

6 Realisierung der digitalen "Ti-TP Kette"

6.1 Realisierung des Filters ohne AD-Wandler und mit DA-Evaluationstionsbord am Ausgang

Da zur Zeit kein Codec bzw. Digital-Analog-Wandler (*DAC*) mit ausreichender Updaterate an der HAW zur Verfügung steht und eine fertige Lösung nach Kapitel 5.6 zu teuer ist, wurde dazu übergegangen, ein System selbst zu entwickeln. Geplant ist eine Realisierung nach dem in Abb. 6.1 gezeigten Prinzipschaltbild.

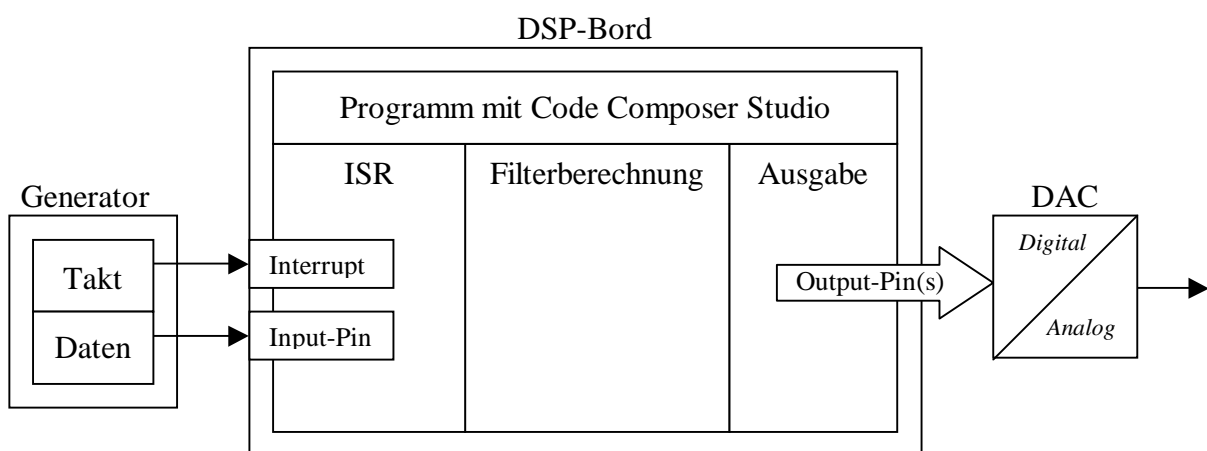


Abb. 6.1:
Prinzipschaltbild des geplanten Kanalmodells

Um das gezeigte System umzusetzen, wurde ein DAC für das vorhandene "TMS320C6713 DSK-Bord" neu entworfen. Der erste Schritt hierbei war, den für einen Codec benötigten Analog-Digital-Wandler (*ADC*) entfallen zu lassen, indem das binäre 5V-Datensignal vom Generator direkt auf einen Eingangspin gelegt wird und dieser nur auf den logischen Zustand 0 oder 1 geprüft wird. Für den speziellen Fall, dass es sich um ein digitales Eingangssignal handelt, ist kein ADC erforderlich. Der normierte Wertebereich für die digitalen Tiefpassfilter der "Ti-TP Kette" liegt zwischen 0 und 1. Allerdings wird das System den üblichen Weg der Digitalisierung für audioteknische Anwendungen mit dem DSK-Bord über das "multi channel buffered seriell Interface" (*McBSP*) nicht einhalten, was für die geplante Aufgabe aber nicht von Bedeutung ist. Daher wurde zur Ausgabe ein DAC mit einer Updaterate von mindestens 6,4Mega Samples per second (*MSPs*) gesucht. Die in Kapitel 5.6 vorgestellten Wandler-Aufsteckmodule verwenden Wandler-ICs der Firma Analog Devices. Da diese Aufsteckmodule mit den geforderten Updatraten angeboten werden, wurde die entsprechende Produktpalette für DAC-ICs näher betrachtet. Dabei ist

zu beachten, dass DACs mit "Voltage Output" nur eine maximale Updaterate von 1,25MSPs aufweisen, während DACs mit "Current Output" weit über 100MSPs schaffen. Auch andere Hersteller bieten keine schnelleren Wandler mit "Voltage Output" an, weshalb ein DAC mit "Current Output" verwendet wurde. Die Ansteuerung des DACs kann entweder über ein serielles oder ein paralleles Interface erfolgen.

6.1.1 Serielle Ansteuerung des DAC

Der "C6713" verfügt, wie bereits erwähnt, über das sogenannte McBSP. Dieses Interface wird am "C6713" verwendet, um die Wertigkeit des Eingangssignals vom AD-Wandler in serieller Form empfangen zu können. Bei dem auf dem Bord befindlichen Codec "AIC23" (Kap. 5.2) und der "PCM3003" Codec-Aussteckkarte aus dem Labor des Informatikbereichs wird der analoge Eingangswert auf diese Weise digitalisiert. Das gleiche gilt für den Ausgang, hierbei wird der auszugebende Wert vom McBSP des "C6713" in serieller Form an den DA-Wandler gesendet und dort umgesetzt.

Als serieller DAC könnte dann z.B. der "AD5444" zum Einsatz kommen, der zu den schnellsten erhältlichen seriellen DACs zählt (Abb. 6.2). Allerdings kann auch dieser nur mit einer max. Updaterate von 2,7MSPs arbeiten, was nicht den geforderten 6,4MSPs entspricht. Zudem wird diese Updaterate nur erreicht, wenn das serielle Interface mit 50MHz betrieben wird, was das McBSP und der DSP (unter Berücksichtigung der anstehenden Filterberechnung) nicht schaffen. Deshalb wurde nach einem DAC mit paralleler Ansteuerung und "Current Output" gesucht.

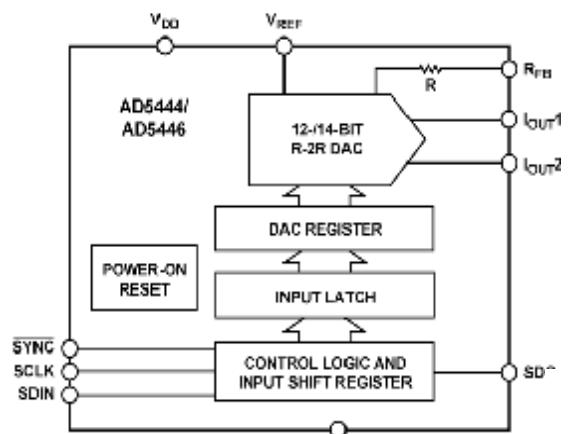


Abb. 6.2:
DAC mit serieller Ansteuerung [12]

6.1.2 Parallele Ansteuerung des DAC per GPIO

Von den in Anhang E.6 gezeigten Wandler-ICs, bietet sich am ehesten der DAC "AD9764" an, da zur Nachbildung des Kanalmodells nur ein Kanal benötigt wird und auch ein interner Taktmultiplizierer nicht zwingend erforderlich ist. Allerdings sind die 14bit Auflösung am Dateneingang ziemlich großzügig dimensioniert, weshalb der annähernd baugleiche "AD9752" (Abb. 6.3) mit 12bit Auflösung und der gleichen Updaterate (125MSPs) auf einem Evaluationsbord bestellt wurde. Die Ansteuerung des Parallelinterface übernehmen

dann "General-Purpose Input/Output Pins" (*GPIO-Pins; Ein-/Ausgangs-Pins für Allgemeine Zwecke*) vom DSP. Da am Programmierbord standardmäßig keine GPIO-Pins vorgesehen sind, würden sich dafür die an der Steckerleiste "J3" (*s.a. Anhang F.1*) bereitgestellten Interrupt-Pins 4 bis 7 eignen.

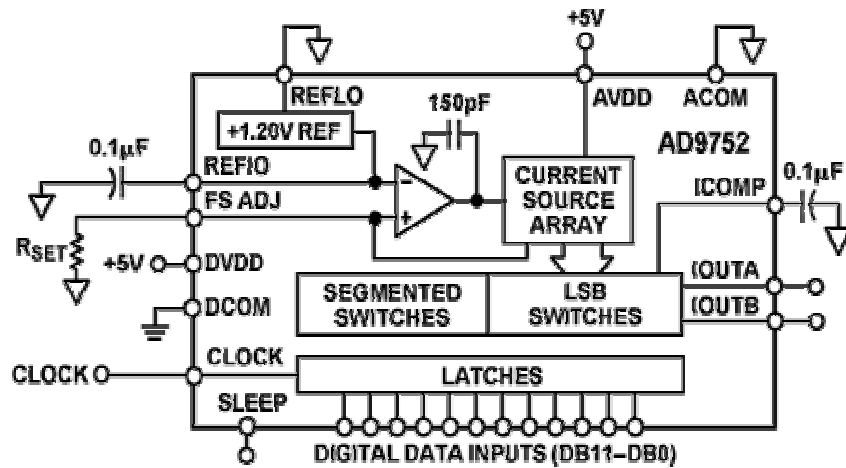


Abb. 6.3:
DAC mit paralleler Ansteuerung [12]

Zusätzlich können die nicht benötigten McBSP-Pins als GPIO verwendet werden. Da weder der McBSP1 noch McBSP0 verwendet werden, stehen dadurch zusätzlich 10 Output-Pins und 4 Input-Pins zur Verfügung. Somit werden die 10 vom McBSP freigewordenen Output-Pins und 2 weitere Interrupt-Pins als 12-bit Datenausgabe zum DA-Wandler vom Programmierbord verwendet. Ein weiterer Interrupt-Pin dient als Taktgenerator für den DAC. Der SLEEP-Pin und alle anderen Pins des DAC werden direkt am IC beschaltet. Zur Ansteuerung des DSK-Bords vom Bitgenerator (*s.a. Abb. 6.1*) werden ebenfalls die an "J3" bereitgestellten Pins verwendet. Der letzte noch freie, an "J3" bereitgestellte Interrupt-Pin, wird nicht als GPIO genutzt, sondern wie vorgesehen als Interrupt. Die entsprechende Interrupt-Service-Routine (*ISR*) wird vom Takt des Generators ausgelöst und dient zur Synchronisierung des Eingangssignals mit der Abtastfrequenz am Programmierbord (*Kap. 5.4*). Der letzte benötigte Pin ist ein Eingangspin der als Dateneingang genutzt wird, um den aktuellen Wert des digitalen Eingangssignals vom Generator zu erfassen. Dafür wird ein Eingangspin vom McBSP verwendet, da nur noch diese von den aufgeführten Pins zur Verfügung stehen.

