

KONSTRUKČNÍ NÁVOD + NÁVOD K OBSLUZE

FKtechnics[®]

CONRAD
partner

velleman[®]-kit HIGH-Q 

Stavebnice experimentální karty USB "INTERFACE BOARD - K8055"

Obj. č.: 19 10 03



Tento montážní návod a návod k obsluze je součástí výrobku. Obsahuje důležité pokyny k uvedení do provozu a k obsluze. Jestliže výrobek předáte nebo prodáte jiným osobám, dbějte na to, abyste jim odevzdali i tento návod k obsluze.

Ponechte si proto tento návod k obsluze, abyste si jej mohli znovu kdykoliv přečíst!

CONRAD
ELEKTRONIKA. TECHNIKA. TRADICE.

Obsah

	Strana
Charakteristické znaky zařízení (technické údaje)	3
Minimální požadavky na operační systém a počítač	3
1. Montáž stavebnice – osazení desky (dříve než začnete)	4
1.1 NÁSTROJE A PŘÍSTROJE, KTERÉ BUDETE K MONTÁŽI POTŘEBOVAT:	4
1.2 DŮLEŽITÉ POKYNY K PROVEDENÍ MONTÁŽE (OSAZENÍ DESKY):	5
1.3 SPRÁVNÉ PROVÁDĚNÍ OSAZOVÁNÍ A PÁJENÍ SOUČÁSTEK	5
BAREVNÁ OZNAČENÍ SOUČÁSTEK (HODNOTY + TOLERANCE).....	6
2. Kusovník součástek	6
1. DRÁTĚNÉ MŮSTKY	6
3. DIODY	6
2. REZISTORY (ODPORY)	6
4. KERAMICKÉ KONDENZÁTORY	7
5. PATICE INTEGROVANÝCH OBVODŮ	7
6. TLAČÍTKA	7
7. POTENCIOMETRY (TRIMRY)	7
8. KONDENZÁTOR	7
9. SVÍTIVÉ DIODY (LED).....	7
10. TRANZISTORY	7
11. USB-KONEKTOR	7
12. ELEKTROLYTICKÝ KONDENZÁTOR	8
13. KŘEMENNÝ KRYSTAL	8
14. KOLÍKOVÁ LIŠTA (VÝVODY).....	8
15. ŠROUBOVÉ SVORKY (KONEKTORY)	8
16. INTEGROVANÉ OBVODY (IC)	9
17. ZVÝŠENÍ VSTUPNÍHO NAPĚTÍ – ZESÍLENÍ ZISKU (REZISTORY R8 A R9)	9
18. PRYŽOVÉ NOŽIČKY	10
3. Vstupy a výstupy na desce s tištěnými spoji	11
Popis vstupů a výstupů na desce s tištěnými spoji	12
4. Zvolení správné adresy v testovacím programu	12
5. Instalace software	13
6. Provedení otestování karty pomocí software	13
7. Schéma zapojení	15
8. Plán osazení desky s tištěnými spoji součástkami	16
9. Úvod do programování a popis souboru K8055.DLL	17
PŘEHLED FUNKCÍ SOUBORU „K8055D.DDL“	18
POUŽITÍ SOUBORU „K8055D.DDL“ V PROGRAMOVACÍM JAZYCE „DELPHI“	26
POUŽITÍ SOUBORU „K8055D.DDL“ V PROGRAMOVACÍM JAZYCE „VISUAL BASIC“	27
POUŽITÍ SOUBORU „K8055D.DDL“ V PROGRAMOVACÍM JAZYCE „C++ BUILDER“	28

Charakteristické znaky zařízení (technické údaje)

- 5 digitálních vstupů [0 = GND (kostra), 1 = otevřený kontakt], testovací tlačítka na desce s tištěnými spoji.
- 2 analogové vstupy s možností tlumení a zesílení (interní test + 5 V).
- 8 digitálních spínacích výstupů s otevřeným kolektorem (max. 50 V / 100 mA). Indikace pomocí svítivých diod (LED) na desce s tištěnými spoji.
- 2 analogové výstupy:
 - 0 až 5 V, výstupní odpor 1,5 k Ω
 - Modulace šířkou impulsů (PWM) 0 až 100 % - výstup s otevřeným kolektorem
 - Max. 100 mA / 40 V
- Indikace pomocí svítivých diod (LED) na desce s tištěnými spoji.
- Doba trvání konverze: 20 ms na každý povel.
- Odběr proudu (napájení) přes USB: cca 70 mA
- Stavebnici dodáváme s diagnostickým software a programem „DLL“ pro komunikaci.

Minimální požadavky na operační systém a počítač

- CPU Pentium
- USB 1.0 nebo vyšší verze
- Windows 98 a vyšší verze (Win NT nelze použít)
- Jednotka CD-ROM a myš

Upozornění:

Počet vstupů a výstupů lze samozřejmě zvýšit, použijete-li až (max.) 4 karty, které připojíte k USB-konektorům počítače. Všechny komunikační programy jsou uloženy do souboru „K8055D.DLL“ (Dynamic Link Library).

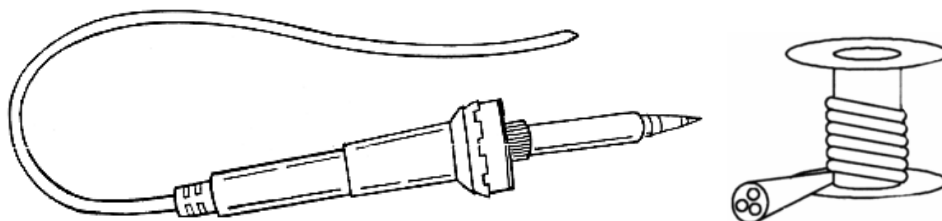
Pro další komunikaci si můžete napsat vlastní aplikační programy podporující „DLL“, například v „Delphi“, „Visual Basic“, „C++ Builder“ atd.

1. Montáž stovebnice – osazení desky (dříve než začnete)

Přečtěte si pozorně následující pokyny, které Vám pomohou sestavit tuto stovebnici.

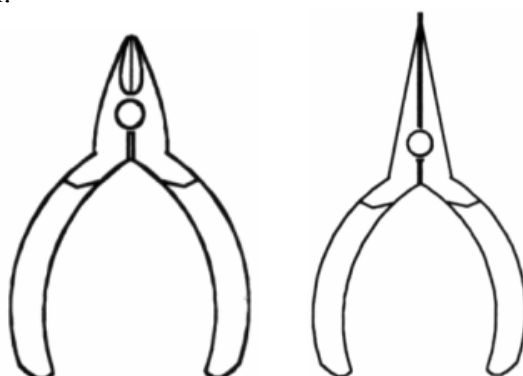
1.1 Nástroje a přístroje, které budete k montáži potřebovat:

- Malou páječku o příkonu 25 až max. 40 W s jemným pájecím hrotem a cínovou pájku 1 mm (trubičkový cín s kalafunou bez pájecího tuku).

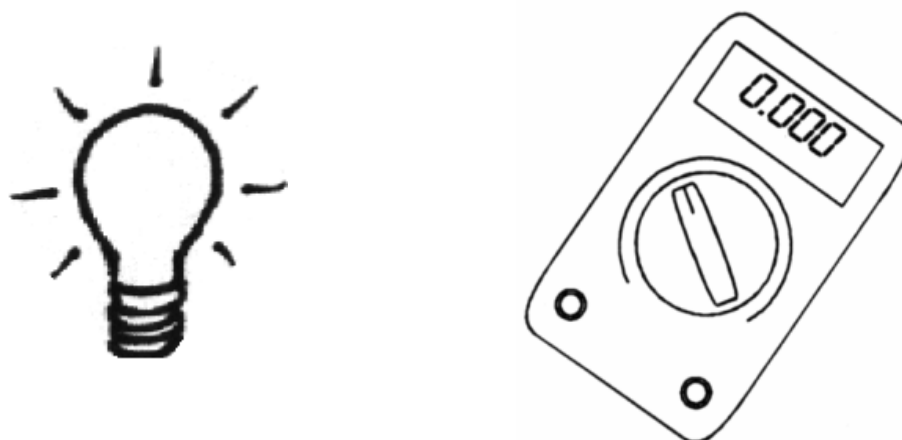


Otírejte pájecí hrot páječky pravidelně mokrým hadříkem nebo o navlhčenou houbičku a udržujte jej stále čistý. Naneste nejdříve malé množství cínu na pájecí hrot. Tímto ochráníte pájecí hrot před znečištěním a provedete snadno příslušná spojení. Pokud by cín s pájecího hrotu odkapával, očistěte jej.

- Malé štípací kleště k odstrižení přebytečných a přečnivajících vývodů (kabelů) a dále malé ploché (jehlové) kleště (pinzety) k ohýbání vývodů a k přidržování součástek při pájení.



- Malé ploché a křížové šroubováky.
- Dobré osvětlení a vhodný měřicí přístroj (digitální multimetr).

