



Listing 1. Fichier Makefile

```
# Libraison (CROSS COMPILED)
export CC = /usr/local/arm/3.4.1/bin/arm-linux-gcc
INCLUDE_DIR = ../microwindows-0.88-nxd/src/include
CFLAGS += -I. -I$(INCLUDE_DIR)
LDFLAGS += -L../microwindows-0.88-nxd/src/lib -lnano-X -lmwengine
             -lmwdrivers -lmwfonts -lc -lm

# List of objects to compile
OBJS = demo.o
all: demo_nanox
clean:
    rm -f $(OBJS) demo_nanox *~ *.gdb
demo_nanox: $(OBJS)
    $(CC) -g -Wall $(OBJS) $(LDFLAGS) -o $@
.c.o:
    $(CC) $(CFLAGS) $< -c
```

Listing 2. Ouverture des p ripherals

```
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <demo.h>
#include <nano-X.h>

static int fd_TS;
static int fd_I2C;

int main(int argc, char **argv)
{
    SetupTouchscreen();
    SetupI2C();
    while(1) {
        usleep(100000);
    }
    close(fd_TS);
    close(fd_I2C);
}

void SetupTouchscreen(void)
{
    fd_TS = open("/dev/input/ts0", O_RDWR | O_NONBLOCK);
    if (!fd_TS) {
        printf("Can't open file /dev/input/ts0 \n");
        exit(1);
    }
    printf("TS peripheral successfully opened\n");
}

void SetupI2C(void)
{
    fd_I2C = open("/dev/i2c/0", O_RDWR);
    if (!fd_I2C) {
        printf("Can't open file /dev/i2c0 \n");
        exit(1);
    }
    printf("I2C peripheral successfully opened \n");
}
```

 cran tactile. Le noyau utilise de base le Frame Buffer, c'est- -dire qu'il r serve une partie de la RAM   la m moire vid o. Ensuite un contr leur graphique int gr  au CPU transfere les donn es vid o de la RAM vers le LCD, et cela au travers d'un acc s DMA.

Nous pouvons donc  crire sur le LCD en modifiant les data sous l'interface /dev/fb0. Toutefois, nous n'allons pas modifier l'interface vid o pixel par pixel, il nous faut passer par une librairie graphique comme Microwindows, plus connue sous le nom de Nano-X dans sa version pour Linux.

Cette librairie vous permettra de tracer des lignes ou des cercles, d'afficher des

Listing 3. Header de l'application

```
#ifndef _DEMO_NANOX_H_
#define _DEMO_NANOX_H_
#define MWINCLUDECOLORS
// Colors
#define FG_COLOR BLACK
#define BG_COLOR WHITE
// SCREEN SIZE
#define SCREEN_WIDTH 320
#define SCREEN_HEIGHT 240
// I2C
#define I2C_SLAVE 0x703
#define I2C_SLAVE_FORCE 0x706
#define I2C_RELAY 0x21
#define I2C_INPUT 0x22
void SetupScreen(void);
void SetupTouchscreen(void);
void SetupI2C(void);
void I2CWrite(int addr,
char data);
char I2CRead(int addr);
void ToggleRelay(int);
void SwitchLedON(int index);
void SwitchLedsOFF(void);
#endif
```

Listing 4. D finition des structures

```
typedef struct _widget {
    int X;
    int Y;
    int ImageId;
} widget;

typedef struct _screenSpec {
    int X;
    int Y;
    int backId;
    int RelayId_OFF;
    int RelayId_ON;
    int LedId_OFF;
    int LedId_ON;
} screenSpec;
```