Verwendung der zur Verfügung stehenden Pins			
Eingangsseite des Programmierbords		Ausgangsseite des Programmierbords	
Datensignal vom Generator	1 Eingangspin vom McBSP	DAC Daten (12bit)	10 Ausgangspins vom McBSP und 2 Ausgänge von den ext. Interrupts
Taktsignal vom Generator	1 Interrupt von den ext. Interrupts	DAC Takt	1 Ausgang von den ext. Interrupts

Tab. 6.1:
Verwendung der zur Verfügung stehenden Pins

Das dazugehörige Programm befindet sich im Anhang F.2. Es folgt in Kapitel 6.1.2.1 eine Erklärung der benötigten RegisterEinstellungen für die gewünschte Konfiguration.

6.1.2.1 Initialisierung der McBSP Register

Die Felddesreibungen für die in diesem Teil beschriebenen Register befinden sich in Anhang F.3. Zwei Bedingungen erlauben es, die seriellen Port Pins (*CLKX*, *FSX*, *DX*, *CLKR*, *FSR*, *DR*, and *CLKS*) als GPIO-Pins zu nutzen:

1. Die mit dem McBSP verbundenen Teile (*transmitter oder receiver*) befinden sich im Reset-Modus: (R/X)RST = 0 im "Serial Port Control Register" (*SPCR*)
2. Der General-Purpose I/O Modus ist für die benötigten Pins des McBSP aktiviert: (R/X)IOEN = 1 im "Pin Control Register" (*PCR*)

D.h. die Bits 0 und 16 im SPCR müssen auf 0 gesetzt werden, um die 1. Bedingung zu erfüllen. Zur Erfüllung der 2. Bedingung müssen im PCR das Bit 11 und 12 auf 1 gesetzt werden. Dadurch sind die Grundvoraussetzungen um den McBSP als GPIO zu nutzen erfüllt. Des weiteren ist die im Anhang F.3 dargestellte Tabelle für die entsprechenden Pins zu beachten, da sich nicht jeder Pin beliebig als Ausgang oder Eingang initialisieren lässt.

Für den Fall, dass z.B. das Bit 10 (*FSRM*) im PCR0 auf 0 gesetzt wird, ist der dazugehörige Pin an "J3" (*PIN29*) ein Eingang (*read only*) und dessen Zustand wird im PCR0 an Bit 2 (*FSRP*) reflektiert. Setzt man das FSRM auf 1 wird der entsprechende PIN ein Ausgang und der in das FSRP geschriebene Wert wird am FSR-Pin ausgegeben.

Genauso lassen sich der CLK(R/X) und FSX nutzen. Der DX hingegen lässt sich nur als Ausgang verwenden, während der DR und CLKS nur als Eingänge verwendet werden können.

Somit stehen durch die Initialisierung des McBSP0 als GPIO max. 5 Output-Pins und min. 2 Input-Pins zur Verfügung. Da ebenfalls der McBSP1 nicht verwendet wird, stehen für das Projekt 10 Output-Pins und 4 Input-Pins zur Benutzung als GPIO zur Verfügung.

Es folgt die Konfiguration der beschriebenen Register im Programm:

```
MCBSP_Config My_MCBSP_Config = {
    0x02000000, /* Serial Port Control Reg. (SPCR) 0x02000000 */
    0x00000000, /* Receiver Control Reg. (RCR) 0x00000000*/
    0x00000000, /* Transmitter Control Reg. (XCR) 0x00000000 */
    0x20000000, /* Sample-Rate Generator Reg. (SRGR) 0x20000000*/
    0x00000000, /* Multichannel Control Reg. (MCR) 0x00000000 */
    0x00000000, /* Receiver Channel Enable(RCER) 0x00000000 */
    0x00000000, /* Transmitter Channel Enable(XCER) 0x00000000 */
    0x00003F00 /* Pin Control Reg. (PCR) 0x00003F00*/
};
```

Abb. 6.4:
Initialisierung der McBSP-Pins als GPIO-Pins

Die Beschreibungen der Register, die zur Lösung der Aufgabe nicht benötigt werden, können [11] entnommen werden.

6.1.2.2 External Interrupt als General-Purpose I/O nutzen

Die in diesem Abschnitt verwendeten Feldbeschreibungen für die benötigten Register befinden sich in Anhang F.4. Da für die Ansteuerung des DAC neben den McBSP-Pins noch 3 weitere Output-Pins (*GPIO*) benötigt werden, wurden an J3 die zur Verfügung gestellten Interrupts (*4 bis 7*) verwendet. Diese lassen sich als GPIO-Pins nutzen, indem man die entsprechenden Bits im GPIO Enable-Register (*GPEN*) setzt. Wird z.B. eine logische 1 an die Bitposition 4 geschrieben, wird der dazugehörige Interrupt-Pin als GPIO eingerichtet (*Standard: Hohe Impedanz*).

Die gewünschte Funktion (*Ein- oder Ausgang*) des Pins wird im GPDIR festgelegt. Eine logische 0 an entsprechender Stelle initialisiert einen Eingang (*Standardeinstellung*). Dementsprechend initialisiert eine logische 1 einen Ausgang. Wird ein externer Interrupt als Ausgang eingerichtet, lässt sich der Interrupt-Pin über das entsprechende Bit im GPVAL-Register steuern. Für den Fall, dass ein Eingang gewählt wurde, lässt sich das aktuell am Pin anliegende Signal im GPVAL auslesen.

Zusätzlich kann bei der Wahl eines Einganges in den GPD-Registern (*GPIO Delta High Register und GPIO Delta Low Register*) festgestellt werden, ob eine Änderung des Eingangssignals von Low auf High (*GPxDH bit = 1*) oder von High auf Low stattgefunden hat (*GPxDL bit = 1*).

Abb. 6.5 zeigt das Funktionsblock-Diagramm vom GPIO-Pin über die Flankenerkennung bis zu den entsprechenden Registern.

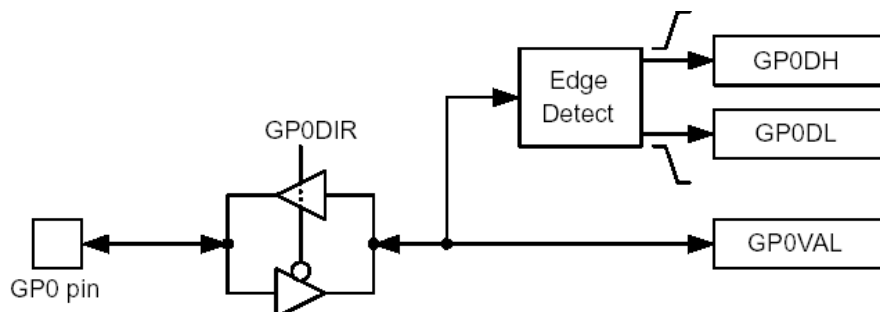


Abb. 6.5:
Funktionsblock Diagramm [11]

Es folgt die Konfiguration der beschriebenen Register im Programm:

```

GPIO_Config My_GPIO_Config = {
    0x00000000, // Uint32 gpgc-register;
    0x000000F0, // Uint32 gpen-register;
    0x000000E0, // Uint32 gpdir-register;
    0x00000000, // Uint32 gpval-register;
    0x00000010, // Uint32 gphm-register;
    0x00000000, // Uint32 gplm-register;
    0x00000010 // Uint32 gppol-register;
};

```

Abb. 6.6:
Initialisierung der ext. Interrupt-Pins als GPIO-Pins

Die restlichen bei der Initialisierung angesprochenen Register werden für die Verwendung von Interrupts benötigt und wurden daher bereits in Kapitel 5.5 beschrieben.

6.1.2.3 Programm mit Interrupt und Ausgabe über GPIO-Pins an "J3"

Das Struktogramm für den programmierten Quelltext ist in Abb. 6.7 für die 2. Ordnung dargestellt.

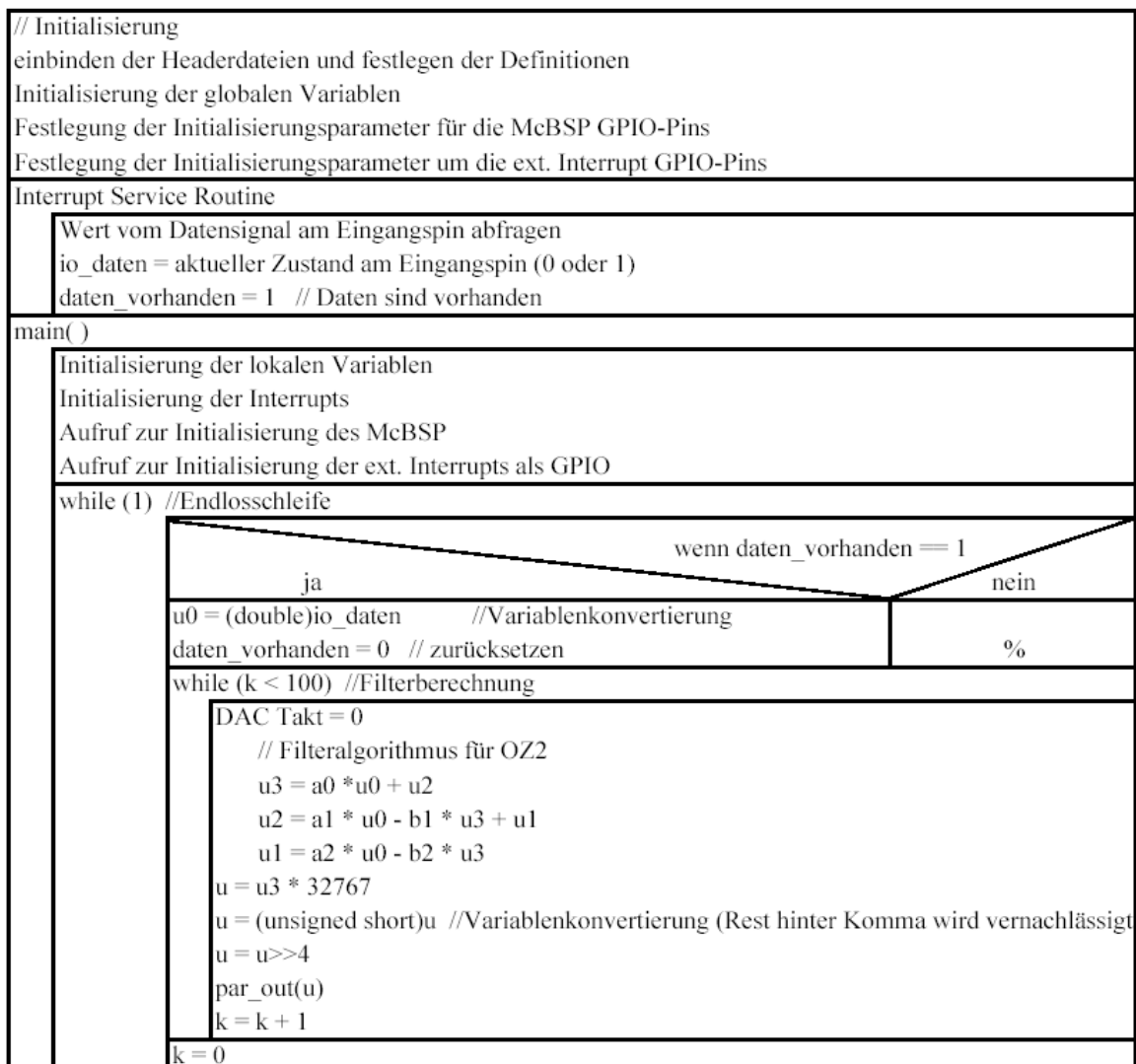


Abb. 6.7a:
Struktogramm für GPIO-Programm (Quelltext)

