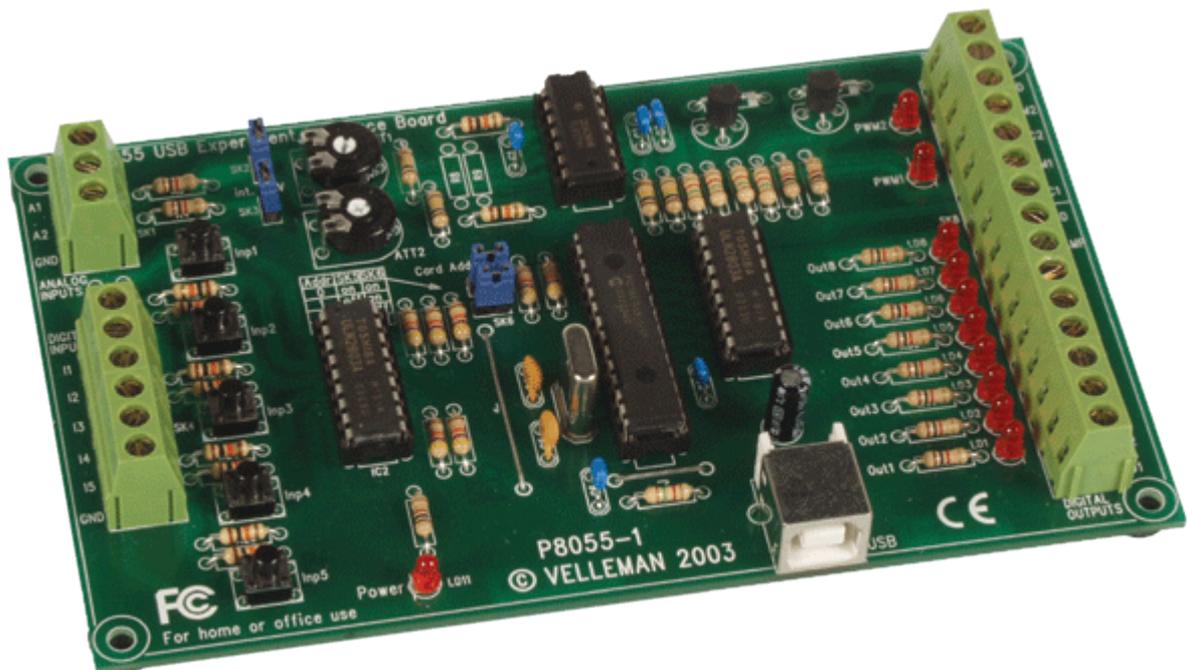


LO22

Mini Projet

Carte interface USB d'expérimentation K8055

Photo 1 : La carte K8055 à l'état final



Sommaire

LO22	1
Mini Projet	1
Carte interface USB d'expérimentation K8055	1
Sommaire	2
Introduction	3
I)La carte	4
a) Le montage de la carte.....	4
b) Les spécifications de la carte.....	5
II)Test du logiciel	5
a)Sous Windows.....	5
b)Sous linux.....	5
c)Mac OS X.....	6
III)Spécifications techniques	6
a)Améliorations apportées au logiciel.....	6
b)Syntaxe : Paramètres.....	6
c)Explication des instructions.....	7
d)Installation sous MAC OS X.....	8
e)Caractéristiques de la carte.....	10
IV)Exploitation du logiciel avec bash	10
Annexes	14

Introduction

Cette carte interface USB d'expérimentation fabriquée par Velleman, est livrée en kit à monter. Un cd est fourni avec, contenant un DLL et divers application (en C, VB, Delphi) permettant de recevoir et d'émettre des signaux (numériques ou analogiques) à partir de cette carte. Le but de ce mini projet est d'étudier le code source mis a disposition pour les plateformes linux (<http://linuxk8055.free.fr/>) de tester les possibilités de la carte sous bash, puis d'effectuer un portage sous Mac OS X (Free BSD). On essayera aussi de déterminer les limites de cette carte (temps de latence, temps de montée...).

