KE-SYSTEM

WAVE AD

Building Extension Moduls

Version 1.1

Safety Instructions

This device fulfills the EU regulations 2004/108/EG (CE-Identification).
The following safety instructions for the operation of the device have to be strictly followed in order to guarantee operational safety of the device and to avoid injuries by voltage or current.
Any liability is excluded for damages caused by the user not following any of these instructions.

- Do not operate the device on moist or wet undergrounds or after the device has come in contact with moistness or wet conditions.
- Do not operate the device in close vicinity to strong magnetic fields (motors, transformers, etc.).
- Do not operate the device by any other voltage source than the USB Interface (5V).
- Do not connect signal outputs to points under voltage.
- Operate signal inputs (trigger inputs) strictly within the specified voltage limits.
- Do not operate the device if damages are visible or the device isn't completely closed.
- WaverAD is designed for operation at room temperature and normal humidity conditions. Exceeding these operational conditions can lead to malfunction or in extreme cases even to damages to the device.
- The device does not contain any components that can be serviced by the customer or any non-authorized repair service. Neither of the two safety seals on the bottom side of the device must be damaged.
- Any warranty expires, if the customer or an unauthorized service provider conducts any kind of intrusion and/or repair without prior, explicit and written approval from KE-System. Also we do not grant warranty for defects or damages caused by inappropriate or incorrect use, neglect of usage instructions or wrong or negligent handling.

KE-System
Ahornstraße 15a
86774 Unterföhring
www.ke-system.de

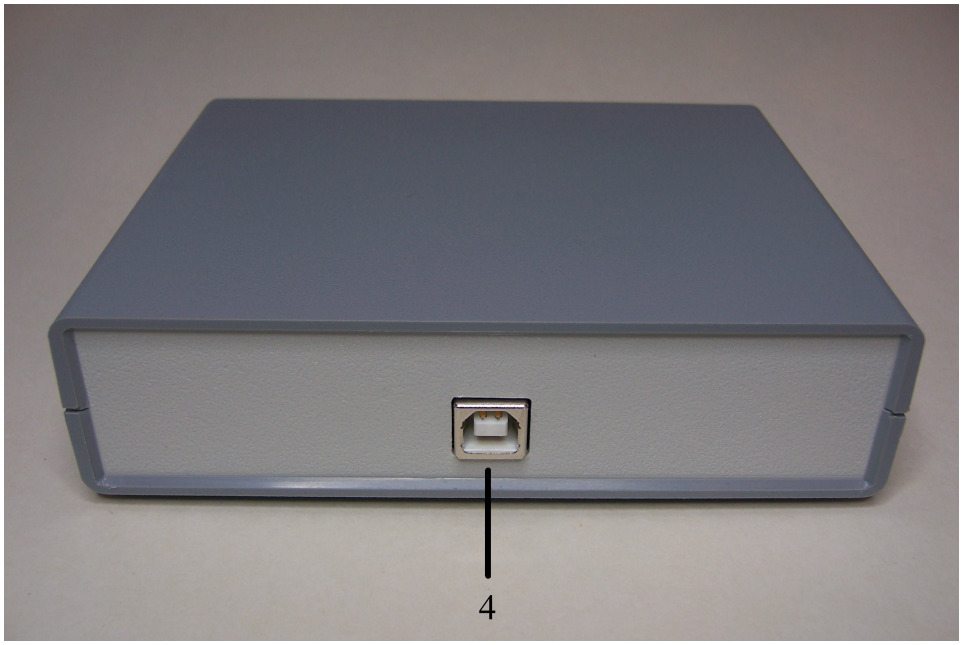
Contents

WaverAD Hardware

Front View

1: Analog Interface (Channel 0)
2: Digital Interface (Channel 0)
3: Digital Interface (Channel 1)

Rear View



4: USB Connection

WaverAD Overview

WaverAD is an arbitrary signal-generator system built for people who are looking for full flexibility however cannot spend huge amounts of well-earned money for a full-featured high-end signal generation system.