<pre> par_out(u) // Ausgabefunktion { An dieser Stelle wurden unterschiedliche, parallele Ausgabeverfahren für den DAC ausprobiert DAC Takt = 1 return 0 } </pre>

Abb. 6.7b:
Struktogramm für GPIO-Programm (Quelltext)

Da mit dem dazugehörigen, in Anhang F.5 angefügten Programm nur eine Durchlaufzeit von $3,03\mu\text{s}$ (330KHz) erreicht wurde und die Ausgabe des Ergebnisses der Filterberechnung die größte Zeitspanne benötigt, wurde dieser Programmteil verbessert. Der Grund weshalb eine Ausgabe in dieser Form relativ lange dauert, ist dass zur Ansteuerung jedes Pins ein Registerzugriff programmiert wurde, wodurch die Durchlaufzeit erheblich vergrößert wird. Daher wurde die Ausgabe bezüglich der Ausgabegeschwindigkeit optimiert und die folgende Ausgabevariante programmiert.

```

int par_out(u)
{
    MCBSP_RSET(PCR0, ((u & 0x2F)+0x3F00));
    GPIO_pinWrite(hGpio0,GPIO_PIN6, ((u & 0x10)>>4));
    u= (u>>6);
    MCBSP_RSET(PCR1, ((u & 0x2F)+0x3F00));
    GPIO_pinWrite(hGpio0,GPIO_PIN7, ((u & 0x10)>>4));
}

```

Abb. 6.8:
Programmoptimierung 1

Mit einer Ausgabe in dieser Form wurde bereits eine Durchlaufzeit von $1,5\mu\text{s}$ (666KHz) erreicht, allerdings lässt sich auch hier noch ein Registerzugriff einsparen:

```

int par_out(u)
{
    MCBSP_RSET(PCR0, ((u & 0x2F)+0x3F00));
    GPIO_RSET(GPVAL, (((u & 0x10)<<2) + ((u & 0x400)>>3)+0xE10));
    u= (u>>6);
    MCBSP_RSET(PCR1, ((u & 0x2F)+0x3F00));
}

```

Abb. 6.9:
Programmoptimierung 2

Auf diese Weise wurde nun bereits eine Durchlaufzeit von $1,2\mu\text{s}$ (830KHz) erreicht. Da eine weitere Optimierung der Ausgabe durch das Einsparen von Registerzugriffen nicht mehr möglich ist (*3 Zugriffe für 3 Register*), wurden die in Kapitel 5.1.2 beschriebenen Optimierungsstufen des "Code Composer Studio" (*-o0 bis -o3*) genutzt.

Dabei wurde festgestellt, dass das Programm mit den Optimierungsstufen "-o3" und "-o2" nicht mehr funktionsfähig ist. Die höchste Optimierungsstufe, die das Programm nicht in

seiner Funktion beeinträchtigt ist der "-o1-Optimizationlevel", dieser verkleinert die Durchlaufzeit auf 1µs (1MHz).

Eine Tabelle für die Belegung der Pins an Steckerleiste "J3" findet sich in Anhang F.5.

Da die geforderte Durchlaufzeit von ca. 156,25ns (6,4MHz) bei einem Programm mit "Fließkommakoeffizienten" und Ausgabe über GPIO-Pins nicht erreicht wird, wurde eine andere Möglichkeit der parallelen Ausgabe zum DAC gesucht.

6.1.3 Parallele Ansteuerung des DA per EMIF

Eine voraussichtlich wesentlich schnellere Art den AD9752 anzusteuern, ist über das "Externe Memory Interface" (EMIF). Das EMIF ist ein 32Bit breites Bussystem und kann mit einer maximalen Taktfrequenz von 90MHz arbeiten, weshalb eine Ausgabe über dieses Interface getestet wurde. Die entsprechenden Pins werden an der Steckerleiste J4 bereitgestellt. Es werden serielle Widerstände an den Ausgängen des EMIF benötigt, was bei der Bestellung berücksichtigt wurde um die angegebene Signalqualität gewährleisten zu können [11].

Die Steuerung des EMIF wird von Speicheradressen gebundenen Registern übernommen.

Um das EMIF nutzen zu können, muss zunächst der Daughtercard detect Pin auf Low gelegt werden, dann lässt sich das EMIF durch einfaches Beschreiben an der richtigen Speicheradresse ansteuern. Die zur Ansteuerung des EMIF benötigten Register sind mit entsprechenden Feldbeschreibungen im Anhang F.10 angefügt und bereits bei der Programmierung mit dem ADDA16-Aufsteckmodul verwendet worden.

Es folgen die verwendeten Befehle zur Nutzung und Initialisierung des EMIF:

```
#include <cs1_emif.h>

#define WRITE_DAC(val)  (*(volatile short*)(0xB0200000) = val)

EMIF_Config MyEmif = {
    0x00003368,          /* gblctl  Global control */
    0x00000030,          /* cectl0  CE0 control */
    0xffffffff03,       /* cectl1  CE1 control */
    0x00000030,          /* cectl2  CE2 control */
    0x15114211,          /* cectl3  CE3 control*///      0x10914221
                          0x1F117C11 0x6F317C11 0x15114211 =1 20 1 ...
    0x07117000,          /* sdctl   SDRAM control register */
    0x00000546,          /* sdtim   SDRAM timing register (refresh period)*/
    0x00054519,          /* sdext   SDRAM extension register*/
};                       /* for a detailed description look at spr433b.pdf*/

EMIF_config(&MyEmif);   // configure EMIF as indicated above

WRITE_DAC((short)u3);   //EMIF Pin D0-D15
```

Abb. 6.10:
Programmausschnitt mit
EMIF-Initialisierung

Es wird also lediglich der Wert für u3 an die Speicheradresse **0xB0200000** geschrieben und die entsprechenden Pins des EMIF werden gesetzt.

Wie erwartet wurde eine wesentlich schnellere Ausgabe über die parallele Ansteuerung des EMIF erreicht, weshalb diese Methode zur Realisierung des Systems verwendet wurde.

6.2 DA-Wandler Evaluationsbord mit AD9752

Da der ausgewählte DAC "AD9752" (s.a. Kap. 5.6 und Kap. 6.1.2) auf einem für mehrere Strom-Spannungswandlungsverfahren vorbereiteten und störungsarmen Evaluationsbord (EB-9752) erhältlich ist und dieses im preislich vorgegebenen Rahmen liegt, wurde ein EB-9752 bestellt. Der dazugehörige Schaltplan ist in Anhang F.7 angefügt. Geliefert wurde ein Bord ohne den im Anhang rot markierten Ausgangsteil (Abb. 6.11).

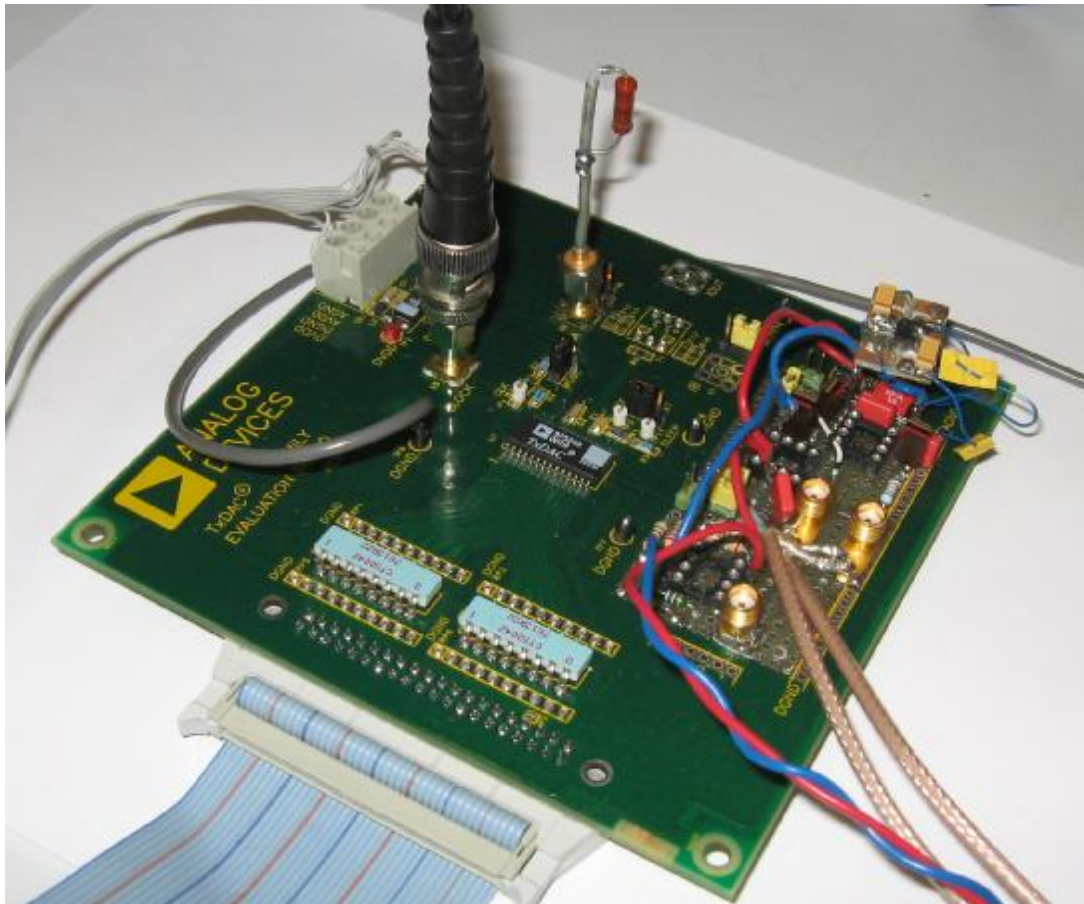


Abb. 6.11:
EB-9752 (geliefert)

Bis zur Nachlieferung der fehlenden Operationsverstärker (AD8047) wurde daher in einem ersten Schritt der fehlende Strom-/Spannungswandlerteil durch einen 50Ω -Widerstand realisiert. Zu erkennen ist in der Abbildung 6.11 die Ansteuerung der parallel liegenden Dateneingänge vom Programmierbord über das Flachbandkabel. Zwischen zwei Datenleitungen ist immer eine Masseleitung vorgesehen worden, da dies eine relativ störungsarme Übertragung sichert. Um weitere Störungen zu vermeiden, wurde die Taktleitung über ein separates, abgeschirmtes Kabel (BNC) geführt.