D) La carte.

a) Le montage de la carte

On reçoit la carte proprement emballée dans un boîte plastique, les composants sont fixés dans l'ordre de montage sur une bande adhésive ce qui facilite grandement le travail. On a réalisé l'inventaire des composants, le condensateur C6 tel qu'il était spécifié dans le manuel était absent, il a été en fait remplacé par un équivalent en céramique. L'inventaire terminé, l'ensemble des composants étant présent, nous avons nous avons procédé au montage.

Photo 2 : Présentation des composants

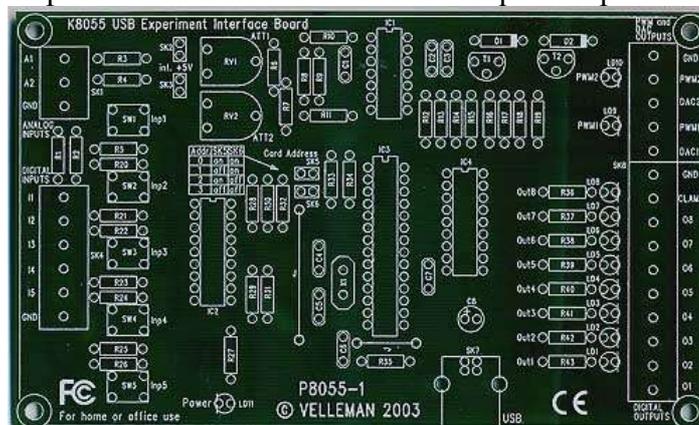


Le montage ne nécessite qu'un fer à souder, de l'étain et un petit tournevis pour le réglage des potentiomètres ainsi que pour fixer les câbles sur les borniers d'entrée et de sortie. Dans le manuel tous les composants sont listés et numérotés, de plus ce numéro ce retrouve sur le circuit imprimé.

Photo 3 : Exemple de page de manuel avec listage et numérotation des composants.

4. Condensateurs □ C1 : 100nF (104, 0,1, ut) □ C2 : 100nF (104, 0,1, ut) □ C3 : 100nF (104, 0,1, ut) □ C4 : 33pF (33) □ C5 : 33pF (33) □ C7 : 100nF (104, 0,1, ut)	6. Boutons-poussoirs □ SW1 □ SW2 □ SW3 □ SW4 □ SW5 KRS0610	8. Condensateurs □ C6 : 220nF/50V~
5. Supports de CI. Faites attention à la position! □ IC1 : 14P □ IC2 : 18P □ IC3 : 28P □ IC4 : 18P	7. Potentiomètres □ RV1 : 100K □ RV2 : 100K	9. LED. Faites attention à la polarité! □ LD1 □ LD2 □ LD3 □ LD4 □ LD5 □ LD6 □ LD7 □ LD8 □ LD9 □ LD10 □ LD11 3mm Rouge

Photo 4 : Le circuit imprimé avec la numérotation de chaque composant.



Il a nécessité environ 3 heures pour souder tous les composants. On a effectué ensuite un contrôle qualité pour vérifier qu'aucune soudure ne court-circuite le circuit imprimé ou

qu'elle ne fasse pas contact correctement. Nous n'avons relevé qu'une soudure d'une forme douteuse qui a été refaite pour lui donner une forme adéquate.

Vient ensuite l'étape critique : le premier branchement, pour des raisons de sécurité nous effectuons ce branchement à travers un HUB USB qui servirait de « fusible » en cas de problème. La DEL d'alimentation s'allume et la DEL : LD4 clignote plusieurs fois nous indiquant que tout est fonctionnel.

Sur cette carte si le signal analogique est trop faible on peut rajouter des résistances aux positions R8 et R9 permettant d'amplifier respectivement A1 et A2.

Voici le calcul pour déterminer le coefficient d'amplification (avec les résistances fournies on peut obtenir un gain de x1, x4, x15) :

Facteur de gain A1 = $1 + (R10/R8)$

Facteur de gain A2 = $1 + (R11/R9)$

b) Les spécifications de la carte

Voici les spécifications que le fournisseur annonce :

- 5 entrées numériques (0 = terre, 1= ouvert). L'appareil est pourvu de boutons de test.
- 2 entrées analogiques avec atténuation et gain optionnelles. Connexion de test interne +5V pourvue.
- 8 interrupteurs de sortie numériques à collecteur ouvert (max 50V/100mA). Indication DEL.
- 2 sorties analogiques
- 0 à 5V, résistance de sortie 1K5.
- PWM 0 à 100% sorties à collecteur ouvert
- max 100mA / 40V.
- Indications LED
- Temps de conversion moyen: 20ms par commande
- Alimentation via USB environ 70mA.

