



```

char *a = NULL;
addTitleElement("Valeur des
    afficheurs 7 segments
    I2C :");
/* GET */
printf("<FORM ACTION=
    \"%s\" METHOD=POST>",
    "/cgi-bin/query-
results.cgi");
printf("<P>7 segments gauche ");
printf("<INPUTNAME=
    \"Valeur_gauche\" VALUE=0
    SIZE=\"1\"MAXLENGTH=
    \"1\"></BR>");
printf("<BR><INPUT TYPE=submit
    VALUE=\"Valid\">");
printf("<INPUT TYPE=reset
    VALUE=\"Reset\">");
printf("</FORM>");
return 0;
}

```

Comme vous pouvez le constater, cette page html présente une *EditBox* dont le but est de saisir une valeur correspondant à un chiffre (de 0 à 9). Pour permettre la lecture de la valeur entrée, un autre script est appelé : *query-result.cgi* présent dans le répertoire *user/cgihtml* :

```

int main() {
    llist entries;
    int status;
    char *p1="Valeur_gauche";
    char *first_input;
    html_header();
    html_begin("Affichage sur carte
        I2C");
    status = read_cgi_input (&entries);
    hl("<small><font color=red>Valeur
        affichee<small><font color=
        black>");
    print_entries(entries);
    html_end();
    first_input = cgi_val(entries, p1);
    html_end();
    list_clear(&entries);
    return 0;
}

```

Afin de convertir la chaîne de caractères saisie en une valeur décimale exploitable par la suite, utilisez la fonction *sscanf* :

```

int value;
sscanf(first_input,"%d", value);

```

Au final, la page web se saisit de valeur ressemble à celle-ci présentée sur la Figure 5.

Interface avec le matériel

Le but étant de pouvoir s'interfacer avec le matériel, nous allons utiliser une carte d'extension I2C qui présente un simple afficheur 7 segments à LED. Cette démonstration est avant tout ludique, et dans une réalité plus industrielle nous utiliserions d'avantage des sorties relais ou des entrées opto-couplées. Le composant utilisé est un IO-expander MCP23016 de la société *Microchip* qui permet de gérer 16 IOs en entrée comme en sortie depuis un simple bus I2C.

Il nous reste à connecter les signaux VCC (5V), GND (0V), SCL (*I2C clock*) et SDA (*I2C data*) pour pouvoir piloter la carte (Figure 6).

Nous chargeons préalablement un module dédié au pilotage du périphérique I2C du processeur :

```
insmod /home/i2c_s3c44.o
```

Voici le code qui pourra être inséré dans le script CGI afin de piloter la carte d'extension I2C :

```

unsigned char tab[] = {0xc0, 0xf9,
0xa4, 0xb0, 0x99, 0x92, 0x82, 0xf8,
0x80, 0x90};
....
fd = open("/dev/i2c0",O_RDWR);
ioctl(fd, I2C_SLAVE, 0x21);
data[0]=0x06;data[1]=0x00;
//IODIRO
write(fd, &data[0], 2);
data[0]=0x07;data[1]=0x00;
//IODIR1
write(fd, &data[0], 2);
data[0]=0x02; data[1]=0xff;
//IOLATO
write(fd, &data[0], 2);
data[0]=0x03; data[1]=0xff;
//IOLATO
write(fd, &data[0], 2);
data[0] = 0x02;
data[1] = tab[value];
write(fd, &data[0], 2);
close(fd);

```

value étant le chiffre à afficher sur le premier afficheur 7 segments par exemple.

La transmission des données au MCP-23016 se fait au travers de 2 octets : le premier correspond au numéro du registre concerné, et le second à la valeur transmise. Certains registres permettent de paramétrer les entrées/sorties, d'autres de valider ou non les pull-up internes, et enfin les derniers servent au contenu des



Concernant l'auteur

Ingénieur en électronique et informatique, *Xavier Montagne* a travaillé en France et aux États-Unis sur des projets de défense et de super-calculateurs. Spécialiste en systèmes temps réels et en Linux embarqué, il est aujourd'hui directeur technique de la société *Pragmatec*. Auteur du noyau temps réel *PICos18* et intervenant auprès d'écoles d'ingénieurs et de formations universitaires, il croit avant tout en l'efficacité des solutions GPL et Open Source.

données elles-mêmes (en entrées ou en sorties). Pour afficher sur le second afficheur (8bits de poids fort du MCP23016), il sera nécessaire de passer *data[0]* à *0x03* juste avant de transmettre la valeur.

Conclusion

Cette article vous a montré qu'il était assez simple de mettre en place un système de contrôle à distance de votre système électronique.

Vous pourrez alors utiliser des radioboutons pour fermer des électrovannes ou des relais, ou allumer des LED. Via une page HTML dédiée, vous pourrez afficher les performances et les statistiques de votre système embarqué.

Pourquoi ne pas faire de votre carte CPU un véritable répondeur enregistreur qui enregistre au format WAV les messages téléphoniques reçus ?! Une page web présenterait les fichiers archivés, que votre navigateur saurait sans difficulté jouer sur votre ordinateur distant... 📡



Sur le réseau

- Site de uClinux, version embarquée de Linux pour microcontrôleurs : <http://www.uclinux.org>
- Site officiel du serveur web BOA : <http://www.boa.org>
- La page web de Patrick Kadionik : <http://www.enseirb.fr/~kadionik/embedded/uclinux/uclinux.html>
- Liste des différents serveurs web existant sous Linux : <http://www.linux-center.org/fr/networking/servers/http-servers/index.html>
- Les excellents ouvrages O'Reilly : <http://www.oreilly.fr>