Die Ansteuerung des Takt-Pins (*Clock*) wird nach dem in Abbildung 6.12 gezeigten Zeitschema, von einem als Ausgang eingerichteten GPIO-Pin des DSP-Bords übernommen.

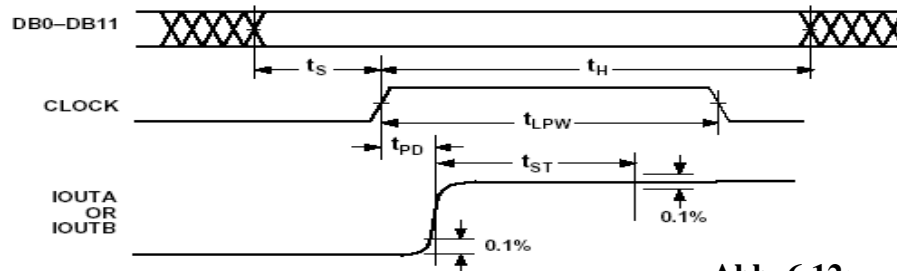


Abb. 6.12:
Zeitliches Ablaufdiagramm zur Ansteuerung des AD9752 [12]

Die entsprechenden Zeitangaben sind dem Anhang F.8 oder [12] zu entnehmen.

Die in Kapitel 6.1.2.3 geforderten seriellen Widerstände befinden sich bereits auf dem gelieferten Evaluationsbord. Die Stromversorgung für den digitalen Teil des EB-9752 (+3,3V) wird von dem an der Steckerleiste "J3" bereitgestellten Pin abgegriffen. Diese Leitung wird gemeinsam mit der Versorgung des analogen Teils (+5V von "J3") an den entsprechenden Klemmblock geführt.

6.2.1 "Unbuffered Voltage Output"-Modus

In diesem Modus wird die Strom-/Spannungswandlung (*I/U-Wandlung*) mit einem 50Ω Widerstand durchgeführt. Für einen ersten Funktionstest ist dies absolut ausreichend. Später wurde der gewünschte Ausgangsteil des bestellten EB-9752 auf dem dafür vorgesehenen Entwicklungsteil des gelieferten EB-9752 mit den nachgesendeten Bauteilen entworfen.

Der Schaltplan für die Strom-/Spannungswandlung mit einem 50Ω Widerstand ist Abb. 6.13 zu entnehmen. Der unbenutzte Ausgang kann entweder direkt an Masse oder über einen "Ladewiderstand", wie in der Abbildung dargestellt, betrieben werden. Dieser Schaltungsaufbau wird als "Unbuffered Voltage Output"-Modus bezeichnet und unterstützt ein unipolares Ausgangssignal.

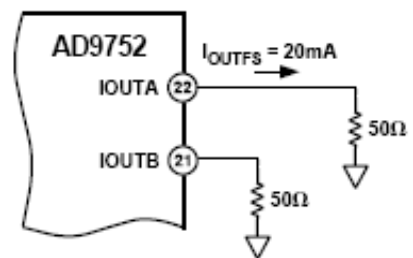


Abb. 6.13:
0V bis 1V "Unbuffered Voltage Output"

Da es zunächst gilt, die Tiefpassfilter der analogen "Ti-TP Kette" nachzubilden und die entsprechenden, zu programmierenden digitalen Tiefpassfilter mit dem normierten Wertebereich von 0 bis 1 ($U_{OUTA} = I_{OUTFS} \cdot R_{LOAD} = 20mA \cdot 50\Omega = 1V$) arbeiten, ist zunächst ein unipolares Ausgangssignal ausreichend.

Hinsichtlich einer optimalen integralen Nicht-Linearität (*INL*) empfiehlt es sich allerdings den "Unipolar, Buffered Voltage Output"-Modus zu verwenden (*s.a. Kap. 6.2.2*).

Beim Aufbau wurde ein $50\ \Omega$ SMD-Widerstand an den dafür vorgesehenen Platz auf dem Evaluationsbrett gelötet. Um die Ergebnisse der Messung für spätere Vergleiche ohne Änderung der Hardware wieder herstellen zu können, wurde ein SMA-Stecker mit einem $50\ \Omega$ Widerstand versehen (Abb. 6.14). Diese Vorgehensweise ist zwar sehr störungsbehaftet, ermöglicht aber eine schnelle Kontrolle der Funktionsfähigkeit des "AD9752".



Abb. 6.14:
50 Ω -SMA Stecker

Das vorgestellte System wurde mit dem in Kapitel 6.1.4 entwickelten Programm in Betrieb genommen und die Funktionsfähigkeit geprüft (Abb. 6.15). In den Abbildungen ist die Nachbildung des Signalverlaufs gut zu erkennen.

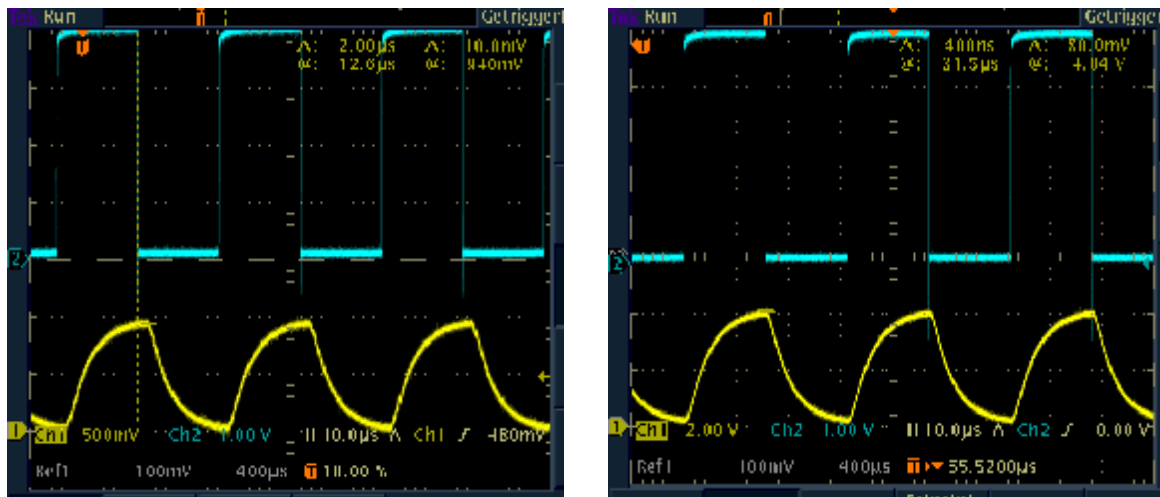


Abb. 6.15:
Das Bild links zeigt die Messung des digitalen IIR-Filters (OZ2) bei 60-fach Oversampling im "Unbuffered Voltage Output"-Modus mit $50\ \Omega$ -SMA Stecker. Rechts ist die Messung des analogen TP-Filters der "Ti-TP Kette" (OZ2) dargestellt.

Der berechnete Ausgangspegel von ca. 1V wird erreicht. Um das Signal auf den gewünschten Spannungspegel zu verstärken reicht es nicht den Widerstand zu erhöhen, da der "AD9752" nur einen begrenzten Strom liefert. Daher wird eine Operationsverstärkerschaltung benötigt. Die nachgelieferten Operationsverstärker sind für eine Betriebsspannung von $\pm 5V$ ausgelegt, weshalb nach der geplanten I/U-Wandlung nur eine Ausgangsspannung von $\pm 3V$ möglich ist. Die gewünschte Spannungsverstärkung wird daher mit Hilfe des Interpolationstiefpasses (s.a. Kap. 6.3) durchgeführt.

Zur Betrachtung der Auflösung des DAC wurde der Signalverlauf genauer betrachtet (Abb. 6.16).

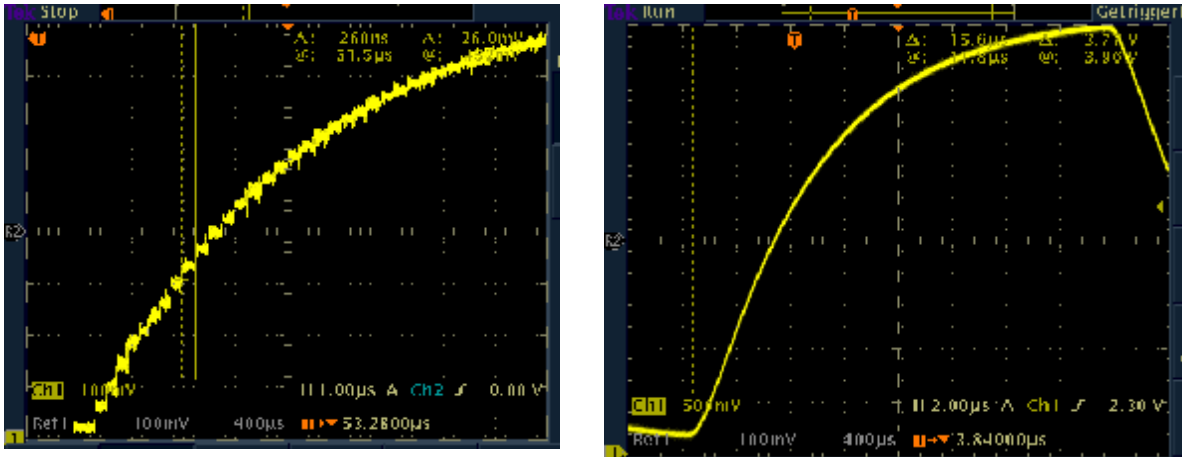


Abb. 6.16:

Das Bild links zeigt die einzelnen Stufen des Ausgangssignals für das digitale IIR-Filter mit maximal 1V (OZ2) bei 60-fach Oversampling im "Unbuffered Voltage Output"-Modus. Rechts die Messung das analoge TP-Filter (OZ2) zum quantitativen Vergleich mit einer Amplitude von 4V.

Deutlich sind beim digitalen Filter die Treppenstufen zu erkennen. Die Dauer einer Stufe beträgt $T = \frac{1}{f_A}$ mit f_A : Abtastfrequenz am Filter.

Diese Stufen gilt es zu interpolieren (Kap. 6.3), damit ein kontinuierliches Signal entsteht. Die Störungen im Signal, bei der Messung mit einem auf dem EB-9752 aufgelöteten SMD-Widerstand, fallen zwar geringer aus, sind aber immer noch deutlich zu erkennen. Da jedoch eine Lösung mit Operationsverstärker (OP) statt Widerstand als Strom-/Spannungsumsetzung bevorzugt wird (auf Grund der INL), wurde dieser Lösungsansatz nicht weiter verfolgt.