II) Test du logiciel

a) Sous Windows

Velleman fournit sur un CD divers logiciel (VB, C, Delphi) basé sur le DLL fournit sur le CD, ils fournissent aussi une documentation sur le DLL et la façon de l'implémenter à travers divers langages. Ils joignent aussi les sources de divers logiciel.

b) Sous linux

Le logiciel fournit par <http://linuxk8055.free.fr/> marche sans aucun problème et la versions que nous développons est inter opérable et marche donc sous linux sans aucun problème.

c) Mac OS X

Le premier test, en ayant simplement porté le logiciel ne fonctionne pas : il n'a pas été possible de lire et d'écrire sur le périphérique, celui-ci étant déjà utilisé par un autre driver. La cause de ce problème est le fait que le périphérique est identifié comme HID (Human Interface Device) et donc le system en prend le contrôle exclusif. Deux solutions était envisageable :

- Trouver un autre méthode pour communiquer avec la carte
- Ou faire en sorte que le noyau ne prennent plus le contrôle de la carte

Ayant des connaissance limités en USB la première méthode est infructueuse, grâce à l'aide trouver sur le forum Libusb nous avons pu concrétiser la deuxième solution. Elle consiste à placer un fichier de configuration dans les extensions du noyau avec les références du matériel et de spécifier qu'un pilote tiers se charge du contrôle de la carte (voir spécifications technique paragraphe Mac OS X).

III) Spécifications techniques

a) Améliorations apportées au logiciel

- Conversion du C++ vers le C
- Implémentation stricte du protocole.
- Affichage au format CSV
- Possibilité d'effectuer plusieurs mesures à la suite, et d'imposer un délai entre chaque mesure.
- Accé au compteur : affichage de l'état, remise à zéro, changement du debounce time
- Compatibilité avec Mac Os X
- Correction de plusieurs bugs relatifs à la selection de la carte

b) Syntaxe : Paramètres

Syntaxe :

- P:(number) Spécifie le numéro de la carte (1 à 4)
- D:(value) Spécifie la valeur de la sortie numérique codé sur 8 bits.
- A1:(value) Spécifie la valeur de la sortie analogique1 entre 0 et 255
- A2:(value) Spécifie la valeur de la sortie analogique2 entre 0 et 255
- NUM:(number) Spécifie le nombre de mesures
- DELAY:(number) Spécifie le délai minimum entre deux mesures (en msec)
- DBT1:(number) Spécifie le debounce time pour le compteur 1 (en msec)
- DBT2:(number) Spécifie le debounce time pour le compteur 2 (en msec)
- debug Affiche toutes les informations de débuggage
- reset1 Reinitialise le compteur 1

-reset2 Réinitialise le compteur 2

Exemple :k8055 -P:1 -D:147 -A1:25 -A2:203

Sortie: (temps);(digital);(analogique 1);(analogique 2);(compteur 1);(compteur 2)

Le temps correspond au nombre de millisecondes à la fin de la lecture des données depuis le début du programme.

La sortie est formatée de sorte qu'elle puisse être assimilée à un format CSV (valeurs séparées par des virgules) avec comme séparateur « ; ». Ainsi en redirigeant la sortie standard vers un fichier vous pouvez ouvrir ce dernier avec un tableur et ainsi facilement traiter vos mesures. (Utilisez pour cela les paramètres -NUM et -DELAY)

Attention le temps d'échantillonnage est en moyenne de 10ms, ainsi les signaux de plus de 10 Hz ne donneront pas de bons résultats.

c) Explication des instructions

L'ensemble des interactions USB se déroulent sur l'interface numéro 1 à travers des interruptions (sur l'endpoint 1).

Ces instructions n'ont pas été communiquées par le constructeur, mais ont été devinées par Bob Dempsey à l'aide d'un logiciel de surveillance des échanges USB.

