ADwin-Gold

Hardware Manual

Version 2.3 November 2002

Table of contents

Typographical Conventions

“Warning” stands for information, which indicate damages caused by incorrect handling of hardware or software, test setup or injury to persons.

Note” stands for:
– information, which have absolutely to be considered in order to guarantee an operation without any errors
– advice for efficient operation

“Information” stands for further information in this documentation or for other sources such as manuals, data sheets, literature, etc.

`<C:\ADwin\ ...>` File names and file paths are indicated in brackets in the font Courier New

`value 1` Program instructions and inputs in the editor window are characterized by the font Courier New

`Var_1` Names of variables in examples are printed in italics

`ENTER` Typography for ***ADbasic*** instructions (in capital letters)

1. Information about this Manual

This manual contains complex information about the operation of the ***ADwin-Gold*** system. Additional information are available in

– the manual „***ADwin*** Driver Installation“, which describes all interface installations for the ***ADwin*** systems. With this manual you begin your installation.
– the description of the configuration program ADconfig, with which you initialize the communication from the corresponding interface to your ***ADwin-Gold*** system.
– the description of the driver installations for the development programs Visual C, Visual BASIC, Delphi, Dia*dem*. LabView, TestPoint, Matlab etc.
– the manual: „***ADbasic***“, which contains all instructions for the compiler ***ADbasic***. With this easy-to-learn real-time development tool you can initialize your ***ADwin-Gold*** optimally for your control processes or for your control or measurement tasks.

Please pay attention to the following information

Please pay attention to the information given in this documentation and in other mentioned manuals so that the ***ADwin*** system works appropriately.

User restrictions

The manufacturer of the systems described in this documentation takes into account that the person who uses the systems is a specialist.

A specialist is a person who, due to his education, experience and training as well as his knowledge in relevant standard norms and regulations and his knowledge in his operational environment, has been authorized by a quality assurance representative to execute all necessary tasks. (VDE 105 and ICE 60364 Definition für Fachkräfte - Definition for specialists).

Availability of the documents

This product documentation and all documents referred to, have always to be available and it is recommended to observe these regulations. For damages caused by disregarding the information in this documentation or in all other additional documentations, no liability is assumed by the company Jäger Computergesteuerte Messtechnik GmbH, Lorsch.

Legal instructions

OEM products are mentioned without referring to possible patent rights, whose existence is not to be excluded.
Subject to change.

Hotline: +49 6251 9632 0, E-Mail: info@*ADwin*.de, Internet: www.*ADwin*.de

2. *ADwin* System Features

2.1 System Concept

ADwin systems guarantee that measurement data acquisition and automation tasks can be executed very fast and accurately under real-time conditions. This is an ideal basis for applications such as:
– very fast digital closed-loop controllers
– very fast open-loop controls
– data acquisition with very fast online analysis of the measurement data
– monitoring of complex trigger conditions and many more
The ***ADwin*** systems are optimized for processes which need **very short process cycle times** of one millisecond up to some microseconds.

Distinctive features

The ***ADwin-Gold*** system is equipped with analog and digital inputs and outputs, a fast processor (32 bit floating point signal processor) and a local RAM. The processor is responsible for the whole real-time processing in the system. The applications are **running independent** of the PC and its workload.

Processor

The processor of the ***ADwin*** system **processes *each* measurement value *instantaneously***. In one cycle the status of the inputs can be acquired, the status can be processed by the help of any mathematical functions, and the system can react to the results, even at very fast process cycle times of some microseconds. Here we have a perfect and logical work sharing: On the PC a program is running for visualizing of data, for input and operation of the processes, together with access to networks and data bases, while on the processor of the ***ADwin*** system all tasks which require real-time, are executed at the same time.

Real-time operating system

The real-time operating system for the DSP of the ***ADwin*** system has been optimized to reach the fastest response times possible. It manages parallel processes in a **multitasking** manner. Low priority processes are managed by time slicing. High priority processes interrupt all low priority processes and are instantaneously and completely executed. High priority processes are executed as time-controlled or event-controlled processes (trigger/event input).

Timing

For the precise calling of high priority processes the built-in **timer** is responsible. It has a resolution of 25 nanoseconds.The ***ADwin*** systems are characterized by an extremely short response time of only 300 nanoseconds during the change from a low priority to a high priority process. A continuously running communication process makes it possible to obtain a continuous data exchange between the ***ADwin*** system and the PC even during applications in process. The communication has no influence on the real-time capability of the ***ADwin*** system, nevertheless, it is possible to exchange data at any time.

ADbasic

With the real-time development tool ***ADbasic*** it is possible to create very easily and quickly time-critical programs für ***ADwin*** systems. ***ADbasic*** is an **integrated development environment** under Windows with possibilities for online debugging. An habitual, easy-to-learn BASIC instruction syntax has been extended by many more functions, in order to get direct access to inputs and outputs as well as by functions for process control and communication with the PC.

Communication between *ADwin* system and PC

Interfaces

The ***ADwin*** system is connected to the PC via an ISA interface (optional PCI, PCMCIA, USB, Ethernet). After power-up the ***ADwin*** system is booted from the PC via this interface. Afterwards the ***ADwin*** operating system is waiting for instructions from the PC which it will process.

Instructions

There are two kinds of instructions: On the one hand instructions, which are used to transfer data from the PC to the ***ADwin*** system, for instance „load process“, „start process“ or „set parameters“, on the other hand instructions which wait for a response from the ***ADwin*** system, for instance „read variables“ or „read data sets“. Both kinds of instructions are processed immediately by the ***ADwin*** system - or respectively they are responding immediately. The ***ADwin*** system never sends data to the PC without being asked to do so! The data transfer to the PC is always a response to an instruction coming from the PC. Thus, embedding the ***ADwin*** system into various programming languages and standard software packages for measurements is easily made, because they have only to be able to call functions and to process the return value.

Software interfaces

Under Windows 95/98/NT/ME/2000/XP a **DLL and an ActiceX interface** are available. Based on this the following drivers for **user interfaces** are also available.
Visual-BASIC, Visual-C, C/C++, Delphi, VBA (Excel, Access, Word), TestPoint, LabVIEW / LabWINDOWS, HP-VEE, InTouch, DIA*dem*, Matlab.
The easy, instruction-oriented communication with the ***ADwin*** system makes it possible that several Windows programs can access the same ***ADwin*** system at the same time. This is of course a great advantage when programs are developed and installed.

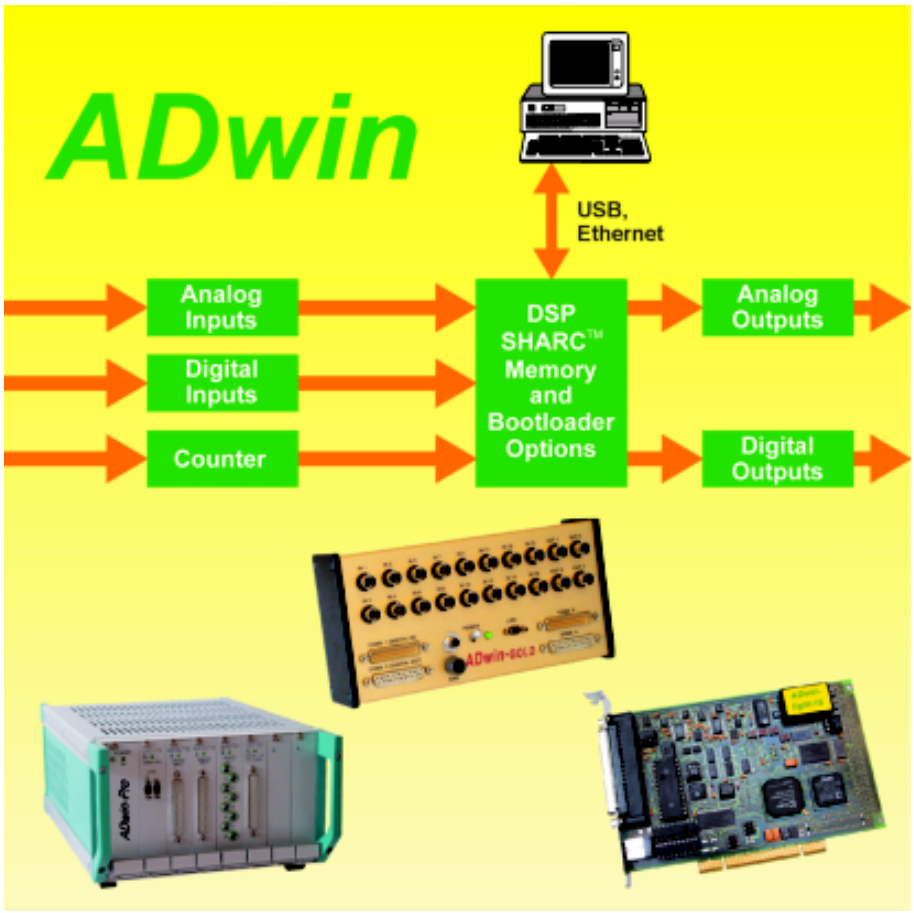


Figure 2-1: Concept of the ***ADwin*** Systems

2.2 The *ADwin-Gold System*

Processor and memory

The ***ADwin-Gold*** system is equipped with the digital **32 bit signal processor** ADSP 21062 (SHARC) from Analog Devices with floating point and integer processing. It is responsible for the complete measurement data acquisition, online processing, and signal output, and makes it possible to process instantaneously sample rates of up to several 100 kHz.
The on-chip **memory with 256 kB** has a very short access time of 25 ns and is large enough to hold the complete ***ADwin*** operating system, the ***ADbasic*** processes and all variables.
In order to get maximum access times, all inputs and outputs are memory-mapped in the external memory section of the DSP. For buffering larger quantities of data the DSP uses an external memory (DRAM) of 4 MB (optional 16 MB or 32 MB).

Analog inputs

The system has **16 analog inputs** with BNC sockets, which are divided into two groups and in each group is connected to one multiplexer. These two outputs are optionally converted by a 12-bit or 16-bit analog-to-digital converter (ADC), (see Figure: Bock diagram ***ADwin-Gold***). With the 12-bit ADCs it is possible to sample **very fast** with the 16-bit ADCs **highly accurately.**

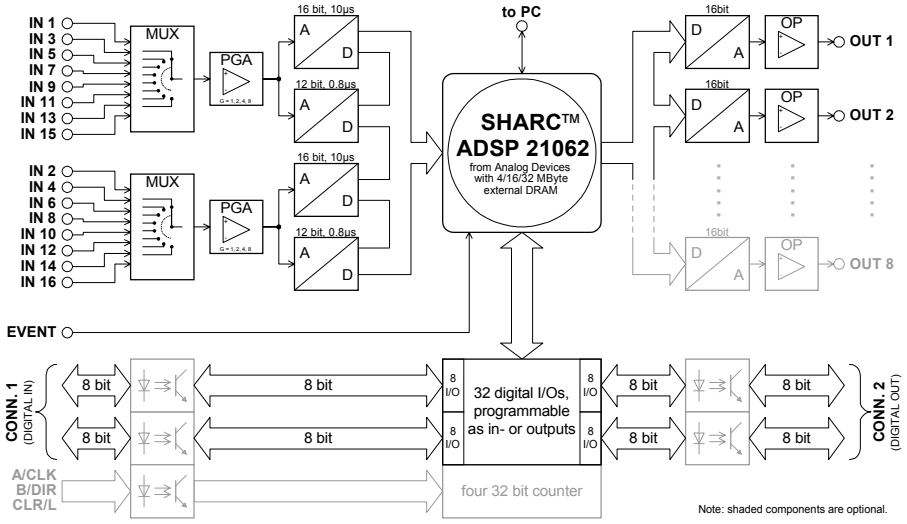


Figure 2-2: Block diagram ***ADwin-Gold***

Analog outputs

The standard version of the ***ADwin-Gold*** system is equipped with **two** (optional eight) **analog outputs** with an output voltage range of -10 to +10 Volt and a 16-bit resolution. You can synchronize the output of the voltage of all DACs per software.
In order to smooth the output signal, it passes through a low-pass filter with a cut-off frequency of f_c = 300 kHz.

Digital inputs and outputs

32 digital inputs or outputs are available on two 25-pin D-Sub connectors. They can be programmed in groups of eight as inputs or outputs. The inputs or outputs are TTL-compatible.

Trigger input (EVENT)

ADwin-Gold has a trigger input (EVENT, see also chapter 5.2 „Digital Inputs and Outputs“). Processes can be triggered by a signal and are completely processed afterwards. (see ***ADbasic*** manual, chapter: Structure of the ***ADbasic*** Program).

ADlink-ISA board

The *ADlink* ISA board is installed in an **ISA slot** of the PC. This is the connection via PC or notebook to the ***ADwin-Gold.*** The *ADlink* cable transfers data from the *ADlink* ISA board to the ***ADwin-Gold***. It has 9-pin D-Sub connectors and it is shielded on both ends. If requested we will deliver an *ADlink* cable which is shielded on one end (optically isolated).
The supplied 3-pin cable is used for the power supply and the earth connection from the ***ADwin-Gold*** to the PC.
Due to technical reasons, pin number 5 has been removed from the subminiature connector.

Standard delivery

The standard delivery items for the ***ADwin-Gold*** system:
– the ***ADwin-Gold*** system
– the *ADlink* ISA board for installation into the PC
– the 3-pin connecting cable for power supply
– the *ADlink* cable, 9-pin data connecting cable, length 2.0 m
– the ***ADwin*** CDROM
– the manual „Driver Installation“
– this hardware manual

2.2.1 Further Interfaces

In addition to the *ADlink* board, the following items are available:
– *Gold-PCI*: board for installation into a **PCI slot** (replaces the *ADlink*-ISA board), in combination with an *ADpcmcia* link adapter
– *ADpcmcia*: link adapter for connection to a **notebook**, with *ADlink* cable
– *Gold-USB*-Set: external **USB adapter**, with connecting cables
– *Gold-ENET*-Set: interface for the connection to the **Ethernet**, with connecting cables

2.2.2 Options (no later upgrading possible)

The following options are available:
– *Gold-D*A: 6 additional analog outputs, 16-bit DACs
– *Gold-CO1*: counter option with four 32 bit counters, which can optionally be used for period width measurement, as impulse counters or as up/down counters with clock/direction or four edge evaluation for quadrature encoders
– *Gold-opt*: optical isolation of the digital inputs/outputs
– *Gold-G-MEM-16/32*: external memory with 16 MB/32 MB instead of 4 MB
– *Gold-G-MEM-512k*: internal CPU memory with 512 kB instead of 256 kB
– *Gold-Boot*: Flash-EPROM bootloader for stand-alone operation without PC
– *Gold-Mount*: kit for installation of the ***ADwin-Gold*** system on a DIN rail

All additional options can be combined with each other.