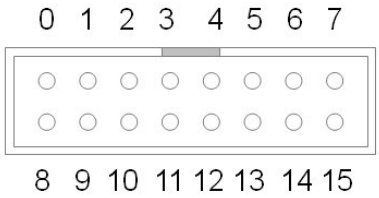
Here are some basic facts about WaverAD:

- Arbitrary Waveform-Generator and Logic-Pattern-Generator in one
- USB Peripheral for Windows PCs
- Up to 100MHz sampling rate
- Frequency, Amplitude, Offset online adjustable
- 3,3V and 5,0V logic level integrated - more possible via extensions
- 8 pages for different individually programmable signals, online switchable
- Single Step, Single Run and Continuous Run Modes
- Manually and externally triggerable
- Intuitive Software-Interface with a powerful waveform calculator
- Multiple WaverAD can be used in parallel to multiply number of channels
- Extendable via external Extension modules

Experienced users can build their own extension modules and hence adjust WaverAD very closely to their own needs. This documentation is intended to be a guideline for such cases.

Extension Module Interface

Apart from the basic functionality to output the 8 bit of a digital signal, the two digital output interfaces of the WaverAD device offer extended functionality.
The following picture shows a detailed pin-out of the two interfaces.

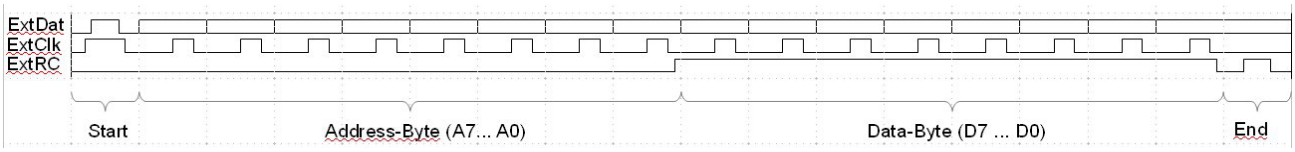


Pin-Number	Pin-Name	Pin-Type	Description
0 ... 7	D0 ... D7	Out	Digital Interface Output (bit lines 0 ... 7)
8	Trig	In	External Trigger Input
9	ExtRC	Out	Extension Module RC
10	ExtClk	Out	Extension Module Clk
11	ExtDat	Out	Extension Module Dat
12	5VOut	Out	5V Output
13	DLOut	Out	Digital Level Output
14	GND	-	Digital GND
15	DClk	Out	Sample Frequency Output (as set in software)

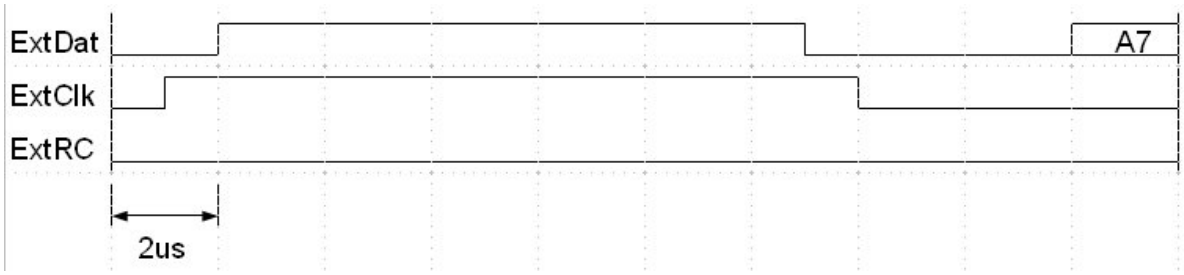
Three of the pins in the pin-out of the each interface build a communication interface by which external extension modules can be controlled by the WaverAD PC Software.
Extension modules can be addressed allowing multiple modules being operated serially or in parallel. To enable this, every extension module can be programmed with a specific address, just as each user-defined control element in the PC software. Single software elements control only those extension modules that are programmed to the same address as the software element.
Using a software control element during operation, an address byte followed by a data byte is sent to every connected hardware extension module. The extension modules will analyze the address byte and execute the data byte accordingly.