– Lecture : La lecture se fait sur 8 octets au endpoint 1 OUT (0x81)

Valeurs :

Octet 1 : unsigned char : Digital

 bit 1 : Entrée 3

 bit 2 à 4 : Réserve

 bit 5 : Entrée 1

 bit 6 : Entrée 2

 bit 7 : Entrée 4

 bit 8 : Entrée 5

Octet 2 : Réserve

Octet 3 : unsigned char : Entrée analogique 1

Octet 4 : unsigned char : Entrée analogique 2

Octet 5 à 6 : unsigned short : Compteur 1

Octet 7 à 8 : unsigned short : Compteur 2

– Ecriture : envoi des instructions (écriture sur l'endpoint 1 IN (0x01))

L'envoi de données se fait sur 8 octets, le premier octet correspond au code de l'instruction.

– Initialisation de l'appareil

Octet 1 : unsigned char : 0x00

Octet 2 à 8 : Réserve

– Définition du debounce time pour le compteur 1

Octet 1 : unsigned char : 0x01

Octet 2 à 5 : Réserve

Octet 6 : unsigned char : Code du debounce time

Octet 7 à 8 : Réserve

- Définition du debounce time pour le compteur 2

Octet 1 : unsigned char : 0x02

Octet 2 à 6 : Réserve

Octet 7 : unsigned char : Code du debounce time

Octet 8 : Réserve

- Mise à zéro du compteur 1

Octet 1 : unsigned char : 0x03

Octet 2 à 8 : Réserve

- Mise à zéro du compteur 2

Octet 1 : unsigned char : 0x04

Octet 2 à 8 : Réserve

- Définition des sorties

Octet 1 : unsigned char : 0x05

Octet 2 : unsigned char : Valeur de la sortie digitale

Octet 3 : unsigned char : Valeur de la sortie analogique 1

Octet 4 : unsigned char : Valeur de la sortie analogique 2

Octet 5 à 8 : Réserve

d) Installation sous MAC OS X

Le logiciel pour fonctionner nécessite l'accès exclusif au périphérique ce qui est impossible sans procéder à quelques modifications de configuration (voir paragraphe sur les tests sous mac os x). Il faut placer un fichier de configuration « k8055.kext » dans le dossier / System/Library/Extensions. Ce fichier est une extension du noyau, qui spécifie que ce dernier ne doit pas charger le driver par défaut sur le périphérique. Cette extension prend en charge les quatre configurations de la carte.

Commandes à exécuter (dans le dossier où se trouve k8055.kext)

```
su
cp -r k8055.kext /System/Library/Extensions/k8055.kext
touch /System/Library
```

Puis redémarrer l'ordinateur pour que les modifications soient prises en charge

Notes :

- Le fichier k8055.kext est en fait un dossier
- Le projet xcode est fourni pour pouvoir compiler le fichier

Source :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>CFBundleDevelopmentRegion</key>
  <string>English</string>
  <key>CFBundleIconFile</key>
  <string></string>
  <key>CFBundleIdentifier</key>
  <string>com.utbm.driver.k8055</string>
```

