

Vážení zákazníci,

dovolujeme si Vás upozornit, že na tuto ukázkou knihy se vztahují autorská práva, tzv. copyright.

To znamená, že ukáзка má sloužit výhradně pro osobní potřebu potenciálního kupujícího (aby čtenář viděl, jakým způsobem je titul zpracován a mohl se také podle tohoto, jako jednoho z parametrů, rozhodnout, zda titul koupí či ne).

Z toho vyplývá, že není dovoleno tuto ukázkou jakýmkoliv způsobem dále šířit, veřejně či neveřejně např. umístováním na datová média, na jiné internetové stránky (ani prostřednictvím odkazů) apod.

redakce nakladatelství BEN – technická literatura
redakce@ben.cz



Tato kapitola ukazuje použití obvodů popsaných v kapitole 6 pro sestavení měřicí desky k počítači. Měřicí deska je připojena přes sériový port k počítači dle standardu PC a je ovládána programem běžícím na operačním systému Windows 95 a vyšším.

Deska obsahuje:

- čtyři 8bitové *analogové vstupy* A/D₀ až A/D₃ se vstupním odporem 400 kΩ pracující buď v bipolárním (−5 V až +5 V) nebo unipolárním (0 až 10 V) rozsahu,
- sedm 6bitových *analogových výstupů* D/A₀ až D/A₆ s maximálním výstupním proudem ±20 mA (vnitřní odpor zhruba 50 Ω) pracujících buď v bipolárním (−5 V až +5 V) nebo unipolárním (0 až 10 V) rozsahu,
- 4 *digitální vstupy* DIN₀ až DIN₃ a 4 *digitální výstupy* DOUT₀ až DOUT₃ slučitelné s TTL LS.

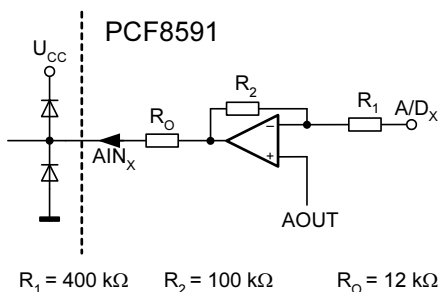
7.1 SCHÉMA ZAPOJENÍ

Úplné schéma zapojení je uvedeno na obr. 7.2. V zapojení jsou použity obvody **TDA8444** (IO₄), **PCF8591** (IO₁) a **PCF8574** (IO₈). Pro analogové vstupy a výstupy je třeba zajistit referenční napětí, proto jsou použity dva obvody **TL431** (IO₃ a IO₅). Protože pracovní rozsah analogových vstupů a výstupů není optimální, byly použity operační zesilovače z pouzder **TL074** (IO₂, IO₆ a IO₇). Pro ochranu a posílení digitálních vstupů a výstupů je použit obvod **74LS244** (IO₉). Konečně napájení číslicové části zajišťuje stabilizátor **7805** (IO₁₀).

Zapojení je kromě operačních zesilovačů a budiče sběrnice velmi jednoduché. Signály SDA a SCL jsou ovládány linkami **RTS** a **DTR** sériového portu. Přizpůsobení pro rozsah napětí 0 až 5 V zajišťují pracovní rezistory R₁, R₂ a omezovací diody D₁ až D₄. Tranzistorový obvod T₁, T₂ je použit pro čtení SDA po sériové lince **CTS** (v režimu vysílání obvodů slouží pro příjem dat z analogových nebo digitálních vstupů).

Deska je napájena ze symetrického zdroje ±12 V (potřebují jej operační zesilovače), proti přepólování napájení je chráněna diodami D₅ a D₆.

Pro ty, kteří jsou začátečníky v analogové technice, jsou připojeny obr. 7.1 a obr. 7.3, které ukazují výpočty operační sítě pro analogové vstupy a výstupy.



