

**N124****8 Kanal, bipol. Stromquelle**

1. Funktion .....	2
1.1. Datenblatt .....	2
1.1.1. Anwendung .....	2
1.1.2. Daten .....	2
1.1.3. Besonderheiten .....	2
1.1.4. Aufbau .....	2
1.1.5. Stromversorgung .....	2
1.2. Blockdiagramm .....	3
1.3. Beschreibung .....	3
2. Betrieb .....	4
2.1. Programmierung .....	4
2.1.1. Bedienung .....	4
2.1.2. Speicherbelegung .....	4
2.1.3. Setzen der Stromwerte .....	4
2.1.4. Rücklesen der Stromwerte .....	4
2.1.5. Lesen der Shunt- und OP-ADCs .....	5
3. Fertigung .....	7
3.1. Mechanik .....	7
3.1.1. Frontplatte .....	7
3.1.2. Gehäuse .....	7
3.1.3. ....	7
3.2. Elektronik .....	7
3.2.1. Schaltbild .....	7
3.2.2. Bestückungsplan .....	7
3.2.3. Stücklisten .....	7
3.2.4. Platinenunterlagen .....	7
3.2.5. ....	7
4. Test .....	8
4.1. Aufbau .....	8
4.2. Ergebnisse .....	8
4.3. ....	8
5. Modifikation .....	9
5.1. Version .....	9
5.2. ....	9
6. Anhang .....	10
6.1. Bausteinunterlagen .....	10
6.2. ....	10

## 1. FUNKTION

### 1.1. Datenblatt

#### 1.1.1. Anwendung

Vielkanal Stromversorgungen, Magnetspulen, Feldkorrekturen

#### 1.1.2. Daten

Parameter	Wert	Dimension
Kanalzahl	8	
max. Strom	+ - 1	A
Auflösung	12	Bit
Stabilität	ca. 10	ppm/°C

#### 1.1.3. Besonderheiten

Hochgenaue Stromregelung, Rücklesung der Stromeinstellung (DAC), Erfassung der Shuntspannung (12 Bit ADC) und der Spulenspannung (12 Bit ADC) für jeden Kanal.

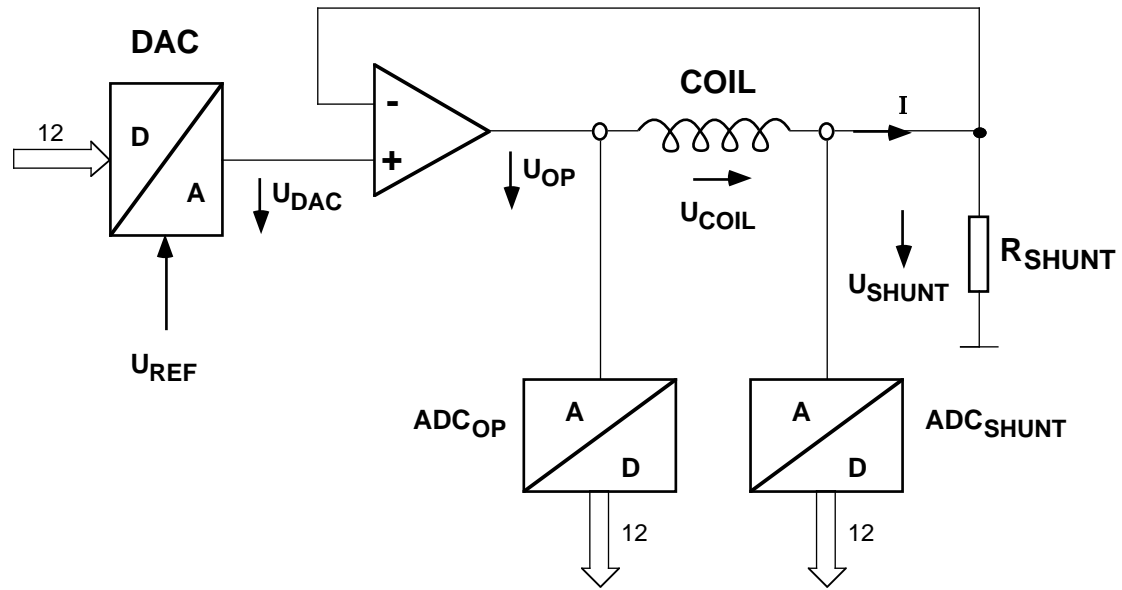
#### 1.1.4. Aufbau

6HE, 4TE VME

#### 1.1.5. Stromversorgung

Spannung	Strom	Leistung
+5V		
-5.2V		
+12V		
-12V		
Gesamt		

## 1.2. Blockdiagramm



## 1.3. Beschreibung

## 2. BETRIEB

### 2.1. Programmierung

#### 2.1.1. Bedienung

Jedes Modul im Überrahmen muß durch eine eigene Nummer (Adresse im SHORT Bereich von VME) festgelegt werden. Dazu dienen zwei HEX-Schalter an der Frontseite. Im folgenden ist die Einstellung dieser Adresse durch den Buchstaben s (zb. \$S<sub>oben</sub>S<sub>unten</sub>01=\$ss01) gekennzeichnet. In den Beispielen sind Speicherzugriffe nach der PASCAL Convention angegeben (Lesen: Wert := Adresse; Schreiben: Adresse:=Wert).

