flash 16 MB.
- 1 écran tactile touch screen.
- 1 capteur de température.
- 1 horloge temps réel RTC.
- 2 Relais.
- 1 serveur web.
- 1 liaison Ethernet Maitre.
- 1 liaison Ethernet esclave.
- 1 liaison I2C.

Ouvrir le projet TP_TCPIP(File > Open Project > TP_ TCPIP)

3. Présentation de MPLAB

1. Présentation générale

MPLAB est un Environnement de Développement Intégré (IDE) qui permet le développement logiciel des microcontrôleurs PIC et les contrôleurs de signal numériques dsPIC de la société Microchip.

MPLAB IDE permet :

- De créer le code source à l'aide de l'éditeur intégré.
- D'assembler, compiler et lier les fichiers sources qui peuvent provenir de langages différents. Un assembleur, un "linqueur" et un gestionnaire de bibliothèques sont fournis avec MPLAB. Un compilateur C est vendu à part par Microchip; des outils de tierces parties peuvent aussi être utilisés.
- De déboguer le code exécutable en observant le déroulement du programme à l'aide du simulateur fourni, de l'émulateur temps réel ICE 2000 ou de l'ICD2 (in circuit debugger) ou encore l'ICD3 développés par Microchip. Des outils de tierces parties peuvent aussi être utilisés.
- D'effectuer des mesures temporelles avec le simulateur ou l'émulateur.
- De voir les variables grâce à des fenêtres d'observation (watch windows).

2. Prise en main de Mplab (Optionnel)

Il faut impérativement avoir fait au préalable le TP1 Prise en main de l'environnement de programmation pour la carte PIC EVAL-ANFA.

4. Projet Serveur Web

1. Serveur Web

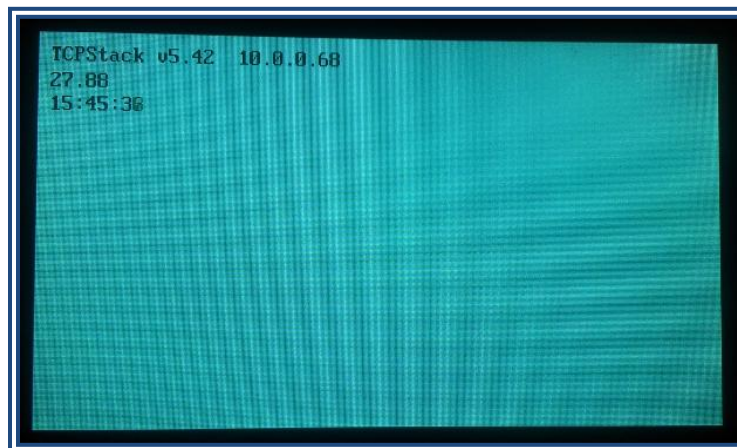
Vous aurez à votre disposition le programme qui génère le site web et le programme qui affiche sur la LCD le temps réel, l'adresse IP et le message envoyé par le PC.

Ouvrez le programme TCPIP Absa nt, compilez-le et lancez-le sur la carte (voir manuel MPLAB_Manuel). On voit sur l'écran LCD de la Pic Eval en première ligne la phrase « TCPStack » avec la version correspondante « v5.42 » et l'adresse IP du serveur de la Pic Eval.

En deuxième ligne on a la température donnée par le capteur de température DS18B20 (voir ci-dessous).



Et en troisième ligne on a l'heure du module RTC. (Voir ci-dessous).



Niveau programmation, on a plusieurs fonctions à initialiser avant toute chose ; comme par exemple :

Fonction TickInit(void) :

```
void TickInit(void){
#if defined(__18CXX)
// Use Timer0 for 8 bit processors
// Initialize the time
TMR0H = 0;
TMR0L = 0;

// Met en place l'interruption de l'horloge
INTCON2bits.TMR0IP = 0;           // Low priority
INTCONbits.TMR0IF = 0;
INTCONbits.TMR0IE = 1;           // Enable interrupt

// Timer0 on, 16-bit, internal timer, 1:256 prescaler
TOCON = 0x87;
#else
// Use Timer 1 for 16-bit and 32-bit processors
// 1:256 prescale
T1CONbits.TCKPS = 3;
// Base
PR1 = 0xFFFF;
// Clear counter
TMR1 = 0;

// Enable timer interrupt
#if defined(__C30__)
IPC0bits.T1IP = 2;           // Interrupt priority 2 (low)
IFS0bits.T1IF = 0;
IEC0bits.T1IE = 1;
#else
IPC1bits.T1IP = 2;           // Interrupt priority 2 (low)
IFS0CLR = _IFS0_T1IF_MASK;
IECOSET = _IECO_T1IE_MASK;
#endif

// Start timer
T1CONbits.TON = 1;
#endif
}
```