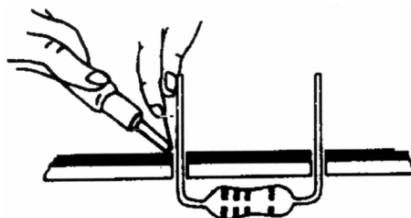


1.2 Důležité pokyny k provedení montáže (osazení desky):

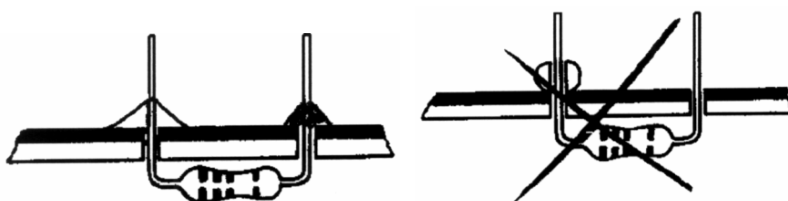
- Pokud nemáte dostatek zkušeností s pájením, svěřte tuto činnost raději odborníkovi, abyste nebyli zklamáni.
- Dodržujte bezpodmínečně všechny pokyny, které jsou uvedeny v tomto návodu k montáži. Osazení desky s tištěnými spoji provádějte v pořadí popsáném v tomto návodu k montáži.
- Osadte desku s tištěným spoji součástkami podle příslušných vyobrazení.
- Dejte pozor na to, že některé hodnoty součástek mohou být nepatrně odlišné (než je uvedeno na příslušných vyobrazeních).

1.3 Správné provádění osazování a pájení součástek

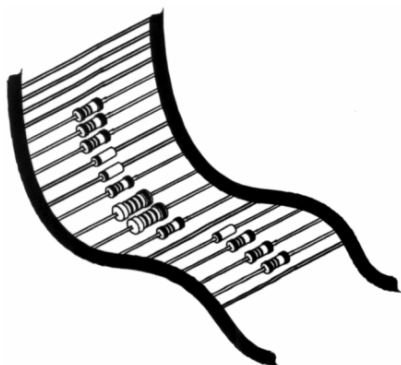
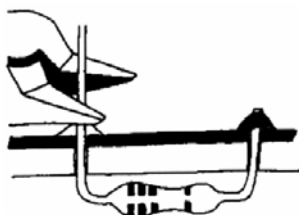
1. Narovnejte případné ohnuté vývody součástek a zatlačte je opatrně do příslušných otvorů na desce s tištěnými spoji. Nyní proveďte připájení vývodů.



2. Připájení vývodů musí být čisté (lesklé) a musí mít kuželovitý tvar.



3. Přechňávající konce vývodů odštípněte (odštípněte) štípacími kleštičkami.

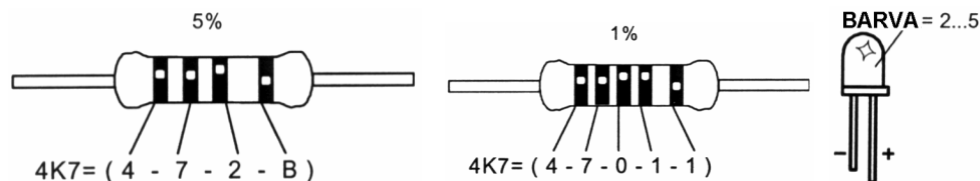


Upozornění:

Mnohé součástky s axiálními vývody jsou sestaveny ve správném pořadí ve svazcích. Toto uspořádání Vám má usnadnit jejich výběr a následné osazení desky a snížit tím i riziko případných chyb.

Při osazování desky s tištěnými spoji vyndávejte součástky z těchto svazků jednu po druhé.

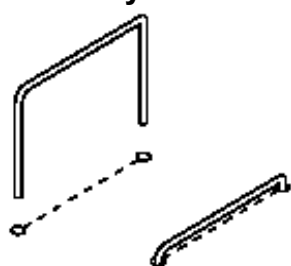
Barevná označení součástek (hodnoty + tolerance)



0 = černá; 1 = hnědá; 2 = červená; 3 = oranžová; 4 = žlutá; 5 = zelená; 6 = modrá;
7 = fialová; 8 = šedá; 9 = bílá; A = stříbrná; B = zlatá

2. Kusovník součástek

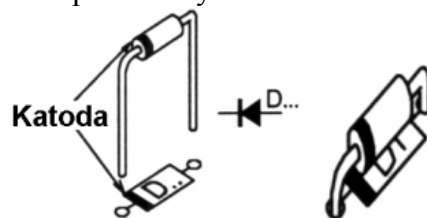
1. Drátěné můstky



J: (2 x)

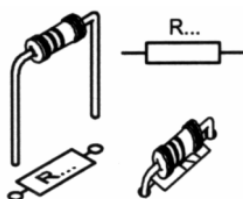
3. Diody

Pozor na polaritu vývodů!



D1: 1N4148; D2: 1N4148

2. Rezistory (odpory)

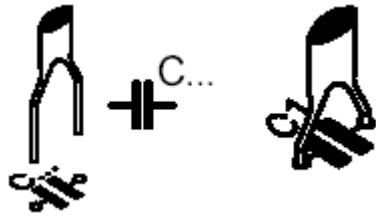


R1: 10K (1-0-3-B); R2: 10K (1-0-3-B); R3: 1K (1-0-2-B); R4: 1K (1-0-2-B);
R5: 10K (1-0-3-B); R6: 100K (1-0-4-B); R7: 100K (1-0-4-B); R10: 10K (1-0-3-B);
R11: 10K (1-0-3-B); R12: 100K (1-0-4-B); R13: 100K (1-0-4-B); R14: 1K5 (1-5-2-B);
R15: 1K5 (1-5-2-B); R16: 1K5 (1-5-2-B); R17: 1K5 (1-5-2-B); R18: 1K (1-0-2-B);
R19: 1K (1-0-2-B); R20: 10K (1-0-3-B); R21: 10K (1-0-3-B); R22: 10K (1-0-3-B);
R23: 10K (1-0-3-B); R24: 10K (1-0-3-B); R25: 10K (1-0-3-B); R26: 10K (1-0-3-B);
R27: 1K (1-0-2-B); R28: 47K (4-7-3-B); R29: 47K (4-7-3-B); R30: 47K (4-7-3-B);
R31: 47K (4-7-3-B); R32: 47K (4-7-3-B); R33: 47K (4-7-3-B); R34: 47K (4-7-3-B);
R35: 1K5 (1-5-2-B); R36 až R43: 1K (1-0-2-B)



Neosazujte prozatím desku s tištěnými spoji rezistory R8 a R9.
K dispozici zůstávají ještě 4 neosazené rezistory 2 x R8 a 2 x R9.

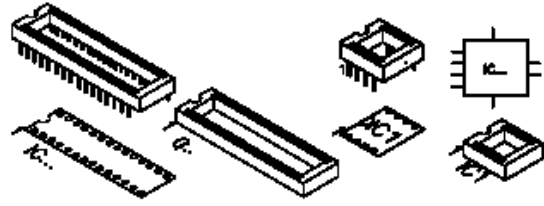
4. Keramické kondenzátory



C1: 100 nF (104); C2: 100 nF (104);
C3: 100 nF (100); C4: 33 pF (33);
C5: 33 pF (33); C7: 100 nF (104);

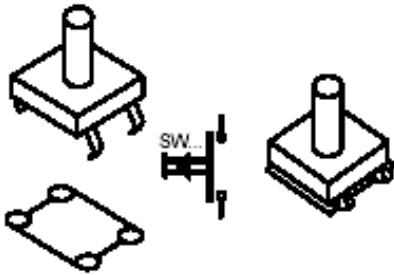
5. Patice integrovaných obvodů

Pozor na polohu vývodů!



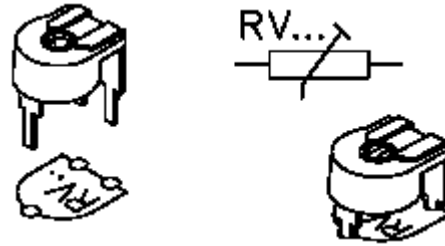
IC1: 14P (14 vývodů); IC2: 18P;
IC3: 24P (88 vývodů); IC4: 18P

6. Tlačítka



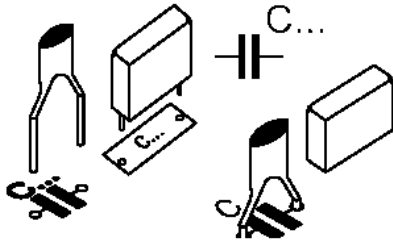
SW1 až SW5: KRS0610

7. Potenciometry (trimry)



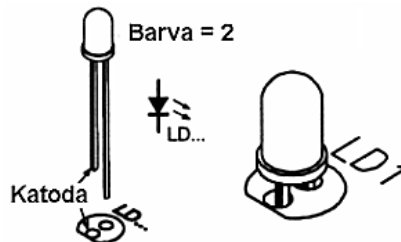
RV1 a RV2: 100 k

8. Kondenzátor



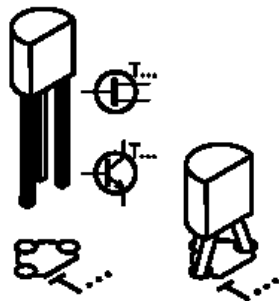
C6: 220 nF / 50 V

9. Svítivé diody (LED)



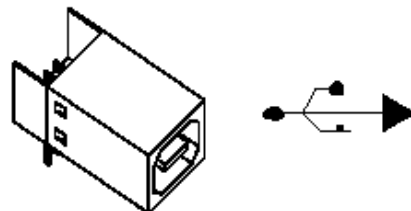
LD1 až LD11: 3 mm, červená (2);

10. Tranzistory



T1: BC337; T2 : BC337

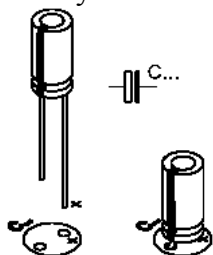
11. USB-konektor



SK7: USBB90

12. Elektrolytický kondenzátor

Pozor na polaritu vývodů!