2.2.3 Accessories

ADbasic

– ***ADbasic***, real-time development tool for all ***ADwin*** systems
– ***ADwin-Gold****-pow*: external power supply (necessary for notebook operation)
– *Gold-cable-5: ADlink* cable, length 5 meters
– *ADlink* cable which is shielded on only one end, length 2m / 5 m
– connector for the external power supply

3. Operating Environment

The ***ADwin-Gold*** electronic is installed in a closed aluminum enclosure and it is only allowed to operate it in this enclosure. (Exeption: Calibration, see chapter with the same name). With the necessary accessories the system can be operated in 19-inch-enclosures or as a mobile system (e.g. in cars). See also chapter 2.2.2 „Options“).

Earth protection

The ***ADwin-Gold*** device **must be earth-protected**, in order to
– build a ground reference point for the electronic
– to conduct interferences to earth.
Connect the GND socket, which is internally connected with the ground reference point and the aluminum enclosure, via a short low-impedance solid-type cable to the central earth connection point of your device.

The 3-pin power supply cable is the galvanic connection between the PC and the ***ADwin-Gold.***
The data lines of the *ADlink* cable are optically isolated by opto-couplers. When using an external power supply with is fully isolated from earth the PC can be completely optically isolated from the ***ADwin-Gold***.
The ***ADwin-Gold****-opt* module has optically isolated digital inputs and outputs.

Excluding transient currents

Transient currents, which are conducted via the aluminum enclosure or the shielding of the *ADlink* cable, have an influence on the measurement signal. You may reduce these transient currents, by ordering (on request) the *ADlink* cable shielded on one end only.
Please, make sure that the shielding is not reduced, for instance by taking measures for bleeding off interferences, such as connecting the shielding to the enclosure just before entering it. The more frequently you earth the shielding on its way to the machine the better the shielding will be.

Use cables with shielding on both ends for **signal lines.**
Here too, you should reduce the bleeding off of interferences via the ***ADwin-Gold*** aluminum enclosure by using screen clips.

BNC cables

The shielding of BNC cables is normally used as differential ground and looses therefore the shielding effect. So BNC cables are influenced by interferences when differential measurements are executed. For signal and data transfer outside of an enclosure it is necessary to use twisted pair data transfer cables, whose channels are shielded, too.

The ***ADwin-Gold*** is operated with a protection low voltage of **10 to 18 Volts** and is not life-threatening. For operation with an external power supply, the instructions of the manufacturer applies.

Ambient temperature

The ***ADwin-Gold*** is designed for operation in dry rooms with a room temperature of +5 °C ... +50 °C and a relative humidity of 0 ... 80 % (see chapter 13: Technical Data).

The temperature of the chassis must not exceed +60 °C, even under extreme operating conditions - e.g. in an enclosure or if the system is exposed to the sun for a longer period of time. You risk damages at the device or not-defined data (values) are output which can cause damages at your measurement device under unfavorable circumstances.

4. Initialization of the Hardware

If you start **initializing** do not connect any cables to the ***ADwin-Gold*** before you have executed the **following steps**:
– carry out the driver installation at the PC or notebook (see manual „***ADwin*** Driver Installation“).
– connect the ***ADwin-Gold*** only with the PC or notebook (s.b.).
– Read chapter 5 “Inputs and Outputs“ in this manual.
– Now you may start connecting the inputs and outputs.

If your application requires an optical isolation between PC and ***ADwin-Gold*** system please, use an external power supply unit and the *ADlink* cable with shielding on one end only, which will be delivered separately.

Providing the power supply

Please pay attention that reliable power soucre is supplied.
This concerns the PC (standard delivery). Otherwise also the external power supply, if operated in a car, the battery voltage.

If using current-limiting power supplies, please pay attention to the fact, that after power-up the current demand can be a multiple of the idle current. More detailed information can be found in the Technical Data.

In case of a power failure ...

– all data which have not been saved are lost. Not-defined data (values) can under unfavorable circumstances cause damages to other equipments.
– the input resistance at the analog inputs will get into the status of low-impedance, so that damages may occur at the ***ADwin-Gold*** system and other connected devices (see also annex A-1 „Technical Data). Therefore avoid absolutely connecting and operating the analog outputs when the ***ADwin-Gold*** system is not powered up.

Connection

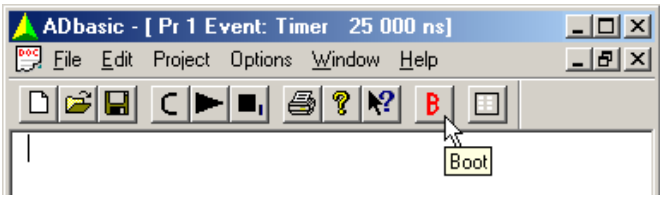
If you have completed the installation of the ***ADwin*** drivers and the configurations in the ***ADbasic*** menue Options ▸ Compiler, than connect the ***ADLink*** **cable and the power supply cable.**

Power-up

In order to avoid switching off the system inadvertently, the switch is equipped with a blocking device. Pull the switch a little bit, than pull it into the direction „Power“. Now the device is switched on and the green LED lights up.

Booting

Start ***ADbasic*** and boot the ***ADwin*** system by clicking on the boot button „B“.



The display in the status line: „***ADwin*** is booted“ shows that the operating system has been loaded appropriately and that via ***ADbasic*** the ***ADwin*** system has been connected. At the same time the **flashing of the green LED** at the ***ADwin-Gold*** shows that it is ready for operation.

Programming

Programming the ***ADwin*** systems is described more detailed in the ***ADbasic*** manual.
Start with the programming examples in the ***ADbasic*** Tutorial.

5. Inputs and Outputs

All inputs and outputs may only be operated according to the specifications given (s. annex A-1 „Technical Data“). In case of doubt, ask the manufacturer of the device, to which you want to connect the ***ADwin-Gold*** system.

Please avoid connecting and operating at the analog inputs when the ***ADwin-Gold* system is switched off.** In this case the input resistance can get in the status of low impedance at the analog inputs, so that damages at the ***ADwin-Gold*** and at other connected devices may occur, (see also annex A-1: „Technical Data, Analog Inputs, Overvoltage - maximum permissible current).

Standard instructions

For fast and easy programming there are standard instructions available in the compiler ***ADbasic***, which enable a user to easily measure or output data (see also ***ADbasic*** manual). Use other instructions only if extremely time-critical or special tasks require to do so.

More detailed information about the analog as well as the digital inputs and outputs can be found in the following chapters.

The pin assignments of CONN.1 and CONN.2 can be found in chapter 5.3.

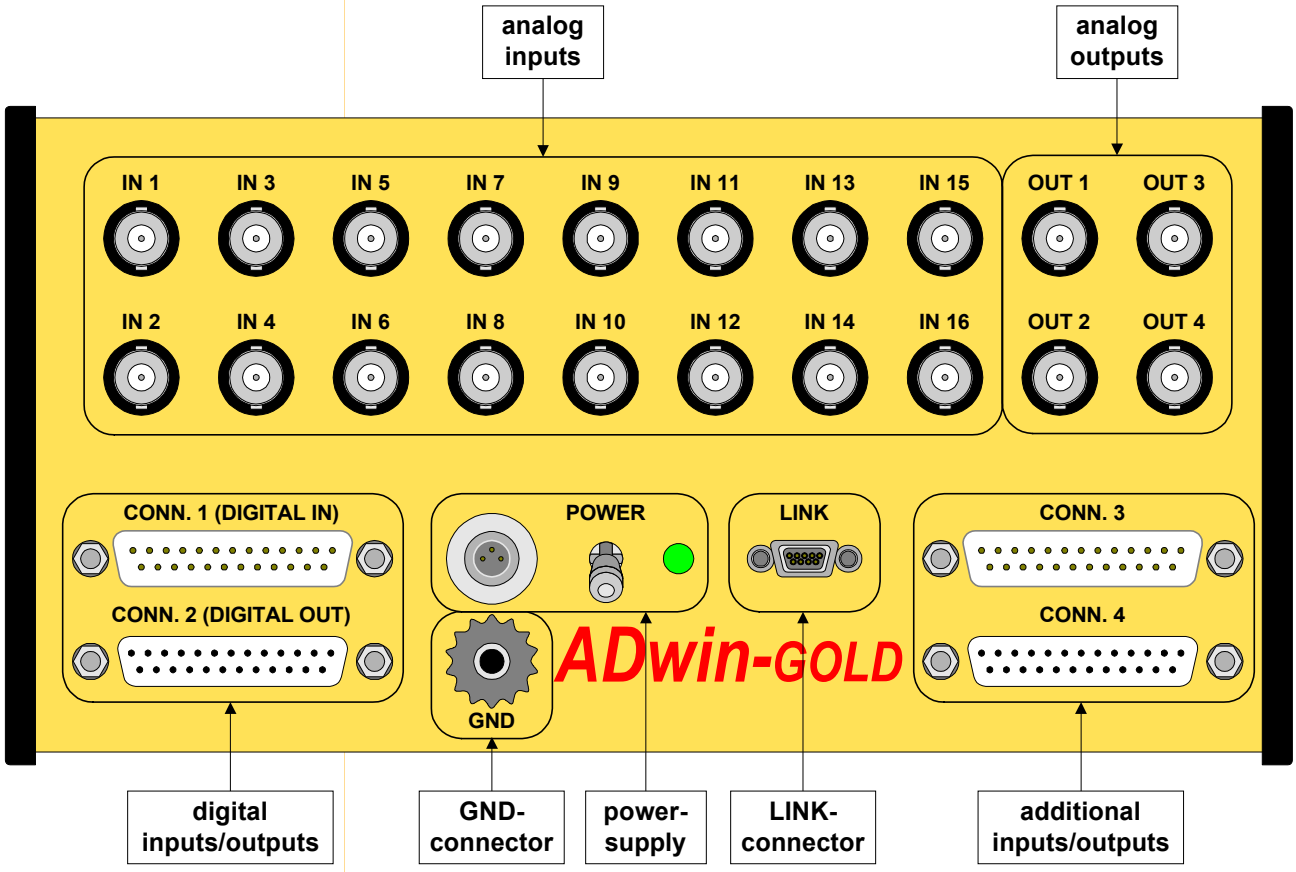


Figure 5-1: Schematic of ***ADwin-Gold***

Power Supply

The power supply connection of the ***ADwin-Gold*** with 12 V (see Annex, „Technical Data“), is made via the built-in connector, at left next to the power switch or above the GND socket (see picture 5-1). Connect there the 3-pin subminiature connector. For the pin assignment see the following picture:

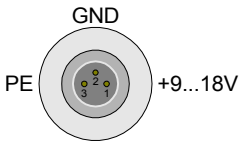


Figure 5-2: Power supply connector (male)

If you operate the ***ADwin-Gold*** together with a PC, the system can get its power supply by using the power-connector of the link adapter.

For using the system with an external power supply unit you need the subminiature connector described above of the series 712, with the article number: 2 99-0406-00-03. The connector is provided by

Franz Binder GmbH + Co. elektrische Bauelemente KG
Rötelstrasse 27
D-74172 Neckarsulm
Tel.: +49 - 7132 - 325 -0
www.binder-connector.de

When using the system with a notebook, power has to be supplied by a separate power supply. Please pay attention to the fact that it is sufficiently dimensioned.

5.1 Analog Inputs and Outputs

In order to operate the system without any interferences, **isolated BNC connectors** are necessary. Otherwise there will be the danger of damages caused by ESD or short circuits at the inputs. This will be the case when using not isolated BNC T-pieces.

The ***ADwin-Gold*** device **has to be connected to earth**, in order to execute measurement tasks without any interferences. Connect the GND socket via a low-impedance solid-type cable with the central earth connection point of your device.

The power supply from the *ADlink* or *ADlink-PCI* board also connects the earth of the ***ADwin-Gold*** with the earth of the PC. If you do not operate the PC and the ***ADwin-Gold*** system in the same place, you should not use the power supplied by the PC but an external power supply unit which if fully earth-free, in order to avoid influences by **different ground reference potentials**.

In addition to the description of the inputs and outputs you will find notes below for the conversion of digits into voltage values and for the input settings of the analog inputs (differential / single-ended).

5.1.1 Inputs

Differential

The default setting of the analog inputs is differential (see chapter 5.1.4). The voltage difference between positive and negative input (inner and outer conducter of the BNC socket) is measured.

Multiplexer

The system has 16 analog inputs (IN1...IN16) with male BNC-sockets, which are arranged in 2 rows. The inputs with odd numbers (upper row) are allocated to multiplexer 1 (and ADC 1), those with even numbers (lower row) to multiplexer 2 (and ADC 2).

16-bit and 12-bit measurements

You can convert the signals at the multiplexer outputs optionally with a 12-bit or a 16-bit analog-to-digital-converter (ADC), (see figure 2-2 „Schematic of the ***ADwin-Gold)***. You are measuring with

- the 12-bit ADC very fast (max. 0.8 µs, resolution 4.8828 mV),
- the 16-bit ADC very accurately (max. 8 µs, resolution 305 µV).

ADC instruction

To simply execute a complete measurement with the ADC use the instructions `ADC()` for the 16-bit ADC and `ADC12()` for the 12-bit ADC.
The ADC instructions consider for instance the settling of the multiplexer and assure perfect measurements (see also ***ADbasic*** manual).

Please pay attention to a small internal resistance of the power supply unit (of the input signals), because it may have influence on the measuring accuracy!

The measuring normally fits to the internal resistance of the ***ADwin-Gold*** system; up from 10 Ω the internal resistance of the power supply unit causes a linear error. Moreover, from about 1 kΩ upwards the multiplexer settling time extends. The waiting time defined in the standard instructions `ADC()` and `ADC12()` shall then fall short, so that unprecise values are recalled. In this case please use the instructions described in chapter 5.3.1.