Fonction MPFSInit(void) :

Définit tous les MPFS en tant que fermées, et initialise l'accès à l'EEPROM si nécessaire.

```
void MPFSInit(void)
{
    BYTE i;

    for(i = 1; i <= MAX_MPFS_HANDLES; i++)
    {
        MPFSStubs[i].addr = MPFS_INVALID;
    }

    #if defined(MPFS_USE_EEPROM)
    // Initialize the EEPROM access routines.
    XEEInit();
    lastRead = MPFS_INVALID;
    #endif

    #if defined(MPFS_USE_SPI_FLASH)
    // Initialize SPI Flash access routines.
    SPIFlashInit();
    #endif

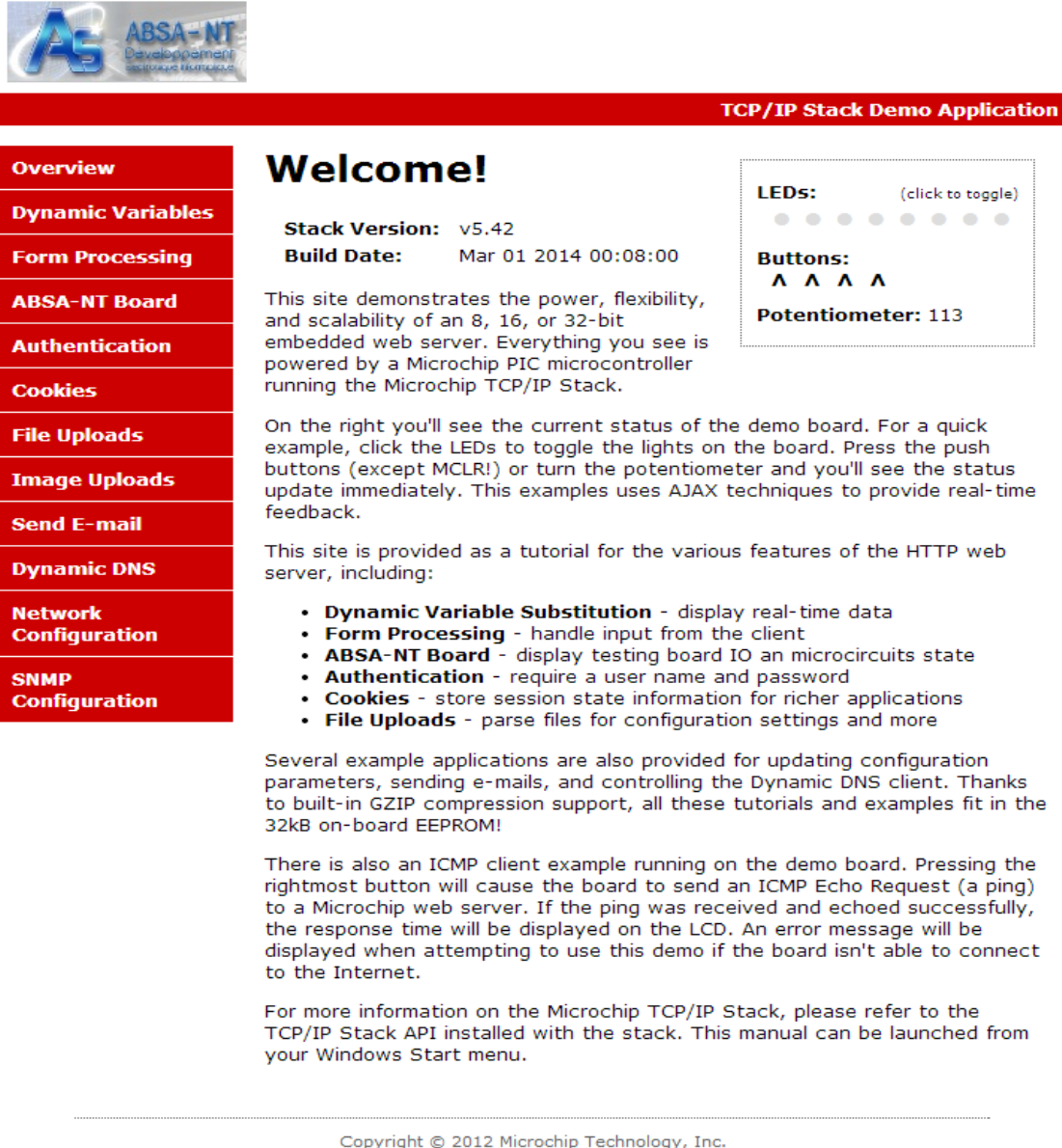
    // Validate the image and load numFiles
    _Validate();

    isMPFSLocked = FALSE;
}
}
```