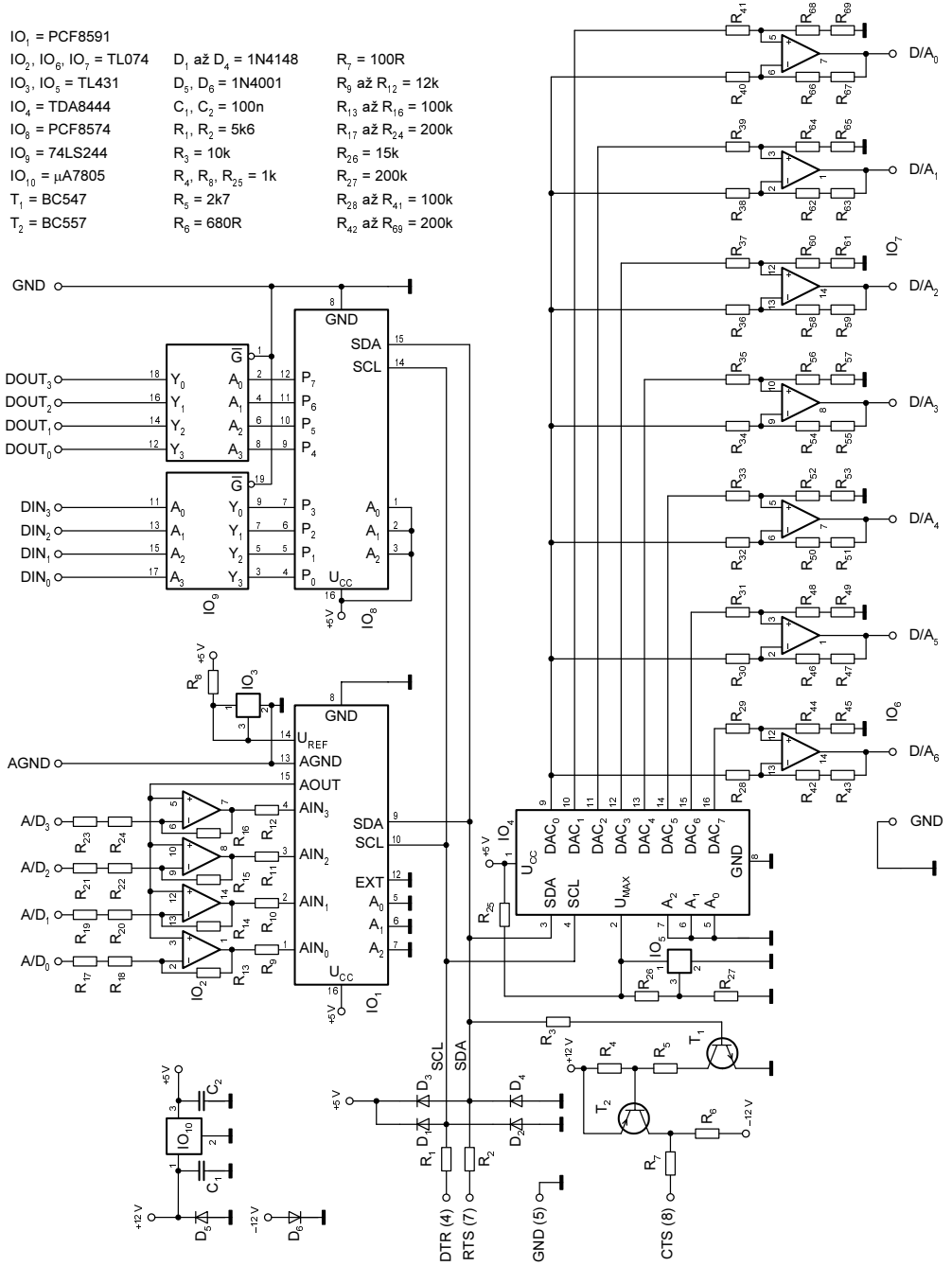
$$U_{AINx} = \frac{R_1 + R_2}{R_1} U_{AOUT} - \frac{R_2}{R_1} U_{A/Dx}$$

Obr. 7.1 Operační síť analogového vstupu

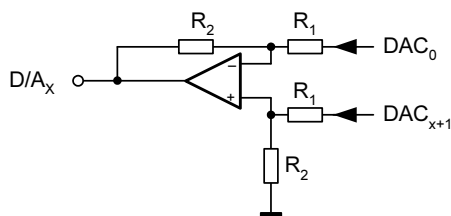
$IO_1 = PCF8591$
 $IO_2, IO_6, IO_7 = TL074$
 $IO_3, IO_5 = TL431$
 $IO_4 = TDA8444$
 $IO_8 = PCF8574$
 $IO_9 = 74LS244$
 $IO_{10} = \mu A7805$
 $T_1 = BC547$
 $T_2 = BC557$