Command Protocol

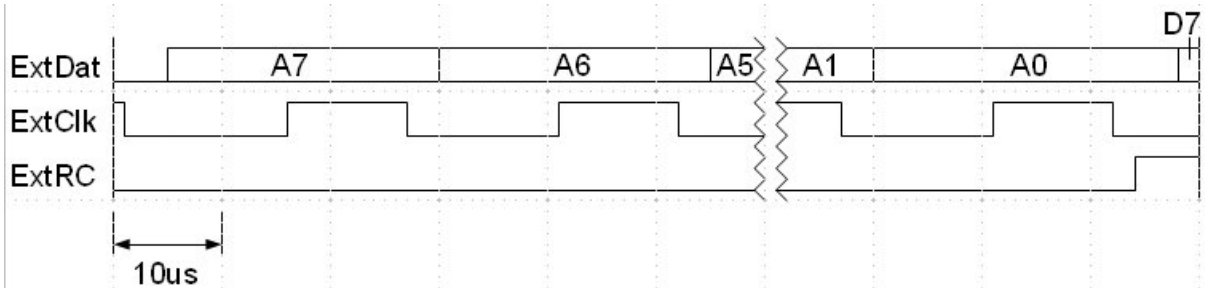
The control interface towards the external extension modules was designed to allow even slowly clocked processors or normal 8-bit serial-in/parallel-out shift registers to be used for analysis of the data stream.
A command sequence has 4 phases:



Start-Phase:
The start phase synchronizes the receiving extension module to the upcoming command transmission. This is established by a level change of the signal ExtDat while ExtClk is set high.

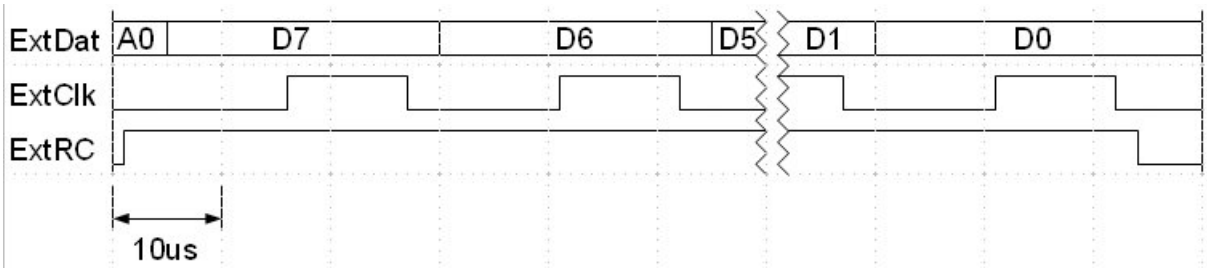


Address-Phase:
During the address phase the address byte is sent to the extension modules. The transmission is MSB first.

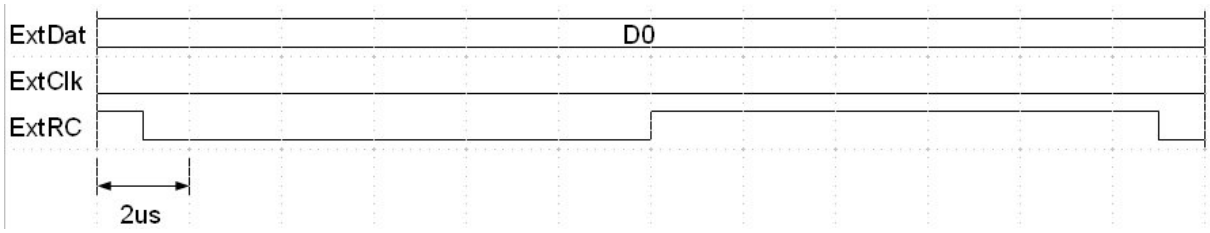


Allowable addresses for extension modules reach from 1 to 254.
Address 0 (b00000000) is a reserved address.
Address 255 (b11111111) is a special case. This one is used as a command to assign an address to the extension modules connected to WaverAD, in order for them to be identifiable during later use. The address to be assigned is transferred in the Data-Phase following this command.