6.2.2 "Unipolar Buffered Voltage Output"-Modus

Zur Verbesserung der INL wurde die Strom-/Spannungswandlung mit der in Abb. 6.17 gezeigten Schaltung mit den nachgelieferten OPs realisiert. Zu beachten ist, dass der entsprechende Jumper, um I_{OUTB} direkt an Masse anzuschließen, gesetzt werden muss.

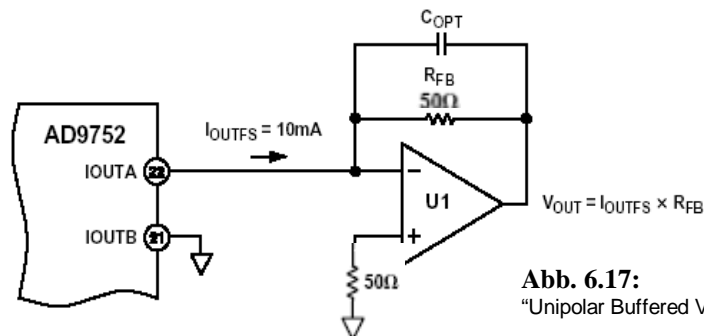


Abb. 6.17:
"Unipolar Buffered Voltage Output"

Zur Erzeugung der Referenzspannung wurde zunächst die interne Referenzquelle genutzt, da u.a. das entsprechende Referenzspannungsbauteil "REF43" noch nicht geliefert wurde.

Somit bestimmt sich der Ausgangsstrom nach:

$$I_{REF} = \frac{V_{REFIO}}{R_{SET}} = \frac{1,2V}{2K\Omega} = 0,6mA \quad \text{à} \quad I_{OUTFS} = 32 \cdot I_{REF} = 32 \cdot 0,6mA = 19,2mA$$

Der entsprechende Ausgangsstrom zum aktuellen Datenwert an I_{OUTA} ergibt sich nach $I_{OUTA} = \left(\frac{DAC_Code}{2^{12}} \right) \cdot I_{OUTFS}$. Der maximale Ausgangsstrom ist gleich I_{OUTFS} . Die Spannung an dem 50Ω Widerstand zur Strom-/Spannungswandlung beträgt demnach max. $U_{OUTA_max.} = I_{OUTFS} \cdot 50\Omega = 0,96V$. Der Aussteuerbereich des DA beträgt somit $0V$ bis $0,96V$.

Ein Nachteil der empfohlenen OPs ist die geringe Ausgangsspannung, was auf die relativ niedrige Versorgungsspannung von $\pm 5V$ zurückzuführen ist.

Die maximale Ausgangsspannung des AD8047 von $\pm 4V$ wird bei einer Versorgungsspannung von $\pm 6V$ erreicht. Laut Datenblatt ist der AD8047 aber sogar nur für $\pm 3V$ Ausgangsspannung vorgesehen, womit die geforderten Ausgangsspannungswerte nicht erreicht werden. Daher wird die Amplitude mit dem sowieso benötigten ITP verstärkt (s.a Kap. 6.3).

Die Messung an der aufgebauten I/U-Wandlung mit OP wird in Abb.6.18 gezeigt.

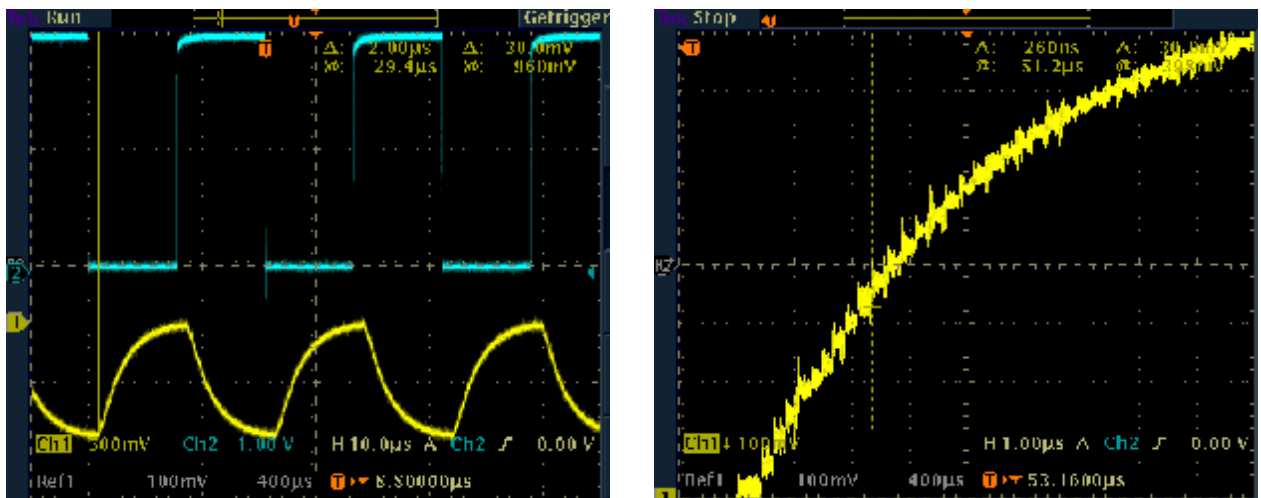


Abb. 6.18:

Das Bild links zeigt die Messung des digitalen IIR-Filters (OZ2) bei 60-fach Oversampling im "Single-Ended Buffered Voltage Output"-Modus. Rechts die Messung der einzelnen, zu interpolierenden Stufen des Ausgangssignals für das digitale IIR-Filter (Messergebnis am Oszilloskop invertiert).

Da die Ausgangsspannung nach der Strom-Spannungswandlung negativ ist, wurde ein Standardinverter mit dem AD8047 nachgeschaltet (Abb. 6.19).

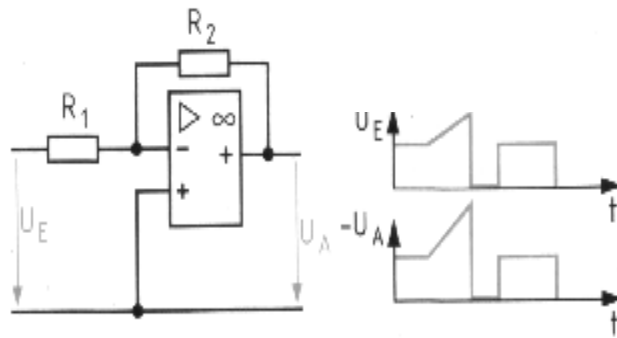


Abb. 6.19:
Standard Inverterschaltung zur
Invertierung des Ausgangssignals

Die Messung am Ausgang des Inverters ist in Abb. 6.20 dargestellt.

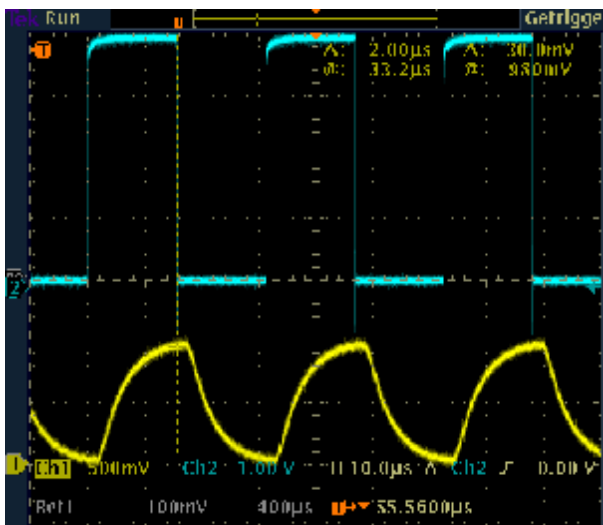


Abb. 6.20:
Digitales IIR-Filters (OZZ) bei 60-fach Oversampling im "Single-Ended Buffered Voltage Output"-Modus hinter dem Inverter-OP.

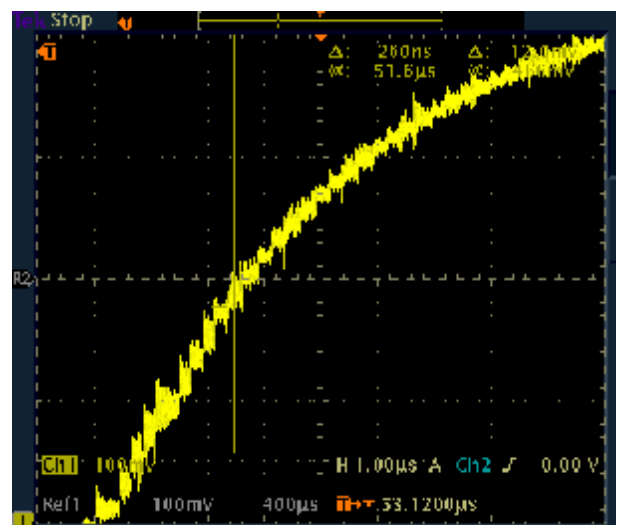


Abb. 6.21:
Zu interpolierende Treppenstufen

Deutlich ist der erhöhte Störungsanteil im rechten Bild zu erkennen. Die wird mit jedem weiteren OP zunehmen. Im Vergleich zur I/U-Wandlung mit dem 50Ω Widerstand sind die Störungen in diesem Schaltungsaufbau stärker, was auf die selbst entwickelten Schaltungsaufbau zurückzuführen ist. Von Bedeutung ist die verbesserte INL.

In Abb. 6.22 ist der gesamte Darstellungsbereich des Oszilloskops ausgenutzt worden, um das digitale mit dem analogen Modell vergleichen zu können. Deutlich sind die erhöhten Störungen im Ausgangssignal des digitalen Systems zu erkennen, was im Wesentlichen auf den störungsbehafteten Schaltungsaufbau zurückzuführen ist. Es ist zu untersuchen, welchen Einfluss der Interpolationstiefpass auf die Störungen hat.