5.1.2 Outputs

DAC instruction

The standard instruction `DAC(number,value)` checks each of the values if it exceeds or falls below of the 16-bit value range (0...65,535). If the value is in the 16-bit value range, the indicated value is output on the output *number*. If it is not in the value range the maximum (65,535) or minimum (0) value is output (see also ***ADbasic*** manual).

5.1.3 Calculation Basis

The voltage range of the ***ADwin-Gold*** at the analog inputs and outputs is between –10 V to +10 V (bipolar 10 V).

Allocation of digits to voltage

The 65,536 (2^{16}) digits are allocated to the corresponding voltage ranges of the ADCs and DACs insofar that
– 0 (zero) digits correspond to the maximum negative voltage and
– 65,535 digits correspond to the maximum positive voltage

The value for 65,536 digits, exactly 10 Volt, is just outside the measurement range, so that you will get a maximum voltage value of 9.999695 Volt for the 16-bit conversion and a voltage value of 9.995117 Volt for the 12-bit conversion.

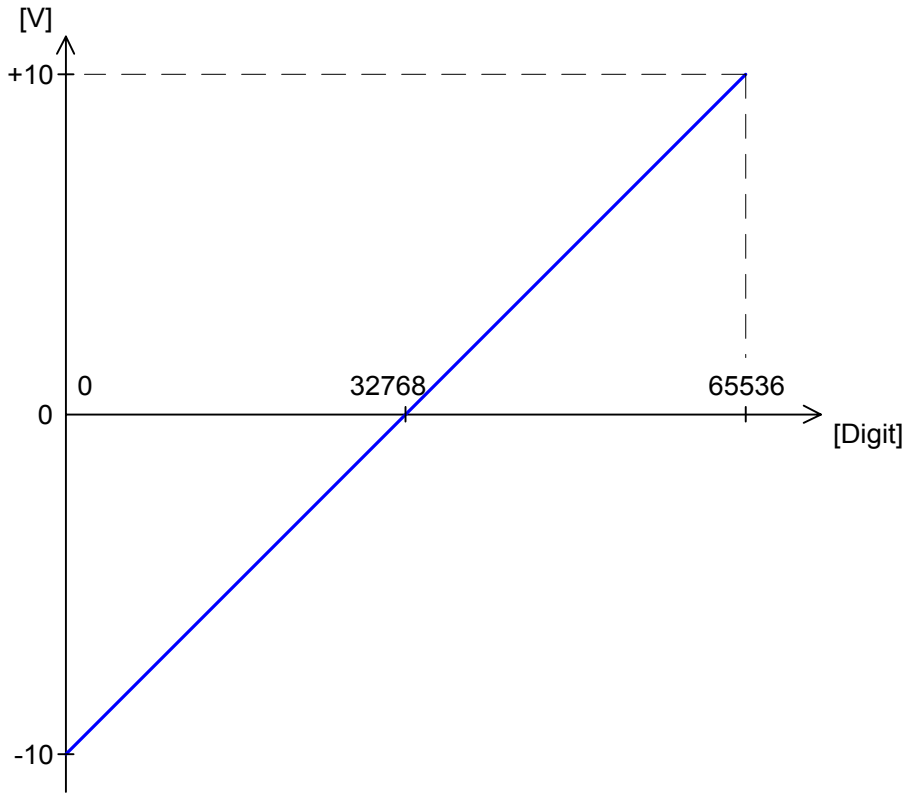


Figure 5-3: Zero offset in the standard setting of bipolar 10 Volt

Zero offset

In the bipolar settings you will get a zero offset, called only offset in the following text.

For the voltage range –10 V to +10 V applies: $U_{OFF} = -10$ V

Gain factor

The ***ADwin-Gold*** has a programmable gain (PGA), with which you can amplify the input voltage by the factors 1, 2, 4, and 8. At the same time the measurement range gets smaller by the corresponding gain factor k (see Technical Data).

Please note that upon applications with k >1 the interference signals are amplified respectively.

Least Significant Bit U_{LSB}

The quantization level (U_{LSB}) is the smallest digitally displayable voltage difference and is equivalent to the voltage of the least significant bit. A U_{LSB} is equivalent to the following formula:

- with 16-bit converters: 20 V / 2^{16} = 305.175 µV,
- with 12-bit converters: 20 V / 2^{12} = 4,882.8 µV.

Allocation of the bits

In order to get the same bit allocation during measurements with the 12-bit ADC and the 16-bit ADC, the converted value is returned left-aligned in one word (16-bit) when using the 12-bit ADC. The lower four bits are always 0 (zero).

Bit	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
12-bit-ADC	12-bit mantissa left-aligned in the lower word												0	0	0	0
16-bit-ADC/DAC	16-bit mantissa in the lower word															

The 4,096 digits of the 12-bit ADC are mapped to the 65,536 digits of the 16-bit ADC. Thus, 16 digits of the 16-bit ADC are equivalent to one digit of the 12-bit ADC.

Therefore the following equations can be used for both ADC types.

DAC

$$U_{OUT} = Digits \cdot U_{LSB} + U_{OFF}$$

$$Digits = \frac{U_{OUT} - U_{OFF}}{U_{LSB}}$$

Conversion Digit ⇆ Voltage

ADC

$$Digits = \frac{k \cdot U_{IN} - U_{OFF}}{U_{LSB}}$$

$$U_{IN} = \frac{Digits \cdot U_{LSB} + U_{OFF}}{k}$$

For the DACs:

For the ADCs (12-bit and 16-bit):

Tolerance Ranges

Slight variations regarding the calculated values may be within the tolerance range of the individual component. Two kinds of variations are possible (in LSB), which are indicated in this hardware manual:

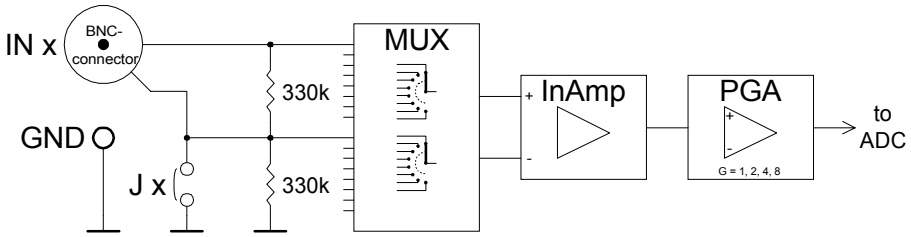
INL

- The integral non-linearity (INL) defines the maximum deviation from the ideal straight line of the conversion characteristics curve, covering the whole input voltage range.

DNL

- The differential non-linearity (DNL) defines the maximum deviation from the ideal quantization level.

5.1.4 Input Circuitry of the Analog Inputs



At **differential inputs** (standard delivery) the voltage difference between inner and outer conductor of the BNC socket is acquired.

At **„single ended" inputs** all outer conductors of the BNC sockets are connected via jumper Jx with the common ground (GND-connection).

Jumper (Jx)

The ***ADwin-Gold*** is delivered with the jumper setting for differential inputs, that means the jumpers (Jx) are only plugged into one of the two pins. For the „single-ended" setting you have to close the corresponding jumper, that means to connect both pins. The following picture of the main printed circuit board shows where you can find the corresponding jumpers. The number added to the „J" corresponds to the number of the analog inputs.

Open the chassis

Proceed as follows:

- You open the chassis by unscrewing the Allen screws at both sides and remove the bottom part. Pull of very carefully and softly the attached printed circuit board off the main printed circuit board.
- After replugging the jumper/s put the printed circuit board/s again back to the analog printed circuit board. Please, pay attention to the fact that all pins of the pin connectors are plugged into the socket strips in the right order.
- Close the chassis and tighten the Allen screws.

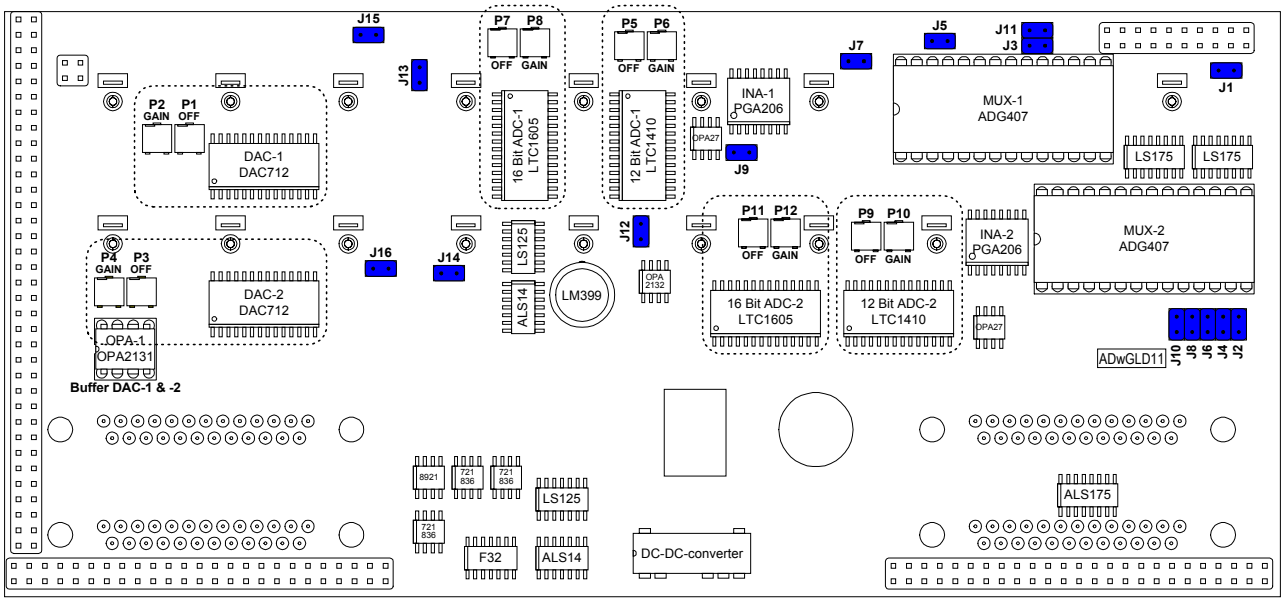


Figure 5-4: Jumper position on the analog printed circuit board

5.2 Digital Inputs and Outputs

32 digital inputs and outputs (abbreviation: DIO) are available on two 25-pin D-SUB connections (see picture below). They are programmable in groups of eight as inputs or outputs.

The digital inputs and outputs are TTL-compatible and not protected against overvoltage.

Do not use connections marked as „reserved“. They are planned for upcoming changes and expansions and can cause damages to your system if you do not pay attention to this fact.

Trigger input (EVENT)

The ***ADwin-Gold*** is equipped with an external trigger input (EVENT). With this trigger input processes are triggered by an external signal (trigger) with rising edge and can completely and immediately be processed. (see ***ADbasic*** manual, chapter „Structure of an ***ADbasic***-Program“).

Power-up configuration

After power-up of the device, all connections are configured as inputs.

The instruction:

`CONF_DIO(12)`

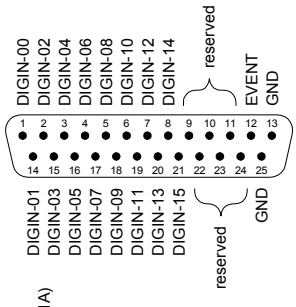
```
CONF_DIO(12)
```

configures DIO 0...DIO 15 as digital inputs and DIO 16...DIO 31 as digital outputs (see pin assignment below).
Only in this configuration will you be able to totally access the inputs and outputs wit the instructions

- `DIGIN`; `DIGIN_WORD`
- `DIGOUT_WORD`; `SET_DIGOUT`; `CLEAR_DIGOUT`

About programming under other configurations the following chapter will give you more detailed information: chapter 5.3.2 „Time-Critical Tasks / Digital Inputs and Outputs“), (see also ***ADbasic*** manual and ***ADbasic*** tutorial).

Inputs / DIGIN (male)



Outputs / DIGOUT (female)

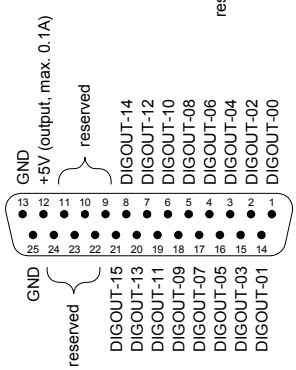


Figure 5-5: Pin assignment under configuration with `CONF_DIO(12)`

5.3 Time-Critical Tasks

For extremely time-critical tasks you can use instructions with which you have direct access to the **control and data registers of the ADC and DAC** (see ***ADbasic*** manual). These registers can be found in the memory address area of the ADSP (memory mapped). These instructions also allow to optimize the program structure (s.b.).

Contrary to the standard instructions `ADC()`, `ADC12()` and `DAC()` the instructions for direct access **do not have any test routines**. Before you use them we recommend to learn more about time sequences, progam structures and functions sequences in an ADC.

5.3.1 Analog Inputs and Outputs

`ADC()` and `ADC12()`

The standard instructions `ADC()` and `ADC12()` consist of a sequence of several instructions (see below). They need a certain time for execution.

```
SET_MUX()
...                 'wait for settling time
START_CONV()
WAIT_EOC()          'wait for end of conversion
READ_ADC() or READ_ADC12()
```

Program structure

You can use (or extend) the waiting times wasted in the standard instructions for other purposes by using the individual instructions. If you apply these instructions skilfully you may be able to execute faster measurements.

It is important to set the `START_CONV()` instruction in a sufficient time-delay from the `SET_MUX()` instruction, in order to consider the multiplexer settling time (see also ***ADbasic*** manual, „Instruction Reference“).

Use the waiting times for instance for arithmetic operations and save CPU time:

– Settling time of the multiplexer: At a maximum voltage jump of 20 Volt it is 6.5 µs (max.) for the 16-bit ADC and 1.5 µs for the 12-bit ADC.

– Conversion time of the ADCs: It is 0.8 µs for the 12-bit ADC and 8 µs for the 16-bit ADC.

Direct Register Access

ADC

A measurement can be executed very fast, when you directly access the control and data registers of the ADC.

DAC

If you have made sure that at the analog outputs the values are within the range limits, you can write very quickly into one or more DAC registers with direct access to the hardware registers, and you can synchronously start the output (see ***ADbasic*** manual).

The hardware addresses for the direct access to the control and data registers are described on the following pages.

