PENDER
ELECTRONIC DESIGN

GR-RTAX-MEZZ Board

User Manual

GAISLER RESEARCH /
PENDER ELECTRONIC DESIGN
Rev. 0.1, 2006-09-19

Pender Electronic Design GmbH
info@pender.ch
GR-RTAX-MEZZ Board User Manual

TABLE OF CONTENTS

1 INTRODUCTION.....5

1.1 Overview.....5
1.2 References.....6
1.3 Handling.....7
1.4 Abbreviations.....7

2 ELECTRICAL DESIGN.....8

2.1 Block Diagram.....8
2.2 LVDS (Spacewire) Interfaces.....10
2.2.1 Spacewire Clock.....11
2.3 Analog Interfaces 11
2.3.1 Power Interface for Analog voltages.....12
2.3.2 ADC Input.....12
2.3.3 DAC Output.....13
2.4 CAN Interface.....14
2.4.1 Configuration of Bus Termination.....14
2.4.2 Configuration of Standby Mode.....15
2.4.3 Other Configuration Aspects.....15
2.5 MIL-STD-1553 Interface.....15
2.6 Other Interfaces 16
2.6.1 GPIO.....16
2.6.2 GENIO.....16
2.6.3 Reset Button.....16

3 INTERFACES AND CONFIGURATION.....17

3.1 List of Connectors.....17

ANNEX: ACCESSING AND USING THE ADC AND DAC ON BOARD.....27

LIST OF TABLES

Table 3-1: List of Connectors....17
Table 3-2: J1 CANBUS interface connections....18
Table 3-3: J2 MIL-STD-1553 interface connections....18
Table 3-4: J3 SPW-0 interface connections....18
Table 3-5: J4 SPW-1 interface connections....19
Table 3-6: J5 SPW-2 interface connections....19
Table 3-7: J6 Analog (DAC) Output....19
Table 3-8: J7 Analog (ADC) Input....19
Table 3-9: Expansion connector P8/J8 Pin-out....20
Table 3-10: Expansion connector P9/J9 Pin-out....21
Table 3-11: Expansion connector P10/J10 Pin-out....22
Table 3-12: Expansion connector P11/J11 Pin-out....23
Table 3-13: J12 Analog Voltage Supply for ADC/DAC Circuits....23
Table 3-14: J14 CPLD programming connections....23
Table 3-15: List of Jumpers....24
Table 3-16: GPIO Header, JP8 pin-out....24
Table 3-17: GENIO Header, JP9 pin-out....24

LIST OF FIGURES

Figure 1-1: GR-RTAX-MEZZ Mezzanine Board....5
Figure 2-1: Block Diagram of GR-RTAX-MEZZ board and connections to GR-CPCI-AX2000....8
Figure 2-2: Installation with GR-CPCI-AX2000 board....9
Figure 2-3: Block Diagram of the SPW interface (one of 3 interfaces shown)....10
Figure 2-4: Circuit and Pin Connections for Spacewire/LVDS interfaces....11
Figure 2-5: Block Diagram of the Analog Control interface....12
Figure 2-6: ADC Circuitry....13
Figure 2-7: DAC Circuit....14
Figure 2-8: Block Diagram of the CAN interface....14
Figure 2-9: Transceiver and Termination Configuration....15
Figure 2-10: MIL-STD-1553 Transceiver and Transformer circuit....16
Figure 3-1: Assembly Drawing - Top View....25
Figure 3-2: Assembly Drawing - Bottom View....25
Figure 3-3: PCB Top view....26
Figure 3-4: PCB bottom view....26
Figure 3-5: VHDL listing for Glue Logic CPLD....28
Figure 3-6: C listing of example DAC and ADC control program....29

REVISION HISTORY

Revision	Date	Page	Description
0.0 DRAFT	2006-09-08	All	New document/draft
0.1	2006-09-07	All	First issue

1 INTRODUCTION

1.1 Overview

The *LEON3FT-RTAX* is an implementation of LEON3FT SPARC V8 processor using the Actel RTAX FPGA Technology. The fault tolerant design of the processor in combination with the radiation tolerant FPGA gives a total immunity to radiation effects. The LEON3FT-RTAX processor is therefore ideally suited for space and other high-rel applications.

Various standard configurations of the *LEON3FT-RTAX* are offered in order to provide an off the shelf processor component for Instrument Controller and Spacecraft controller applications.

The *GR-RTAX-MEZZ* accessory board is a mezzanine board which can be connected to the *GR-CPCI-AX2000* FPGA development board (RD-3) and which, when an appropriate LEON3FT-(RT)AX Fault-tolerant Processor (RD-5) is installed in the Actel AX2000 FGPA on the main board, provides a convenient platform for the early development and evaluation of Leon3-FT systems and interfaces.

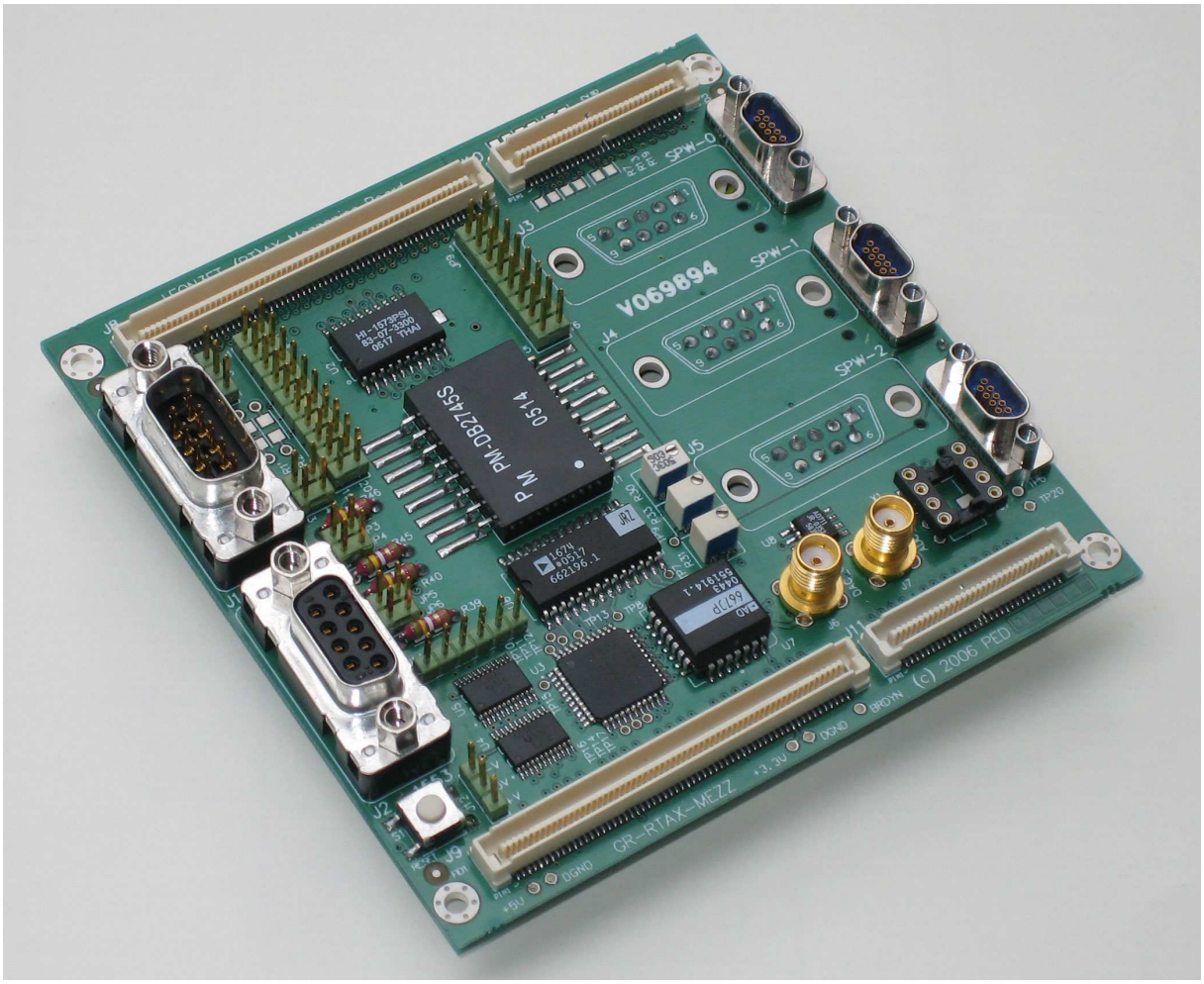

Figure 1-1: GR-RTAX-MEZZ Mezzanine Board

This interface board has been designed in order to support various configurations of LEON3FT-(RT)AX Fault-tolerant Processor.

The *GR-RTAX-MEZZ* accessory board therefore provides circuits and connectors for:

- three LVDS (Spacewire type) electrical interfaces (D9 Female or MDM9S)
- 1 channel analog output with 12 bit DAC (AD667)
- 1 channel analog input with 12 bit ADC (AD1674)
- one CAN interface (D9 Male)
- MIL-STD-1553 interface
- General Purpose I/O (0.1” Headers)

To enable convenient connection to the interfaces, the connector types and pin-outs are compatible with the standard connector types for these types of interfaces.

This board implements only the electrical driver/receiver devices with the appropriate connectors. The logic for control of the interfaces (CAN, Spacewire or ADC/DAC controller) is required to be implemented in the FPGA on the development board.