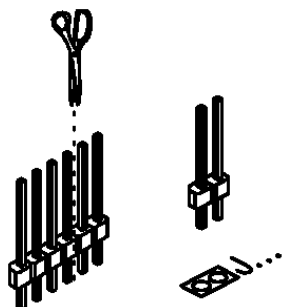
C8: 4 μ 7 (4,7 μ F / 50 V)

13. Křemenný krystal



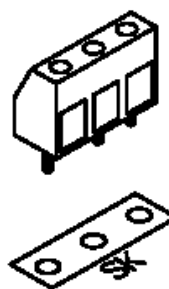
X1: 6 MHz

14. Kolíková lišta (vývody)



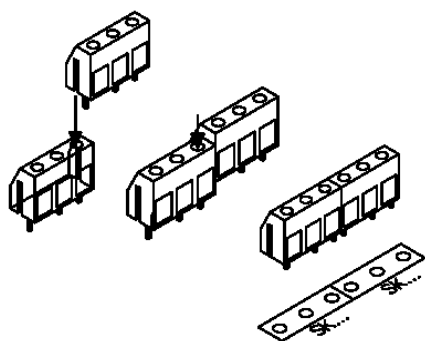
SK2 až SK6: 2P (2 vývody)

15. Šroubové svorky (konektory)



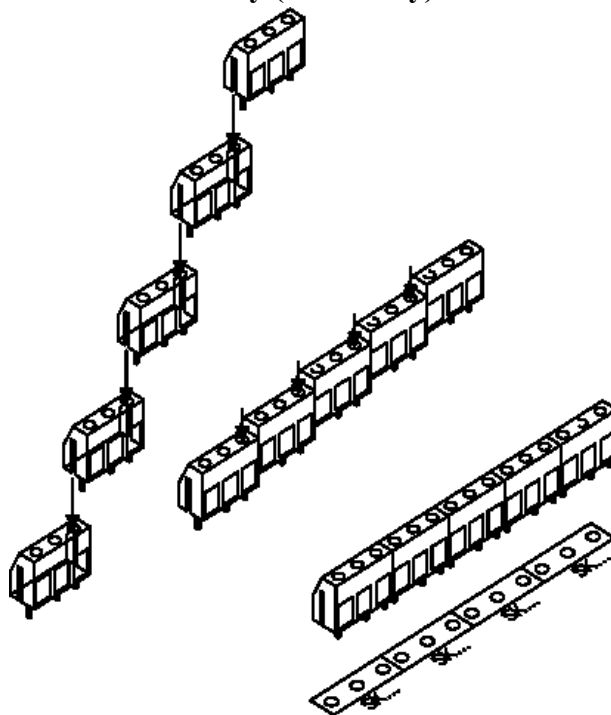
SK1: 3 P (vývody)

15. Šroubové svorky (konektory)



SK4: 2 x 3 P (vývody)

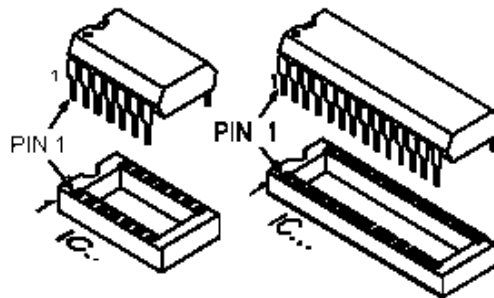
15. Šroubové svorky (konektory)



SK8: 5 x 3 P (vývody)

16. Integrované obvody (IC)

Pozor na polohu vývodů (pinů)!



IC1: TLV274IN

IC2: ULN2803

IC3: VK8055 = programový mikrořadič (μ C): (PIC16C745-IP)

IC2: ULN2803

17. Zvýšení vstupního napětí – zesílení zisku (rezistory R8 a R9)

Pokud bude vstupní napětí příliš nízké, můžete toto napětí zvýšit „2 x“, „4 x“ nebo až „15 x“.

K 4-násobnému zvýšení vstupního napětí použijte pro vstupní signál 1 rezistor [R8] 3k3 (3,3 k Ω), pro vstupní signál 2 rezistor [R9] 3k3 (3,3 k Ω).

K 15-násobnému zvýšení vstupního napětí použijte pro vstupní signál 1 rezistor [R8] 820E (820 Ω), pro vstupní signál 2 rezistor [R9] 820E (820 Ω).

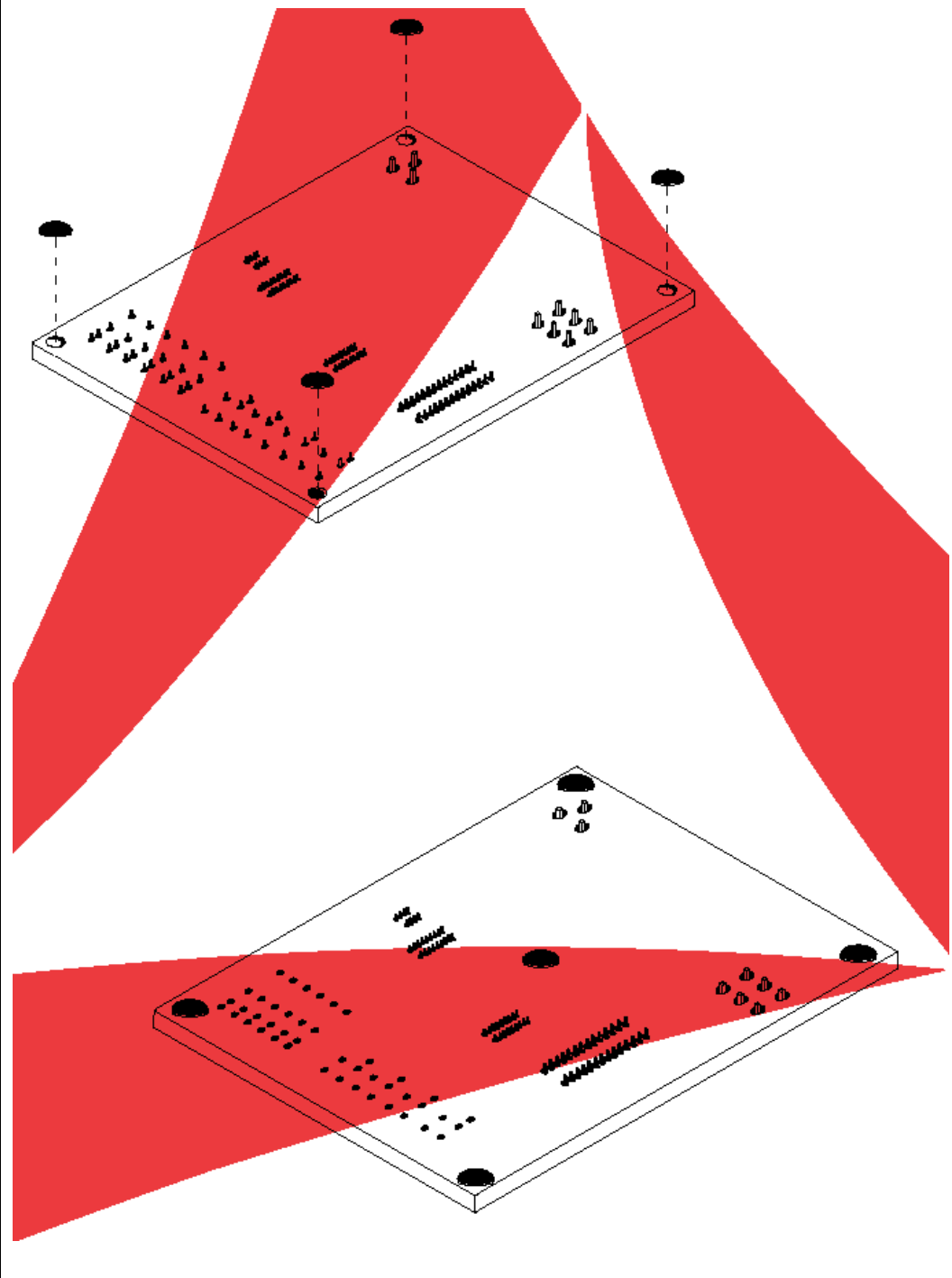
Budete-li potřebovat jinou úroveň zesílení zisku na jednotlivých vstupech, použijte k výpočtu následující rovnice:

Zesílení zisku vstupu 1: $A_1 = 1 + (R_{10} / R_8)$

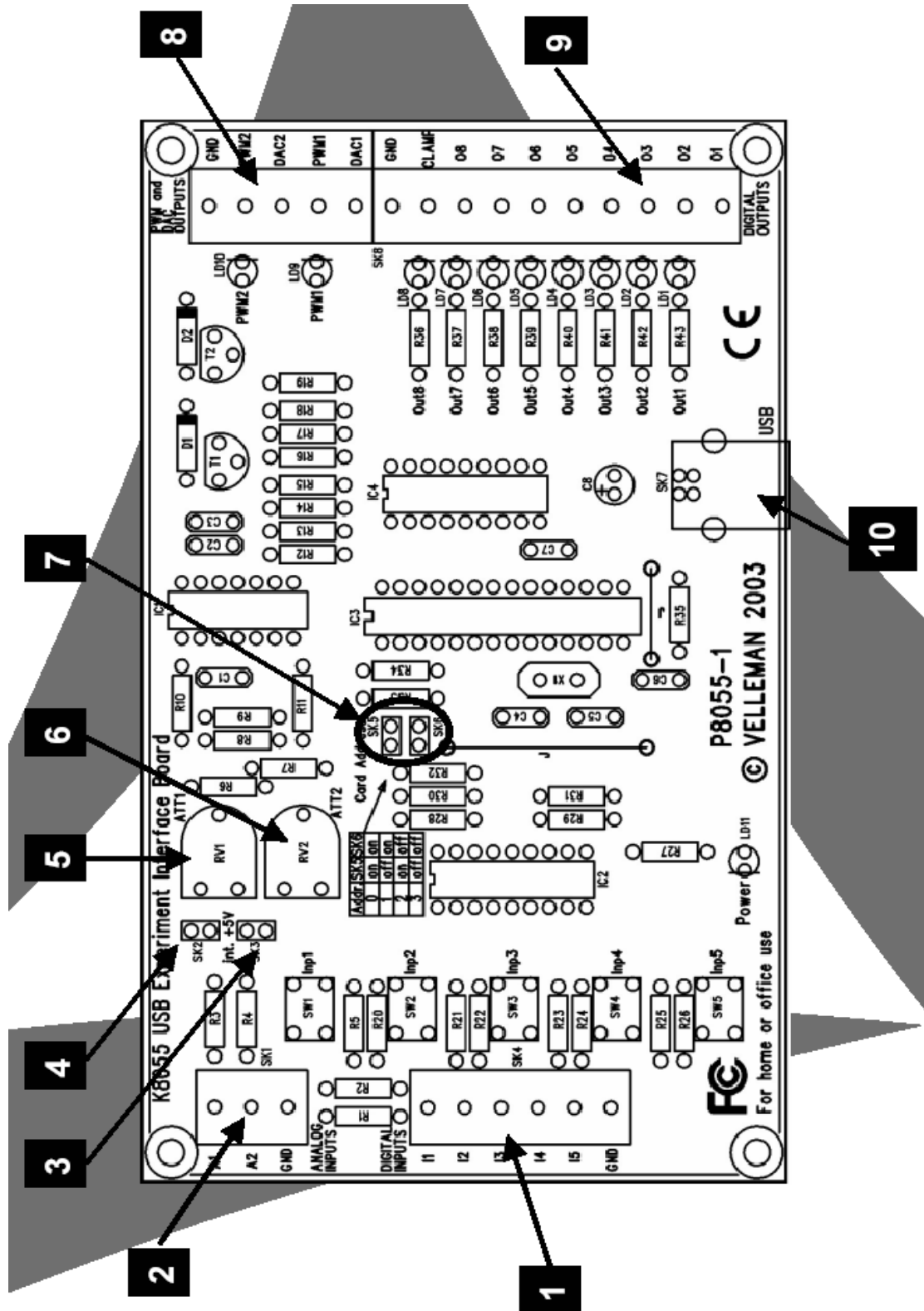
Zesílení zisku vstupu 2: $A_2 = 1 + (R_{11} / R_9)$

18. Pryžové nožičky

Nasaďte na pájecí straně desky s tištěnými spoji pryžové nožičky (opěrky) podle následujícího vyobrazení:



3. Vstupy a výstupy na desce s tištěnými spoji



Popis vstupů a výstupů na desce s tištěnými spoji

- [1] 5 digitálních vstupů (například: tlačítko, spínač, kontakt relé ...). Vstupy jsou většinou nastaveny na logickou jedničku „high“ (1), spojení s GND“ (s kostrou, uzemněním) přepne výstupy logickou nulu „low“ (0).
- [2] Analogové vstupy (například: senzor měření teploty, potenciometr ...).
- [3] Pokud osadíte desku s tištěnými spoji zástrčkovým můstkem, můžete použít interní napětí, které lze dále přizpůsobit (nastavit) pomocí trimrů (potenciometrů) RV2 a RV1. Pokud neosadíte desku s tištěnými spoji zástrčkovým můstkem, pak musíte použít externí napětí.
- [4]
- [5] Nastavení interního napětí vstupu A1.
- [6] Nastavení interního napětí vstupu A2.
- [7] Nastavení adresy desky: 1 = rozpojený kontakt, 0 = spojený kontakt.
- [8] Analogové výstupy.
- [9] Digitální výstupy.
- [10] Propojení pomocí USB s počítačem.

4. Zvolení správné adresy v testovacím programu

Digitální výstupy

8 kontaktů s otevřeným kolektorem pro propojení s příslušnými (odpovídajícími) vstupy.

Analogové výstupy

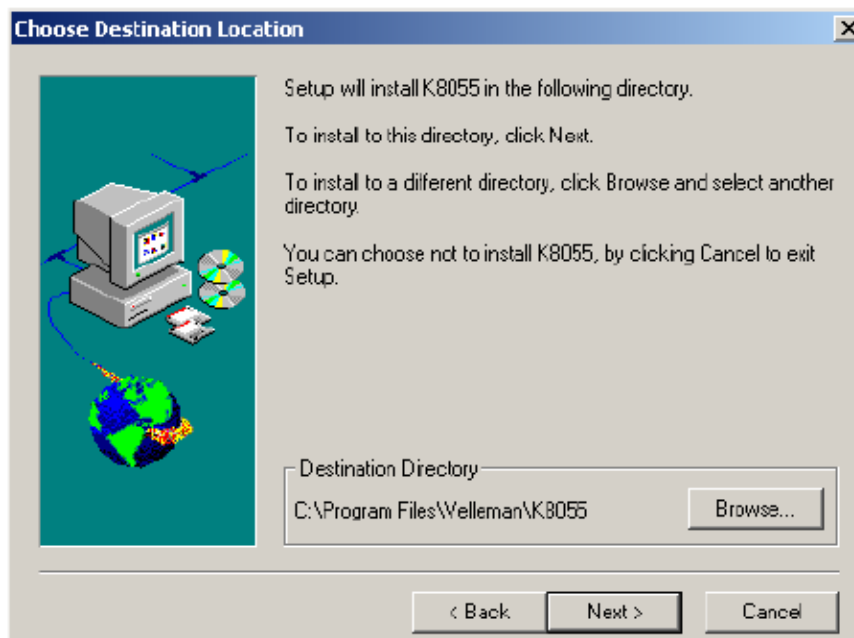
- 2 analogové výstupy „0“ a „+ 5 V“ (impedance 1,5 k Ω).
- 2 výstupy s modulací šířkou impulsů (PWM = pulse width modulation) 0 až 100 %.



Upozornění: Analogové výstupy a výstupy PWM jsou vždy aktivovány nebo deaktivovány společně.

5. Instalace software

- Klikněte na „setup.exe“. Pokud nemáte příslušný software (pokud Vám nebyl dodán), můžete si tento software stáhnout z internetové adresy: www.velleman.be.
- Příslušná nápověda Vás provede postupem při instalaci software.
- Standardní instalace software: „C:\Program Files\Velleman\K8055“.



6. Provedení otestování karty pomocí software

Příložený software (DEMO) Vám zajistí, abyste mohli snadno provádět různé experimenty.

Nastavte nejprve správnou adresu karty (její identifikační číslo):

Vyháčkujte (rozpojte) kontakty kolíkové lišty SK5 a/nebo SK6 (viz následující tabulka).