Data-Phase:
During the data phase the address byte is sent to the extension modules. The transmission is MSB first.



End-Phase:
The end phase closes the command sequence. The closing pulse on ExtRC can be used to latch the serially received data byte into a parallel register (e.g. using a SN74xx595).



It needs to be noticed that the signals ExtDat and ExtRC can randomly toggle outside of a command sequence. ExtClk is guaranteed to toggle only during a command sequence in the specified way.

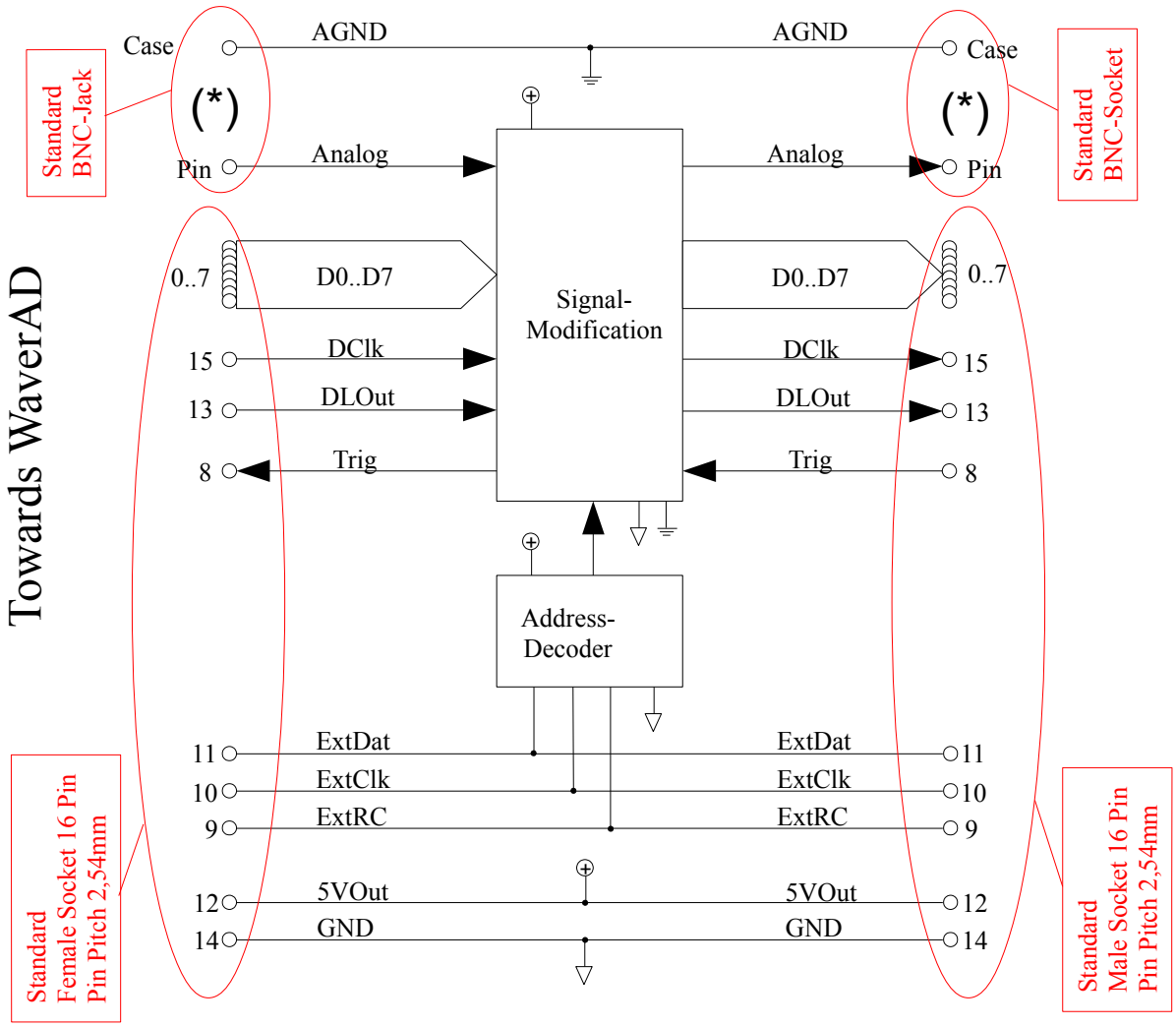
Building Extension Modules

In general, all extension modules should be built with the idea behind, that it can be cascaded with other modules. In order to reach this, it needs to be taken care that all extension modules offer output interfaces which are compatible to the ones from WaverAD. It is necessary that the pin-out of the output interface is identical to the pin-out of the WaverAD interface. Especially the following pins of the interface should be directly routed from input to output through the extension module, eventually buffering them:

- GND
- 5VOut