```

<key>CFBundleInfoDictionaryVersion</key>
<string>6.0</string>
<key>CFBundlePackageType</key>
<string>KEXT</string>
<key>CFBundleSignature</key>
<string>????</string>
<key>CFBundleVersion</key>
<string>1.0.0d1</string>
<key>IOKitPersonalities</key>
<dict>
  <key>K8055 Card1</key>
  <dict>
    <key>CFBundleIdentifier</key>
    <string>com.apple.kernel.iokit</string>
    <key>IOClass</key>
    <string>IOService</string>
    <key>IOProbeScore</key>
    <integer>11000</integer>
    <key>IOProviderClass</key>
    <string>IOUSBInterface</string>
    <key>bConfigurationValue</key>
    <integer>1</integer>
    <key>bInterfaceNumber</key>
    <integer>0</integer>
    <key>idProduct</key>
    <integer>21760</integer>
    <key>idVendor</key>
    <integer>4303</integer>
  </dict>
  <key>K8055 Card2</key>
  <dict>
    <key>CFBundleIdentifier</key>
    <string>com.apple.kernel.iokit</string>
    <key>IOClass</key>
    <string>IOService</string>
    <key>IOProbeScore</key>
    <integer>11000</integer>
    <key>IOProviderClass</key>
    <string>IOUSBInterface</string>
    <key>bConfigurationValue</key>
    <integer>1</integer>
    <key>bInterfaceNumber</key>
    <integer>0</integer>
    <key>idProduct</key>
    <integer>21761</integer>
    <key>idVendor</key>
    <integer>4303</integer>
  </dict>
  <key>K8055 Card3</key>
  <dict>
    <key>CFBundleIdentifier</key>
    <string>com.apple.kernel.iokit</string>
    <key>IOClass</key>
    <string>IOService</string>
    <key>IOProbeScore</key>
    <integer>11000</integer>
    <key>IOProviderClass</key>
    <string>IOUSBInterface</string>
    <key>bConfigurationValue</key>
    <integer>1</integer>
    <key>bInterfaceNumber</key>
    <integer>0</integer>
    <key>idProduct</key>
    <integer>21762</integer>
    <key>idVendor</key>
    <integer>4303</integer>
  </dict>
  <key>K8055 Card4</key>
  <dict>
    <key>CFBundleIdentifier</key>
    <string>com.apple.kernel.iokit</string>
    <key>IOClass</key>
    <string>IOService</string>
    <key>IOProbeScore</key>
    <integer>11000</integer>
    <key>IOProviderClass</key>
    <string>IOUSBInterface</string>
    <key>bConfigurationValue</key>
    <integer>1</integer>
    <key>bInterfaceNumber</key>
    <integer>0</integer>
    <key>idProduct</key>

```

```

        <integer>21763</integer>
        <key>idVendor</key>
        <integer>4303</integer>
    </dict>
</dict>
<key>OSBundleLibraries</key>
<dict>
    <key>com.apple.iokit.IOUSBFamily</key>
    <string>1.8</string>
</dict>
</dict>
</plist>

```

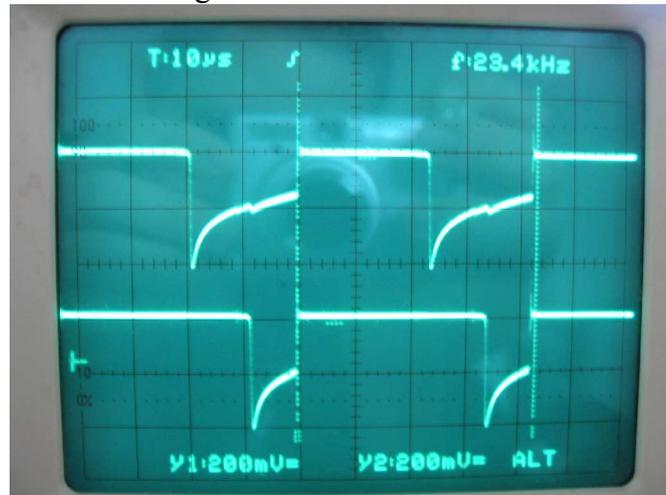
e) Caractéristiques de la carte

Le contrôleur utilisé par la carte est PIC16C745-I/SP. Cette puce est équipée d'une mémoire programmable de 8 Ko, et d'une mémoire vive de 256 octets. Il devrait être possible de modifier le programme de la puce afin d'obtenir de meilleurs résultats, ou proposer de nouvelles fonctionnalités.

La documentation de la puce nous indique que l'on peut choisir un quartz de 6 ou 24 Mhz. On en déduit qu'en remplaçant le quartz original par un quartz de 24 Mhz on obtiendrait des meilleures fréquences de lecture et d'écriture améliorées.

Le signal analogique est générée grâce à un signal PWM. La sortie analogique correspondant à chaque PWM passe en fait par un circuit qui en fait la moyenne obtenant une tension continu.

Photo 5 : oscillogramme de deux signaux PWM.



Le signal en haut correspond à 50 % (128)

Le signal du bas correspond à 75% (192)

Remarque : Le premier signal est perturbé par le passage de 0 à -400mv du second.

IV) Exploitation du logiciel avec bash

Nous avons créé sous bash avec ncurses un programme comprenant certaines fonctionnalités que le logiciel proposé par Velleman sur le Cd-rom joint avec la carte. En effet notre programme permet un affichage numérique et graphique (vumètre) des entrées analogiques, ainsi que des valeurs des entrées numériques (case cochée ou non).

Cependant à chaque boucle le programme en C doit procéder à la connexion vers l'appareil, ce qui sous mac os x prends énormément de temps (½ seconde). Ceci ajouté au temps d'affichage, rends les performance du programme très limitées.

Source :