#### 2.1.2. Speicherbelegung

	D15	HighByte	D7	LowByte	D0
R/W. \$ss00	Strom(...D11)		Strom		Lesen DACKanal I/Übern. I-4

z.B hier.: Wert:=\$ss15;

### 2.1.5 Lesen der Shunt- und OP-ADCs

Bei diesen ADCs handelt es sich um serielle Bausteine, die auch von VME her seriell ausgelesen werden müssen. Jeweils ein ADC mit 8 Kanälen liefert die Spannungsinformationen für alle Shunts ( $U_{\text{shunt}}$ ) und ein ADC mit 8 Kanälen für alle OPs ( $U_{\text{OP}}$ ) (siehe Blockschaltbild). Bei funktionsfähigem Modul sollte die Spannung  $U_{\text{shunt}}$  natürlich gleich der Spannung  $U_{\text{DAC}}$  sein! Sie dient also im wesentlichen der Funktionskontrolle des Stroms!

Da beide ADCs gemeinsam 'enabled/disabled' und ge'clockt' werden ist es sinnvoll auch beide ADCs gemeinsam zu beschreiben bzw. auszulesen. Das folgende Beispielprogramm zeigt das Prinzip dafür (siehe auch Datenblatt MAX186):

```

var SSTRB @ $ss20;
    EnableADC @ $ss20;
    DisableADC @ $ss24;
    DataADC @ $ss28;

procedure ADC (channelShunt: Integer;    {Kanal: Zählweise 0..7 !}
               channelOP: Integer;      {Kanal: Zählweise 0..7 !}
               var dataShunt: LongInt;   {Daten: -2048...2047}
               var dataOP: LongInt);     {Daten: -2048...2047}

    procedure LoadADC (model, mode2: Integer); {Lade ModeRegister}
    var
        data: Longint;
        n: Integer;
    begin
        for n := 7 downto 0 do                {8 Bits seriell nacheinander}
            begin
                if BTST(model, n) then        {setze entspr. Bits in data}
                    BSET(data, 1)
                else
                    BCLR(data, 1);
                if BTST(mode2, n) then
                    BSET(data, 4)
                else
                    BCLR(data, 4);
                WriteADC := data;            {schreibe Bits in ADC}
            end;
        end;

    var
        data: Longint;
        n: Integer;
    begin
        EnableADC := 0;                       {Enable}

        channelShunt := BSL(channelShunt, 3); {setze Kanal#}
        if BTST(channelShunt, 3) then
            channelShunt := BOR(channelShunt, $40)
        else
            channelShunt := BAND(channelShunt, $0BF);
        channelShunt := BOR(BAND(channelShunt, $70), $86); {bip/sgl/intClk}

        channelOP := BSL(channelOP, 3);       {setze Kanal#}
        if BTST(channelOP, 3) then
            channelOP := BOR(channelOP, $40)
        else
            channelOP := BAND(channelOP, $0BF);
    
```

```
channelOP := BOR(BAND(channelOP, $70), $86);           {bip/sgl/intClk}
LoadADC(channelShunt, channelOP);                     {Laden}
WriteADC := 0;                                       {Start ADC}
repeat                                               {warte auf Fertig}
  data := SSTRB;
  data := BAND(data, $24);
until data = $24;

dataShunt := 0;                                     {Aufbau DATEN}
dataOP := 0;
for n := 11 downto 0 do                             {12 Bits}
  begin
    data := ReadADC;
    if BTST(data, 0) then
      BSET(dataShunt, n);
    if BTST(data, 3) then
      BSET(dataOP, n);
  end;

  if dataShunt > 2047 then                            {Vorzeichen}
    dataShunt := dataShunt - 4096;
  if dataOP > 2047 then
    dataOP := dataOP - 4096;

  DisableADC := 0;                                  {Disable}

  EnableADC := 0;
  LoadADC($81, $81);
  DisableADC := 0;
end;
{setze ADC auf}
{fast power down}
```

### **3. FERTIGUNG**

#### **3.1. Mechanik**

**3.1.1. Frontplatte**

**3.1.2. Gehäuse**

**3.1.3. ...**

#### **3.2. Elektronik**

**3.2.1. Schaltbild**

**3.2.2. Bestückungsplan**

**3.2.3. Stücklisten**

**3.2.4. Platinenunterlagen**

**3.2.5. ...**

## **4. TEST**

### **4.1. Aufbau**

### **4.2. Ergebnisse**

### **4.3. ...**



## 5. MODIFIKATION

### 5.1. Version

### 5.2. ...

## **6. ANHANG**

### **6.1. Bausteinunterlagen**

### **6.2. ...**