- ExtRC
- ExtClk
- ExtDat

Depending on the functionality of the extension modules, also other signal lines that are not intended to be modified by the extension module, should be routed from input to output.

Following the above, the general function should look like shown in the below:



Depending on the function of the extension module, not all of the shown signals need to be routed through the modification block. Such signals should then be routed directly from input to output of the module, eventually buffering them.

(*) Even though the analog signal of the WaverAD is not part of the digital interface of the WaverAD, extension modules can be built, which are controlled over the digital interface, but modify the analog signal (BNC interface). In this case, a separation of the analog GND (AGND, Steel case of the BNC socket) and the digital GND (GND, Pin 14 of the digital interface) is recommended.

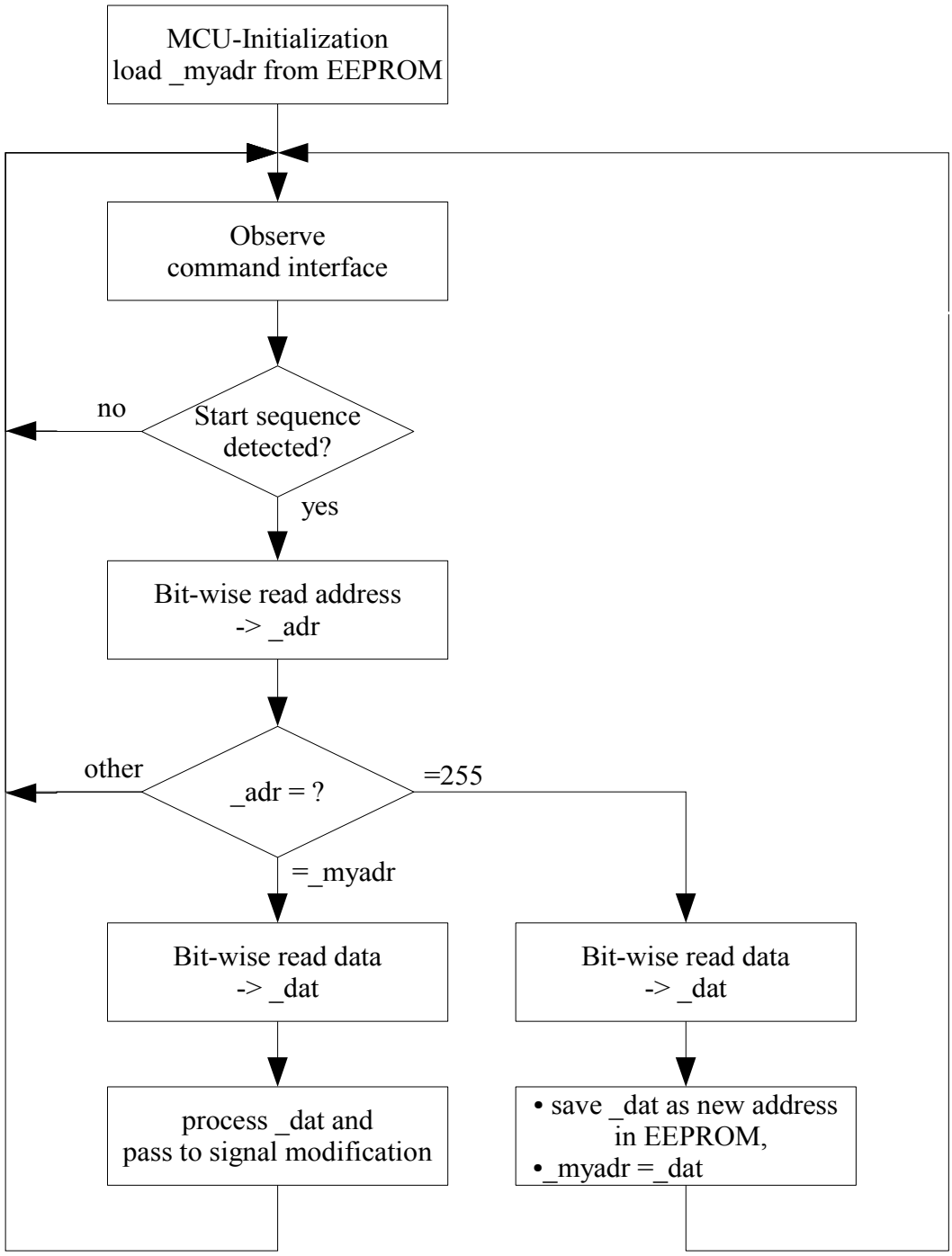
WaverAD uses the following output interfaces.
Digital Interface: Standard male socket, 16 Pin, pin-pitch 2,54mm
Analog Interface: Standard BNC socket, 50 Ohm