Address [HEX]	Function	Bit												Comments
		31-16	15-10	9	8	7	6	5	4	3	2	1	0	
20 40 00 00	set MUX-#1: channels 1, 3, 5, ..., 15	-	-	-	-	-	-	-	-	-	n	n	n	"nnn" binary = 0...7 decimal, selected ch. = nnn + 1
	set MUX-#2: channels 2, 4, 6, ..., 16	-	-	-	-	-	-	n	n	n	-	-	-	"nnn" binary = 0...7 decimal, selected ch. = 2(nnn + 1)
	gain PGA-#1	-	-	-	-	g	g	-	-	-	-	-	-	"gg" binary = 0...3 decimal, selected gain = 2^{gg}
	gain PGA-#2	-	-	g	g	-	-	-	-	-	-	-	-	
20 40 00 10	start conversion: ADC-#1 (16-bit)	-	-	-	-	-	-	-	-	-	1	-	s	s = 0 : start conversion s = 1 : no effect
	start conversion: ADC-#2 (16-bit)	-	-	-	-	-	-	-	-	-	1	s	-	
	start conversion: ADC-#1 (12-bit)	-	-	-	-	-	-	-	-	s	1	-	-	
	start conversion: ADC-#2 (12-bit)	-	-	-	-	-	-	-	s	-	1	-	-	
20 40 00 20	EOC status: ADC-#1 (16-bit)	-	-	-	-	-	-	-	-	-	-	-	e	e = 0 : end of conversion e = 1 : conversion is running
	EOC status: ADC-#2 (16-bit)	-	-	-	-	-	-	-	-	-	-	e	-	
	EOC status: ADC-#1 (12-bit)	-	-	-	-	-	-	-	-	e	-	-	-	
	EOC status: ADC-#2 (12-bit)	-	-	-	-	-	-	-	e	-	-	-	-	
20 40 00 30	read out register: ADC-#1 (16-bit)	-	x	x	x	x	x	x	x	x	x	x	x	x : result of conversion
20 40 00 40	read out register: ADC-#2 (16-bit)	-	x	x	x	x	x	x	x	x	x	x	x	
20 40 01 30	read out register: ADC-#1 (12-bit)	-	x	x	x	x	x	x	x	0	0	0	0	
20 40 01 40	read out register: ADC-#2 (12-bit)	-	x	x	x	x	x	x	x	0	0	0	0	
20 40 01 00	read out register and start conversion: ADC-#1 (16-bit)	-	x	x	x	x	x	x	x	x	x	x	x	
20 40 01 10	read out register and start conversion: ADC-#2 (16-bit)	-	x	x	x	x	x	x	x	x	x	x	x	
20 40 01 20	read out register and start conversion: ADC-#1 (12-bit)	-	x	x	x	x	x	x	x	x	x	x	x	
20 40 0 1D0	read out register and start conversion: ADC-#2 (12-bit)	-	x	x	x	x	x	x	x	x	x	x	x	

Table 5-1: Hardware addresses of the control and data registers for the **ADC**

Address [HEX]	Function	Bit												Comments
		31-16	15-10	9	8	7	6	5	4	3	2	1	0	
20 40 00 10	start conversion: all DACs synchronously	-	-	-	-	-	-	-	1	1	s	1	1	s = 0 : start conversion s = 1 : no effect
20 40 00 50	write only to the register: DAC-#1	-	x	x	x	x	x	x	x	x	x	x	x	x : digital value to be converted
20 40 00 60	write only to the register: DAC-#2	-	x	x	x	x	x	x	x	x	x	x	x	
20 40 00 70	write only to the register: DAC-#3	-	x	x	x	x	x	x	x	x	x	x	x	
20 40 00 80	write only to the register: DAC-#4	-	x	x	x	x	x	x	x	x	x	x	x	
20 40 00 90	write only to the register: DAC-#5	-	x	x	x	x	x	x	x	x	x	x	x	
20 40 00 A0	write only to the register: DAC-#6	-	x	x	x	x	x	x	x	x	x	x	x	
20 40 01 90	write only to the register: DAC-#7	-	x	x	x	x	x	x	x	x	x	x	x	
20 40 01 A0	write only to the register: DAC-#8	-	x	x	x	x	x	x	x	x	x	x	x	
20 40 02 00	write to the register and start conversion immediately: DAC-#1	-	x	x	x	x	x	x	x	x	x	x	x	
20 40 02 10	write to the register and start conversion immediately: DAC-#2	-	x	x	x	x	x	x	x	x	x	x	x	
20 40 02 20	write to the register and start conversion immediately: DAC-#3	-	x	x	x	x	x	x	x	x	x	x	x	
20 40 02 30	write to the register and start conversion immediately: DAC-#4	-	x	x	x	x	x	x	x	x	x	x	x	
20 40 02 40	write to the register and start conversion immediately: DAC-#5	-	x	x	x	x	x	x	x	x	x	x	x	
20 40 02 50	write to the register and start conversion immediately: DAC-#6	-	x	x	x	x	x	x	x	x	x	x	x	
20 40 02 60	write to the register and start conversion immediately: DAC-#7	-	x	x	x	x	x	x	x	x	x	x	x	
20 40 02 70	write to the register and start conversion immediately: DAC-#8	-	x	x	x	x	x	x	x	x	x	x	x	

Table 5-2: Hardware addresses of the control and data registers for the **DAC**

5.3.2 Digital Inputs and Outputs

After power-up of the device all connections are configured as inputs; this corresponds to the instruction `CONF_DIO(0)`. With the `CONF_DIO()` instruction, according to the following table, you can configure the inputs and outputs.

CONF_DIO( )	DIO 31 to DIO 24	DIO 23 to DIO 16	DIO 15 to DIO 08	DIO 07 to DIO 00
0	IN	IN	IN	IN
1	IN	IN	IN	OUT
2	IN	IN	OUT	IN
3	IN	IN	OUT	OUT
4	IN	OUT	IN	IN
5	IN	OUT	IN	OUT
6	IN	OUT	OUT	IN
7	IN	OUT	OUT	OUT
8	OUT	IN	IN	IN
9	OUT	IN	IN	OUT
10	OUT	IN	OUT	IN
11	OUT	IN	OUT	OUT
12	**OUT**	**OUT**	**IN**	**IN**
13	OUT	OUT	IN	OUT
14	OUT	OUT	OUT	IN
15	OUT	OUT	OUT	OUT
Applicable instructions:	`DIGOUT_WORD, CLEAR_DIGOUT, SET_DIGOUT`		`DIGIN_WORD, DIGIN`	
Instruction applicable for DIO nn, at	configuration "OUT"		configuration "IN" At configuration "OUT" the register contents of this byte is returned	

Please pay attention to the following restriction:

Only if the inputs/outputs are configured with `CONF_DIO(12)` will you be able to fully access the inputs/outputs with the instructions `DIGOUT_WORD`, `SET_DIGOUT`, `CLEAR_DIGOUT`, `DIGIN_WORD` and `DIGIN`. Otherwise you have to read out or write into the corresponding hardware register.

Address [HEX]	Function	Bit 31-16	15-10	9	8	7	6	5	4	3	2	1	0	Comments
20 40 01 E0	configre DIO 00 to DIO 07	0	0	0	0	0	0	0	0	-	-	-	c	c = 0: inputs; c = 1: outputs
	configure DIO 08 to DIO 15	0	0	0	0	0	0	0	0	-	-	c	-	
	configure DIO 16 to DIO 23	0	0	0	0	0	0	0	0	-	c	-	-	
	configure DIO 24 to DIO 31	0	0	0	0	0	0	0	0	c	-	-	-	
20 40 00 B0	Input registers DIO 00 to DIO 15	-	x	x	x	x	x	x	x	x	x	x	x	x : digital value read in
20 40 01 B0	Input registers DIO 16 to DIO 31	-	x	x	x	x	x	x	x	x	x	x	x	
20 40 01 C0	Output registers DIO 00 to DIO 15	-	x	x	x	x	x	x	x	x	x	x	x	x : digital value to be output
20 40 00 C0	Output registers DIO 16 to DIO 31	-	x	x	x	x	x	x	x	x	x	x	x	

Table 5-3: Hardware addresses of the digital inputs/outputs

6. Calibration

6.1 General Information

The digital-to-analog (DAC) and analog-to-digital (ADC) converters of the ***ADwin*** systems have been **calibrated in factory**. In accordance with the regulations for keeping the measurement accuracy in your field of application, the systems must be calibrated in regular time intervals.

Tools

The following tools are necessary for the calibration:
- insulated (!) alignment tool
- a reference voltage source with a resolution of
  - 30 µV when using 16 bit converters
  - 500 µV when using 12 bit converters
- a digital multimeter with a resolution of
  - 30 µV when using 16 bit converters
  - 500 µV when using 12 bit converters
- connecting cables from the inputs/outputs to the reference voltage source and to the measurement device
- Allen key 2 mm

Calibration steps

Calibrate according to the following steps:
1. Initialize hardware (chapt. 6.2)
2. Initialize software (chapt. 6.3)
3. Adjust DAC and ADC (chapt. 6.4)

6.2 Initializing the Hardware

Calibration has to be made when the ***ADwin-Gold*** system reaches its **operating temperature**. With a power-up temperature of the device of approx. 20 to 25 degrees Celsius (room temperature), the system reaches the operating temperature approx. 30 minutes after power-up.

Calibration is only possible when the device is opened. Please pay attention to the safety notes at the beginning of this documentation.

Open device

Open the system **just before calibration**, because the opened system cools down quickly. Remove the lower four Allen screws with the 2 mm Allen key, remove the upper four screws and remove also the chassis. Please lay down the system in such a way that it is stable and cannot fall down and that you can reach the potentiometers easily from above.

Reassembling the device is made in reverse order.

The analog (printed circuit) board (see the following figure), is directly located below the front panel, below the front panel there is the digital (printed circuit) board with the ADSP. If the ***ADwin-Gold*** system is equipped with an add-on, this component will be located between the analog and digital (printed circuit) boards.

Connect now the power supply cable and the *ADlink* cable and power-up the system. Connect the digital multimeter with the output of DAC (OUT) which you want to calibrate and connect the reference voltage source with the input (IN 1, 3, ..., 15 for ADC1; IN 2, 4, ..., 16 for ADC2) which you want to calibrate.

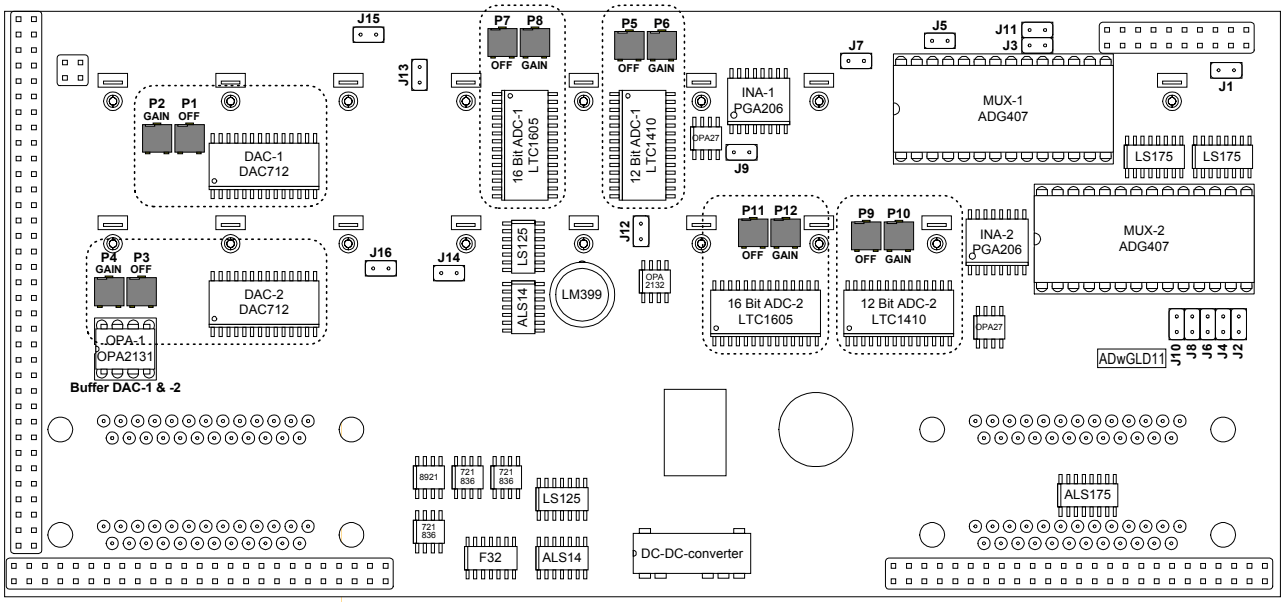


Figure 6-1: Position of the potentiometers for calibration

Position of the trimmers

The position of the gain and offset trimmers is described in the figure above. The dotted frame shows how the trimmers are allocated to the converters, that is, an offset and gain potentiometer for each of the converters. The trimmers can be found **between the printed circuit boards** and have lateral adjusting screws (in the picture they are adjusted in upward position).

The position of the trimmers for the DA add-on can be found in the description of the DA add-on.

6.3 Initializing the Software

– Start ***ADbasic***, boot the system and open the window: „Options/Parameter“.
– Enter in ***ADbasic*** the programs described in chapter 6.5. Afterwards compile and start these programs.
You can execute the programs either individually or all together (with different process numbers).

In the window „Options/Parameter“ you can set and compare the parameters:

Output channel DAC:	`PAR_10`
Output value DAC:	`PAR_9` (default value 32,768 => 0 V)
Input channel ADC:	`PAR_3`
Input value ADC:	`PAR_1`, mean value in `FPAR_1` for 16-bit ADC
	`PAR_2`, mean value in `FPAR_2` for 12-bit ADC

Please pay attention to the fact that the set channel corresponds to the connected measurement cables during the following adjustment.

The values indicated in the programs refer to the voltage range of 20 Volt (-10V...+10V) of the ***ADwin-Gold*** at the analog inputs and outputs.

Conversion of digits into voltage

Notes for the conversion of voltage into digits, digits into voltage as well as information about the quantization levels (U_{LSB}) and the offset (U_{OFF}) can be found in chapter 5.1.3.
Please pay also attention to the information about INL and DNL in chapter 5.1.3.

6.4 Adjustment

Use the test values of the table below for the adjustment.
An offset and gain trimmer are allocated to each of the converters (previous figure). Adjust the trimmers for each converter type implicitly according to the order indicated, even if you work with different test values.