SK5	SK6	Adresa
Přemostěné kontakty	Přemostěné kontakty	0
Rozpojené kontakty	Přemostěné kontakty	1
Přemostěné kontakty	Rozpojené kontakty	2
Rozpojené kontakty	Rozpojené kontakty	3

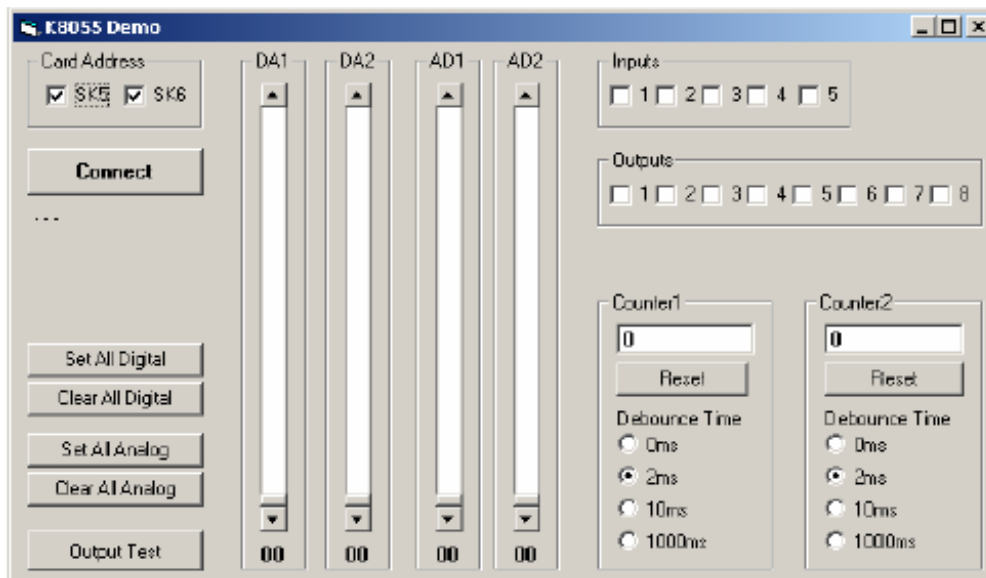
POZOR: Proved'te příslušná nastavení dříve, než spojíte kartu s počítačem (nebo dříve než počítač zapnete).

- Propojte počítač a desku pomocí USB-kabelu.
- Zapněte počítač. Po jeho zapnutí se musí na desce s tištěnými spoji rozsvítit svítivá dioda „LD3“ (Power) = zapnutí napájení.
- Po spuštění počítače krátce zabliká kontrolka „LD4“ (výstup 4), což znamená správnou funkci obvodu.

Spusťte program „**K8055_demo.exe**“.

Klikněte myší na „**Connect**“, čímž provedete spojení počítače s deskou „K8055“.

Hlášení „**Card x connected**“ znamená, že bylo úspěšně navázáno spojení.



Nyní můžete provést simulaci vstupů kliknutím na „**Inputs 1**“ až „**Inputs 5**“ (Inp1 až Inp5). Pokud podržíte některé z těchto tlačítek (ikon) stisknuté, zůstane příslušné políčko zaškrtnuté.

Zaškrtněte vždy zakřížkované políčko odpovídajícího výstupu, pokud chcete provést kontrolu příslušného digitálního výstupu (Outputs). Tento test můžete též nechat proběhnout automaticky: Klikněte myší na „**Output Test**“ nebo proveďte aktivaci všech výstupů kliknutím na „**Set All Digital**“. Klikněte myší na „**Output Test**“ za účelem otestování všech digitálních výstupů.

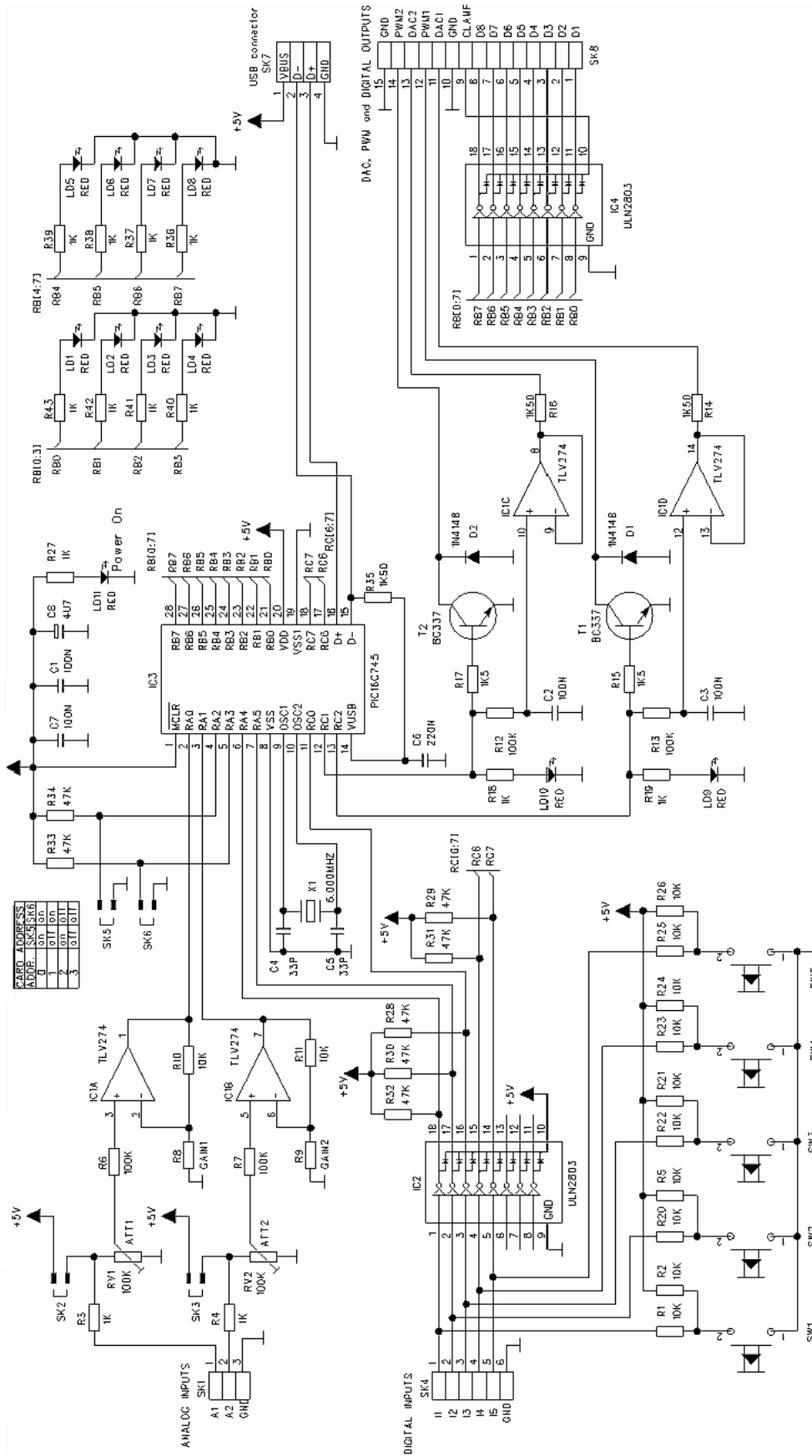
Otestování analogových výstupů provedete kliknutím na „**Set All Analog**“. Proveďte poté přizpůsobení (nastavení) výstupního napětí pomocí „**DA1**“ a „**DA2**“.

Pomocí „**Inp1**“ a „**Inp1**“ můžete provést otestování obou čítačů „**Counter1**“ a „**Counter2**“. Tyto čítače provedou připočítání času, pokud podržíte některé z obou „tlačítek“ stisknuté. Odskok „kontaktů“ (Debounce) Vám dovolí nastavit dobu spouštění (reakční čas) v rozmezí „0 ms – 2 ms – 10 ms – 1000 ms“.

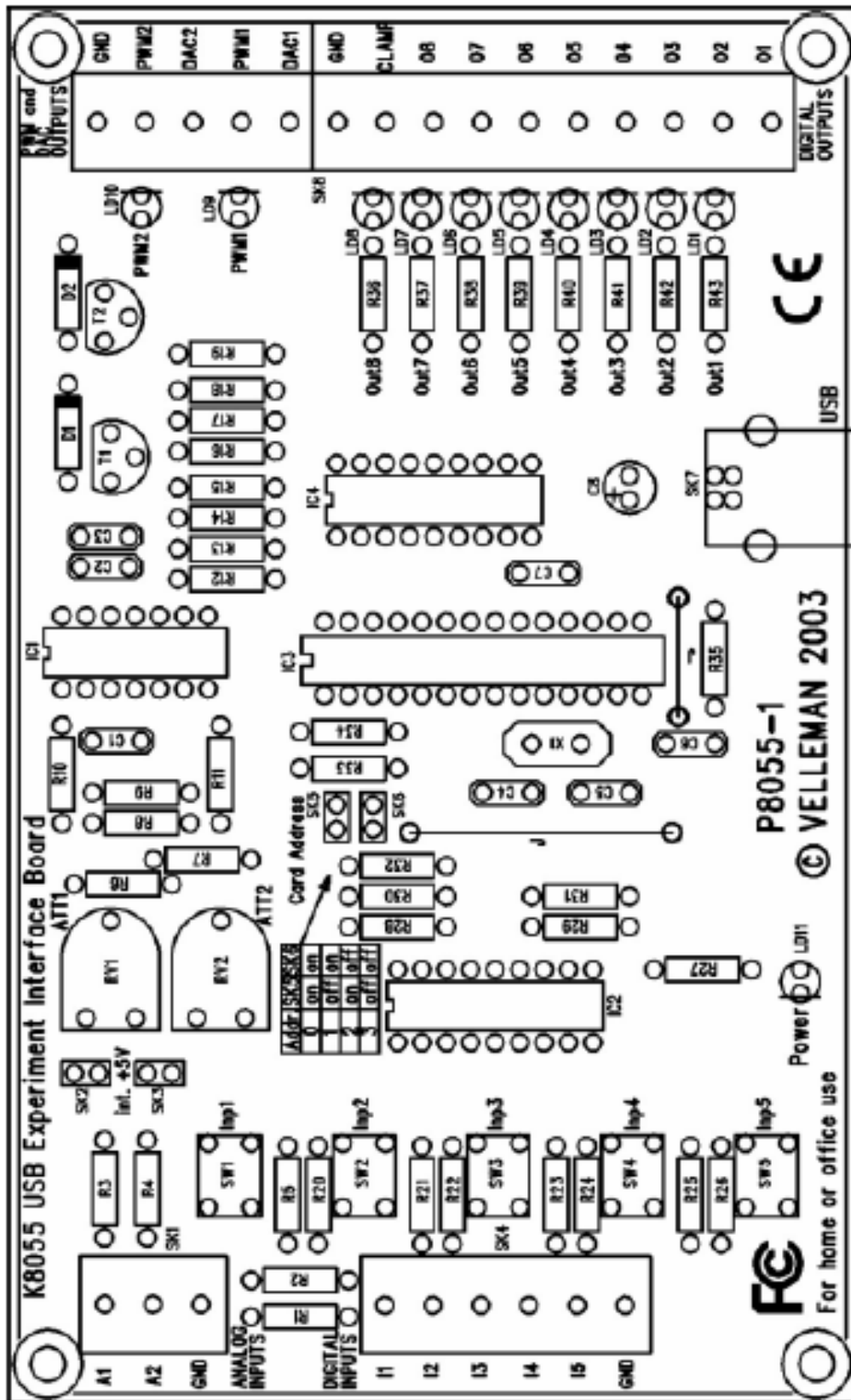
Interní analogové napětí můžete použít k simulaci analogového vstupu pomocí trimrů „**RV1**“ a „**RV2**“.