Please note that, depending on the exact configuration implemented in the FGPA design, some interfaces and features of the Mezzanine board may or may not be useable, or may have to be appropriately configured to be used.

1.2 References

RD-1 GR-RTAX-MEZZ_schematic.pdf, Schematic

RD-2 GR-RTAX-MEZZ_assy_drawing.pdf, Assembly Drawing

RD-3 GR-CPCI-AX2000 Leon Development Board Users Manual

RD-4 GRLIB IP Core User’s Manual, Gaisler Research

RD-5 LEON3-FT SPARC V8 Processor Data Sheet, Gaisler Research

RD-6 ECSS-E-50-12A Specification, Spacewire Nodes Links, Routers and Networks

1.3 Handling

ATTENTION : OBSERVE PRECAUTIONS FOR

HANDLING ELECTROSTATIC SENSITIVE DEVICES

This board contains sensitive electronic components which can be damaged by Electrostatic Discharges (ESD). When handling or installing the board observe appropriate precautions and ESD safe practices.

When not in use, store the board in an electrostatic protective container or bag.

When configuring the jumpers on the board, or connecting/disconnecting cables, ensure that the board is in an unpowered state.

1.4 Abbreviations

ADC	Analog to Digital Converter
CPLD	Complex Programmable Logic Device
DAC	Digital to Analog Converter
ESD	Electro-Static Discharge
FIFO	First-In First-Out
FPGA	Field Programmable Gate Array
FT	Fault-Tolerant
GPIO	General Purpose Input / Output
I/O	Input/Output
LVDS	Low Voltage Digital Signaling
PCB	Printed Circuit Board
SPW	Spacewire

2 ELECTRICAL DESIGN

2.1 Block Diagram

The *GR-RTAX-MEZZ* provide the electrical functions and interfaces as represented in the block diagram, Figure 2-1.

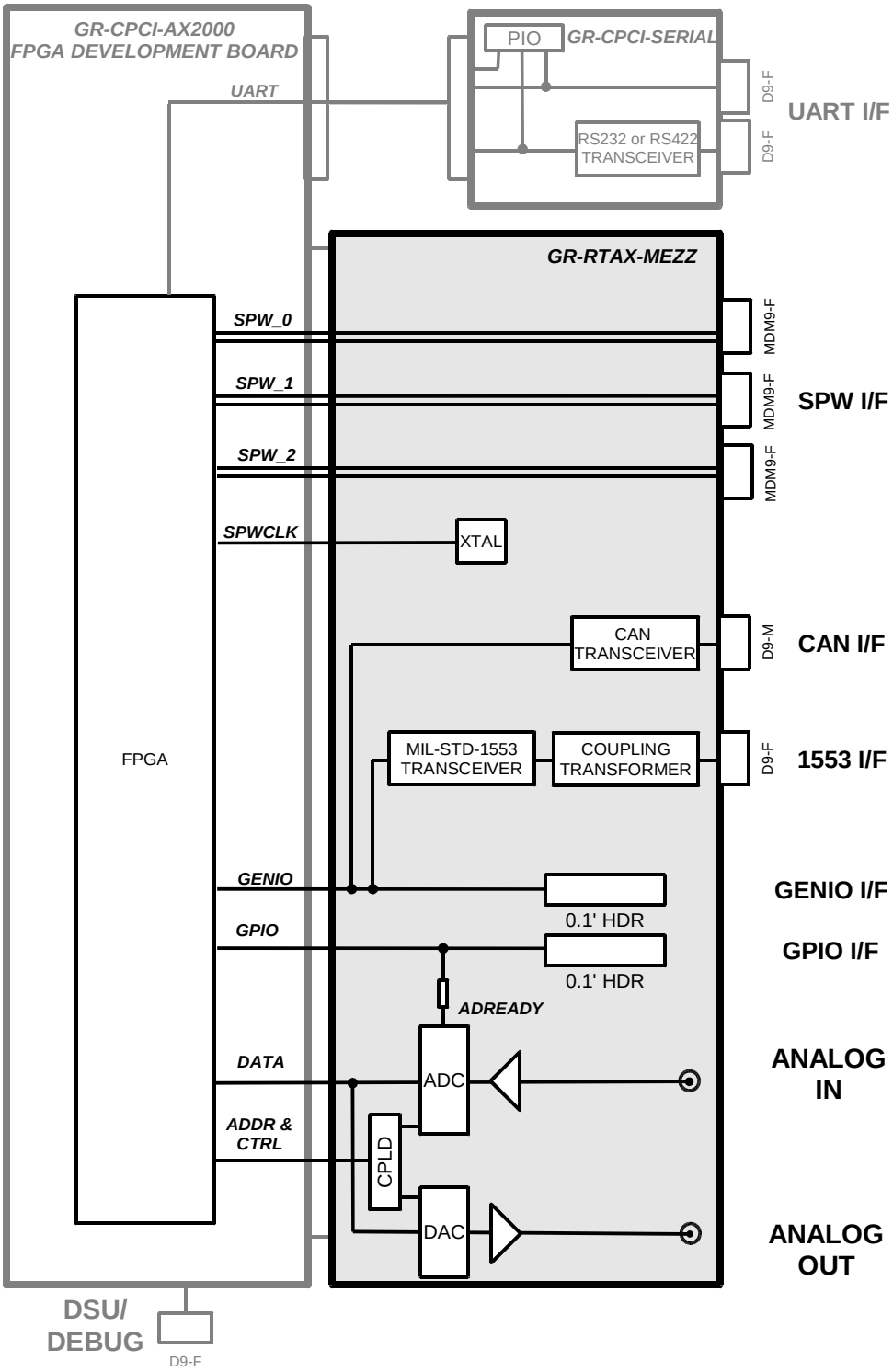

Figure 2-1: Block Diagram of *GR-RTAX-MEZZ* board and connections to *GR-CPCI-AX2000*

The board is intended to be mounted as a mezzanine board connected to the J8/J9/J10/J11 Expansion connectors of the *GR-CPCI* board.

The board, so mounted, can be used 'stand-alone' on the bench top, or can be mounted in slot of a 3U high Compact PCI rack.

Note that, when using this mezzanine board, it is still possible to mount to connector J4 of the *GR-CPCI-AX2000* board a *GR-CPCI-RS232* or *GR-CPCI-RS422* accessory board to allow the connection of two serial (RS232 or RS422) interfaces to the system (as shown in Figure 2-1).

Figure 2-2 shows the *GR-RTAX-MEZZ* board mounted with the *GR-CPCI-AX2000* development board.

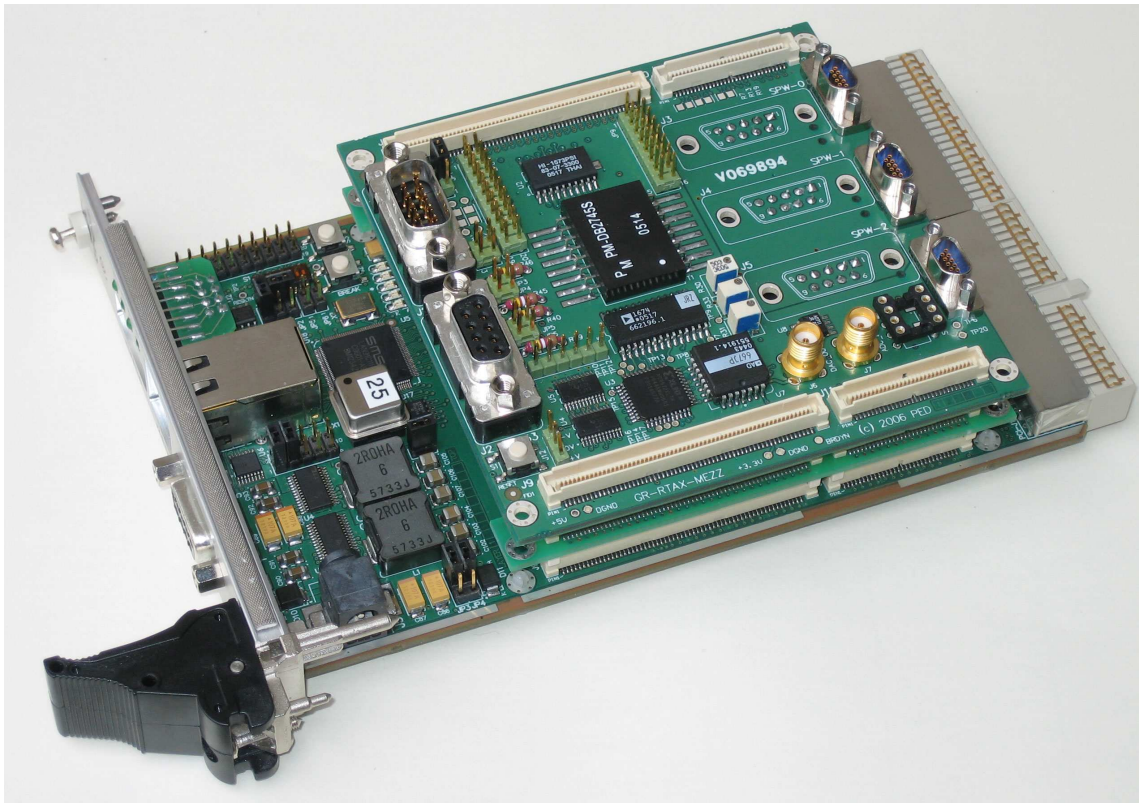

Figure 2-2: Installation with GR-CPCI-AX2000 board

2.2 LVDS (Spacewire) Interfaces

Three Spacewire type LVDS interfaces are implemented. No LVDS interface transceiver circuits are provided by the *GR-RTAX-MEZZ* board and therefore it is necessary to implement the Spacewire logic inside the FPGA on the *GR-CPCI-AX2000* Development Board and to configure the appropriate outputs/inputs of the FPGA to be LVDS types.