6.4.1 DAC

1. Enter in the parameter window for PAR_9 the **test value min** (digits) and confirm by pressing the „send“ button or [RETURN]. Check with the multimeter the voltage which has been output and adjust it with the **offset potentiometer** until the voltage value of the table is displayed.

2 Now enter similarly the **test value max** (digits) for PAR_9. Check again the voltage value at the multimeter and adjust it with the **gain potentiometer.**

3. Check as **control** the voltage values of all test values.

6.4.2 ADC 16-bit

1. Set at the selected input the voltage for the **test value max** with the reference voltage source. Check in the parameter window the converted digital value (`PAR_1`) and adjust it with the **gain trimmer**, until the digital value for the test value max. is displayed.

2. Set the voltage for the **test value min.** at the input. Check again the converted digital value (`PAR_1`) and adjust it with the **offset trimmer**.

3. Check as **control** the digital values of all test values.

6.4.3 ADC 12-bit

1. Set at the selected input the voltage for the **test value mid.** with the reference voltage source. Check in the parameter window the converted digital value (`PAR_2`) and adjust it with the **offset trimmer**, until the digital value for the test value mean is displayed.

2. Set the voltage for the **test value max.** at the input. Check again the converted digital value (`PAR_2`) and adjust it with the **gain trimmer**.

3. Check as **control** the digital values for all test values.

	Maximum value	Test value max.	Test value mid.	Test value min.	Minimum value
Digits decimal hex	65,535 FF FFh	64,080 FA 50h	32,768 08 00h	1,456 05 B0h	0 0h
Voltage -10 V ... +10 V					
16-bit ADC	+9.9996948 V	+9.5556641 V	0 V	-9.5556641 V	-10 V
12-bit ADC	+9.9951172 V				

U_{LSB} 305.1758 µV at 16-bit; U_{LSB} 4,882.81 µV at 12-bit

Table 6-1: Test values for the adjustment of the DACs/ADCs

6.5 Processes for Calibration

The ***ADbasic*** processes for calibration can be found as <*.bas>-files on the ***ADwin***-CD-ROM beginning from version 3.00.30xx in the directory <C:\ADwin\Tools\Calibration\...>.

1. Output voltage with the DAC (Process 1)

```
'Process for the ADwin-Gold in order to
'output a voltage with the 16 bit DAC.
'Last modification on July 31st, 2000
'Usage of the variables:
'
'PAR_10: DAC channel number (1 or 2, with DA add-on: 1...8)
'PAR_9 : output value (0...65,535)
'#########################################################################
INIT:
  GLOBALDELAY=200000
  IF (PAR_10=0) then PAR_10=1 'prevent channel number 0 (not allowed)
  IF (PAR_9=0) then PAR_9=32768
    '64080 => +9.555664V  (at a voltage range of ±10V)
    '32768 => 0V
    ' 1456 => -9.555664V

EVENT:
  DAC(PAR_10,PAR_9)            'output value
```

2. Read voltage with the 16-bit ADC (Process 2)

```
'Process for the ADwin-Gold in order to
'read a voltage with a 16 bit ADC.
'A mean value is calculated in FPAR_1.
'Last modification on August 08, 2000
'Usage of the variables:
'
'PAR_1 : value which has been read (0...65,535)
'PAR_3 : channel number (1...16)
'FPAR_1: mean value
'#########################################################################
INIT:
  GLOBALDELAY=20000
  IF (PAR_3=0) then PAR_3=1   'prevent channel! number 0 (not allowed)
  IF (PAR_1=0) then PAR_1=32768
    '64080 => +9.555664V  (at a voltage range of ±10V)
    '32768 => 0V
    ' 1456 => -9.555664V

EVENT:
  PAR_1 = ADC(PAR_3)               'read value
  FPAR_1= FPAR_1*0.95 + PAR_1*0.05 'calculate mean value
```

3. Read voltage with the 12-bit ADC (Process 3)

```
'Process for the ADwin-Gold in order to
'read a voltage with a 12 bit ADC.
'A mean value is calculated in FPAR_2.
'Last modification on August 08  2000
'Usage of the variables:
'
'PAR_2 : value which has been read (0...65,535)
'PAR_3 : channel number (1...16)
'FPAR_1: mean value
'#########################################################################
INIT:
  GLOBALDELAY=20000
  IF (PAR_3=0) then PAR_3=1   'prevent channel number 0 (not allowed)
  IF (PAR_2=0) then PAR_2=32768
    '64080 => +9.555664V  (at a voltage range of ±10V)
    '32768 => 0V
    ' 1456 => -9.555664V

EVENT:
  PAR_2 = ADC12(PAR_3)                  'read value
  FPAR_2 = FPAR_2*0.95 + PAR_2*0.05     'calculate mean value
```

7. DA Add-on

Connectors

With the DA add-on you will get six additional analog outputs with a resolution of 16 bit. Here, the DACs 3 and 4 are located on the BNC sockets OUT 3 and OUT 4. The outputs of DACs 5 to 8 are located on pins 1 to 4 of the 25-pin D-Sub socket CONN.4 (see Figure).

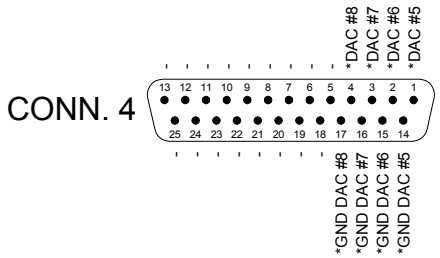


Figure 7-1: Pin assignment of the DA add-on (female)

Programming and calibration

Programming and calibration (see chapters 5 and 6 as well as the ***ADbasic*** manual) are made according to the instructions for the DACs 1 and 2. The DA add-on is located on an intermediate PCB (on the second, seen from above). The position of the trimmers can be seen in the following picture:

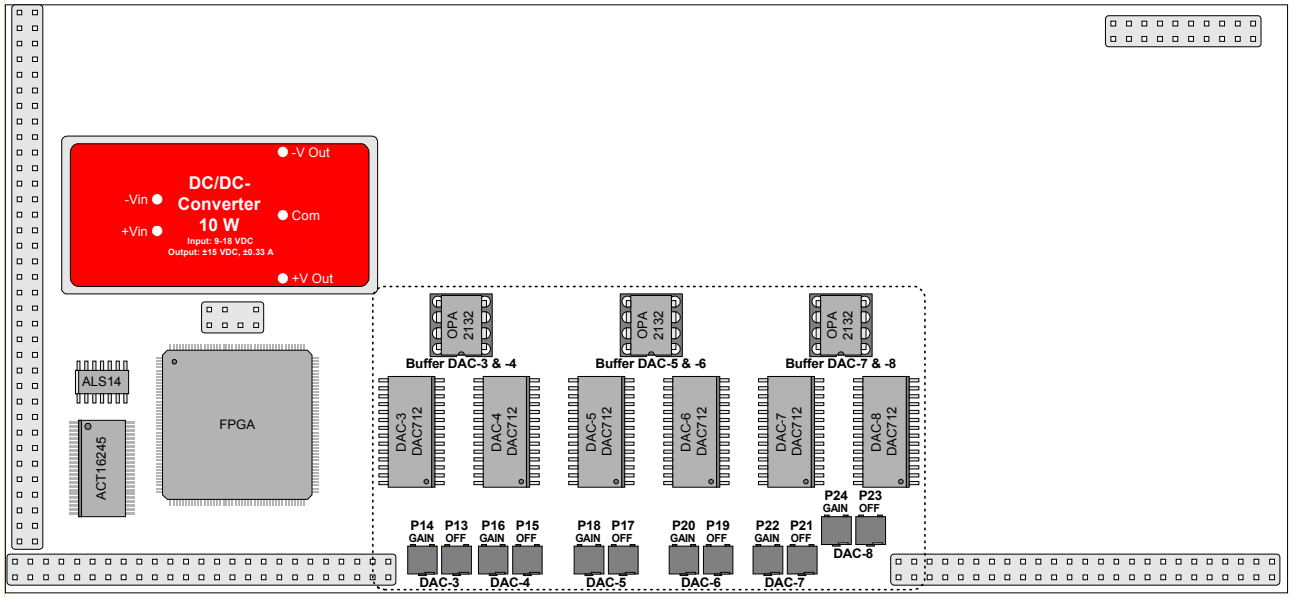


Figure 7-2: Position of the trimmers for calibration (DA add-on)

8. OPT Add-on

The OPT add-on offers:

– **optical isolation** between the ***ADwin-Gold*** system and the digital inputs and outputs.
– Three input voltage ranges of 5, 12 and 24 Volt at the digital inputs.
– an enhanced driving capability (voltage and current) at the digital outputs.

An optical isolation of the channels among each other is not possible. All digital inputs and outputs have a common reference potential (OPT-GND).

Reference potential

If you have purchased an ***ADwin-Gold*** system with OPT add-on, the digital inputs and outputs have to be configured by the instruction `CONF_DIO(12)`, which sets DIO 0...15 as inputs and DIO 16...31 as outputs.

Configuration

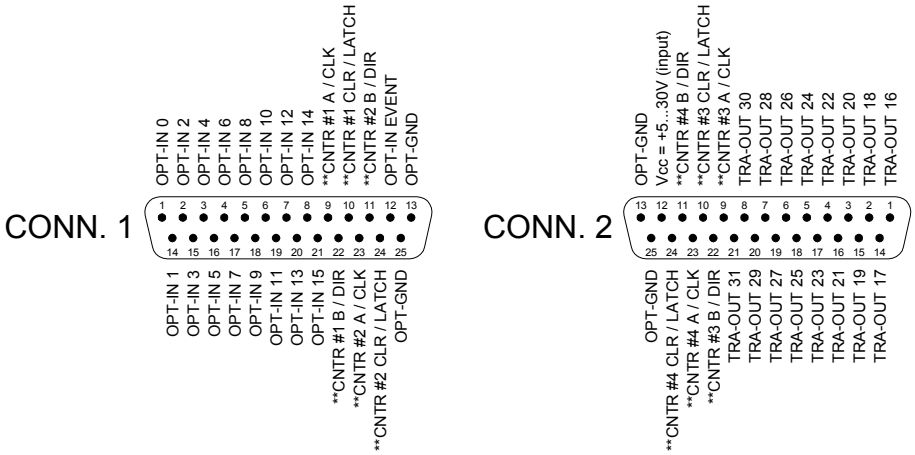


Figure 8-1: Pin assignment of the OPT add-on
(CONN. 1: male; CONN. 2: female)

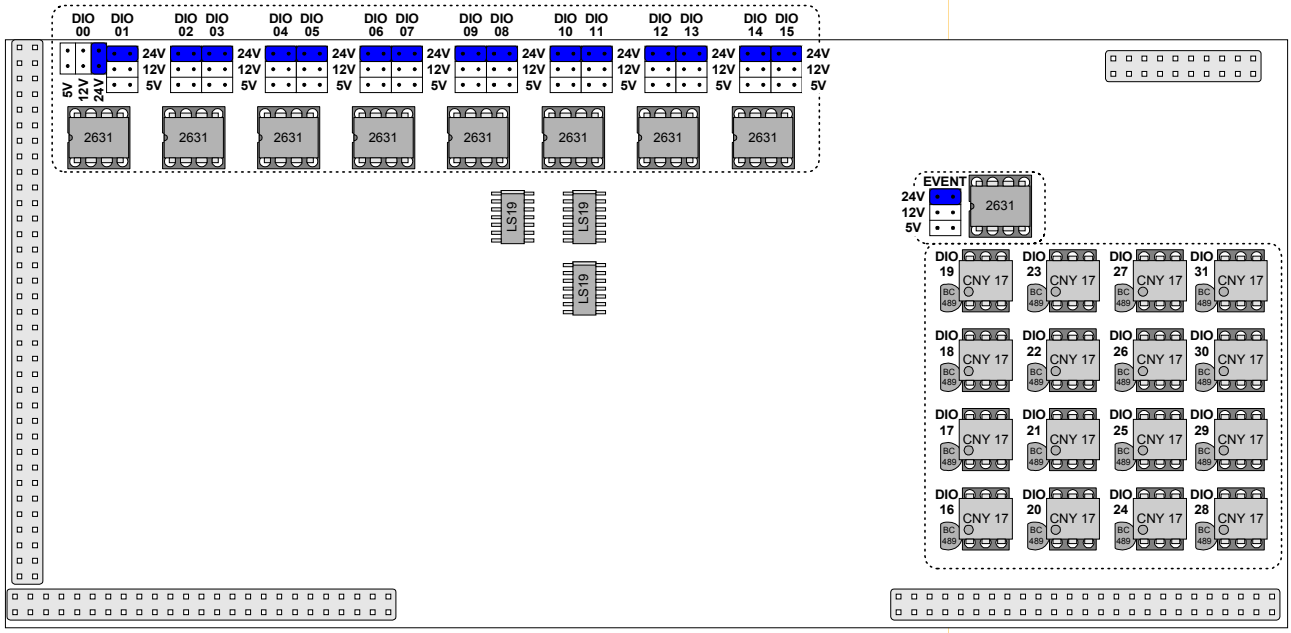
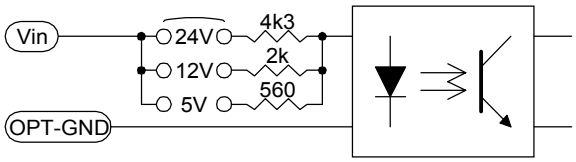


Figure 8-2: Position of the jumpers for the input voltage range (OPT add-on)

8.1 Digital Inputs

Voltage ranges

If not requested otherwise, the input voltage range of the digital inputs is set to 24 Volt. The setting can be changed by jumpers to 5 or 12 Volt (see right). The switching thresholds for the logic level can be found in the chapter about the technical features of the OPT add-on.



For adjusting the jumpers you have to open the chassis (see chapter 6.2). The position of the jumpers is shown in the picture below at left.

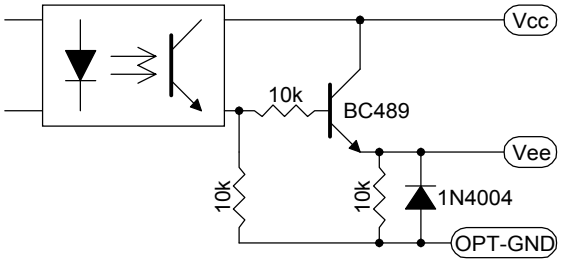
Input current

The system, which is connected to the digital input, must be capable of driving a current of at least 7 mA, in order to light up the LED in the optocoupler.