a) Prise en main du site Web

Dans un premier temps on va commander les Leds via une première page du site web et afficher l'état des boutons poussoirs ainsi que la valeur du potentiomètre.

Voici une image de cette page :



TCP/IP Stack Demo Application

Overview

Dynamic Variables

Form Processing

ABSA-NT Board

Authentication

Cookies

File Uploads

Image Uploads

Send E-mail

Dynamic DNS

Network Configuration

SNMP Configuration

Welcome!

Stack Version: v5.42
Build Date: Mar 01 2014 00:08:00

This site demonstrates the power, flexibility, and scalability of an 8, 16, or 32-bit embedded web server. Everything you see is powered by a Microchip PIC microcontroller running the Microchip TCP/IP Stack.

On the right you'll see the current status of the demo board. For a quick example, click the LEDs to toggle the lights on the board. Press the push buttons (except MCLR!) or turn the potentiometer and you'll see the status update immediately. This examples uses AJAX techniques to provide real-time feedback.

This site is provided as a tutorial for the various features of the HTTP web server, including:

- **Dynamic Variable Substitution** - display real-time data
- **Form Processing** - handle input from the client
- **ABSA-NT Board** - display testing board IO an microcircuits state
- **Authentication** - require a user name and password
- **Cookies** - store session state information for richer applications
- **File Uploads** - parse files for configuration settings and more

Several example applications are also provided for updating configuration parameters, sending e-mails, and controlling the Dynamic DNS client. Thanks to built-in GZIP compression support, all these tutorials and examples fit in the 32kB on-board EEPROM!

There is also an ICMP client example running on the demo board. Pressing the rightmost button will cause the board to send an ICMP Echo Request (a ping) to a Microchip web server. If the ping was received and echoed successfully, the response time will be displayed on the LCD. An error message will be displayed when attempting to use this demo if the board isn't able to connect to the Internet.

For more information on the Microchip TCP/IP Stack, please refer to the TCP/IP Stack API installed with the stack. This manual can be launched from your Windows Start menu.

Copyright © 2012 Microchip Technology, Inc.

Dans la partie droite supérieure on voit la zone où on peut voir l'état des Leds, boutons poussoirs et potentiomètre.

Pour les Leds, lorsqu'on appuie dessus, la Led correspondante devient de couleur verte sur la page Web.

Pour les boutons poussoirs, il suffit de les appuyer sur la carte Pic Eval et la figure qui représente chaque bouton sera inversée.

Enfin dans la partie Potentiomètre, il vous affiche la valeur du potentiomètre, il suffit de le faire varier.

b) Affichage sur LCD

Dans cette partie, on va écrire du texte via le pc qui sera ensuite affiché sur l'écran LCD

Pour cela allez sur la partie « Form Processing ». Cela vous affichera la page suivante :



Overview
Dynamic Variables
Form Processing
ABSA-NT Board
Authentication
Cookies
File Uploads
Image Uploads
Send E-mail
Dynamic DNS
Network Configuration
SNMP Configuration

Form Processing

Your application will likely need to accept data from the user. Forms allow you to do just that. Forms can be submitted in one of two methods (*GET* and *POST*), and this server supports both.

The **GET** method appends the data to the end of the URI. You'll see this data following the question mark (?) in your browser's address bar. For Microchip's embedded server, this data is limited to around 80 bytes. However, this sort of submission is generally easier to process. Data sent via GET is automatically decoded, and stored in the array `curHTTP.data`. Your application will handle the data in the `HTTPExecuteGet` callback. The functions `HTTPGetArg` and `HTTPGetROMArg` provide an easy method to retrieve submitted values for processing.