Address-Decoder

The address decoder observes the command bus for the extension modules. It reads all transmissions on this bus and verifies whether the address that has been transmitted during an address phase equals its own programmed address. In such case, the data byte of a transmission will be read, eventually processed and passed to the signal modification block of the extension module.
Allowable addresses for extension modules reach from 1 to 254.
Address 0 is reserved.
Address 255 is a special case. This one is used as a command to assign an address to the extension modules connected to WaverAD, in order for them to be identifiable during later use.

Realization by an MCU

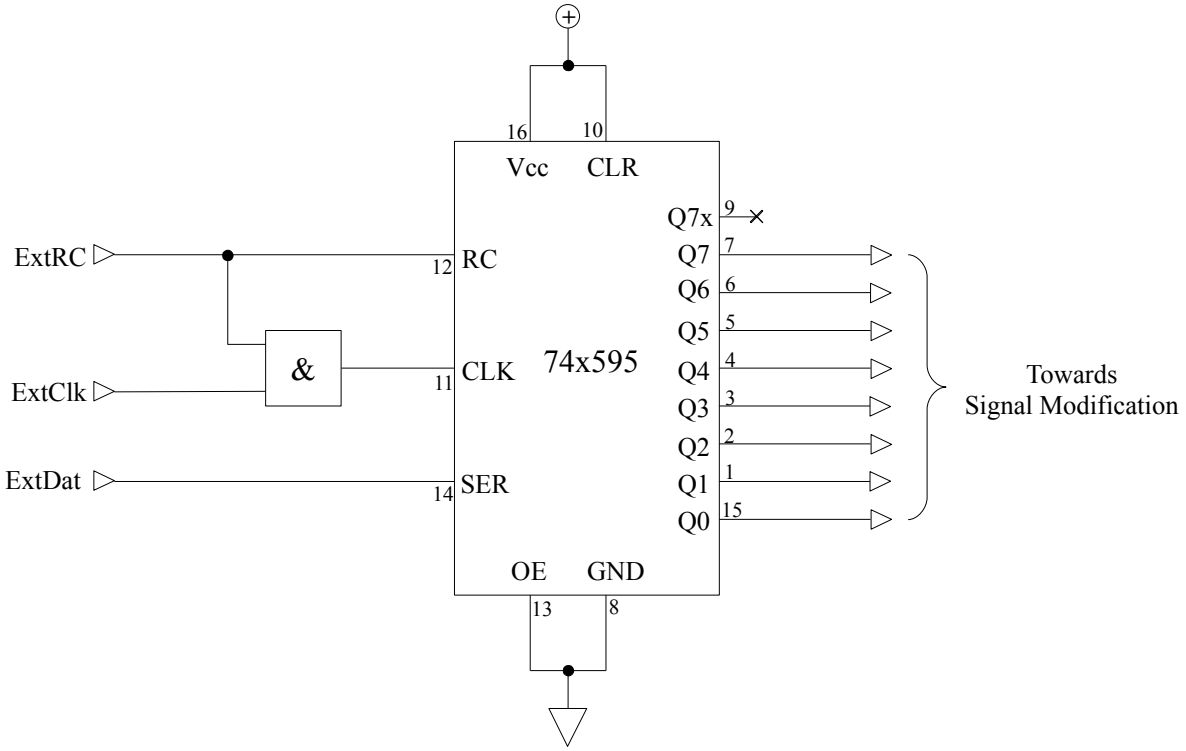
The most flexible way to realize an address decoder is by using a small micro controller unit. Ideally this MCU supports at least 3 digital inputs, the necessary outputs to control the modification block and a small non-volatile memory (e.g. EEPROM) to store the programmed address of the module (1 Byte).
As the command protocol is driven by a very low data rate, constraints to the clock rate of the processor are very low. Typically (depending on the processor architecture) 1MHz is sufficient.
The following flow chart shows the recommended structure of the program code.