To implement LVDS compatible inputs and outputs on the Actel (RT)AX FGPA's devices resistors are required to be implemented on the signals line external to the FPGA (For more information on LVDS, please refer to the Actel datasheets and tech notes for the (RT)AX2000.

100 Ohm Termination resistors for the LVDS receiver signals are mounted on the mezzanine board.

140 Ohm parallel termination resistor are mounted on the Mezzanine board, which together with 160 Ohm serial resistors on the FPGA main board provide the require resistor networks for the LVDS transmitter signals of the Spacewire interfaces.

The pin out and connector types for these Spacewire interfaces conforms to the Spacewire standard RD-5, as shown in Figure 2-4. The layout of the Mezzanine PCB allows either D9-Female connectors or micro-miniature MDM9S connectors to be installed on the board for the Spacewire interfaces as required.

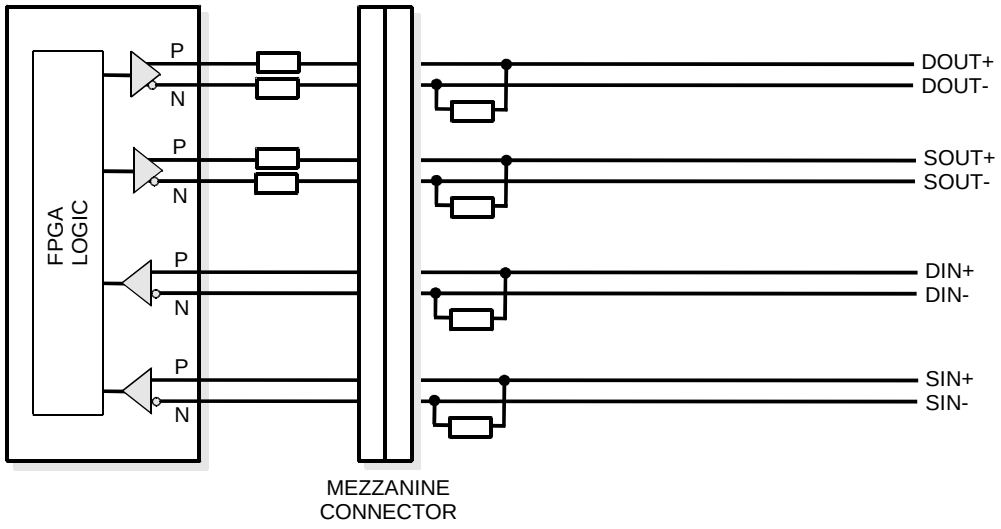

Figure 2-3: Block Diagram of the SPW interface (one of 3 interfaces shown)

During the design of the PCB care has been taken to ensure that has SIN/DIN and SOUT/DOUT pairs have matched lengths, to ensure equal propagation time for signals. This equalisation has been done, taking account not only the length of the traces on the Mezzanine, but also the length of the trace on the *GR-CPCI-AX2000* board (i.e. the total length of the signal traces).

The PCB traces on the *GR-RTAX-MEZZ* board have been laid out with 100-Ohm differential impedance design rules.

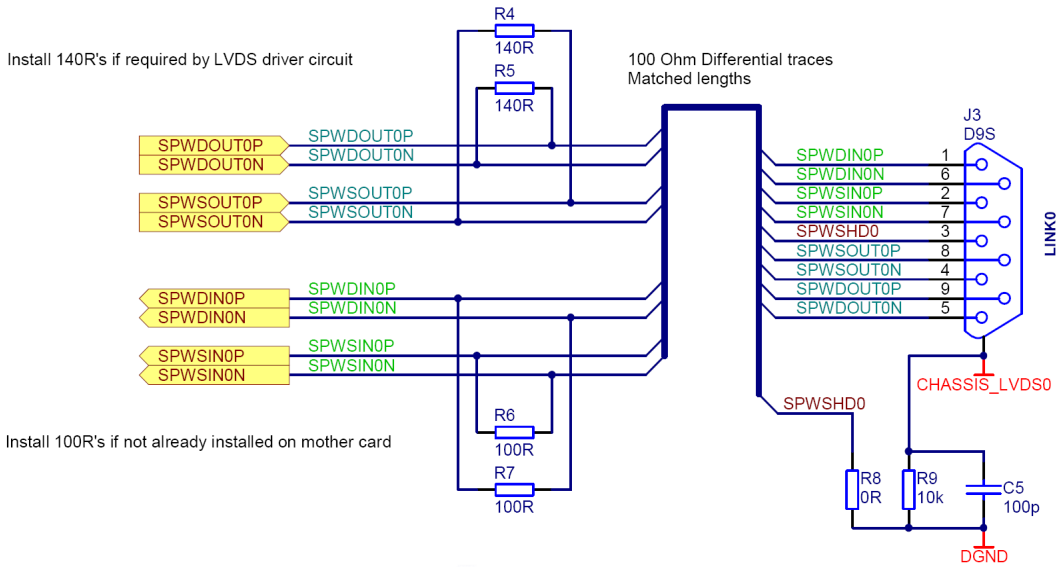

Figure 2-4: Circuit and Pin Connections for Spacewire/LVDS interfaces

(one of three interfaces shown)

The inner shield pin (pin3 of the connector) is connected to DGND via a Zero-ohm resistor.

The outer shield (connector chassis) is connected to the board DGND with a 10k resistor in parallel with a 100pF capacitor. If a 'hard' connection to DGND is desired, the resistor can be replaced with a Zero-ohm resistor.

2.2.1 Spacewire Clock

Depending on the clock configuration required on the main FGPA board, it may be desirable or necessary to use an additional separate SPW clock frequency.

This can be provided by installing a user defined Crystal Oscillator in the 8 pin DIL socket (X1) and providing this as an input to the FGPA on the Main board.

By means of installing/removing resistors, this SPW clock input frequency can be injected either through the JP7 (PCI_CLK) input or via I/O connection *PCIIO31* (please refer to the schematic for the details of the configuration options).

2.3 Analog Interfaces

The *GR-RTAX-MEZZ* board includes an ADC and DAC circuit on board, intended to be used to demonstrate the operation of ADC/DAC.

These devices are not controlled by specific logic in the FPGA/Leon, but are instead accessed by the Leon processor as memory mapped peripherals in the I/O space of the memory map.

This address decoding of the I/O Memory space is performed by a CPLD on the board to generate the necessary control and chip select signals for the ADC and DAC devices.

Connection to the ADC input and DAC output is by means of SMA connectors mounted on

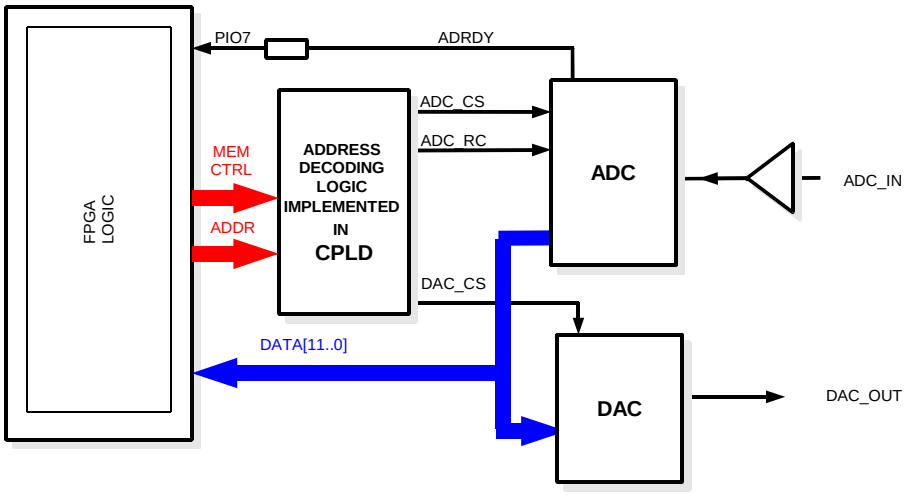

Figure 2-5: Block Diagram of the Analog Control interface

the mezzanine board.

2.3.1 Power Interface for Analog voltages.

The analog circuits of the *GR-RTAX-MEZZ* board require a +/-12V (nominal) power supply in addition to the +5V and +3.3V power supplies normally provided to the logic via the mezzanine connectors.

Most conveniently, if the *GR-PCI-XC2V* board is installed in a Compact PCI rack, this voltage supply will be provided by the Compact PCI backplane, through the mezzanine connector J9, to the *GR-RTAX-MEZZ* mezzanine board.

However, if the board is used in a 'stand-alone' configuration, the user must provide an external +/-12V power supply, connected to the jumper connection J12.

2.3.2 ADC Input

The ADC circuitry comprises a 12 bit ADC (AD1674) with a high speed precision op-amp (AD711). For detailed operation of the ADC please refer to its datasheet.

The default configuration of the ADC is set-up to measure a bipolar -5V to +5V input voltage range. However, the circuit on the board can also be configured to provide a uni-polar 0 to +10V or input voltage range, by reconfiguration of a number of the resistors on-board.

Note that, since the logic part of the ADC device operates from a +5V digital supply, the output data must be buffered with a level shifter circuit to ensure compatibility with the 3.3V LVTTL inputs of the FPGA.