```
# Affiche une barre de progression
# Arguments : (y) (titre) (valeur)
#   y : ligne du terminal à partir de la quelle sera affichée la barre
#   titre : Titre affiché au dessus de la barre
#   valeur : valeur à afficher comprise entre 255 et 0
function AfficheBarre ()
{
    # Calcule la largeur effective de la barre à l'écran
    largeur=$((($3 * $LargeurEcran) / 255 )

    #Positionne le curseur
    tput cup $1 0

    #Affiche le titre
    echo "$2 : $3      "

    # Affiche la partie pleine de la barre
    for i in $(seq 1 $largeur); do
        echo -n "#"
    done

    #Affiche la partie vide de la barre (sert à effacer l'ancienne position)
    for i in $(seq $largeur $($LargeurEcran-1)); do
        echo -n " "
    done
}

# Affiche l'état des bits d'une valeurs numériques (5 bits affichés)
# Arguments : (valeur)
#   valeur : valeur à afficher
function AfficheValeurNumeric () {
    d=$1

    echo "Numeric"

    if [ "$d" -ge 16 ]; then
        echo -n "[#]"
        d=$((d-16))
    else
        echo -n "[ ]"
    fi

    if [ "$d" -ge 8 ]; then
        echo -n "[#]"
        d=$((d-8))
    else
        echo -n "[ ]"
    fi

    if [ "$d" -ge 4 ]; then
        echo -n "[#]"
        d=$((d-4))
    else
        echo -n "[ ]"
    fi

    if [ "$d" -ge 2 ]; then
        echo -n "[#]"
        d=$((d-2))
    else
        echo -n "[ ]"
    fi

    if [ "$d" -ge 1 ]; then
        echo -n "[#]"
        d=$((d-1))
    else
        echo -n "[ ]"
    fi
}
```

```

}

# Boucle générale du programme
function Boucle ()
{
    while true ; do

        # Lit les données grace au programme
        donnees=$(./k8055)

        # Sépare les valeurs en utilisant cut
        d=$(echo $donnees | cut -f2 -d";")
        a1=$(echo $donnees | cut -f3 -d";")
        a2=$(echo $donnees | cut -f4 -d";")
        c1=$(echo $donnees | cut -f5 -d";")
        c2=$(echo $donnees | cut -f6 -d";")

        # Affiche la barre pour l'entrée analogique 1
        AfficheBarre 4 "Analog 1" "$a1"

        # Affiche la barre pour l'entrée analogique 2
        AfficheBarre 6 "Analog 2" "$a2"

        # Affiche l'état des compteurs
        echo "Counter 1 : $c1"
        echo "Counter 2 : $c2"

        # Affiche l'état de l'entrée numérique
        AfficheValeurNumeric $d

        # Lit l'entrée clavier
        read
        case $REPLY in
            q) return # Quitte le programme si l'entrée est "q"
              ;;
            *)
              ;;
        esac

    done
}

# Intitialise le terminal pour le programme
function initialiser ()
{
    # Mémoire la largeur du terminal
    LargeurEcran=$(tput cols)

    # Initialise
    if !tput init ; then
        echo "Impossible d'initialiser le terminal"
        exit 1
    fi

    # Defini le mode de saisie sur l'entrée standart :
    # temps d'attente = 0sec, 0 caractères minimum à lire
    stty -icanon -echo time 0 min 0

    # Nettoie le terminal
    tput clear

    # Cache le curseur
    tput civis
}

# Restaure les paramètres par défaut du terminal
function deinitialiser ()
{
    # Remet les paramètres par défaut de l'affichage du terminal
    tput reset