Flow chart of a complete address decoder using a micro controller

Realization by Logic

Significantly less flexible than the realization by a micro controller, but much simpler and without the need to program a processor, an address decoder can be built by standard logic devices.
Users who can completely do without an address decoding (e.g. in cases when only a single extension module is connected to WaverAD) can use the following schematic based on the standard shift register 74595 to extract the data byte from the command protocol.



Simple schematic to extract the data byte from the command protocol. An extension module that uses this block will accept data from any software control element, independent from addressing.

Signal Modification

The signal modification block is the place where the actual function of the extension module is defined. A designer of an extension module will use this block to build the desired and controllable function of the module. Potential options could be for example:

- Continuous adjustment of the digital output voltage level
- Configurable trigger input (e.g. adjustable trigger level and/or slope)
- Adjustable signal amplification for the analog signal, beyond the maximum output voltage of the WaverAD (e.g. by an external voltage source)

General Considerations

Developers of own extension modules should specifically take care of the electrical parameters of the WaverAD specification (see the WaverAD user guide for details)..

In particular these are:

- Correct circuitry
  In all cases it needs to be taken special care, that voltage-driving pins of the WaverAD interfaces (output signals, voltage pins) are not connected to external points that also drive a voltage. Doing so can lead to malfunction or in extreme cases even to damages to the device.
- Output voltages of the signal and voltage pins
  Should those not be sufficient for the extension module, the needed voltage needs to be generated inside of the extension module or it even needs to be supplied by an external voltage supply.
- Output currents of the signal and voltage pins
  The output currents from the pins of the digital interface are strictly limited. This is due to the WaverAD being supplied through the USB interface, which is limited in its ability to deliver electrical energy. To not overstress the maximum currents, even when cascading multiple extension modules, it is recommended to avoid power-consuming components or to support the extension module by an external power supply.
- Signal voltage and output resistance of the command signals (ExtRC, ExtClk, ExtDat)
  The signal voltage of the command signals is 3,3V. Connected address decoders must be compatible to this voltage. Equally it must be assured that a connected address decoder provides high impedance inputs to not overly stress the signal source. Eventually a buffering of the command signals needs to be taken into consideration.
- Maximum trigger input voltage
  Extension modules that pass a trigger signal along to the WaverAD, need to adhere to the maximum allowable input voltage of the trigger input. Exceeding such voltage can lead to malfunction or in extreme cases even to damages to the device.

Developers of extension modules that modify the analog output signal (BNC interface) should take care to separate the analog GND (steel case of the BNC socket) and the digital GND (Pin 14 of the digital interface) within their module.