The conversion speed of the AD674A device is typically in the order of 15 us. Completion of the conversion is indicated by the AD_RDY signal output of the ADC.

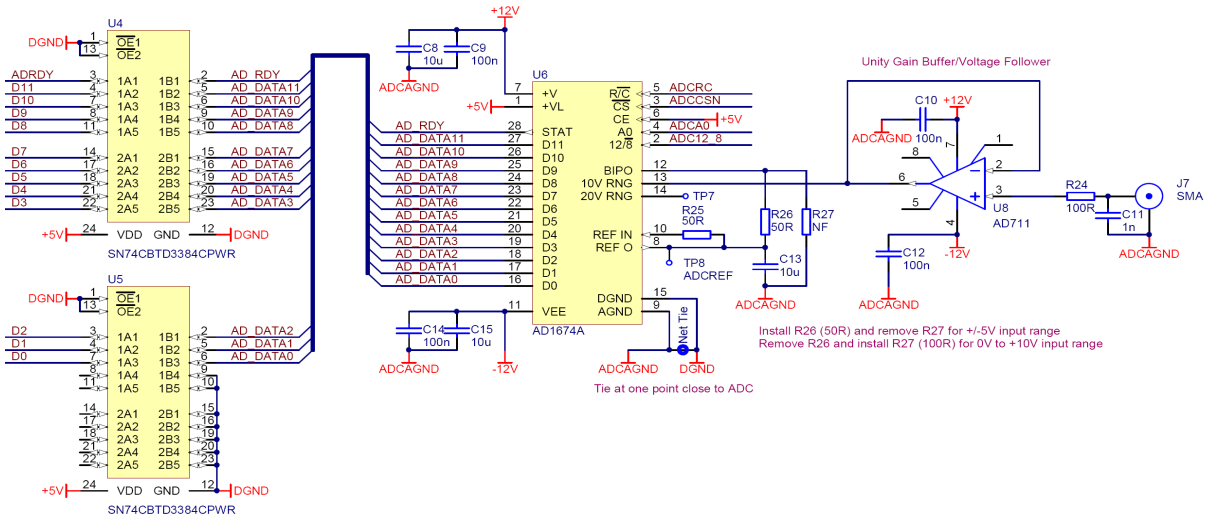

Figure 2-6: ADC Circuitry

The AD_RDY signal is connected on the board to the PIO7 input of the FGPA. In this manner the state of the signal can be easily read by the processor, by reading the state of the GPIO port. Alternatively if the GPIO port can be configured so that a change of state of the AD_RDY can generate a processor interrupt.

2.3.3 DAC Output

The DAC circuitry comprises a single channel 12 bit DAC (AD667) .

The default configuration of the DAC is set-up to provide a bipolar -5V to +5V output voltage range. However, the circuit on the board can also be configured to provide a uni-polar 0 to +10V or 0 to +5V output voltage range, by reconfiguration of a number of the resistors on-board. Additionally Offset and Gain of the DAC circuit can be adjusted by means of the on-board potentiometers.

For detailed operation of the DAC please refer the schematic and the datasheet of the DAC device.

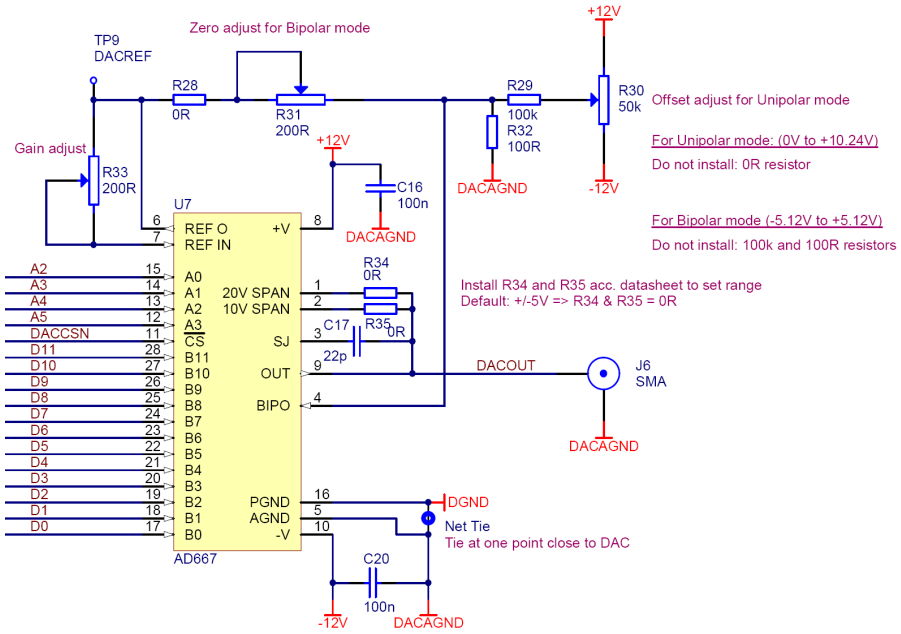

Figure 2-7: DAC Circuit

2.4 CAN Interface

The mezzanine board provides the electrical interfaces for a CAN bus interface, as represented in the block diagram, Figure 2-8.

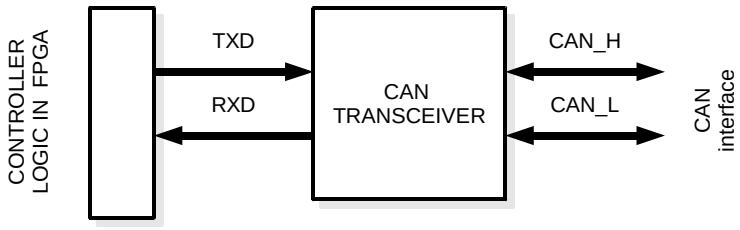

Figure 2-8: Block Diagram of the CAN interface

The CAN bus transceiver IC on this board is a *SN65HVD230* device from Texas Instruments which operates from a single +3.3V power supply.

2.4.1 Configuration of Bus Termination

The board can be configured for either end node or stub-node operation by means of the jumper JP2, as shown in Figure 2-9.

For normal end-node termination with a nominal 120 Ohm insert the jumper in position 1-3.

However, if a split termination is desired (if required for improved EMC performance), insert the jumpers in positions 1-2 and 3-4.

For stub nodes, if termination is not required, do not install any jumpers.

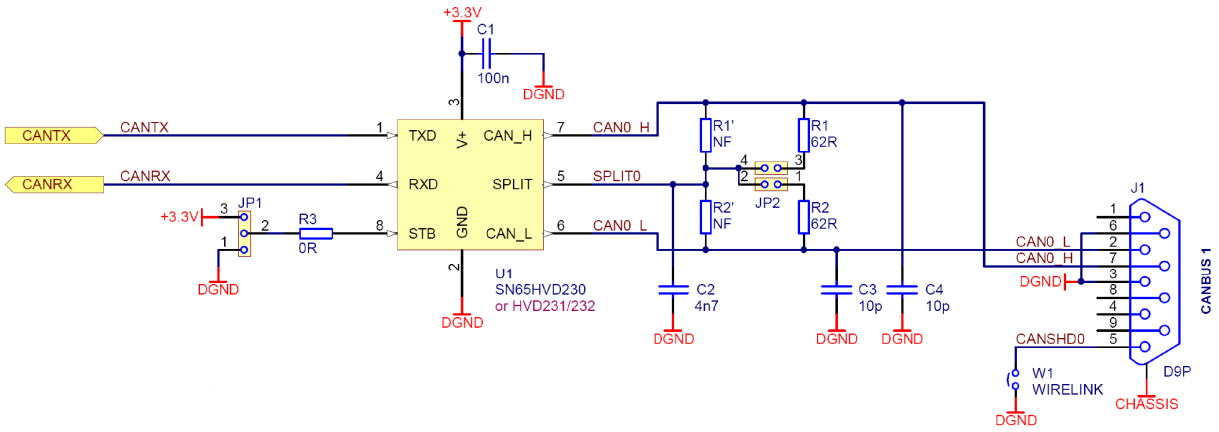

Figure 2-9: Transceiver and Termination Configuration

For stub nodes, if some form of custom termination is required, pads are provided on the board to enable user supplied resistors R1' – R2' to be installed.

2.4.2 Configuration of Standby Mode

The SN65HVD230 transceiver device used on the board has the facility to set the device into *STANDBY or RUN* mode, by connecting an external signal to pin 8 of the device (refer to Figure 2-9)

Insert the jumper 1-2 (default position) to enable the transceiver device.

Insert the jumper 2-3 to force the transceiver into Standby mode (for example if the CAN interface is not to be used).

A further feature provided by the SN65HVD230 device is the capability to adjust the transceiver slew rate. This can be done by modifying the values of resistor R3.

The default value of 0 ohms is compatible with 1Mbps operation.

From the data sheet the following resistor values give the following slew rates:

10kOhm => 15V/us

100kOhm => 2V/us

2.4.3 Other Configuration Aspects

To use the CAN interface, it is necessary to compile a CAN controller in your FGPA configuration. Furthermore, ensure that the CAN bus output signals are assigned in the Actel pin definition file (.pdc) to the appropriate I/O pin of the FPGA as listed in Table 3-9.

If the shield from the CAN bus cable is required to be grounded, install zero-ohm (0805 size) resistors for the jumper W1.

2.5 MIL-STD-1553 Interface

The board implements a Dual MIL-STD-1553 interface with a 3.3V Transceiver and Transformer circuits as shown in Figure 2-10.