    # Remet les paramètres par défaut de saisie du terminal
    stty icanon echo

    # Nettoie le terminal
    clear
}

initialiser

```

```
Boucle  
deinitialiser
```

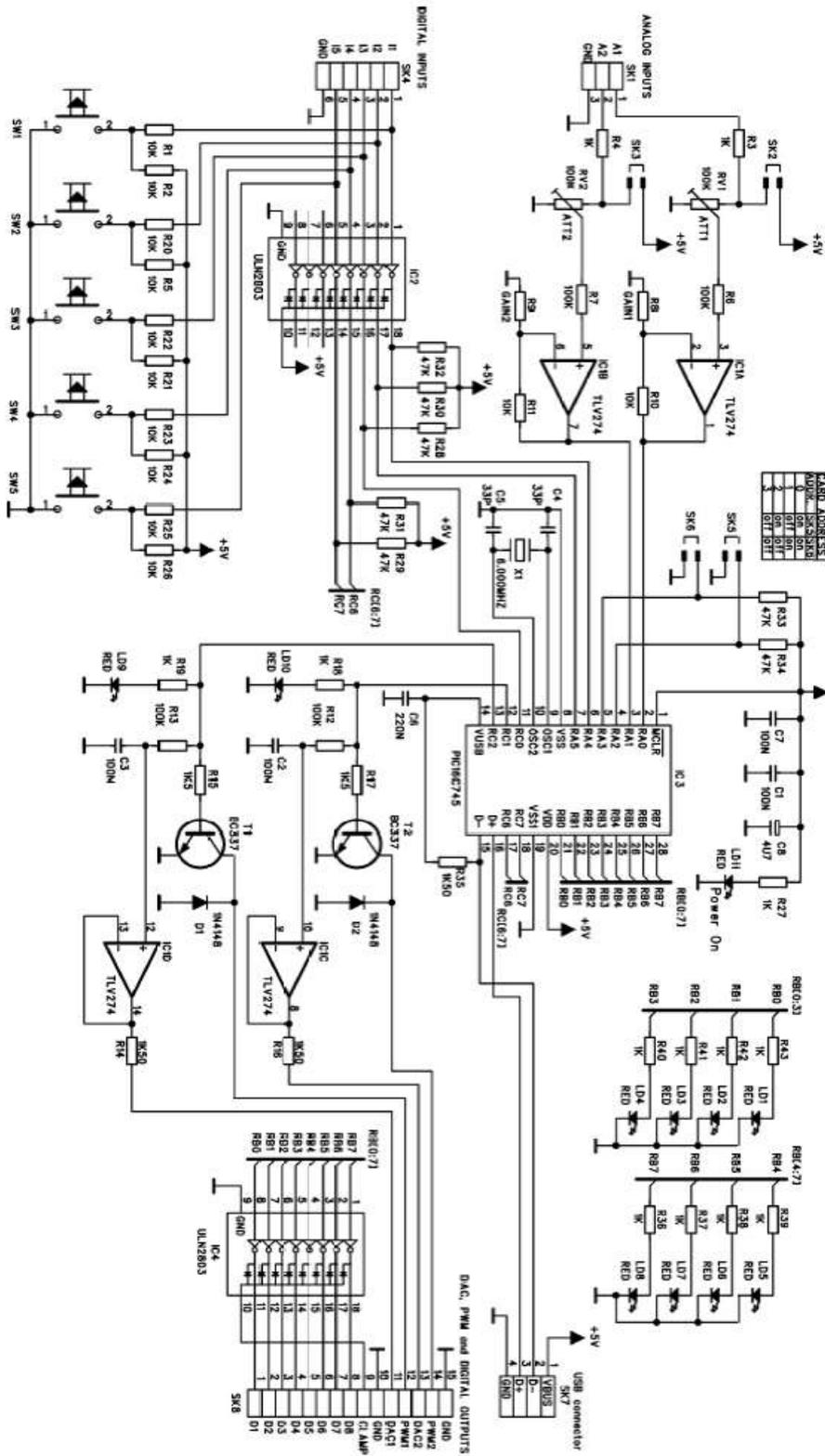
Petit exemple pour faire clignoter les diodes sur la carte

Source :

```
# allumage successif de chaque diode  
for i in 0 1 2 4 8 16 32 64 128; do  
    ./k8055 -D:$i  
done  
  
# compteur numérique de 0 à 255  
for i in $(seq 0 255); do  
    ./k8055 -D$i  
done
```

Annexes

Annexe 1 : Schéma de raccordement



Annexe 1 : Résumé des fonctionnalités de la carte.