D_1 až $D_4 = 1N4148$
 $D_5, D_6 = 1N4001$
 $C_1, C_2 = 100n$
 $R_1, R_2 = 5k6$
 $R_3 = 10k$
 $R_4, R_8, R_{25} = 1k$
 $R_5 = 2k7$
 $R_6 = 680R$

$R_7 = 100R$
 R_9 až $R_{12} = 12k$
 R_{13} až $R_{16} = 100k$
 R_{17} až $R_{24} = 200k$
 $R_{26} = 15k$
 $R_{27} = 200k$
 R_{28} až $R_{41} = 100k$
 R_{42} až $R_{89} = 200k$



Obr. 7.2 Úplné schéma zapojení měřicí desky (napájení všech IO je blokováno kondenzátory 100n)



$$U_{D/A_X} = \frac{R_2}{R_1} [U_{DAC_{x+1}} - U_{DAC_0}]$$

$$R_1 = 100 \text{ k}\Omega$$

$$R_2 = 400 \text{ k}\Omega$$

Obr. 7.3 Operační síť analogového výstupu

Pro *analogový vstup* (po dosažení hodnot odporů) platí:

$$U_{AIN_X} = 1,25U_{AOUT} - 0,25U_{A/D_X}$$

Tedy: vstupní napětí je děleno 4 a poté sečteno s výstupem D/A převodníku násobeným 1,25. Nastavením napětí U_{AOUT} na 1 V nebo 2 V se dosahuje práce v bipolárním nebo unipolárním rozsahu.

Rezistor R_O je použit pro ochranu vstupů obvodu PCF8591. Jak je z obr. 7.2 zřejmé, je každý vstup obvodu opatřen ochrannými diodami (jsou zaintegrované v obvodu), které omezí vstupní napětí na rozsah 0 V až U_{CC} . Je však třeba připojit pracovní rezistor R_O , který sníží proud tak, aby nedošlo k jejich zničení.

Pro *analogový výstup* (po dosažení hodnot odporů) platí:

$$U_{D/A_X} = 4 \cdot [U_{DAC_{x+1}} - U_{DAC_0}]$$

Tedy: výstupní napětí je dáno rozdílem napětí aktivního výstupu a výstupu DAC_0 . Nastavením napětí U_{DAC_0} na minimum nebo na polovinu maxima se dosahuje práce výstupu v unipolárním nebo bipolárním rozsahu.

7.2 PROGRAMÁTORSKÉ ROZHRAŇÍ V C++ BUILDERU

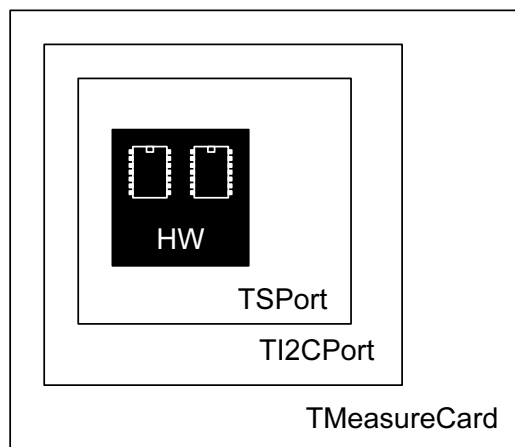
Pro značný rozsah nebude vypsán zdrojový text rutin pro ovládání měřicí desky (je umístěn na CD. Kromě toho jej lze stáhnout z: www.vosji.cz/~matousek). Provedu pouze popis jednotlivých metod a vlastností implementovaných tříd.

Třída TSPort (soubory SPORT.H a SPORT.CPP)

Třída TSPort slouží pro přímé řízení linek sériového portu, popis byl proveden již v kapitole 1.4.1.

Třída TI2CPort (soubory ISCPort.H a I2CPort.CPP)

Třída TI2CPort je specializována na ovládání obvodů se sběrnici I²C, které jsou připojeny na sériový port počítače. V jejím pojetí je hodinový signál sběrnice (SCL) ovládán linkou DTR a datový signál (SDA) řídí linky RTS (výstup) a CTS (vstup).