Since there are various 'standard' connectors defined for the connection to MIL-STD-1553

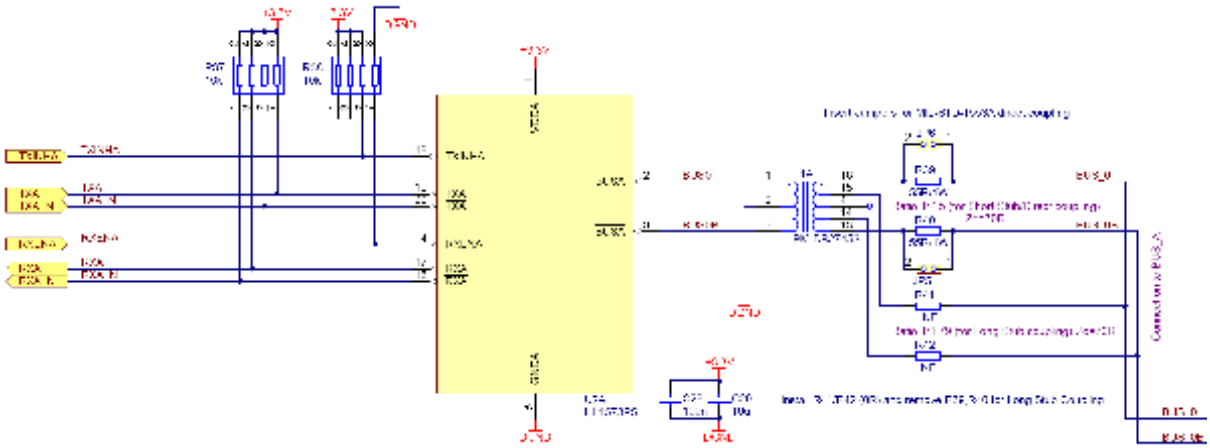

Figure 2-10: MIL-STD-1553 Transceiver and Transformer circuit
(one of two interfaces shown)

bus, and because of limited PCB area it has instead been decided to implement a D9-Female connected for the connector on the board as this can be most easily adapted to suit the bus connector configuration of the user.

2.6 Other Interfaces

2.6.1 GPIO

16 signals from the FPGA are connected to provide General Purpose Input/Output signals (3.3V LVTTL voltage levels), on a 0.1” standard header connector (JP8).

2.6.2 GENIO

16 signals from the FPGA are connected to provide Input/Output signals (3.3V LVTTL voltage levels) on a 0.1” standard header connector (JP9). This connector gives the user the opportunity to monitor the control/data signals of the CAN and 1553 interfaces with a logic analyser.

These pins could also be used for user defined general purpose I/O signals. However, if the board is equipped with the CAN-bus and 1553 interfaces please note that the following signals are connected by default as inputs from the transceiver circuits on the board:

GENIO25 / CANRX
GENIO30 / RXA
GENIO31 / RXA_N
GENIO36 / RXB
GENIO37 / RXB_N

If it is desired to utilise these signals as general purpose I/O signals, the CAN /1553 transceiver circuits must be removed, or otherwise disabled to prevent signal contention.

2.6.3 Reset Button

As the mezzanine board partially obscures the Reset button mounted on the *GR-CPCI-AX2000* board, a push-button switch connected to the RESETN signal is provided on the mezzanine board.

3 INTERFACES AND CONFIGURATION

3.1 List of Connectors

Name	Function	Type	Description
J1	CANBUS	D9-P (male)	Connections for CANBUS interface
J2	MIL1553	D9-S (female)	Connections for MIL-STD-1553 interface
J3	SPW-0	MDM9-S (female)	LVDS connections for Spacewire Interface-0
J4	SPW-1	MDM9-S (female)	LVDS connections for Spacewire Interface-1
J5	SPW-2	MDM9-S (female)	LVDS connections for Spacewire Interface-2
J6	ADC_IN	SMA-JACK	Analog ADC input
J7	DAC_OUT	SMA-JACK	Analog DAC Out
J8	Mezzanine I/O	AMP 177984-5	Expansion connector signals fed-thru (120 pin, 0.8mm pitch PCB connector)
J9	Mezzanine I/O	AMP 177984-5	Expansion connector signals fed-thru (120 pin, 0.8mm pitch PCB connector)
J10	Mezzanine I/O	AMP 177984-2	Expansion connector signals fed-thru (60 pin, 0.8mm pitch PCB connector)
J11	Mezzanine I/O	AMP 177984-2	Expansion connector signals fed-thru (60 pin, 0.8mm pitch PCB connector)
J12	V_ANALOG	0.1” Header 1x3 pin	Analog Voltage supply for ADC/DAC circuits
J13	not assigned	---	---
J14	CPLD Program	0.1” Header 1x6 pin	JTAG programming connection for CPLD
P8	Mezzanine I/O	AMP 5-179009-5	Expansion connector signals from FPGA (120 pin, 0.8mm pitch PCB connector)
P9	Mezzanine I/O	AMP 5-179009-5	Expansion connector signals from FPGA (120 pin, 0.8mm pitch PCB connector)
P10	Mezzanine I/O	AMP 5-179009-2	Expansion connector signals from FPGA (60 pin, 0.8mm pitch PCB connector)
P11	Mezzanine I/O	AMP 5-179009-2	Expansion connector signals from FPGA (60 pin, 0.8mm pitch PCB connector)

Table 3-1: List of Connectors

Pin	Name	Comment
1		No connect
6	DGND	Ground
2	CAN0_L	CAN Dominant Low
7	CAN0_H	CAN Dominant High
3	DGND	Ground
8		No connect
4		No connect
9		No connect
5	CANSHD0	Shield

Table 3-2: J1 CANBUS interface connections

Pin	Name	Comment
1	BUS_0	BUS_0 positive
6	DGND	Ground
2	BUS_0B	BUS_0 negative
7		No connect
3		No connect
8		No connect
4	BUS_1	BUS_1 positive
9	DGND	Ground
5	BUS_1B	BUS_1 negative

Table 3-3: J2 MIL-STD-1553 interface connections

Pin	Name	Comment
1	DIN0+	Data In +ve
6	DIN0-	Data In -ve
2	SIN0+	Strobe In +ve
7	SIN0-	Strobe In -ve
3	SHIELD	Inner Shield
8	SOUT0+	Strobe Out +ve
4	SOUT0-	Strobe Out -ve
9	DOUT0+	Data Out +ve
5	DOUT0-	Data Out -ve

Table 3-4: J3 SPW-0 interface connections

Pin	Name	Comment
1	DIN1+	Data In +ve
6	DIN1-	Data In -ve
2	SIN1+	Strobe In +ve
7	SIN1-	Strobe In -ve
3	SHIELD	Inner Shield
8	SOUT1+	Strobe Out +ve
4	SOUT1-	Strobe Out -ve
9	DOUT1+	Data Out +ve
5	DOUT1-	Data Out -ve

Table 3-5: J4 SPW-1 interface connections

Pin	Name	Comment
1	DIN2+	Data In +ve
6	DIN2-	Data In -ve
2	SIN2+	Strobe In +ve
7	SIN2-	Strobe In -ve
3	SHIELD	Inner Shield
8	SOUT2+	Strobe Out +ve
4	SOUT2-	Strobe Out -ve
9	DOUT2+	Data Out +ve
5	DOUT2-	Data Out -ve