8.2 Digital Outputs

Design and configuration

After you have connected an external voltage between 5 and 30 Volt to V_{CC} (Pin 12, CONN. 2) and OPT-GND (Pin 13, CONN. 2), the output transistor will be able to supply the voltage to the relevant output (CONN. 2). The positive power rail V_{CC} is switched and not the ground rail (GND) as commonly used and known as open-collector output.



Depending on the output current I_{MAX} and the connected V_{IN} (=Vcc) the transistor causes a voltage drop between 0.7 and 1.2 Volt. An inverse diode protects the transistor during switching of an inductive load against overvoltage.

OPT add-on						
Parameters	**Symbol**	**Conditions**	**min**	**typ.**	**max**	**Unit**
Opto-coupler inputs						
Number	16					
Isolation	Ch - Ch				500	V
	Ch - GND				500	
Speed	f_{IN}	V_{IN}= 5 V		8	10	MHz
LED curent	I_{LED}				20	mA
5 V voltage range	V_{IL} (Low)				0.8	V
	V_{IH} (High)		4.5			
	I_{LED}	V_{IN}= 5 V		6.4		mA
12 V voltage range	V_{IL} (Low)				1.6	V
	V_{IH} (High)		10			
	I_{LED}	V_{IN}= 12 V		5.4		mA
24 V voltage range	V_{IL} (Low)				3.2	V
	V_{IH} (High)		20			
	I_{LED}	V_{IN}= 24 V		5.3		mA
Transistor outputs						
Number	16					
Isolation	Ch - GND				500	V
Ext. collector voltage	V_{CC}		5		30	
Output current	I_{EE}	per channel			300	mA
	$I_{CONN. 2, Pin 12}$	total			3	A
Saturation voltage	$V_{CE sat.}$	V_{CC}= 5 V			0.7	V
		V_{CC}= 12 V			0.8	
		V_{CC}= 24 V			1.2	
Slope steepness	t_{rise}*	V_{CC}= 5 V			2	µs
		V_{CC}= 12 V			3.8	
		V_{CC}= 24 V			6.7	
	t_{fall}*	V_{CC}= 5 V			32	
		V_{CC}= 12 V			29	
		V_{CC}= 24 V			26	
Power-up status	V_{EE}			0		V

* t_{rise} and t_{fall} are defined between 10% and 90% of the amplitude end points

Table 8-1: Technical data of the OPT add-on

9. CO1 Counter Add-On

9.1 Hardware

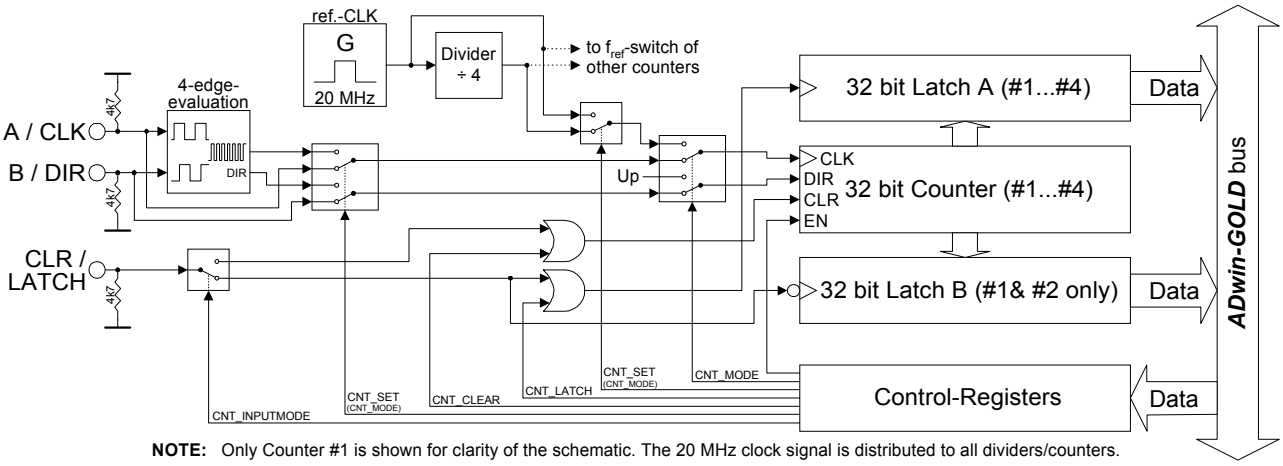
Counter

The counter add-on CO1 has **four 32 bit counters**, which you can configure and read out individually or all 4 together by software (the block diagram below shows the design of a single counter).

Latch register

The counters can be **internally or externally clocked** and are read out via accompanying latches. All counters have each a latch register A, the counters 1 and 2 have additionally a latch B.

The counter values can be cleared or transferred in a latch by using programming commands or (at special configurations) when there is an external signal at CLR/LATCH.



NOTE: Only Counter #1 is shown for clarity of the schematic. The 20 MHz clock signal is distributed to all dividers/counters.

There are the following operating modes: event counting (external clock) and pulse width measurement (internal clock); see also chapter 9.3/9.4:

External clock input

1. **Event counting**: Incrementing/decrementing of the counter is caused by external square-wave signals at the inputs A/CLK and B/DIR. A signal at CLR/LATCH has the effect that either the counter is set to zero (CLR) or that the counter values are written into the latch (LATCH).
   There are the modes:
   – **Clock and direction**: Every positive edge at CLK increments or decrements the counter values by one. The signal at DIR determines the counting direction (0 = count down; 1 = count up).
   – **Four edge evaluation**: Every edge of the signals (phase-shifted by 90 degrees) at A/CLK and B/DIR causes the counter to increment/decrement. The counting direction is determined by the sequence of the rising/falling edges of these signals. This mode is particularly used for quadrature encoders.

Internal clock input

2. **Pulse width measurement**: Incrementing/decrementing of the counter is caused by an internal reference clock generator with a signal frequency of 20 MHz (optionally 5 MHz after scaler). The square-wave signal at CLR/LATCH is evaluated: With every positive edge the counter values are written to latch A, with a negative edge to latch B.
   You can calculate:
   – the period duration of the input signal at CLR/LATCH from the values in latch A or latch B.
   – the impulse width and pause time from the values in latch A and latch B (only counters 1 and 2).

The counters are controlled by ***ADbasic*** instructions via a control register (instructions, see table in chapter 9.2).

Connect inputs

TTL-compatible signals are necessary at the inputs A/CLK, B/DIR and CLR/LATCH (exception: OPT add-on; see also the information at the end of this chapter).
More details and limit values can be found in the „Technical Data“.

Although all inputs for the CO1 add-on have a pull-down resistor, not-connected inputs can cause errors in an environment which is not protected against interferences. In this case, connect the CLR/LATCH inputs to GND, for safety reasons.

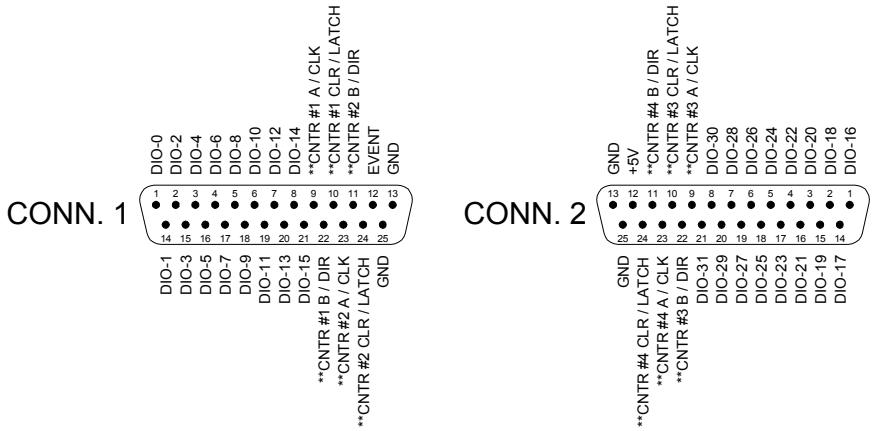


Table 9-1: Pin assignment of the CO1 add-on
(CONN. 1: male; CONN. 2: female)

9.2 Software

The functions necessary for accessing the counters can be found in the include file:

Include file

```
<ADWGCNT.INC>
```

Therefore programming has to start with the include file, so that you can use the instructions in the following table.

Counter no.	4	3	2	1	Comments
Bit	3	2	1	0	
CNT_CLEAR( )	0	0	0	0	no effect
	1	1	1	1	clear counter *
CNT_ENABLE( )	0	0	0	0	disable counter
	1	1	1	1	enable counter (note the already running counters, too)
CNT_INPUTMODE( )	0	0	0	0	set CLR/LATCH input to CLR mode
	1	1	1	1	set CLR/LATCH input to LATCH mode
CNT_LATCH( )	0	0	0	0	no effect
	1	1	1	1	write counter value into Latch A *
CNT_MODE( )	0	0	0	0	external clock input
	1	1	1	1	internal reference clock (20MHz / 5MHz)
CNT_SET( )	0	0	0	0	CNT_MODE-bit = 0 : four edge (quadrature) evaluation
					CNT_MODE-bit = 1 : internal reference clock of 20MHz
	1	1	1	1	CNT_MODE-bit = 0 : clock and direction inputs (CLK & DIR)
					CNT_MODE-Bit = 1 : internal reference clock of 5MHz
CNT_READ(#)					latch counter values into latch A and read out (# = counter no., 1...4)
CNT_READLATCH(#)					read out latch A (triggered by rising edge), (# = counter no. 1...4)
CNT_READFLATCH(#)					read out latch B (triggered by falling edge), (# = counter no. 1, 2)

* these functions are reset after they have been executed. All other functions are reset by opposing functions.

With the instructions in the table matrix you are always effecting all counters. Therefore pay attention to the fact which bits you are setting or deleting. You will be able to effect every counter individually or all together.

Sequence of instructions

Please configure the counters according to the following order:
1. disable specified counter (CNT_ENABLE)
2. if necessary, set operating mode (CNT_MODE, CNT_SET, CNT_INPUTMODE)
3. clear counter (CNT_CLEAR)
4. enable counter (CNT_ENABLE)

For further processing of the values in the ***ADbasic*** program, transfer the values into the latch register and read them out there.

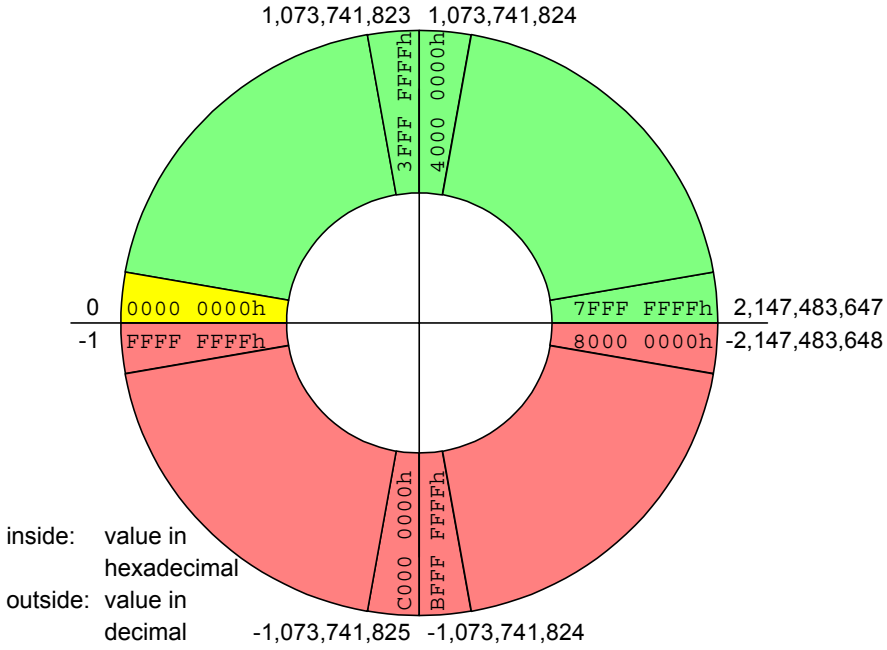
Please pay attention to the fact that the CNT_SET instruction depends on the CNT_MODE instruction.

If you disable or enable a specified counter, then you also enable the running counters (= set bits), otherwise they will be disabled.

9.2.1 Evaluation of the counter contents

The binary counters of the CO1 add-on generate 32-bit values, which are interpreted by ***ADbasic*** as numerical values according to the model of the circle below: The most significant bit (MSB) is interpreted as a sign, the highest positive number (2^{31}-1) follows the highest negative number (-2^{31}) and the lowest positive number (0) follows the highest negative number (-1).
Therefore note the following rules during programming.

Circle



Process the read 32-bit value only with variables of the type INTEGER or LONG. ***ADbasic*** then keeps internally the read bit pattern unmodified and automatically considers the transition from the positive to the negative range of numbers. Then you get:

Count direction

The count direction (up or down) can reliably be derived from the

sign of the difference: [new counter value] minus [old counter value]

and not from the comparison of the counter values.
Calculate the difference only with integer variables (INTEGER, LONG).

„Overflow“

For programming please remember that an „overflow“ between the reading out of two counts - i.e. the current counter value „laps“ the last counter value which has been read out - is not registered.
Such a lap overflow occurs after some 3½ minutes with an input frequency of 20 MHz or after more than 14 minutes with 5 MHz.

Example programs

You will find several example programs for the CO1 add-on in the directory <C:\ADwin\ADbasic3\samples_ADwin_Gold> (Standard installation).

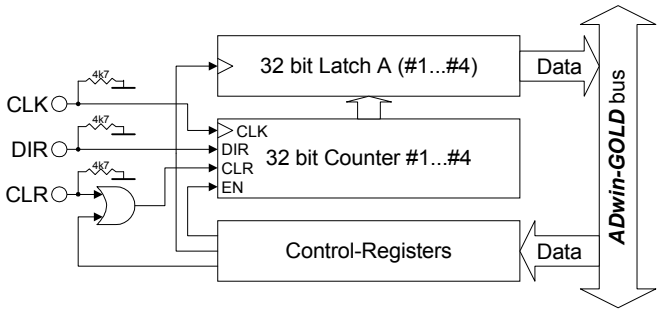
9.3 Mode Impulse/Event Counting

External square-wave signals at the inputs A/CLK and B/DIR clock the counters in this mode. With `CNT_SET` you either activate the mode for determining the clock frequency and direction or the four edge evaluation.