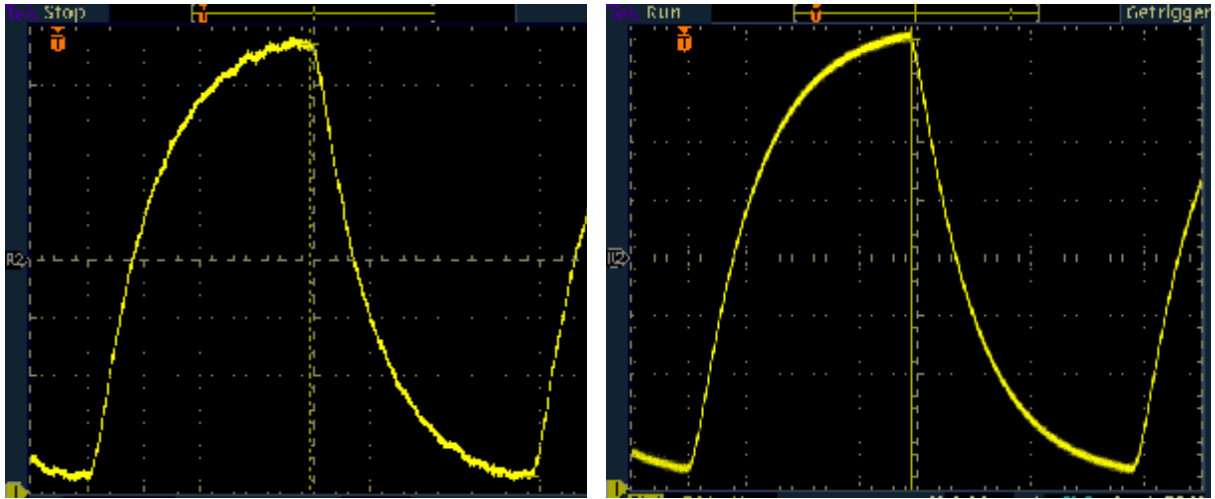


Abb. 6.22:
Das Bild links zeigt die Messung des digitalen IIR-Filters (OZ2) bei 60-fach Oversampling im "Single-Ended Buffered Voltage Output"-Modus hinter dem Inverter-OP mit variabel vergrößerter Darstellung. Rechts die Messung des analogen Filters zum Vergleich mit 500mV/div.

6.3 Realisierung des ITP bzw. Rekonstruktionsfilter

Da die nachgelieferten OPs (*AD8047*), wie bereits erwähnt, mit keiner ausreichenden Ausgangsspannung arbeiten, wurde beim ITP (*Interpolationstiefpass*) der TL074 mit einer Versorgungsspannung von $\pm 15V$ verwendet. Laut Datenblatt wird mit diesem OP eine Ausgangsspannung von $\pm 12V$ erreicht. Der TL074 zeichnet sich neben seinem geringen Preis durch die entscheidende Eigenschaft eines Verstärkungs-Bandbreite-Produkt¹ von 3MHz aus.

B ₁	Unity-gain bandwidth	25°C	3	MHz
----------------	----------------------	------	---	-----

Abb. 6.23:
Verstärkungsbandbreiteprodukt des TL074

Der Vorteil Verstärkungs-Bandbreite-Produkts von 3MHz ist, das die periodischen Fortsetzungen des Signalspektrums durch die erreichte Abtastfrequenz am Filter von 3,84MHz automatisch gedämpft werden. Zudem wird das eigentliche Signalspektrum kaum durch die relativ weit auseinanderliegenden Fortsetzungen beeinflusst.

Für den ITP wurde daher nach ausführlicher Simulation mit MatLab (*für alle Ordnungszahlen der "Ti-TP Kette" mit $f_a = 3,84MHz$*) ein Butterworth-TP-Filter 2. Ordnung mit einem TL074 als OP aufgebaut.

¹ auch *gain-bandwidth-product* oder *unity gain frequency*.

Die Entwicklung der dazugehörigen Filter-Schaltung nach dem Sallen-Key Algorithmus findet man z.B. in "Das Aktiv Kochfilter Buch" von Don Lancaster [8].

Der letztendlich Entwurf erfolgte mit Hilfe eines Filter Design Tools der Firma Analog Devices [13]. Auf der Internetseite (*s.a. Anhang F.9*) lässt sich ein "Aktiv Filter Tool" nutzen, welches die benötigten Bauteile nach dieser Methode bestimmt.

Die dazugehörige aufgebaute Schaltung ist in Abb. 6.24 zu sehen und wies, nachdem ein einstellbarer Kondensator eingelötet wurde, die erwarteten Eigenschaften auf. Der Frequenzgang des aufgebauten ITP ist in Abb. 6.25 dargestellt.

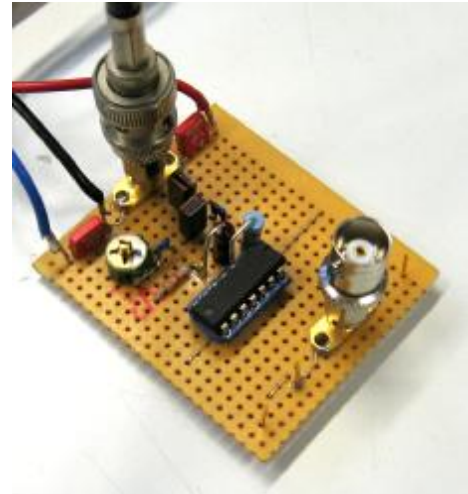


Abb. 6.24:
ITP mit TL074

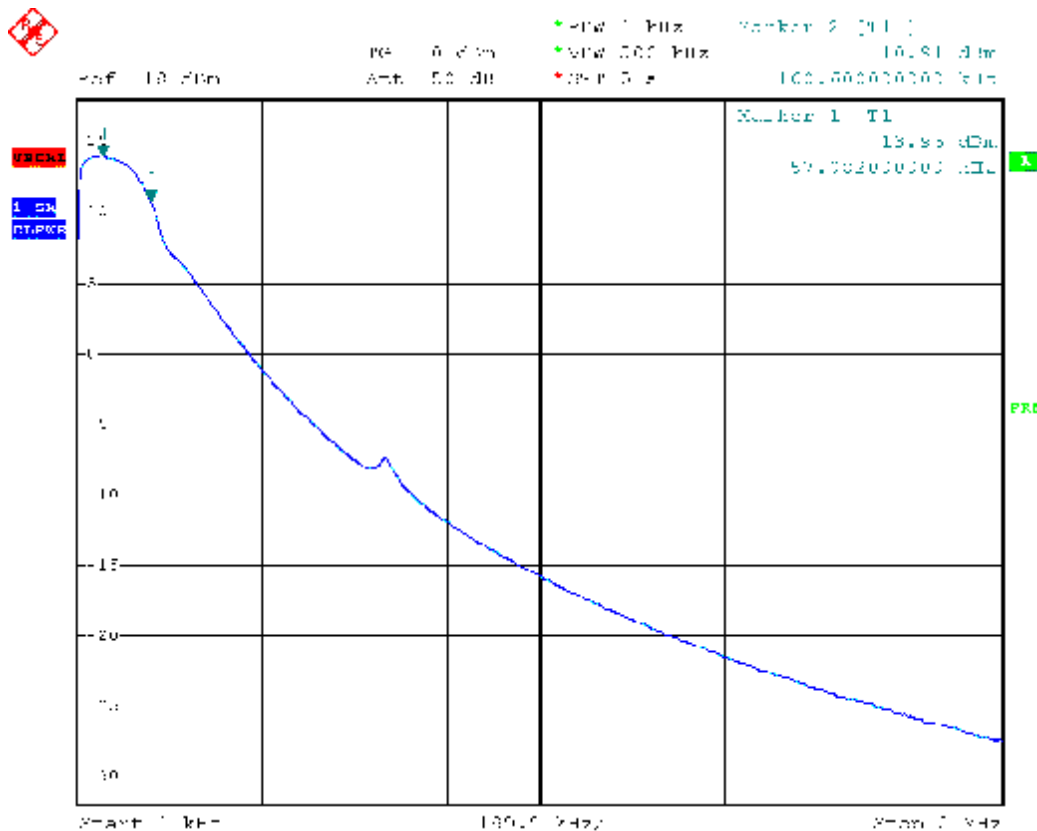


Abb. 6.25:
Amplitudengang des ITP

Zwar ist im Amplitudengang bei ca. 650KHz ein Abweichung vom gewünschten Verlauf zu erkennen. Dies hat allerdings keine Auswirkungen auf die Funktion, da die in diesem Bereich auftretenden Signalfrequenzen nicht signifikant zur Signalbildung beitragen.

Es gilt die in Abb 6.26 gut zu erkennenden hochfrequenten Störungen zu beseitigen und die Treppenstufen zu interpolieren.

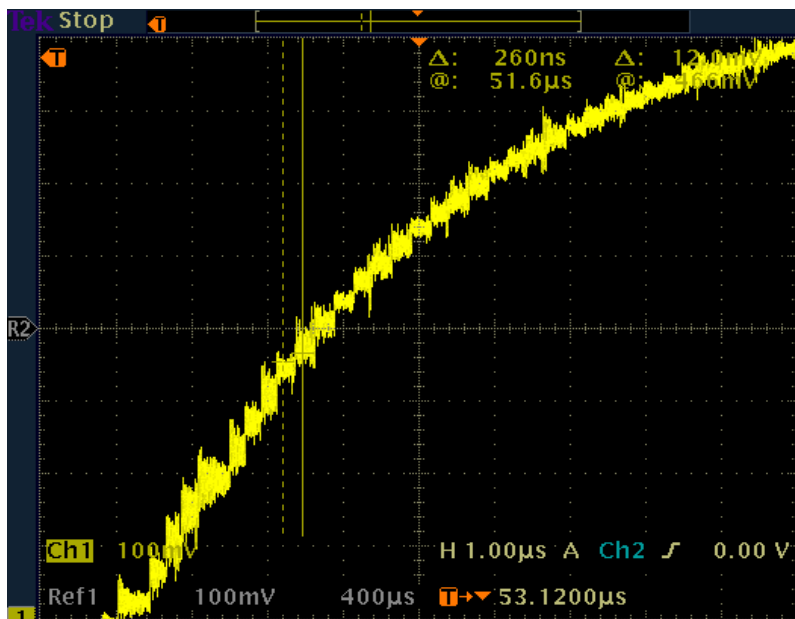


Abb. 6.26:
Ausgangssignal vor dem ITP

Es folgt in Abb. 6.27 ein Vergleich des analogen Filters mit dem digitalen IIR-Filter 2. Ordnung.