Table 3-6: J5 SPW-2 interface connections

Pin	Name	Comment
1	DAC_OUT	Analog (DAC) Output
outer	SHD	DGND

Table 3-7: J6 Analog (DAC) Output

Pin	Name	Comment
1	ADC_IN	Analog (DAC) Input
outer	SHD	DGND

Table 3-8: J7 Analog (ADC) Input

FUNCTION	FPGA PIN CG624	FPGA PIN FG896	CONNECTOR PIN		FPGA PIN CG624	FPGA PIN FG896	FUNCTION
DGND			1	120			DGND
PIO14 / GPIO14	F20	B27	2	119	F19	A27	PIO15 / GPIO15
PIO12 / GPIO12	G21	G23	3	118	G20	G22	PIO13 / GPIO13
PIO10 / GPIO10	E18	D25	4	117	F18	C25	PIO11 / GPIO11
PIO8 / GPIO8	H20	F23	5	116	H19	E23	PIO9 / GPIO9
+3.3V			6	115			+3.3V
DGND			7	114			DGND
PIO6 / GPIO6	G22	D29	8	113	M17	E29	PIO7 / GPIO7
PIO4 / GPIO4	H22	H25	9	112	J22	H26	PIO5 / GPIO5
PIO2 / GPIO2	K18	F28	10	111	L18	G28	PIO3 / GPIO3
PIO0 / GPIO0	F24	F30	11	110	G24	G30	PIO1 / GPIO1
+3.3V			12	109			+3.3V
DGND			13	108			DGND
RXD1	J23	H28	14	107	L19	L23	TXD1
CTSN1	J21	K25	15	106	J20	K24	RTSN1
RXD2	D25	J27	16	105	K20	K26	TXD2
CTSN2	K19	L24	17	104	E25	K27	RTSN2
+3.3V			18	103			+3.3V
DGND			19	102			DGND
GENIO54 / not used	M18	L25	20	101	M19	M25	GENIO55 / not used
GENIO52 / not used	H24	K28	21	100	J24	K30	GENIO53 / not used
GENIO50 / not used	N16	M24	22	99	L23	M23	GENIO51 / not used
GENIO48 / not used	K22	L27	23	98	L22	M27	GENIO49 / not used
+3.3V			24	97			+3.3V
DGND			25	96			DGND
GENIO46 / not used	F25	L28	26	95	G25	M28	GENIO47 / not used
GENIO44 / not used	L20	N23	27	94	L21	N22	GENIO45 / not used
GENIO42 / not used	K24	L29	28	93	L24	M29	GENIO43 / not used
GENIO40 / not used	M20	N28	29	92	N17	N26	GENIO41 / not used
+3.3V			30	91			+3.3V
DGND			31	90			DGND
GENIO38	N19	N25	32	89	M21	N27	GENIO39
GENIO36 / RXB	J25	N29	33	88	N18	N24	GENIO37 / RXB_N
GENIO34 / TXB_N	M24	P24	34	87	N24	P25	GENIO35 / RXENB
GENIO32 / TXINHB	K25	P27	35	86	L25	P28	GENIO33 / TXB
+3.3V			36	85			+3.3V
DGND			37	84			DGND
GENIO30 / RXA	M22	P26	38	83	N22	R26	GENIO31/ RXA_N
GENIO28/ TXA_N	M23	P30	39	82	N23	P29	GENIO29/ RXENA
GENIO26 / TXINHA	P17	R25	40	81	P18	R24	GENIO27/ TXA
GENIO24 / CANTX	M25	R30	41	80	N25	R29	GENIO25 / CANRX
+3.3V			42	79			+3.3V
DGND			43	78			DGND
no connect			44	77			no connect
no connect			45	76			no connect
no connect			46	75			no connect
no connect			47	74			no connect
+3.3V			48	73			+3.3V
DGND			49	72			DGND
no connect			50	71			no connect
no connect			51	70			no connect
no connect			52	69			no connect
no connect			53	68			no connect
+3.3V			54	67			+3.3V
DGND			55	66			DGND
no connect			56	65			no connect
no connect			57	64			no connect
no connect			58	63			no connect
no connect			59	62			no connect
DGND			60	61			DGND

Table 3-9: Expansion connector P8/J8 Pin-out

FUNCTION	FPGA PIN CG624	FPGA PIN FG896	CONNECTOR PIN		FPGA PIN CG624	FPGA PIN FG896	FUNCTION
DGND			1	120			DGND
+5V			2	119			+5V
DGND			3	118			DGND
-12V			4	117			-12V
DGND			5	116			DGND
+12V			6	115			+12V
DGND			7	114			DGND
D15	L8	H6	8	113	W1	W3	D31
D7	D2	E2	9	112	R4	W6	D23
+3.3V			10	111			+3.3V
DGND			11	110			DGND
D14	K8	G6	12	109	T8	V6	D30
D6	H6	F1	13	108	V2	AA2	D22
D13	G3	G5	14	107	R8	V7	D29
D5	H5	F2	15	106	U7	AA5	D21
D12	G4	F5	16	105	AB1	Y2	D28
D4	J4	H4	17	104	T7	Y5	D20
D11	E3	F3	18	103	AA1	Y1	D27
D3	H4	G4	19	102	T6	Y8	D19
+3.3V			20	101			+3.3V
DGND			21	100			DGND
D10	F3	F4	22	99	R5	V8	D26
D2	F2	D2	23	98	R6	W8	D18
D9	J5	J7	24	97	R3	Y4	D25
D1	E2	D1	25	96	Y2	AE2	D17
D8	J6	H7	26	95	T4	W7	D24
D0	H2	K6	27	94	W2	AB1	D16
A26	F1	J4	28	93	G1	J3	A27
A24	M9	K5	29	92	M8	L5	A25
+3.3V			30	91			+3.3V
DGND			31	90			DGND
A22	L7	J1	32	89	M7	K1	A23
A20	J2	K4	33	88	J1	L4	A21
A18	L4	N9	34	87	J3	L6	A19
A16	L2	M3	35	86	K4	N8	A17
A14	L5	N6	36	85	K2	L3	A15
A12	M5	M4	37	84	L6	N7	A13
A10	K1	L1	38	83	L1	M1	A11
A8	N10	P7	39	82	N9	P6	A9
+3.3V			40	81			+3.3V
DGND			41	80			DGND
A6	L3	P4	42	79	M3	P5	A7
A4	N7	R6	43	78	M4	P9	A5
A2	M6	R4	44	77	N8	R7	A3
A0	P9	R8	45	76	N6	R9	A1
WRITEN	P7	T5	46	75	R7	U5	READ
OEN	N1	V2	47	74	M1	U2	IOSN
ROMSN0	N2	T2	48	73	R1	V3	ROMSN1
RAMSN4	P5	U7	49	72	R2	U9	RAMOEN4
+3.3V			50	71			+3.3V
DGND			51	70			DGND
RAMSN3	P6	U4	52	69	T1	W1	RAMOEN3
RAMSN2	P1	U3	53	68	U1	W2	RAMOEN2
RAMSN1	P4	T6	54	67	N4	T7	RAMOEN1
RAMSN0	M2	R2	55	66	P3	U1	RAMOEN0
RWEN2	E1	H2	56	65	D1	G2	RWEN3
RWEN0	H3	J6	57	64	G2	K7	RWEN1
BRDYN	N3	T1	58	63	P8	T8	BEXCN
RESETN	D24	H24	59	62	C12	D15	CLK
DGND			60	61			DGND

Table 3-10: Expansion connector P9/J9 Pin-out

FUNCTION	FPGA PIN CG624	FPGA PIN FG896	CONNECTOR PIN		FPGA PIN CG624	FPGA PIN FG896	FUNCTION
DGND			1	60			DGND
GENIO92 / SPWDIN0P	P19	T24	2	59	P23	U27	GENIO93 / SPWSIN0P
GENIO94 / SPWDIN0N	P20	U24	3	58	R23	V27	GENIO95 / SPWSIN0N
GENIO96 / SPWSOUT0N	R22	V25	4	57	R25	W29	GENIO97 / SPWDOUT0N
GENIO98 / SPWSOUT0P	P22	U25	5	56	P25	V29	GENIO99 / SPWDOUT0P
GENIO100 / SPWDIN1P	R18	V24	6	55	R24	W28	GENIO101 / SPWSIN1P
GENIO102 / SPWDIN1N	T18	W24	7	54	T24	W27	GENIO103 / SPWSIN1N
GENIO104 / SPWSOUT1N	T20	Y28	8	53	U25	Y30	GENIO105 / SPWDOUT1N
GENIO106 / SPWSOUT1P	R20	Y27	9	52	T25	W30	GENIO107 / SPWDOUT1P
DGND			10	51			DGND
+3.3V			11	50			+3.3V
GENIO108 / SPWDIN2P	T19	W23	12	49	W25	AA30	GENIO109 / SPWSIN2P
GENIO110 / SPWDIN2N	U19	Y23	13	48	Y25	AB30	GENIO111 / SPWSIN2N
GENIO112 / SPWSOUT2N	V20	AA24	14	47	AB25	AF29	GENIO113 / SPWDOUT0N
GENIO114 / SPWSOUT2P	U20	Y24	15	46	AA25	AF30	GENIO115 / SPWDOUT2P
PCIiO0 / not used	U23	AD29	16	45	Y24	AC27	PCIiO1 / not used
PCIiO2 / not used	U24	AE29	17	44	AA24	AD27	PCIiO3 / not used
PCIiO4 / not used	V22	AC26	18	43	V23	AE28	PCIiO5 / not used
PCIiO6 / not used	U22	AB26	19	42	V24	AD28	PCIiO7 / not used
DGND			20	41			DGND
+3.3V			21	40			+3.3V
PCIiO8 / not used	V21	AE26	22	39	U21	AD26	PCIiO9 / not used
PCIiO10 / not used	Y23	AD25	23	38	AA23	AC25	PCIiO11 / not used
PCIiO12 / not used	W22	AF27	24	37	W23	AE27	PCIiO13 / not used
PCIiO14 / not used	Y22	AB23	25	36	Y21	AA23	PCIiO15 / not used
PCIiO16 / not used	N20	T27	26	35	P24	T30	PCIiO17 / not used
PCIiO18 / not used	P21	T22	27	34	R19	V23	PCIiO19 / not used
PCIiO20 / not used	R21	V26	28	33	T22	Y25	PCIiO21 / not used
PCIiO22 / not used	W24	AD30	29	32	AB24	AG28	PCIiO23 / not used
DGND			30	31			DGND

Table 3-11: Expansion connector P10/J10 Pin-out

FUNCTION	FPGA PIN CG624	FPGA PIN FG896	CONNECTOR PIN		FPGA PIN CG624	FPGA PIN FG896	FUNCTION
DGND			1	60			DGND
GENIO56 / CB6	T2	Y6	2	59	U2	AA6	GENIO57 / CB7
GENIO58 / CB4	U5	AC3	3	58	U3	AD2	GENIO59 / CB5
GENIO60 / CB2	W5	AE3	4	57	U6	AB3	GENIO61 / CB3
GENIO62 / CB0	V4	AD4	5	56	Y5	AD3	GENIO63 / CB1
GENIO64 / RAMBEN2	K7	L8	6	55	K6	L7	GENIO65 / RAMBEN3
GENIO66 / RAMBEN0	Y1	W4	7	54	P2	U8	GENIO67 / RAMBEN1
no connect			8	53			no connect
no connect			9	52			no connect
DGND			10	51			DGND
+3.3V			11	50			+3.3V
no connect			12	49			no connect
no connect			13	48			no connect
no connect			14	47			no connect
no connect			15	46			no connect
no connect			16	45			no connect
no connect			17	44			no connect
no connect			18	43			no connect
no connect			19	42			no connect
DGND			20	41			DGND
+3.3V			21	40			+3.3V
no connect			22	39			no connect
no connect			23	38			no connect
PCIIO24 / not used	AA3	AF4	24	37	V3	AB7	PCIIO25 / not used
PCIIO26 / not used	W3	AC7	25	36	AA2	AD5	PCIIO27/ not used
PCIIO28/ not used	AB2	AE5	26	35	V6	AF1	PCIIO29 / not used
PCIIO30 / not used	W4	AF2	27	34	U4	AC4	PCIIO31 / not used
no connect			28	33			no connect
no connect			29	32			no connect
DGND			30	31			DGND