Clearing Latching

The input CLR/LATCH (at high-signal) can be used to
- clear the counter (CLR)
- latch the counter values into latch register A (LATCH).

9.3.1 Clock and Direction



Every positive edge of a square-wave signal at the CLK input (clock) is counted (incremented or decremented) up to a maximum frequency of 20 MHz. The direction is derived from a high signal (count up) or low signal (count down) at the DIR input (direction); This signal can be static, for a fixed count direction, or dynamic, for changing directions.

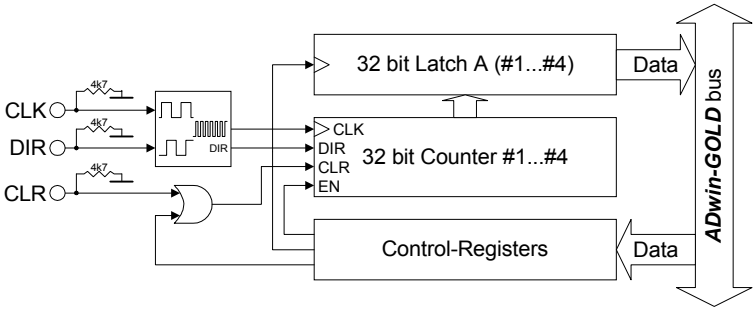
Programming example

```
INIT:                          Initialize
. . .
CNT_ENABLE(0)                  disable counter
CNT_CLEAR(1)                   clear counter
CNT_MODE(0)                    external clock input at the CLK input
CNT_SET(1)                     set mode clock and direction
CNT_INPUTMODE(0)               set input CLR/LATCH to CLR mode
CNT_ENABLE(1)                  enable counter
. . .

EVENT:
CNT_LATCH(1)                   latch current counter values to latch registerA
CNT_READLATCH(1)               read out latch register A
. . .                          evaluate counter values in program
```

9.3.2 Four Edge Evaluation

This mode determines clock and direction of two signals, which are intput phase-shifted by 90 degrees to the inputs A and B. The count direction is determined by the temporal sequence of the rising and falling edges of the two input signals.



Please note:
- The counter counts 4 edges in a cycle.
- The maximum count frequency is 20 MHz. Together with the 4 edges per cycle it will result in a maximum input frequency of 5 MHz.
- The time between an edge at A and an edge at B must not be shorter than 50 ns. Impulse widths or pause durations shorter than 100 ns are not incremented.
- Changing the phase-shift will have an effect on the maximum input frequency. If it differs from 90 degrees, the maximum input frequency of 5 MHz decreases for instance to 45 degrees at 2.5 MHz.

Programming example

`INIT:`	Initialize
`. . .`	
`CNT_ENABLE(0)`	disable counter
`CNT_CLEAR(1)`	clear counter
`CNT_MODE(0)`	external clock input at CLK input
`CNT_SET(0)`	activate mode four edge evaluation
`CNT_INPUTMODE(0)`	set input CLR/LATCH to CLR mode
`CNT_ENABLE(1)`	enable counter
`. . .`	

`EVENT:`	
`CNT_LATCH(1)`	latch current counter values to latch register A
`CNT_READLATCH(1)`	read out latch register A
`. . .`	evaluate counter values in program

9.4 Mode Pulse Width and Period Width Measurement

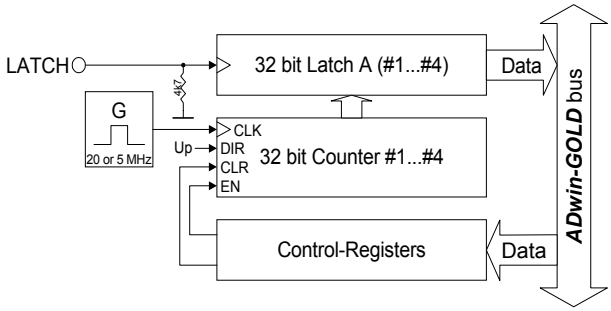
Reference clock generator

In this operating mode an internal reference clock generator clocks the counter with a signal frequency of 20 MHz or (after a prescaler) 5 MHz. All counters have a switch in order to change the signal frequency. The period duration or pulse width of a square-wave signal at input CLR/LATCH can be measured.

In this mode you have to consider at high frequencies that your `GLOBALDELAY` remains smaller than a signal period, in order to acquire a cycle.

9.4.1 Period Duration Measurement

All four counters can execute period duration measurements.



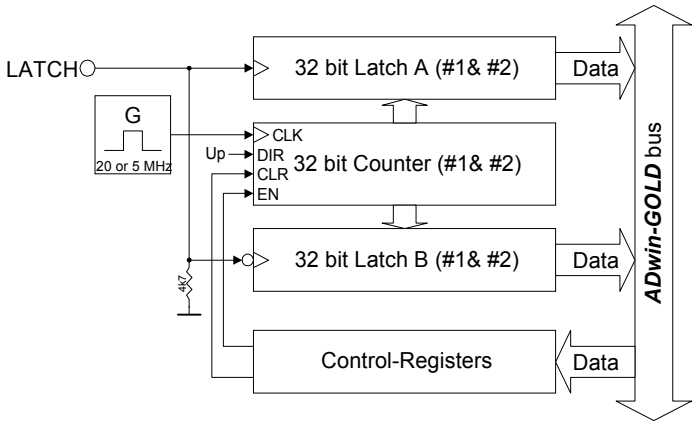
In this mode, the counter values are latched into latch A at every positive edge, and the previous data are overwritten. The pulse width will be derived from the counter value difference multiplied by the period duration of the reference clock.

Programming example

```
INIT:                          Initialize
. . .
CNT_ENABLE(0)                  disable counter
CNT_CLEAR(1)                   clear counter
CNT_MODE(1)                    mode internal reference clock at internal CLK
                               input of the counter ...
CNT_SET(0)                     ... with 20 MHz or
    CNT_SET(1)                 ... with 5 MHz
CNT_INPUTMODE(1)               set input CLR/LATCH to LATCH-mode
CNT_ENABLE(1)                  enable counter
. . .
EVENT:
CNT_READLATCH(1)               read out latch register A
. . .                          evaluate counter values in program
```

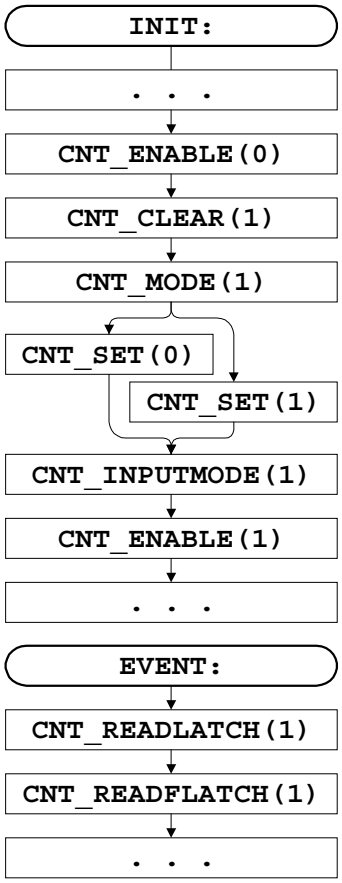
9.4.2 Impulse Width and Pause Duration Measurement

The measurement of impulse width and pause duration can be made with **counters 1 and 2** only .



The counters 1 and 2 have two latches for positive (latch A) and negative edges (latch B). Thus, pulse and pause duration can be evaluated separately by calculating the differences of the latches.

Programming example



Initialize

disable counter

clear counter

mode internal reference clock at internal CLK-input of the counter ...

... with 20 MHz or

... with 5 MHz

set input CLR/LATCH to LATCH-mode

enable counter

read out latch register A

read out latch register B

evaluate counter values in program

9.5 Hardware Addresses (CO1 Add-On)

A process can be executed very quickly if you access directly the control and data register (see chapter 5.3 as well as ***ADbasic*** manual). The hardware addresses of the CO1 add-on can be found in the following table (see command index table in chapter 9.2).

Address [hex]	Function	Bit											Comments
		31-16	15-10	9	8	7	6	5	4	3	2	1	0	
20400204	read out latch A: counter-#1	x	x	x	x	x	x	x	x	x	x	x	x	x : contents of latch
20400208	read out latch B: counter-#1	x	x	x	x	x	x	x	x	x	x	x	x	x : contents of latch
20400214	read out latch A: counter-#2	x	x	x	x	x	x	x	x	x	x	x	x	x : contents of latch
20400218	read out latch B: counter-#2	x	x	x	x	x	x	x	x	x	x	x	x	x : contents of latch
20400224	read out latch A: counter-#3	x	x	x	x	x	x	x	x	x	x	x	x	x : contents of latch
20400234	read out latch A: counter-#4	x	x	x	x	x	x	x	x	x	x	x	x	x : contents of latch
20400300	enable counter	-	-	-	-	-	-	-	-	x	x	x	x	x = 0 : disable counter x = 1 : enable counter
20400310	clear counter	-	-	-	-	-	-	-	-	x	x	x	x	x = 0 : no effect x = 1 : clear counter
20400320	latch counter	-	-	-	-	-	-	-	-	x	x	x	x	x = 0 : no effect x = 1 : latch counter
20400330	Input: CLR or LATCH	-	-	-	-	-	-	-	-	x	x	x	x	x = 0 : CLR input x = 1 : LATCH input
20400340	Impulse/event counter or pulse width/period duration measurement	-	-	-	-	-	-	-	-	x	x	x	x	x = 0 : ext. clock input x = 1 : int. ref. clock (20MHz / 5MHz)
20400350	4 edge evaluation/CLK+DIR or 20 MHz/5 MHz reference clock	-	-	-	-	-	-	-	-	x	x	x	X	CNT_MODE=0: x=0: 4-Fl.; x=1: CLK+DIR CNT_MODE=1: x=0: 20 MHz; x=1: 5 MHz

Table 9-2: Hardware addresses of the CO1 add-on

9.6 Technical Data (CO1 Add-On)

CO1 counter add-on						
Parameters	Symbol	Conditions	min.	typ.	max.	Unit
TTL inputs* (with Schmitt-trigger, with opto-couplers see data OPT add-on),						
edge recognition, pos.	V_{T+} (Low)	V_{CC}= 5 V	1.65	1.9	2.15	V
edge recognition, neg.	V_{T-} (High)	V_{CC}= 5 V	0.75	1.0	1.25	
switching hysteresis	V_{T+} - V_{T-}		0.4	0.9		
input current	I_{IH}	V_I= 2.7 V			20	µA
	I_{IL}	V_I= 0.4 V			-50	
reference crystal oscillator						
reference frequency	f_{ref}			20		MHz
prescaler by 4	$f_{ref/4}$			5		
accuracy and drift					100	ppm
counter						
counter width				32		Bit
count frequency	f_{CLK}	input CLK		20		MHz
		input A/B		5		
count frequency with OPT add-on (V_{IN} = 5V)	f_{CLK}	input CLK		8	10	
		input A/B		5		
latch width	LATCH			32		Bit

* see also data sheet for 74LS19 from TI

Table 9-3: Technical data of the CO1 add-on

9.7 CO1 Add-On with opto-couplers

If your ***ADwin-Gold*** system is equipped with an OPT add-on, then the inputs of the CO1 add-on have also opto-couplers.
The notes given in the chapter 8: „OPT add-on“ apply also for the CO1 add-on.

Depending on the counter mode the maximum input frequencies are changing (see chapter 9.6 „Technical Data“ (CO1 add-on).

10. *ADwin-Gold-Boot*

ADwin-Gold-Boot starts a previously programmed application, automatically after power-up. After installation of this application an **operation without PC** is possible.

With ***ADwin-Gold-Boot*** the following steps are executed after power-up:

- Loading the operating system
- Loading the processes built with the ***ADbasic*** compiler (max. 10)
- Automatic starting of the **process no. 10.** Here you have also to program the start of all other processes.

Disable bootloader

If you do not wish to work with the bootloader option:

Boot the system after power-up and the previously saved processes are disabled.
After switching off and powering up anew, the bootloader option is enabled again.

As memory a Flash-EPROM with the size of 512 kByte is used.

With the installation of the ***ADwin*** *Developer-Software* from the supplied ***ADwin*** CD-ROM, the utility programs for the bootloader are automatically copied.

The version of the ***ADwin*** CD-ROM should be 3.00.2735 or higher.

Link or Ethernet ?

The utitliy programs for the bootloader option are different, depending on the interface. Use the program <`ADbootload.exe`> for an ***ADwin*** system with link interface, and the program <`ADethflash.exe`> with an Ethernet interface.

At standard installation you will find the **utility programs and help files** in the main path <`C:\ADwin\Tools\`> in the directories

Link:	<`ADbootload\..`>
Ethernet:	<`Ethernet Interface\..`>

Help file for the link

For the bootloader with link interface you will find a help file called <`ADBOOTLOAD.HLP`>. Install the bootloader option according to the description given here. At the end of the file there is an example program.

Help file for the Ethernet

Notes for the bootloader with Ethernet interface can be found in the ***ADwin*** driver installation.

11. Accessories

The following accessories are available for the ***ADwin-Gold***:

– ***ADwin-Gold-pow***: external 12-V dc power supply
– various power supply and link cables
– cable connector for an external power supply

ADwin-Gold-pow

On the secondary side ***ADwin-Gold-pow*** provides 12 Volt at a maximum load of 2 Ampere. The power supply unit is rated for the highest load and maximum expansions of the ***ADwin-Gold***.

***ADlink* cable**

For larger distances *ADlink* cables with 5 m can be provided. All *ADlink* cables can also be delivered with a shielding on one end only.

Please pay attention to the fact that the *ADlink* cable is sufficiently shielded, in order to avoid interferences in the data lines. Interferences have to be passed before entering the chassis via GND (ground) (see also chapter 3).

Cable connector

In case you want to use an external power supply, you need the cable connector for the correct connection to the ***ADwin-Gold.***

Annex

A-1 Technical Data - General Overview

All technical data refer to a powered-up ***ADwin-Gold*** system.