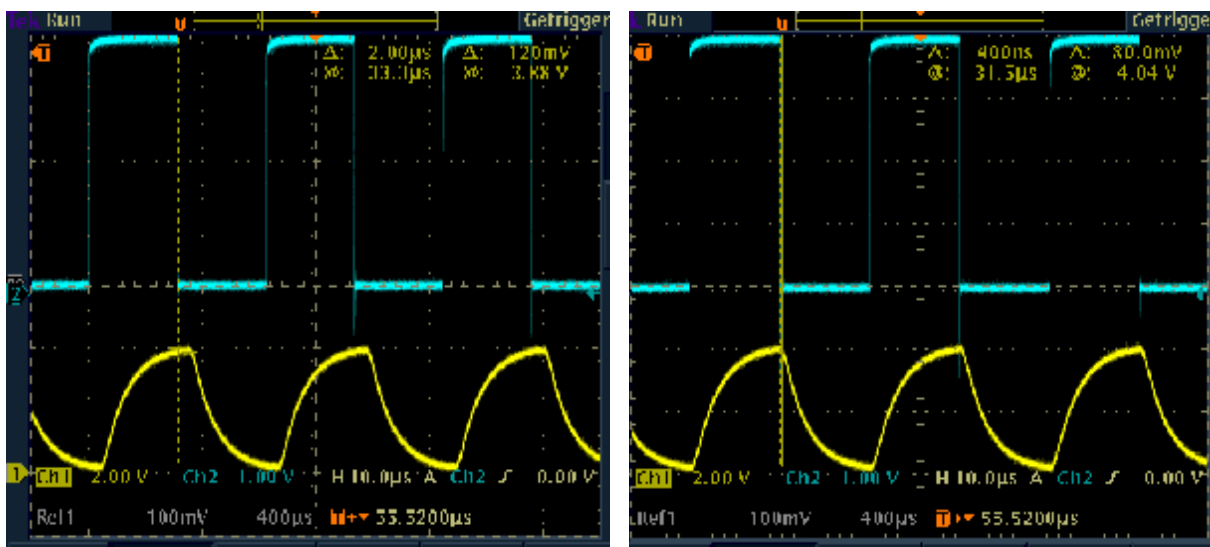


Abb. 6.27:
Das Bild links zeigt die Messung des digitalen IIR-Filters (OZ2) bei 60-fach Oversampling und rechts die Messung des analogen Filters zum Vergleich

Daher ist in der Abb. 6.27 die Nachbildung des geforderten Signalverlaufs mit diesem System gut zu erkennen. Die hochfrequenten Störungen sind, neben der erfolgreichen Interpolation durch den ITP, erheblich geringer geworden (*im Vergleich zu Kap. 6.2.2*), was auch genau aus Abb. 6.28 hervorgeht.

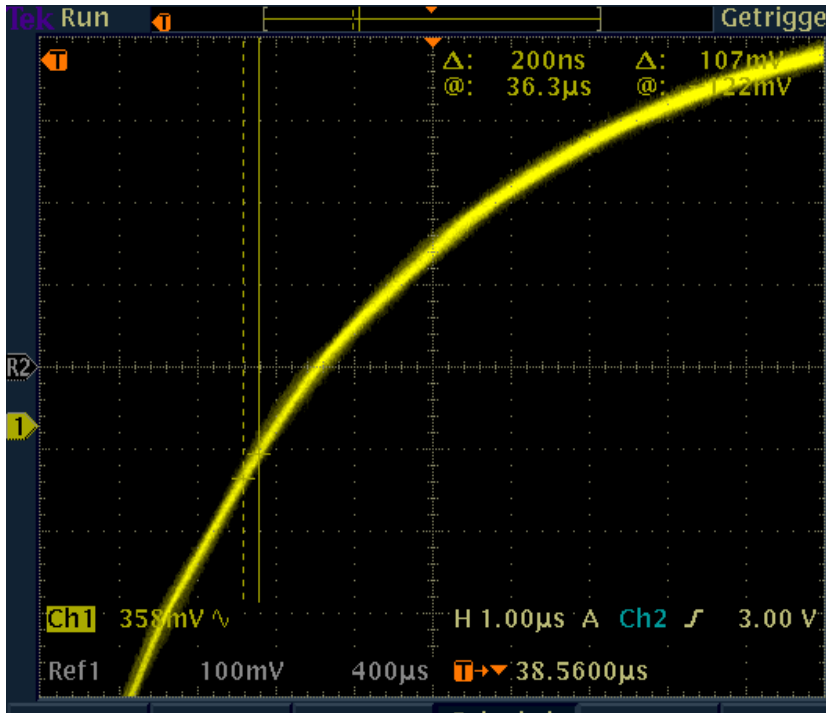


Abb. 6.28:
Interpoliertes Ausgangssignal

Das entworfene System kann also zur vollständigen Nachbildung des im Labor der Übertragungstechnik genutzten Kanalmodells, in Bezug auf digitale Eingangssignale ersetzt werden.

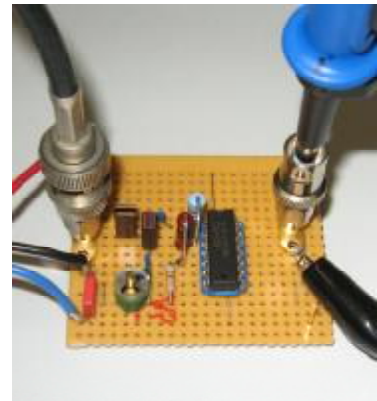
Als nächstes wurde die unipolare Ausgabe um den negativen Spannungsbereich erweitert. Dies ist notwendig, da das entworfene System auch die HP-Einsteckkarte nachbilden soll und dieses die Ausgabe von negativen Spannungswerten erforderlich macht.

Es folgt auf der nächsten Seite eine Übersicht des erstellten Systems mit Hilfe von Abbildungen (Abb. 6.29).

DAC EB-9752



ITP



Zu interpolierendes Signal vom DAC nach der I/U-Wandlung



Stromversorgung + 12bit Datensignal (+ Taktsignal über hier nicht dargestelltes BNC-Kabel)



TMS320C6713 DSK-Board

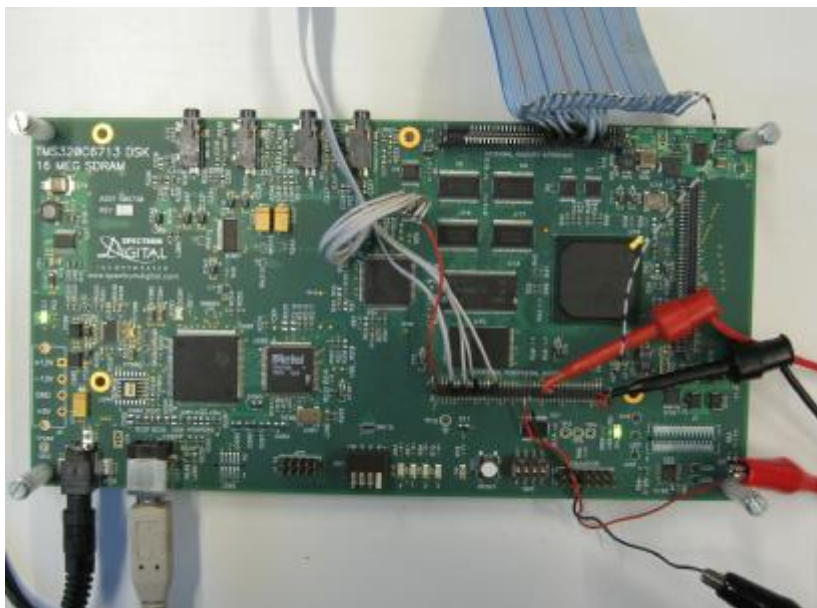


Abb. 6.29: Übersichtplan des Testaufbaus für die "Ti-TP Kette" (rote und schwarze Klemmen sind der Eingangs-takt und das Datensigna vom Generator)

Der Amplitudengang des Systems kann nicht ermittelt werden, da dazu ein ADC benötigt werden würde. Daher wurde im nächsten Kapitel versucht das Signalspektrum zu ermitteln. Zu beachten ist bei den Messungen, dass sich der benutzte GPIO-Pin bei negativen Spannungen am Eingang verhält wie ein Kurzschluss.

6.4 Spannungsversorgung mit dem LM2662

Da für den Betrieb des gesamten Aufbaus bereits 3 externe Spannungsversorgungen benötigt werden (230V/50Hz des DSK-Netzteils, +/-15V für ITP-OP TL074, +/-5V für IU-Wandlungs- und Inverter-OP AD8047), wurde mit Hilfe des LM2662 und der +5V Versorgungsspannung des DSK-Bords eine eigene -5V "on-board"-Quelle aufgebaut.

So kann schon mal auf die ext. Quelle zur Versorgung des AD8047 verzichtet werden. Die ext. +/-15V Versorgung für den ITP wollte ich ebenfalls durch die am Stecker J3 des DSK-Bords bereitgestellten +/-12V Versorgungspins ersetzen. Leider sind diese jedoch nur aktiv, wenn das Bord nicht über das mitgelieferte Netzteil an J5 versorgt wird, sondern über die für ein Computernetzteil vorbereiteten Steckerkontakte an J6. Deshalb wurde zunächst die externe Versorgung des ITP über ein Netzteil beibehalten.

6.5 Hochpass

Da zusätzlich auch ein Hochpass auf dem DSK Bord programmiert werden soll und dieser einen normierten Wertebereich von -1 bis +1 aufweist wird auch die Ausgabe von negativen Spannungswerten erforderlich. Die nach Kapitel 3.8 ermittelten Koeffizienten für ein die HP-Einstereckkarte sind im Anhang E3 angefügt.

Zu beachten ist beim Hochpass, dass der Gleichanteil nicht übertragen wird.

Der Filteralgorithmus für den HP 1. Ordnung gleicht dem für den TP, bis auf die Werte der Koeffizienten. Für die 2. Ordnung rechnet der Algorithmus mit zwei aufeinanderfolgenden Filteralgorithmen 1. Ordnung (*Abb. 6.30*).

```
u0 = (float)io_daten;

// HP 2.Ordnung mit 60fach Oversampling
while (k<60)
{
    WRITE_GPVAL(0xE10);
    u4 = a01*u0+u3;
    u3 = a11*u0-b11*u4;

    u2 = a02*u4+u1;
    u1 = a12*u4-b12*u2;

    u = (u2+1)*32267;

    WRITE_DAC((unsigned short)u2); //float to unsigned short conversion
    WRITE_GPVAL(0xE30);
    k=k+1;
};
k=0;
```

Abb. 6.30:
HP-Algorithmus 2. Ordnung

Um auch negative Spannungswerte ausgeben zu können, bietet sich die in Abb. 6.31 gezeigte Strom-Spannungswandler Schaltung mit einem Spannungsteiler am positiven Eingang zur Offsetspannungserzeugung an.

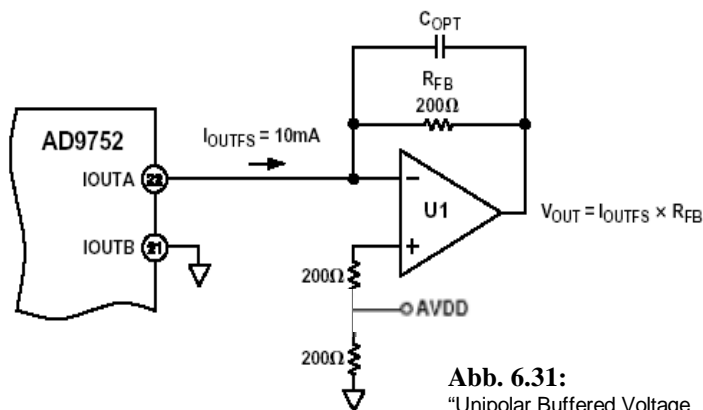


Abb. 6.31:
"Unipolar Buffered Voltage Output" Modus mit Offset

Da die Ausgangsspannung aufgrund der Strom-Spannungswandlung wie in Kap. 6.2.2 wieder invertiert ausgegeben wird, wurde die bereits verwendete Inverterschaltung nachgeschaltet. Allerdings tritt nun das Problem auf, dass der Inverter beim Wechsel von negativen nach positiven Spannungen nicht richtig arbeitet. Daher wurde eine Strom-Spannungswandlung mit einem Differenzverstärker bevorzugt, um einen Tiefpass und Hochpass über den selben Signalausgangspfad zu betreiben.