Table 3-12: Expansion connector P11/J11 Pin-out

Pin	Name	Comment
1	+V	(Nominal +12V)
2	GND	Ground
3	-V	(Nominal -12V)

Table 3-13: J12 Analog Voltage Supply for ADC/DAC Circuits

Pin	Name	Comment
1	+3.3V	V+
2	DGND	Ground
3	JTAG_TCK	JTAG Test Clock
4	JTAG_TDO	JTAG Test Data Out
5	JTAG_TDI	JTAG Test Data In
6	JTAG_TMS	JTAG Test Mode Select

Table 3-14: J14 CPLD programming connections

Name	Function	Type	Description
JP1	CAN_EN	0.1” Header 1x3 pin	Jumper to Enable CAN Transceiver 0
JP2	CAN_TERM	0.1” Header 2x2 pin	Jumper to Configure CAN termination
JP3	BUS1B-DIRECT	0.1” Header 1x2 pin	1553 direct coupling jumper
JP4	BUS1-DIRECT	0.1” Header 1x2 pin	1553 direct coupling jumper
JP5	BUS0B-DIRECT	0.1” Header 1x2 pin	1553 direct coupling jumper
JP6	BUS0-DIRECT	0.1” Header 1x2 pin	1553 direct coupling jumper
JP7	SPW_CLK	0.1” Socket 1x2 pin	Socket connects to PCI_CLK Header on CPCI board
JP8	PIO	0.1” Header 2x10 pin	Header for (G)PIO interface expansion
JP9	GENIO	0.1” Header 2x8 pin	Header for GENIO[39..24] signals

Table 3-15: List of Jumpers

FUNCTION	CONNECTOR PIN		FUNCTION
GPIO0	1	2	GPIO1
GPIO2	3	4	GPIO3
GPIO4	5	6	GPIO5
GPIO6	7	8	GPIO7
GPIO8	9	10	GPIO9
GPIO10	11	12	GPIO11
GPIO12	13	14	GPIO13
GPIO14	15	16	GPIO15
+3.3V	17	18	+3.3V
DGND	19	20	DGND

Table 3-16: GPIO Header, JP8 pin-out

FUNCTION	CONNECTOR PIN		FUNCTION
GPIO24 / CANTX	1	2	GPIO25 / CANRX*
GPIO26 / 1553_TXINHA	3	4	GPIO27 / 1553_TXA
GPIO28 / 1553_TXA_N	5	6	GPIO29 / 1553_RXENA
GPIO30 / 1553_RXA*	7	8	GPIO31 / 1553_RXA_N*
GPIO32 / 1553_TXINHB	9	10	GPIO33 / 1553_TXB
GPIO34/ 1553_TXB_N	11	12	GPIO35 / 1553_RXENB
GPIO36/ 1553_RXB*	13	14	GPIO37 / 1553_RXA_N*
GPIO38	15	16	GPIO39

Table 3-17: GENIO Header, JP9 pin-out

* If board is equipped with 1553 and CAN bus circuits some I/O pins cannot be used as general purpose I/O pins - see section 2.6.2