General Data/Limit Values						
	Symbol	Conditions	min.	typ.	max.	Unit
Supply Voltage/Supply Current						
voltage	U_b		10	12	18	V
idle current	I_{idle}	U_b=10 V		0.9		A
		U_b=12 V		0.8		
		U_b=18 V		0.6		
Gold System		U_b=12 V		0.8		
Gold + DA Add-on				1.2		
Gold + other Add-ons				0.8		
inrush current	$I_{power-on}$	U_b=12 V				
Gold system			1.5			
Gold + DA add-on			2.5			
Gold + other add-ons			1.5			
Operation						
chassis temperature	T_{case}		0		+60	°C
relative humidity	F_{rel}	no condensation	0		90	%
Storage						
temperature	T		-20		+70	°C
dimensions (as desktop unit)						
width	w			214		mm
height	h			67		
		(with DA- or OPT)		97		
depths	d			109		
Mounting						
standard	desktop unit					
optional	DIN rail mounting and wall mounting					
Weight						
net	m_{net}					g
Gold System				1,320		
Gold + DA add-on				1,760		
Gold + other add-ons				1,320		
+ clips*				32		
* Clips for DIN rail mounting available as accessories for ***GOLD-Mount***.						

Analog Inputs/Outputs						
Parameters	Symbol	Conditions	min.	typ.	max.	Unit
inputs						
number	2x 8 (via MUX) differential or single-ended (individually selectable by jumpers for each of the inputs)					
input resistance*	R_i		323.4	330	336.6	kΩ
overvoltage	$U_{in\ max.}$	powered-up		±15**	±17**	V
overvoltage		switched off		0**	±2**	
MUX settling time	t_{MUX}	1 LSB 12-bit		1.5		µs
		1 LSB 16-bit		6.5		
* only if the system is powered-up. If the system is switched off the input has very low impedance. **and max. 20 mA are allowed						
ADC 12-bit						
conversion time	t_{conv}				0,8	µs
measurement range (k = gain factor)	U_{in}	k = 1	-10		+9.995117	V
		k = 2	- 5		+4.997559	
		k = 4	- 2.5		+2.498779	
		k = 8	- 1.25		+1.24939	
diff. common mode voltage					±2.5	
integral non-linearity	INL			±0.3	±1	LSB
differential non-linearity	DNL			±0.3	±1	
offset	drift				±15	ppm/°C
	error	adjustable				
gain	drift				±15	ppm/°C
	error	adjustable				
ADC 16-bit						
conversion time	t_{conv}				10	µs
measurement range (F_G = gain factor)	U_{in}	F_V = 1	-10		+9.999695	V
		F_V = 2	- 5		+4.999847	
		F_V = 4	- 2.5		+2.499924	
		F_V = 8	- 1.25		+1.249962	
diff. common mode voltage					±2.5	
integral non-linearity	INL			±1	±3	LSB
differential non-linearity	DNL			±0.25	±0.5	
offset	drift			±2		ppm/°C
	error	adjustable				
gain	drift			±5		ppm/°C
	error	adjustable				

Outputs DAC 16-bit						
Parameters	Symbol	Conditions	min.	typ.	max.	Unit
number	2 (with DA add-on 8)					
output voltage	U_{out}		-10		+9.999695	V
settling time	t_{settle}	2 V step		3		µs
		FSR* (20 V)		10		
maximum current				±25		mA
integral non-linearity	INL				±2	LSB
differential non-linearity	DNL				±1	
offset	error	adjustable				
gain	error	adjustable				
*FSR = Full Scale Range						

Digital Inputs/Outputs						
Parameters	Symbol	Conditions	min.	typ.	max.	Unit
I/O-lines						
number	DIO00:DIO31	32 (programmable in groups of 8 as inputs or outputs)				
	EVENT	ext. Trigger Input (pos. TTL-Logic)				
as inputs *						
max. input voltage		TTL-Level	-0,5		+5.5	V
logic-input voltage	V_{IH} (high)	V_{CC} = 5 V	2			
	V_{IL} (low)	V_{CC} = 5 V			0.8	
logic-input current	I_I	V_{CC} = 5 V		±0.01	±5	µA
as outputs *						
logic-output voltage	V_{OH} (high)	I_{OH} = -6 mA	3.84	4.3		V
	V_{OL} (low)	I_{OL} = +6 mA		0.17	0.33	
logic-output current	I_O				±35	mA
	I_{TOTAL}				±70	
* see also data sheet for SN74HCT245 from Texas Instruments						

Processor: ADSP21062 (SHARC™) from Analog Devices						
Parameters	Symbol	Conditions	min.	typ.	max.	Unit
CPU						
type	ADSP21062 (SHARC™)					
manufacturer	Analog Devices					
clock frequency	f_{CLK}			40		MHz
register width				32		Bit
internal memory (SRAM)		for program		128	256*	kB
		for data		128	256*	
external memory (DRAM)				4	16 / 32 **	MB
* with ADSP 21060 (*ADwin*-G-MEM 512 k) ** with memory expansion (*ADwin*-G-MEM/16 or. .../32)						

A-2 Hardware Addresses - General Overview

Address [HEX]	Function	Bit 31-16	15-10	9	8	7	6	5	4	3	2	1	0	Comments
20400000	set MUX-#1: channels 1, 3, 5, ..., 15	-	-	-	-	-	-	-	-	-	n	n	n	"nnn" binary = 0...7 decimal, selected channel = nnn + 1
	set MUX-#2: channels 2, 4, 6, ..., 16	-	-	-	-	-	-	n	n	n	-	-	-	"nnn" binary = 0...7 decimal, selected channel = 2(nnn + 1)
	gain PGA-#1	-	-	-	-	g	g	-	-	-	-	-	-	"gg" binary = 0...3 decimal, selected gain = 2^{gg}
	gain PGA-#2	-	-	g	g	-	-	-	-	-	-	-	-	
20400010	start conversion: ADC-#1 (16-bit)	-	-	-	-	-	-	-	-	-	-	-	s	s = 0 : start conversion s = 1 : no effect
	start conversion: ADC-#2 (16-bit)	-	-	-	-	-	-	-	-	-	-	s	-	
	start conversion: all DACs synchronically	-	-	-	-	-	-	-	-	-	s	-	-	
	start conversion: ADC-#1 (12-bit)	-	-	-	-	-	-	-	-	s	-	-	-	
	start conversion: ADC-#2 (12-bit)	-	-	-	-	-	-	-	s	-	-	-	-	
20400020	EOC status: ADC-#1 (16-bit)	-	-	-	-	-	-	-	-	-	-	-	e	e = 0 : end of conversion e = 1 : conversion is running
	EOC status: ADC-#2 (16-bit)	-	-	-	-	-	-	-	-	-	-	e	-	
	EOC status: ADC-#1 (12-bit)	-	-	-	-	-	-	-	-	e	-	-	-	
	EOC status: ADC-#2 (12-bit)	-	-	-	-	-	-	-	e	-	-	-	-	
20400030	read out register: ADC-#1 (16-bit)	-	x	x	x	x	x	x	x	x	x	x	x	x : result of the conversion
20400040	read out register: ADC-#2 (16-bit)	-	x	x	x	x	x	x	x	x	x	x	x	
20400050	only write into register: DAC-#1	-	x	x	x	x	x	x	x	x	x	x	x	x : digital value to be converted
20400060	only write into register: DAC-#2	-	x	x	x	x	x	x	x	x	x	x	x	
20400070	only write into register: DAC-#3	-	x	x	x	x	x	x	x	x	x	x	x	
20400080	only write into register: DAC-#4	-	x	x	x	x	x	x	x	x	x	x	x	
20400090	only write into register: DAC-#5	-	x	x	x	x	x	x	x	x	x	x	x	
204000A0	only write into register: DAC-#6	-	x	x	x	x	x	x	x	x	x	x	x	
204000B0	input registers DIO 00 to DIO 15	-	x	x	x	x	x	x	x	x	x	x	x	x : read digital value
204000C0	output register DIO 16 to DIO 31	-	x	x	x	x	x	x	x	x	x	x	x	x : digital value to be output
20400100	read out register and start conversion: ADC-#1 (16-bit)	-	x	x	x	x	x	x	x	x	x	x	x	x : digital value to be converted
20400110	read out register and start conversion: ADC-#2 (16-bit)	-	x	x	x	x	x	x	x	x	x	x	x	
20400120	read out register and start conversion: ADC-#1 (12-bit)	-	x	x	x	x	x	x	x	x	x	x	x	
20400130	read out register: ADC-#1 (12-bit)	-	x	x	x	x	x	x	x	0	0	0	0	x : result of the conversion
20400140	read out register: ADC-#2 (12-bit)	-	x	x	x	x	x	x	x	0	0	0	0	
20400190	only write into register: DAC-#7	-	x	x	x	x	x	x	x	x	x	x	x	x : digital value to be converted
204001A0	only write into register: DAC-#8	-	x	x	x	x	x	x	x	x	x	x	x	

Address [HEX]	Function	Bit											Comments
		31-16	15-10	9	8	7	6	5	4	3	2	1	0	
204001B0	input register DIO 16 to DIO 31	-	x	x	x	x	x	x	x	x	x	x	x	x : read digital value
204001C0	output register DIO 00 to DIO 15	-	x	x	x	x	x	x	x	x	x	x	x	x : digital value to be output
204001D0	read out register and start conversion: ADC-#2 (12-bit)	-	x	x	x	x	x	x	x	x	x	x	x	x : digital value to be converted
204001E0	configure DIO 00 to DIO 07	0	0	0	0	0	0	0	0	-	-	-	c	c = 0: inputs; c = 1: outputs
	configure DIO 08 to DIO 15	0	0	0	0	0	0	0	0	-	-	c	-	
	configure DIO 16 to DIO 23	0	0	0	0	0	0	0	0	-	c	-	-	
	configure DIO 24 to DIO 31	0	0	0	0	0	0	0	0	c	-	-	-	
20400200	write into register and start conversion immediately: DAC-#1	-	x	x	x	x	x	x	x	x	x	x	x	x : digital value to be converted
20400204	read out latch A: counter-#1	x	x	x	x	x	x	x	x	x	x	x	x	x : contents of the latch
20400208	read out latch B: counter-#1	x	x	x	x	x	x	x	x	x	x	x	x	x : contents of the latch
20400210	write into register and start conversion immediately: DAC-#2	-	x	x	x	x	x	x	x	x	x	x	x	x : digital value to be converted
20400214	read out latch A: counter-#2	x	x	x	x	x	x	x	x	x	x	x	x	x : contents of the latch
20400218	read out latch B: counter-#2	x	x	x	x	x	x	x	x	x	x	x	x	x : contents of the latch
20400220	write into register and start conversion immediately: DAC-#3	-	x	x	x	x	x	x	x	x	x	x	x	x : digital value to be converted
20400224	read out latch A: counter-#3	x	x	x	x	x	x	x	x	x	x	x	x	x : contents of the latch
20400230	write into register and start conversion immediately: DAC-#4	-	x	x	x	x	x	x	x	x	x	x	x	x : digital value to be converted
20400234	read out latch A: counter-#4	x	x	x	x	x	x	x	x	x	x	x	x	x : contents of the latch
20400240	write into register and start conversion immediately: DAC-#5	-	x	x	x	x	x	x	x	x	x	x	x	x : digital value to be converted
20400250	write into register and start conversion immediately: DAC-#6	-	x	x	x	x	x	x	x	x	x	x	x	
20400260	write into register and start conversion immediately: DAC-#7	-	x	x	x	x	x	x	x	x	x	x	x	
20400270	write into register and start conversion immediately: DAC-#8	-	x	x	x	x	x	x	x	x	x	x	x	
20400300	enable counter	-	-	-	-	-	-	-	-	x	x	x	x	x = 0 : disable counter x = 1 : enable counter
20400310	clear counter	-	-	-	-	-	-	-	-	x	x	x	x	x = 0 : no effect x = 1 : clear counter
20400320	latch counter	-	-	-	-	-	-	-	-	x	x	x	x	x = 0 : no effect x = 1 : latch counter
20400330	input: CLR or LATCH	-	-	-	-	-	-	-	-	x	x	x	x	x = 0 : CLR input x = 1 : LATCH input
20400340	impulse/event counter or pulse width/period duration measurement	-	-	-	-	-	-	-	-	x	x	x	x	x = 0 : ext. clock input x = 1 : int. ref. clock (20 MHz / 5MHz)
20400350	4 edge evaluation/CLK+DIR or 20 MHz / 5 MHz reference clock	-	-	-	-	-	-	-	-	x	x	x	x	CNT_MODE=0: x=0: 4-Fl.; x=1: CLK+DIR CNT_MODE=1: x=0: 20 MHz; x=1: 5 MHz

A-3 List of common abbreviations

A/D	Analog to digital
ADC	Analog to digital converter
ADSP	Analog Devices signal processor
b	Binary number (trailing)
CLK	Clock
CLR	Clear
CMOS	Complementary metal oxide semiconductor
CMRR	Common mode rejection ratio
D/A	Digital to analog
DAC	Digital to analog converter
DIL	Dual inline
DIO	Digital input / output
DIR	Direction
DMA	Direct memory access
DMM	Digital multi-meter
DNL	Differential non linearity
DRAM	Dynamic random access memory
DSP	Digital signal processor
EOC	End of conversion
EMC	Electromagnetic compatibility
ESD	Electrostatic discharge
FPGA	Field programmable gate array
FSR	Full scale range
GND	Ground
h / Hex	Hexadecimal number (trailing)
I/O	Input/Output
IC	Integrated circuit
InAmp	Instrumentation amplifier
INL	Integral non linearity
IRQ	Interrupt request
kB	kilo-Byte (= 1024 Byte)
kByte	see kB
LED	Light emitting diode
LSB	Least significant bit
MB	Mega-Byte (= 1024 kB)
MByte	see MB
MUX	Multiplexer
OpAmp	Operational amplifier
PC	Personal computer
PGA	Programmable gain amplifier
S&H	Sample & Hold
SRAM	Static random access memory
TTL	Transistor-transistor logic
Vcc	Voltage collector-collector
Vee	Voltage emitter-emitter
U/D	Up/Down

Manufacturers

AD	Analog Devices
BB	Burr-Brown
LT	Linear Technology
TI	Texas Instruments

A-4 List of Figures

A-5 List of Tables