Rolovací sloupce (na monitoru počítače) „**AD1**“ a „**AD2**“ mění neustále svojí polohu, pokud budete měnit polohu trimrů „**RV1**“ a „**RV2**“. Digitální hodnotu (0 až 255) interního analogového napětí (úrovně) odečtete pod těmito rolovacími sloupci.

7. Schéma zapojení



8. Plán osazení desky s tištěnými spoji součástkami



9. Úvod do programování a popis souboru K8055.DLL

Karta „Interface Board K8055“ je vybavena 5 digitálními vstupy, 8 digitálními výstupy, 2 analogovými vstupy, 2 analogovými napětovými výstupy a 2 výstupy s modulací šířkou impulsů (PWM) a 8-bitovým rozlišením.

Počet vstupů a výstupů lze samozřejmě zvýšit, použijete-li až (max.) 4 karty, které připojíte k USB-konektorům počítače. Každé kartě přiřadíte vlastní identifikační číslo neboli adresu pomocí 2 zastrkovacích můstků SK5 a SK6.

SK5	SK6	Adresa karty
Přemostěné kontakty	Přemostěné kontakty	0
Rozpojené kontakty	Přemostěné kontakty	1
Přemostěné kontakty	Rozpojené kontakty	2
Rozpojené kontakty	Rozpojené kontakty	3

POZOR: Proved'te příslušná nastavení dříve, než spojíte kartu s počítačem (nebo dříve než počítač zapnete).

Všechny komunikační programy jsou uloženy do souboru „K8055D.DLL“ (Dynamic Link Library). Tento následující popis Vás seznámí se všemi funkcemi souboru „DLL“, které budete moci použít pro své aplikační programy.

Pro další komunikaci si můžete napsat vlastní aplikační programy podporující „DLL“, například v „Delphi“, „Visual Basic“, „C++ Builder“ atd. V těchto programovacích jazycích uvádíme i mnohé příklady aplikací. Většina příkladů a popisů funkcí či metod je uvedena v programovacím jazyce „Delphi“.

Úplný přehled všech funkcí a metod, které můžete exportovat pomocí souboru „K8055D.DLL“, naleznete dalších stránkách.

Přehled funkcí souboru „K8055D.DDL“

Otevření a uzavření karty:

OpenDevice (CardAddress): Otevření komunikace s kartou.

CloseDevice : Ukončení (uzavření) komunikace s kartou.

Konverze analogový / digitální vstup:

ReadAnalogChannel (Channelno): Načtení stavu jednoho analogového vstupu.

ReadAllAnalogChannel (Data1, Data2): Načtení stavu obou analogových vstupů.

Konverze digitální / analogový výstup:

OtputAnalogChannel (Channel, Data): Nastavení jednoho analogového výstupu podle dat.

OutputAllAnalogChannel (Data1, Data2): Nastavení obou analogových výstupů podle dat.

ClearAnalogChannel (Channel): Nastavení jednoho analogového výstupu na minimum (výmaz).

ClearAllAnalog : Nastavení všech analogových výstupů na minimum (výmaz).

SetAnalogChannel (Channel): Nastavení jednoho analogového výstupu na maximum.

SetAllAnalog : Nastavení všech analogových výstupů na maximum.

Digitální výstupy (zápis):

WriteAllDigital (Data): Nastavení všech digitálních výstupů podle dat.

ClearDigitalChannel (Channel): Vymazání jednoho digitálního výstupu.

ClearAllDigital : Vymazání všech digitálních výstupů.

SetDigitalChannel (Channel): Nastavení (zapnutí) 1 digitálního výstupu.

SetAllDigital : Nastavení (zapnutí) všech digitálních výstupů.

Digitální vstupy:

ReadDigitalChannel (Channel): Načtení stavu jednoho digitálního vstupu.

ReadAllDigital (Buffer): Načtení stavu všech digitálních vstupů.

Funkce čítače:

ResetCounter (CounterNr): Nastavení 16-bitového čítače 1 nebo 2 na nulu.

ReadCounter (CounterNr): Načtení obsahu čítače 1 nebo 2.

SetCounterDebounceTime (CounterNr, DebounceTime): Nastavení doby trvání odskoku příslušného čítače.

OpenDevice (otevření karty)

Syntaxe:

```
FUNCTION OpenDevice(CardAddress: Longint):Longint;
```

Parametr:

CardAddress: Hodnota mezi 0 až 3, identifikační adresa karty přidělená kartě pomocí 2 zastrkovacích můstků SK5 a SK6.

Výsledek: Longint: Bude-li akce otevření úspěšná, vrácená hodnota adresy karty K8055 se bude rovnat načtenému číslu.

Popis: Otevření komunikace (linku) s kartou K8055. Načtení ovladačů, které jsou nutné pro komunikaci s kartou pomocí USB. Jedná se první krok programu.

Příklad:

```
var h:longint;  
BEGIN  
    h:=OpenDevice (0); // otevření karty č. 0  
END;
```

CloseDevice (uzavření karty)

Syntaxe:

```
PROCEDURE CloseDevice;
```

Popis: Ukončení komunikace (linku) s kartou K8055. Uvolnění ovladačů, které jsou nutné pro komunikaci s kartou pomocí USB. Poslední krok programu.

Příklad:

```
BEGIN  
    CloseDevice (0); // uzavření karty č. 0  
END;
```

ReadAnalogChannel (konverze 1 analogového / digitální vstupu + načtení)

Syntaxe:

```
FUNCTION ReadAnalogChannel(Channel: Longint):Longint;
```

Parametr:

Channel: Hodnota 1 nebo 2, která souhlasí (musí souhlasit) se stavem (číslem) příslušného AD-kanálu, který má být načten.

Výsledek: Longint: Načtení dat konverze analogový / digitální vstup.

Popis: Konverze (přepočít) hodnoty vstupního (analogového) napětí zvoleného 8-bitového kanálu na digitální hodnotu 0 až 255.

Příklad:

```
var data:longint;  
BEGIN  
    data := ReadAnalogChannel (1); // Načtení AD-kanálu 1 do  
    variabilní proměnné data  
END;
```

ReadAllAnalog (konverze obou analogových / digitální vstupů + načtení)

Syntaxe:

```
PROCEDURE ReadAllAnalog(var Data1, Data2: Longint);
```

Parametry (popis):

Data1, Data2: Načtení dat (stavů) obou konverzí analogových / digitálních vstupů jako hodnoty „longint“ do těchto datových polí.

Příklad:

```
Procedure TForm1.ButtonClick (Sender: TObject);
var Data1, Data2:longint;
begin
  ReadAllAnalog(Data1, Data2); // Načtení dat z K8055
  Label1.caption:=inttostr(Data1); // Zobrazení dat CH1
  Label2.caption:=inttostr(Data2); // Zobrazení dat CH2
end;
```

OutputAnalogChannel (konverze 1 digitálního / analogového výstupu)

Syntaxe:

```
PROCEDURE OutputAnalogChannel(Channel:Longint;Data:Longint);
```

Parametry:

Channel: Hodnota 1 nebo 2, která souhlasí s 8-bitovým číslem kanálu DA, jehož data mají být nastavena.