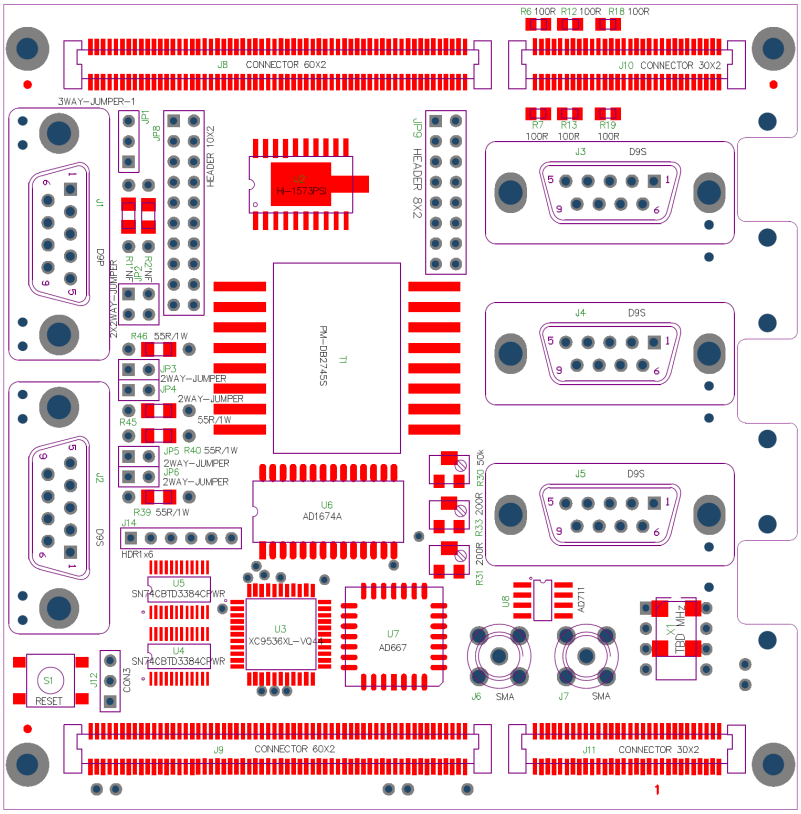

Figure 3-1: Assembly Drawing - Top View

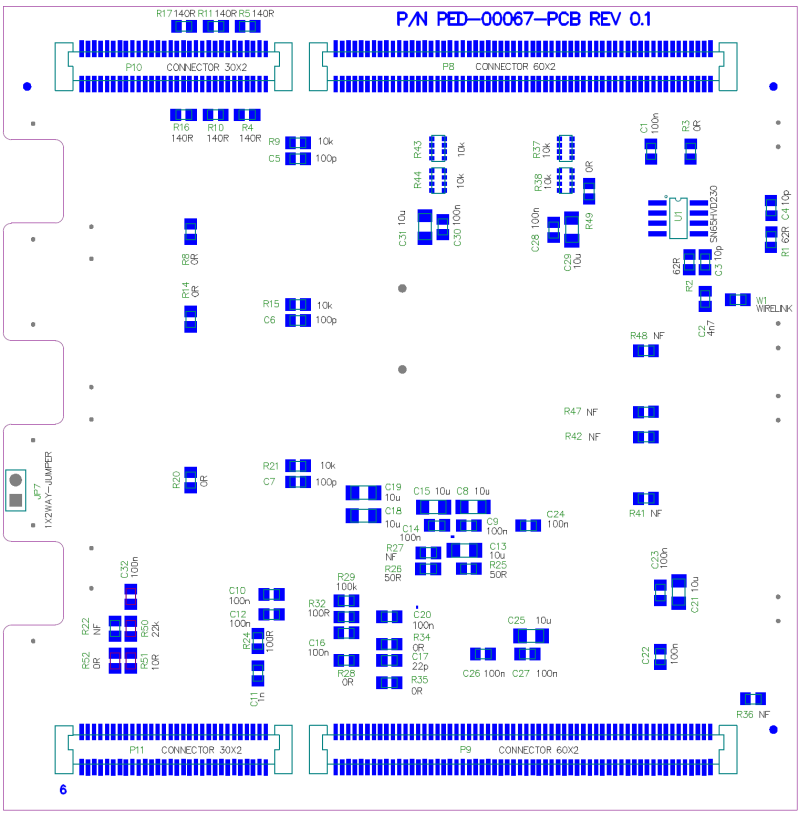

Figure 3-2: Assembly Drawing - Bottom View

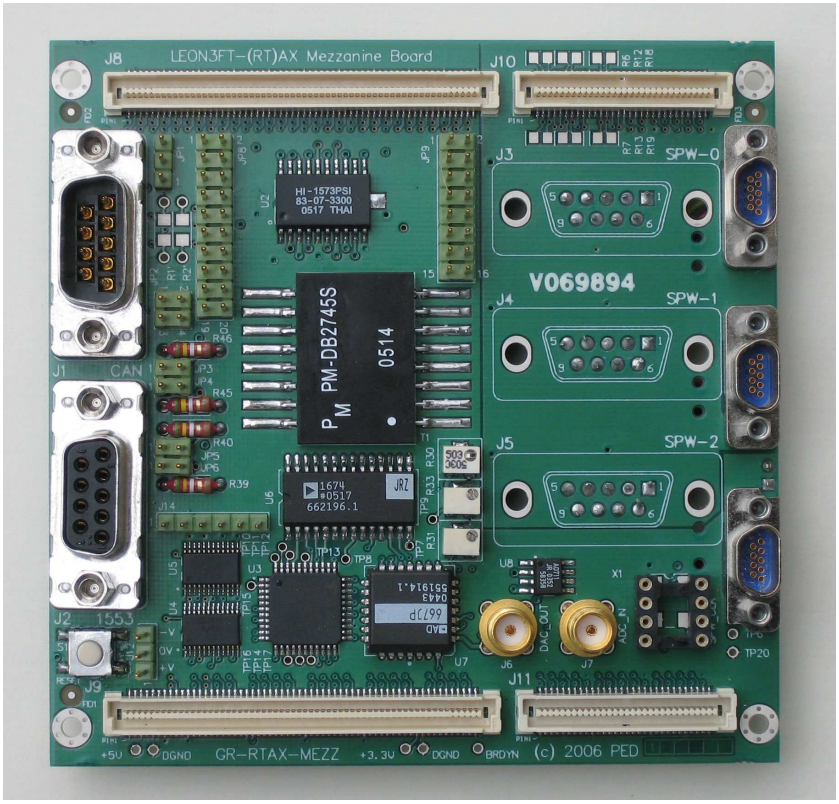

Figure 3-3: PCB Top view

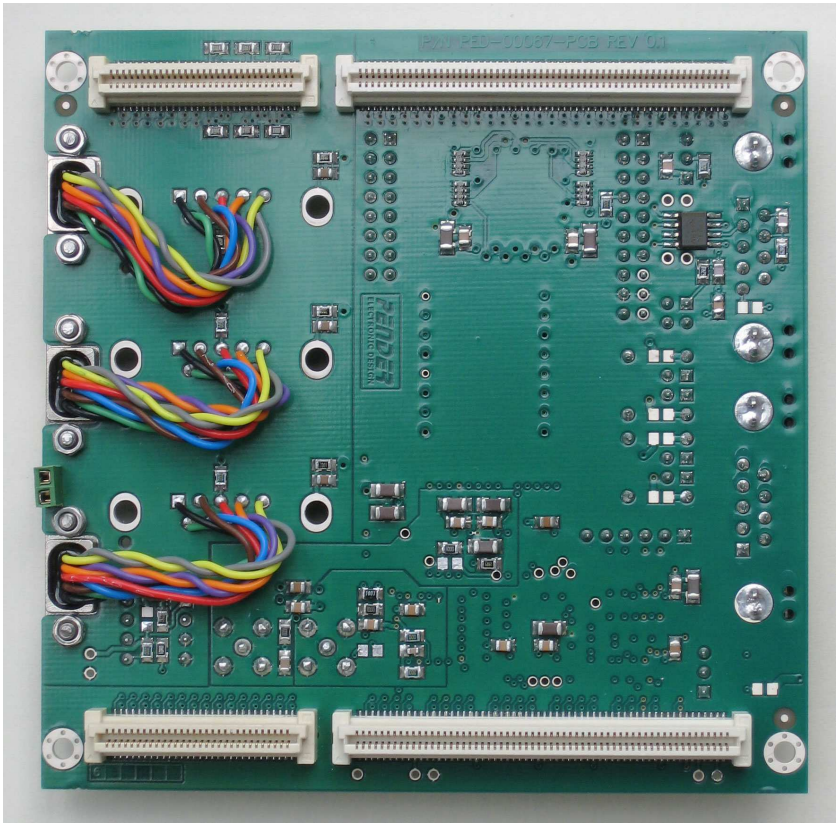

Figure 3-4: PCB bottom view

Annex: Accessing and Using the ADC and DAC on board

For demonstration purposes, the mezzanine board is equipped with a 12 bit DAC and a 12 bit ADC of types which are typically available also in space qualified versions.

No additional logic is required inside the FPGA in order to access and use these devices on the mezzanine board as these devices are intended to be memory mapped peripherals in the I/O address space of the Leon3 processor. However, in order control the devices it is necessary to include glue logic to perform the address decoding. This address decoding is incorporated into the Xilinx CPLD device (U3) on the mezzanine, and simply decodes the address and read/write control signals of the Leon processor memory bus in order to generate the DAC Chip select, ADC chip Select and the ADC Read/Convert signals.

Figure 3-5, provides the VHDL listing for the glue logic which has been implemented.

After compilation and generation of the *.jed* file, this can be programmed into the CPLD using a JTAG programming cable attached to connector J14 on the board.

In this example implementation:

- Writing of a DAC value is achieved by writing to the memory location 0x20100000.
- Triggering of an ADC conversion is achieved by writing to the memory location 0x20200000.
- Since the ADC conversion time of the device used is in the order of 10 us, it is necessary to check the ADC Ready status output of the ADC before reading the result. On this board the ADC Ready signal has been connected to the GPIO[7] input signal. When the GPIO[7] bit goes low the conversion is complete.
- Reading of an ADC result is achieved by reading to the memory location 0x20200000.

A simple example program which performs DAC and ADC conversions is given in Figure 3-6. This program simply sets the DAC output in 16 steps between zero and full-scale and at each step makes 256 ADC measurements. By connecting an SMA cable between the DAC-OUT and ADC-IN connectors a simple demonstration of the DAC and ADC operation can therefore be made using this software.

```
--------------------------------------------------------------------------------
-- Glue logic for connecting AD1674 ADC and AD667 DAC to LEON3 I/O Bus
--------------------------------------------------------------------------------
-- R. Pender  /  2006/08/31   /  richard@pender.ch
--------------------------------------------------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity mezz_cpld is
    Port ( CLK    : in std_logic;
           A      : in std_logic_vector(27 downto 20);
           RESETN : in std_logic;
           READ   : in std_logic;
           WRITEN : in std_logic;
           OEN    : in std_logic;
           IOSN   : in std_logic;
           RWEN   : in std_logic_vector(3 downto 0);

           BRDYN  : out std_logic;
           DACCSN : out std_logic;
           ADCCSN : out std_logic;
           ADCA0  : out std_logic;
           ADCRC  : out std_logic;
           ADC12_8: out std_logic;
           RESET  : out std_logic;
           SPARE  : out std_logic_vector(7 downto 0)
         );

end mezz_cpld;

architecture Behavioral of mezz_cpld is

begin

test : process(RESETN, CLK)

begin
  if RESETN ='0' then
     ADCCSN <= '1';
     DACCSN <= '1';
     ADCA0  <= '0';
     ADCRC  <= '1';
     ADC12_8<= '1';
     BRDYN  <= 'Z';
     RESET  <= '0';
     SPARE  <= (others => '0');
  elsif rising_edge(clk) then


  -- generate DAC chip select (low) if memory access is in IO space and
  -- in range 0x20100000 to 0x201fffff

    if ((IOSN='0') and (WRITEN = '0') and (A(27 downto 20)="00000001")) then
      DACCSN <= '0';
    else
      DACCSN <= '1';
    end if;

  -- generate ADC chip select (low) if memory access is in IO space and
  -- in range 0x20200000 to 0x202fffff

    if ((IOSN='0') and (A(27 downto 20)="00000010")) then
      ADCCSN <= '0';
    else
      ADCCSN <= '1';
    end if;

  -- generate ADC R/C =>

    if ((IOSN='0') and READ = '1' and (A(27 downto 20)="00000010")) then
      ADCRC <= '1';
    else
      ADCRC <= '0';
    end if;

    RESET <= not(RESETN);

  end if;
end process;

end Behavioral;
```

Figure 3-5: VHDL listing for Glue Logic CPLD

```
/*****************************************************************************/
/*  Project :    GR-RTAX-MEZZ                                                */
/*  Version :    Flash Memory Test Software    PENDER ELECTRONIC DESIGN GmbH  */
/*                                             Waffenplatzstrasse 92          */
/*  File    :    dac_adc_test.c                CH-8002 Zurich                 */
/*                                                                           */
/*  Authors :    R. Pender                                                   */
/*                                                                           */
/*                                                                           */
/*****************************************************************************/
//#include "leon.h"
#include <stdio.h>
//#include "IO_access.h"


#define MEMCFG1  0x80000000
#define PIOreg   0x80000500

#define DACreg 0x20100000
#define ADCreg 0x20200000

#define DACsteps 16
#define ADCsamples 64



/* ------------------------------------------------------------------------- */
/* Module Name : Main                                                        */
/* ------------------------------------------------------------------------- */
/* Creation                                                                  */
/* Date        : 18/09/06                                                    */
/* Author      : R. Pender                                                   */
/* Description :                                                             */
/*                                                                           */
/* Inputs      : -                                                           */
/* Outputs     : -                                                           */
/* ------------------------------------------------------------------------- */
/* Modification                                                              */
/* Date        :                                                             */
/* Author      :                                                             */
/* Description :                                                             */
/*                                                                           */
/* ------------------------------------------------------------------------- */

main()
{

  int i,j,z, ADCresult;

  printf("Start test...\n");

  // enable I/O space and set 15 I/O wait states
  *(volatile unsigned int*) MEMCFG1 = 0x10f802ff;    /* write to memory location  */

  for (j=0; j<DACsteps; j++)
  {
  // set dac with value
  *(volatile unsigned int*) DACreg= (j * 4096/DACsteps);

    for (i=0; i<ADCsamples; i++)
    //take a number of measurements
    {
      //trigger ADC
      *(volatile unsigned int*) ADCreg=0;
      //wait for ADC status to indicated ready (read PIO7)

      do
      {
        z = *(unsigned int*) PIOreg;    /* read from memory location */
      } while ((z & 0x00000080) != 0);


      // read ADC result
      ADCresult = ((*(unsigned int*) ADCreg) & 0xfff);  /* read from memory location */

      // print result
      printf("DAC = %1x ADC = %2x\n",(j * 4096/DACsteps), ADCresult);
    }
  }
}
```

Figure 3-6: C listing of example DAC and ADC control program