



Listing 1. Création du programme

```

#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <signal.h>
#include "can.h"
#define DEV_DEFAULT "/dev/can0"
int fd;
/*--- handler on SIGINT signal: quit with CTL-C ---*/
void sortie(int sig) {
    close(fd);
    printf("Terminated by user\n");
    exit(0);
}
int main(int argc, char**argv) {
    int n,ret;
    unsigned long i=0;
    struct canmsg_t readmsg={0,5,0,{0,}};
    struct sigaction act;
    char can_device[255];
    /*----- register handler on SIGINT signal -----
    -*/
    act.sa_handler=sortie;
    sigemptyset(&act.sa_mask);
    act.sa_flags=0;
    sigaction(SIGINT,&act,0);
    /*-----*/
    /* By default => /dev/can0 */
    if (argc > 1) strcpy(can_device, argv[1]);
    else strcpy(can_device, DEV_DEFAULT);
    if ((fd=open(can_device,O_RDWR|O_NONBLOCK)) < 0) {
        perror("open");
        printf("Error opening %s\n", can_device);
        exit(1);
    }
    printf("device opened is %s.\n", can_device);
    while (1) {
        usleep(50000);
        printf("\n");
        do {
            readmsg.flags = 0;
            ret = read(fd, &readmsg, sizeof(struct
                canmsg_t));
            if (ret < 0) {
                //perror("read");
                continue;
            } else if (ret != sizeof(struct canmsg_t)) {
                printf("Erreur read Retour=%d\n",ret);
                continue;
            } else {
                printf("Received message #%lu: id=%1X dlc=
                    %u",i,readmsg.id,readmsg.length);
                for (n = 0; n < readmsg.length ; n++)
                    printf(" %.2X", (unsigned char)
                        readmsg.data[n]);
                printf("\n");
            }
            i++;
        } while (ret >= 0);
        close(fd);
        return 0;
    }
}

```

C'est ce que nous avons fait en nous connectant à un véhicule de marque française, et nous avons finalement pu constater que lorsque le véhicule avançait plus ou moins vite, les 2 premiers octets de la trame CAN *0x101* bougeait eux aussi. Il ne nous a fallu que quelques essais pour deviner qu'il s'agissait bel et bien



Concernant l'auteur

Xavier Montagne est un ingénieur en électronique et informatique. Il 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.

de la vitesse du véhicule sur 16 bits, allant de 0 à 655 km/h (0 à 65535 en décimal sur 16 bits).

Avec un peu de patience, et surtout un logiciel programmé pour rechercher les variantes dans les trames interceptées, on arrive facilement à trouver le niveau de carburant, le nombre de tours/min, la vitesse engagée, l'action de freinage, ... bref toutes les informations utiles liées à notre véhicule et qui pourront être transmises via le réseau GPRS.

Conclusion

Une fois notre hardware mis en place et le logiciel adéquate réalisé, il devient facile de s'interfacer avec un réseau CAN quelconque. Dans le cas de notre véhicule, nous avons pu retrouver la plupart des paramètres importants, si bien qu'il devient possible d'envisager des réalisations diverses et variées comme celle de créer de toutes pièces son propres tableaux de bord numériques !

Prenez un LCD STN ou TFT couleurs, associez-y un framebuffer, ajoutez une pincée

de Nano-X, saupoudrez le tout d'un quelques lignes de code bien choisies et hop ! Vous parviendrez finalement à vos fins, et en tous les cas l'informatique embarquée dans votre voiture n'aura plus de secret pour vous ... ☺



Sur le réseau

- Site de uClinux, version embarquée de Linux pour microcontrôleurs : www.uclinux.org
- Site officiel de Microchip (MCP2551) : www.microchip.com
- Datasheet du SJA1000 : www.nxp.com/acrobat_download/datasheets/SJA1000_3.pdf
- Cours sur le bus CAN par Patrice Kadionik : <http://uuu.enseirb.fr/~kadionik/formation/canbus/canbus.html>
- Site officiel du serveur nano-X : www.microwindows.org