Data:Digitální hodnota 0 až 255, která má být poslána 8-bitovému konvertoru z digitálního formátu na formát analogový.

Popis:

Uvedený 8-bitový kanál konverze z digitálního formátu na formát analogový bude přepsán (změněn) obsahem nových dat. To znamená, že tato data budou odpovídat specifickému napětí. Hodnota „0“ odpovídá minimálnímu výstupnímu napětí (0 V), hodnota „255“ odpovídá maximálnímu výstupnímu napětí (+ 5 V). Hodnota proměnné „Data“ mezi výše uvedenými hodnotami odpovídá výstupnímu napětí, které lze vypočítat podle následující rovnice: Napětí = (Data / 255) x 5 V.

Příklad:

```
BEGIN
  OutputAnalogChannel(1, 127); // Výstupní napětí kanálu 1
  bude nastaveno na 2,5 V
END;
```

OutputAllAnalog (konverze obou digitálních / analogových výstupů)

Syntaxe:

```
PROCEDURE OutputAllAnalog(Data1:Longint;Data2:Longint);
```

Parametry:

Data1, Data2: Digitální hodnota 0 až 255, která má být poslána 8-bitovému konvertoru z digitálního formátu na formát analogový.

Popis: Oba 8-bitové kanály konverze z digitálního formátu na formát analogový budou přepsány (změněny) obsahem nových dat. To znamená, že tato data budou odpovídat specifickým napětím. Hodnota „0“ odpovídá minimálnímu výstupnímu napětí (0 V), hodnota „255“ odpovídá maximálnímu výstupnímu napětí (+ 5 V). Hodnota proměnné „Data1 nebo Data2“ mezi výše uvedenými hodnotami odpovídá výstupnímu napětí, které lze vypočítat podle následující rovnice: $\text{Napětí} = (\text{Data} / 255) \times 5 \text{ V}$.

Příklad:

```
BEGIN
  OutputAllAnalog(127, 255); // Výstupní napětí kanálu 1
  bude nastaveno na 2,5 V, napětí kanálu 2 na 5 V
END;
```

ClearAnalogChannel (vymazání 1 analogového výstupu, nastavení na nulu)

Syntaxe:

```
PROCEDURE ClearAnalogChannel(Channel:Longint);
```

Parametr:

Channel: Hodnota 1 nebo 2, která souhlasí s 8-bitovým číslem kanálu DA, jehož data mají být vynulována (nastavena na nulu).

Popis: Zvolený kanál DA bude nastaven na minimální výstupní napětí (0 V).

Příklad:

```
BEGIN
  ClearAnalogChannel(1); // Výstupní napětí kanálu 1 bude
  nastaveno na 0 V
END;
```

ClearAllAnalog (vymazání obou analogových výstupů, nastavení na nulu)

Syntaxe:

```
PROCEDURE ClearAllAnalog;
```

Popis: Oba kanály DA budou nastaveny na minimální výstupní napětí (0 V).

Příklad:

```
BEGIN
  ClearAllAnalog; // Výstupní napětí kanálu 1 a 2 bude
  nastaveno na 0 V
END;
```

SetAnalogChannel (nastavení 1 analogového výstupu na maximum)

Syntaxe:

```
PROCEDURE SetAnalogChannel(Channel:Longint);
```

Parametr:

Channel: Hodnota 1 nebo 2, která souhlasí s 8-bitovým číslem kanálu DA, jehož data mají být nastavena na maximum.

Popis: Zvolený kanál DA bude nastaven na maximální výstupní napětí (5 V).

Příklad:

```
BEGIN
  SetAnalogChannel(1); // Výstupní napětí kanálu 1 bude
  nastaveno na + 5 V
END;
```

SetAllAnalog (nastavení obou analogových výstupů na maximum)

Syntaxe:

```
PROCEDURE SetAllAnalog;
```

Popis: Oba kanály DA budou nastaveny na maximální výstupní napětí (5 V).

Příklad:

```
BEGIN
  SetAllAnalog; // Výstupní napětí kanálu 1 a 2 bude
  nastaveno na + 5 V
END;
```

WriteAllDigital (nastavení všech digitálních výstupů)

Syntaxe:

```
PROCEDURE WriteAllDigital(Data:Longint);
```

Parametr:

Data: Digitální hodnota 0 až 255, která má být přiřazena digitálnímu výstupnímu portu (8 kanálům).

Popis: Kanály digitálního výstupního portu budou aktualizovány stavem odpovídajícího počtu bitů. Vyšší úroveň (1) znamená zapnutí kanálů (na výstupu integrovaného obvodu IC1), nižší úroveň (0) znamená uvolnění (vypnutí) výstupu.

Příklad:

```
BEGIN
  WriteAllDigital(7); // Zapnutí výstupních kanálů 1 až 3 a
  vypnutí (uvolnění) výstupních kanálů 4 až 8
END;
```

ClearDigitalChannel (vymazání – uvolnění 1 digitálního výstupu)

Syntaxe:

```
PROCEDURE ClearDigitalChannel(Channel:Longint);
```

Parametr: Channel:Číslo výstupního digitálního kanálu (1 až 8), který má být uvolněn (vymazán).

Popis: Zvolený kanál bude uvolněn (vymazán).

Příklad:

```
BEGIN  
  ClearIOchannel(4); // Vypnutí výstupního kanálu č. 4  
END;
```

ClearAllDigital (vymazání – uvolnění všech digitálních výstupů)

Syntaxe:

```
PROCEDURE ClearAllDigital;
```

Popis: Budou uvolněny (vymazány) všechny digitální výstupy (8 kanálů).

Příklad:

```
BEGIN  
  ClearAllDigital; // Vypnutí výstupního kanálu č. 1 až 8  
END;
```

SetDigitalChannel (zapnutí 1 digitálního výstupního kanálu)

Syntaxe:

```
PROCEDURE SetDigitalChannel(Channel:Longint);
```

Parametr:

Channel:Číslo výstupního digitálního kanálu (1 až 8), který má být zapnut (nastaven).

Popis: Zvolený výstupní kanál bude zapnut.

Příklad:

```
BEGIN  
  SetDigitalChannel(3); // Zapnutí digitálního výstupního  
  kanálu č. 3  
END;
```

SetAllDigital (zapnutí všech digitálních výstupních kanálů)

Syntaxe:

```
PROCEDURE SetAllDigital;
```

Popis: Dojde k zapnutí všech digitálních výstupních kanálů.

Příklad:

```
BEGIN  
  SetAllDigital; // Zapnutí všech výstupních kanálů  
END;
```

ReadDigitalChannel (načtení 1 digitálního vstupního kanálu)

Syntaxe:

```
FUNCTION ReadDigitalChannel(Channel: Longint): Boolean;
```

Parametr:

Channel: Hodnota 1 až 5, která souhlasí (musí souhlasit) se stavem (číslem) příslušného vstupního kanálu, který má být načten.

Výsledek: Boolean: „TRUE“ (pravdivá hodnota) znamená, že byl kanál nastaven (zapnut), „FALSE“ (nepravdivá hodnota) znamená, že byl kanál uvolněn (vypnut).

Popis: Dojde k načtení stavu zvoleného kanálu.

Příklad:

```
var status: boolean;  
BEGIN  
    status := ReadIOChannel (2); //Načtení vstupního kanálu 2  
END;
```

ReadAllDigital (načtení všech digitálních vstupních kanálů)

Syntaxe:

```
FUNCTION ReadAllDigital: Longint;
```

Výsledek: Longint: 5 nejnižších platných bitů je v souladu se stavem vstupních kanálů. Logická jednička (1) znamená, že je kanál nastaven na „HIGH“ (vysokou úroveň), nula (0) znamená, že je kanál nastaven na „LOW“ (nízkou úroveň).

Popis: Tato funkce reprodukuje stav digitálních vstupů.

Příklad:

```
var status: longint;  
BEGIN  
    status := ReadAllDigital; //Načtení všech vstup. kanálů  
END;
```

ResetConter (vynulování čítačů)

Syntaxe:

```
PROCEDURE ResetCounter(CounterNumber: Longint);
```

Parametr: CounterNumber: Číslo čítače (1 nebo 2), který má být vynulován.

Popis: Zvolený čítač bude vynulován (nastaven na nulu).

Příklad:

```
BEGIN  
    ResetCounter(2); // Reset čítače č. 2  
END;
```


ReadCounter (načtení čítačů)

Syntaxe:

```
PROCEDURE ReadCounter (CounterNumber:Longint) : Longint;
```

Parametr:

CounterNumber : Číslo čítače (1 nebo 2), který má být načten.

Výsledek: Longint : Obsah zvoleného 16-bitového čítače impulsů.

Popis: Tato procedura načte stav zvoleného 16-bitového čítače impulsů. Čítač 1 načítá impulsy vstupu č. 1, čítač 2 vstupu č. 2.

Příklad:

```
var pulses: longint;  
BEGIN  
    pulses := ReadCounter (2); // Načtení obsahu čítače č. 2  
END;
```

SetCounterDebounceTime (nastavení doby trvání odskoku)

Syntaxe:

```
PROCEDURE SetCounterDebounceTime (CounterNr, DebounceTime:Longint);
```

Parametry:

CounterNr : Číslo čítače (1 nebo 2), kterého se toto nastavení týká.