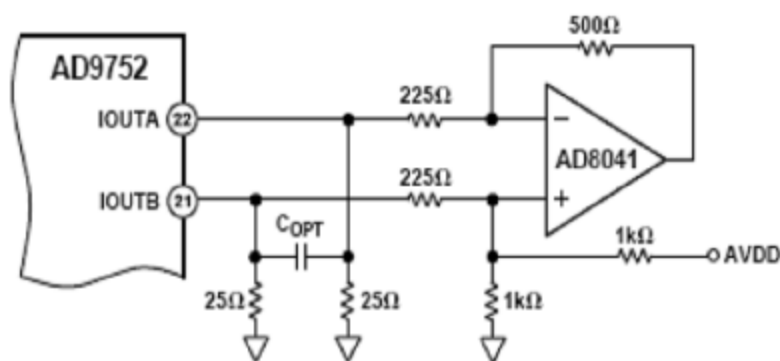


Abb. 6.32:
"Single-Supply DC Differential". Modus

Eine Berechnung im Programm sollte mit einem Zahlentyp erfolgen, der negative und positive Werte verarbeiten kann (*kein unsigned verwenden*). Der berechnete Wert wird dann durch eine Addition auf einen im jeden Fall positiven Wert "gehoben" (*unsigned verwenden möglich*) und z.B. als 16bit-Wert ausgegeben.

Dieser kleinste vorzeichenlose 12Bit-Ausgabewert (0000 0000 0000) entspricht 0mA und soll dann dem maximalen negativen Spannungsausgabewert am Ausgang erzeugen (*mehr dazu in Kap. 7 "Verarbeitungskette"*).

7 Realisierungskonzept der "Ti-TP Kette" und HP-Einsteckkarte

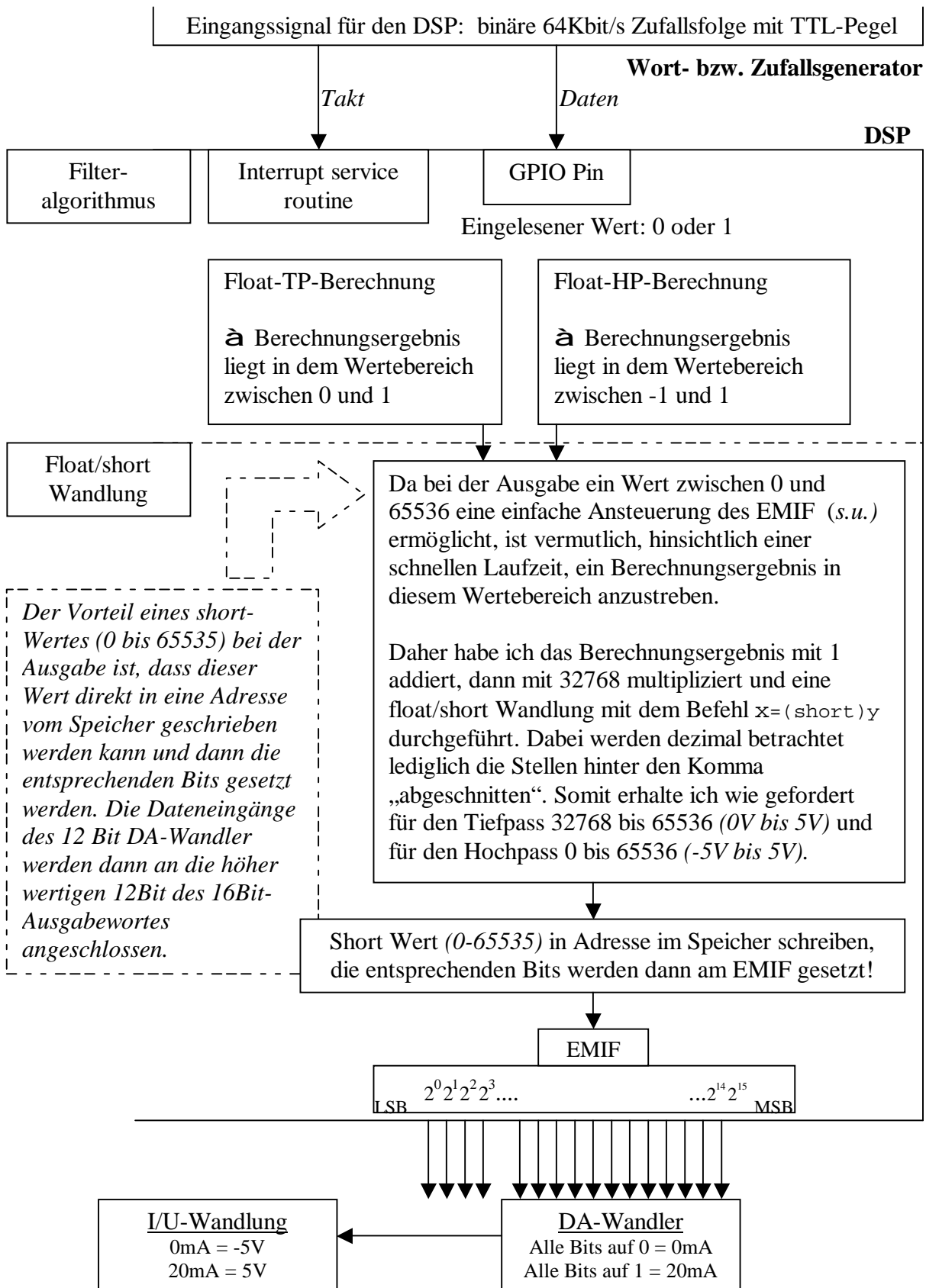


Abb. 7.1:
Verarbeitungskette für das digitale Kanalmodell

8 Zusammenfassung

In Abb. 7.1 ist das Flussdiagramm für das realisierte System dargestellt. Zu erkennen ist der vom Generator kommende Takt, der die Interrupt Service Routine (ISR) auslöst. Die ISR beinhaltet das Einlesen des am GPIO-Pin anliegenden Datensignal. Der eingelesene Wert fließt mit in die Filterberechnung ein. Der berechnete Wert wird dann in einen Short-Wert gewandelt, der eine schnelle Ausgabe über das EMIF ermöglicht.

Versuche mit einem Pseudozufallsgenerator zeigten die identische Beeinflussung des Eingangssignals durch das digitale Kanalmodell im Vergleich zum bestehenden System. Nur die relativ lange Zeit bis zur Abarbeitung der ISR lässt eine Unterscheidung der Modelle zu. Eine Koeffizientenquantisierung in Short hat einen Berechnungsgeschwindigkeitsgewinn von ca. 50% gebracht, allerdings wurde bisher nur ein Filter 1. Ordnung mit quantisierten Koeffizienten erfolgreich in Betrieb genommen.

Der SNR wird mit Hilfe folgender Formeln bestimmt ($b=12bit$),

$$SNR = b \cdot 6,02dB + 1,76dB = 12 \cdot 6,02dB + 1,76dB = 74dB .$$

Zur Steuerung der Filterordnungszahlen u.a. wurde eine Benutzeroberfläche in Visual C++ entworfen, die bis zur Abgabe dieser Arbeit leider nicht funktionsfähig war (Abb. 8.1).

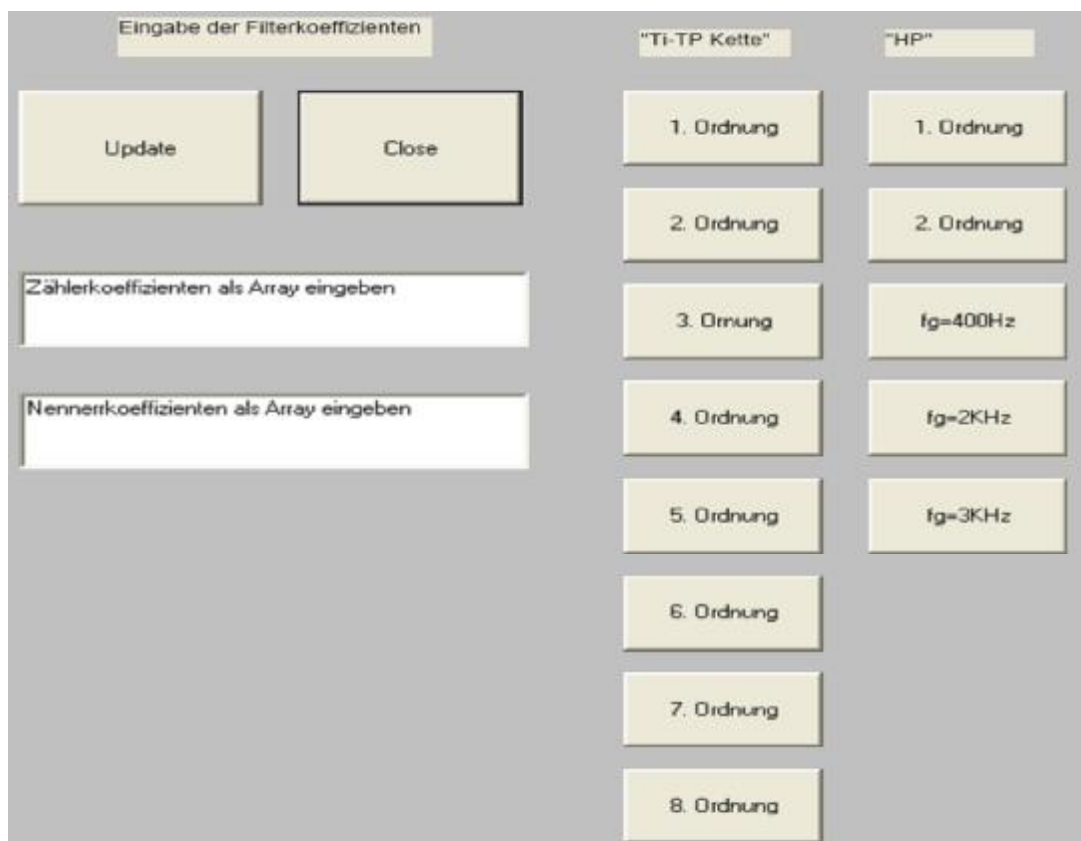


Abb. 8.1:
Benutzeroberfläche für
das digitale Kanalmodell