Obr. 7.4 Znáornění vzájemné závislosti jednotlivých vrstev rozhraní měřicí desky

Metody

- __fastcall TI2CPort(int SerialPortNumber);** konstruktor, SerialPortNumber udává číslo portu, se kterým chceme pracovat (např.: SerialPortNumber = 2 pro COM2),
- __fastcall ~TI2CPort();** destruktork, zavře sériový port,
- bool __fastcall TestPort();** otestuje funkčnost portu pro účely I²C sběrnice tak, že nastaví SCL = 0 a mění SDA na 1 nebo 0 a čte stav SDA zpět. Pokud je port funkční, musí přečtená hodnota SDA odpovídat dříve nastavené, v tom případě vrací funkce true,
- void __fastcall StartBit();** vyšle START (SCL = 1 a na SDA je připojena sestupná hrana),
- void __fastcall StopBit();** vyšle STOP (SCL = 1 a na SDA je připojena náběžná hrana),
- void __fastcall SendByte(byte Byte);** vyšle 8 datových bitů a přečte 9. bit odpovídající ACK do vlastnosti **Acknowledge** (START a STOP je třeba vytvořit voláním metod StartBit a StopBit),
- byte __fastcall ReceiveByte(bool Acknowledge);** přečte 8 datových bitů a v místě odpovídajícím 9. bitu generuje ACK nebo N podle parametru **Acknowledge**.

Vlastnosti

- DWORD Delay** doba určující trvání půlperiody SCL hodin v milisekundách (výchozí nastavení Delay = 0 většinou postačuje, protože zpracování příkazů počítačem je dostatečně pomalé),

- ❑ **bool Acknowledge** (R/O) obsahuje informaci o tom, zda byl při posledním volání **SendByte** podřízeným obvodem generován ACK nebo ne.

Třída **TMeasureCard** (soubory **TMCARD.H** a **TMCARD.CPP**)

TMeasureCard se již specializuje na ovládaní obvodů, ze kterých je sestavena měřicí deska. Při vytvoření instance **TMeasureCard** jsou zároveň vytvořeny instance dalších tříd (**TDigPort**, **TADCPort** a **TDACPort**), které přímo ovládají konkrétní obvody.

Metody

- ❑ **__fastcall TMeasureCard(int SerialPortNumber)**; konstruktor, **SerialPortNumber** udává číslo portu, se kterým chceme pracovat (např.: **SerialPortNumber** = 2 pro COM2). Dále jsou vytvořeny instance tříd, které se specializují na ovládaní konkrétních obvodů desky: **TDigPort**, **TADCPort** a **TDACPort**,
- ❑ **__fastcall ~TMeasureCard()**; destruktory, ruší i instance pomocných tříd,
- ❑ **bool __fastcall TestCard()**; otestuje připojení desky k sériovému portu stejným způsobem, jako metoda **TestPort** třídy **TI2CPort**.

Vlastnosti

- ❑ **TDigPort *DigPort** (R/O) ukazatel na instanci třídy **TDigPort** pro řízení digitálních vstupů a výstupů (viz níže),
- ❑ **TADCPort *ADCPort** (R/O) ukazatel na instanci třídy **TADCPort** pro řízení analogových vstupů (viz níže),
- ❑ **TDACPort *DACPort** (R/O) ukazatel na instanci třídy **TDACPort** pro řízení analogových výstupů (viz níže).

Třída **TDigPort**

Třída **TDigPort** ovládá digitální vstupy a výstupy poskytované obvodem **PCF8574**. Instanci této třídy není možno vytvořit samostatně, ale musí být použit ukazatel poskytovaný vlastností **DigPort** třídy **TMeasureCard**!

Metody

- ❑ **bool __fastcall SetPort(const byte &Data)**; nastaví stav výstupů **DOUT₀** až **DOUT₃**, význam mají jen spodní 4 bity **Data**. Při úspěchu vrací true (pokud libovolná z metod vrátí false, znamená to, že komunikace s obvodem **PCF8574** neproběhla správně),
- ❑ **bool __fastcall GetPort(byte &Data)**; přečte stav vstupů **DIN₀** až **DIN₃**, význam mají jen spodní 4 bity **Data**. Při úspěchu vrací true,
- ❑ **bool __fastcall Init()**; inicializuje obvod **PCF8574** tak, aby byly vývody **P₀** až **P₃** brány jako vstupy a **P₄** až **P₇** jako výstupy. Výstupy jsou nastaveny do log. 0. Při úspěchu vrací true.