DebounceTime : Doba trvání odskoku zvoleného čítače impulsů. Tato doba se zadává v milisekundách (ms) a může se pohybovat v rozmezí od 0 do 5000.

Popis: U vstupů čítačů jsou prováděny pomocí software odskoky, aby nemohlo docházet k nesprávnému spouštění, pokud používáte mechanické spínače nebo reléové vstupy. Tato doba trvání odskoku je stejná pro dobu náběhu i doběhu čela impulsu. Standardní doba odskoku činí 2 ms. To znamená, že musí být vstup čítače stabilní alespoň po dobu 2 ms, dříve než dojde k jeho načtení a pokud bude zohledněna maximální rychlost (četnost impulsů) asi 200 načítání za sekundu. Nastavíte-li tento čas na „0“, pak to znamená maximální rychlost (četnost impulsů) asi 2000 načítání za sekundu

Příklad:

```
BEGIN  
    SetCounterDebounceTime (1, 100);  
    // Nastavení doby trvání odskoku čítače 1 na 100 ms  
END;
```

Použití souboru „K8055D.DDL“ v programovacím jazyce „Delphi“

V tomto příkladu aplikace jsou vysvětleny funkce a způsob činnosti souboru „K8055D.DDL“ včetně „OpenDevice“ a „CloseDevice“.

```
unit K8055;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls, ComCtrls;

type
  TForm1 = class(TForm)
    GroupBox1: TGroupBox;
    SK6: TCheckBox;
    SK5: TCheckBox;
    Button1: TButton;
    Label1: TLabel;
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure Button1Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  timed:boolean;

implementation

{$R *.DFM}
function OpenDevice(CardAddress: Longint): Longint; stdcall; external 'K8055d.dll';
procedure CloseDevice; stdcall; external 'K8055d.dll';
function ReadAnalogChannel(Channel: Longint): Longint; stdcall; external 'K8055d.dll';
procedure ReadAllAnalog(var Data1, Data2: Longint); stdcall; external 'K8055d.dll';
procedure OutputAnalogChannel(Channel: Longint; Data: Longint); stdcall; external
'K8055d.dll';
procedure OutputAllAnalog(Data1: Longint; Data2: Longint); stdcall; external 'K8055d.dll';
procedure ClearAnalogChannel(Channel: Longint); stdcall; external 'K8055d.dll';
procedure ClearAllAnalog; stdcall; external 'K8055d.dll';
procedure SetAnalogChannel(Channel: Longint); stdcall; external 'K8055d.dll';
procedure SetAllAnalog; stdcall; external 'K8055d.dll';
procedure WriteAllDigital(Data: Longint); stdcall; external 'K8055d.dll';
procedure ClearDigitalChannel(Channel: Longint); stdcall; external 'K8055d.dll';
procedure ClearAllDigital; stdcall; external 'K8055d.dll';
procedure SetDigitalChannel(Channel: Longint); stdcall; external 'K8055d.dll';
procedure SetAllDigital; stdcall; external 'K8055d.dll';
function ReadDigitalChannel(Channel: Longint): Boolean; stdcall; external 'K8055d.dll';
function ReadAllDigital: Longint; stdcall; external 'K8055d.dll';
function ReadCounter(CounterNr: Longint): Longint; stdcall; external 'K8055d.dll';
procedure ResetCounter(CounterNr: Longint); stdcall; external 'K8055d.dll';
procedure SetCounterDebounceTime(CounterNr, DebounceTime: Longint); stdcall; external
'K8055d.dll';

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  CloseDevice;
end;

procedure TForm1.Button1Click(Sender: TObject);
var h, CardAddr: longint;
begin
  CardAddr:= 3-(integer(SK5.Checked) + integer(SK6.Checked) * 2);
  h:= OpenDevice(CardAddr);
  case h of
    0..3: label12.caption:='Card '+ inttostr(h)+' connected';

-1: label12.caption:='Card '+ inttostr(CardAddr)+' not found';
  end;
end;

end.
```

Použití souboru „K8055D.DDL“ v programovacím jazyce „Visual Basic“

```
Option Explicit
Private Declare Function OpenDevice Lib "k8055d.dll" (ByVal CardAddress As Long) As Long
Private Declare Sub CloseDevice Lib "k8055d.dll" ()
Private Declare Function ReadAnalogChannel Lib "k8055d.dll" (ByVal Channel As Long) As Long
Private Declare Sub ReadAllAnalog Lib "k8055d.dll" (Data1 As Long, Data2 As Long)
Private Declare Sub OutputAnalogChannel Lib "k8055d.dll" (ByVal Channel As Long, ByVal Data As Long)
Private Declare Sub OutputAllAnalog Lib "k8055d.dll" (ByVal Data1 As Long, ByVal Data2 As Long)
Private Declare Sub ClearAnalogChannel Lib "k8055d.dll" (ByVal Channel As Long)
Private Declare Sub SetAllAnalog Lib "k8055d.dll" ()
Private Declare Sub ClearAllAnalog Lib "k8055d.dll" ()
Private Declare Sub SetAnalogChannel Lib "k8055d.dll" (ByVal Channel As Long)
Private Declare Sub WriteAllDigital Lib "k8055d.dll" (ByVal Data As Long)
Private Declare Sub ClearDigitalChannel Lib "k8055d.dll" (ByVal Channel As Long)
Private Declare Sub ClearAllDigital Lib "k8055d.dll" ()
Private Declare Sub SetDigitalChannel Lib "k8055d.dll" (ByVal Channel As Long)
Private Declare Sub SetAllDigital Lib "k8055d.dll" ()
Private Declare Function ReadDigitalChannel Lib "k8055d.dll" (ByVal Channel As Long) As Boolean
Private Declare Function ReadAllDigital Lib "k8055d.dll" () As Long
Private Declare Function ReadCounter Lib "k8055d.dll" (ByVal CounterNr As Long) As Long
Private Declare Sub ResetCounter Lib "k8055d.dll" (ByVal CounterNr As Long)
Private Declare Sub SetCounterDebounceTime Lib "k8055d.dll" (ByVal CounterNr As Long, ByVal DebounceTime As Long)

Private Sub Connect_Click()
    Dim CardAddress As Long
    Dim h As Long
    CardAddress = 0
    CardAddress = 3 - (Check1(0).Value + Check1(1).Value * 2)
    h = OpenDevice(CardAddress)
    Select Case h
        Case 0, 1, 2, 3
            Label1.Caption = "Card " + Str(h) + " connected"
        Case -1
            Label1.Caption = "Card " + Str(CardAddress) + " not found"
    End Select
End Sub

Private Sub Form_Terminate()
    CloseDevice
End Sub
```



Nezapomeňte zkopírovat soubor „K8055D.DDL“ do adresáře Windows „System32“.

Použití souboru „K8055D.DDL“ v programovacím jazyce „C++ Builder“

```
//Listing K8055D.h
#ifdef __cplusplus
extern "C" {
#endif
#define FUNCTION __declspec(dllimport)

FUNCTION long __stdcall OpenDevice(long CardAddress);
FUNCTION __stdcall CloseDevice();
FUNCTION long __stdcall ReadAnalogChannel(long Channel);
FUNCTION __stdcall ReadAllAnalog(long *Data1, long *Data2);
FUNCTION __stdcall OutputAnalogChannel(long Channel, long Data);
FUNCTION __stdcall OutputAllAnalog(long Data1, long Data2);
FUNCTION __stdcall ClearAnalogChannel(long Channel);
FUNCTION __stdcall ClearAllAnalog();
FUNCTION __stdcall SetAnalogChannel(long Channel);
FUNCTION __stdcall SetAllAnalog();
FUNCTION __stdcall WriteAllDigital(long Data);
FUNCTION __stdcall ClearDigitalChannel(long Channel);
FUNCTION __stdcall ClearAllDigital();
FUNCTION __stdcall SetDigitalChannel(long Channel);
FUNCTION __stdcall SetAllDigital();
FUNCTION bool __stdcall ReadDigitalChannel(long Channel);
FUNCTION long __stdcall ReadAllDigital();
FUNCTION long __stdcall ReadCounter(long CounterNr);
FUNCTION __stdcall ResetCounter(long CounterNr);
FUNCTION __stdcall SetCounterDebounceTime(long CounterNr, long DebounceTime);

#ifdef __cplusplus
}
#endif

//Listing Unit1.cpp
//-----
#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
#include "K8055D.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----

__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
//-----

void __fastcall TForm1::Connect1Click(TObject *Sender)
{
    int CardAddr = 3 - (int(CheckBox1->Checked) + int(CheckBox2->Checked) * 2);
    int h = OpenDevice(CardAddr);
    switch (h) {
        case 0 :
        case 1 :
        case 2 :
        case 3 :
            Labell->Caption = "Card " + IntToStr(h) + " connected";
            break;
        case -1 :
            Labell->Caption = "Card " + IntToStr(CardAddr) + " not found";
    }
}

void __fastcall TForm1::FormClose(TObject *Sender, TCloseAction &Action)
{
    CloseDevice;
}

```