As an example, this GET form controls several LEDs on the demo board:

Exercise: Modify this form to support LED 5.

The **POST** method submits the data after the request headers are sent. This allows the data to be virtually unlimited in length, as long as your application can process it in chunks. However, your application must manage the receipt of data, so it is generally more complicated.

As an example, this POST form sets the text shown on the LCD display:

Copyright © 2012 Microchip Technology, Inc.

Dans la partie « LCD : » on écrit le code qu'on veut voir affiché sur l'écran LCD de la carte Pic Eval.

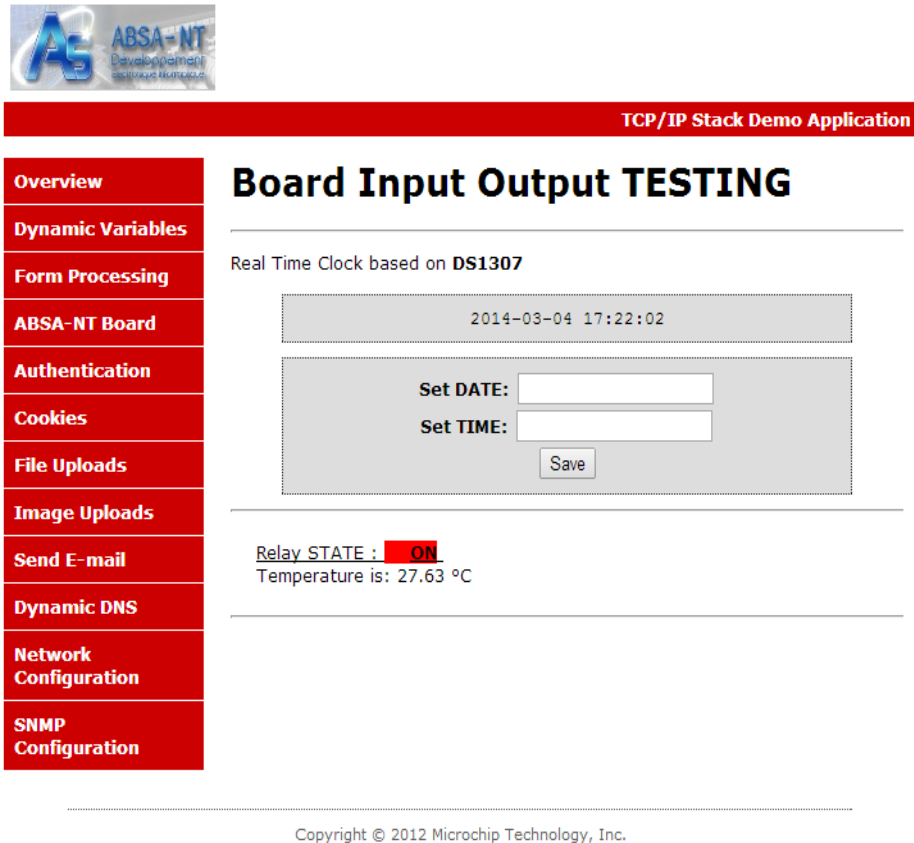
Exercice :

D'après l'exemple fait et d'après le programme, modifiez le programme pour qu'il affiche dès le reset de la carte la phrase suivante en première ligne : « Hello world », en deuxième ligne « Il fait x degrés » où x est la température et en dernière ligne afficher la date donnée en plus de l'heure par le module RTC.

c) ABSA-NT Board

Dans cette partie, on va modifier la date et l'heure du RTC et activer/désactiver le Relay.

Pour cela allez sur la partie « ABSA-NT Board ». Cela vous affichera la page suivante :



Board Input Output TESTING

Real Time Clock based on **DS1307**

2014-03-04 17:22:02

Set DATE:

Set TIME:

Save

Relay STATE : **ON**

Temperature is: 27.63 °C

Copyright © 2012 Microchip Technology, Inc.

Exercice :

Changez la date et l'heure via cette page, et au lieu d'afficher « Set DATE : » et « Set TIME : », affichez « On est le : » et « Il est : ».

Exercice :

- Modifiez la page d'accueil en ajoutant le logo de votre université dans la partie supérieure droite.
- Ajoutez une page Web, dans celle-ci vous donnerez le chiffre que vous voulez voir affiché sur le 7 segments.

