Neurotechnology Ultrasonic Array Controller UAC128
User Guide

User Guide v1.0
June, 2018

Introduction

Welcome to the user guide of Neurotechnology's Ultrasonic Array Controller UAC128. It overviews the main features and specifications of the controller and walks you through the set-up procedure for the controller. Should you have more questions not covered in this guide please contact us at ultrasound@neurotechnology.com .

Contents

1 Overview

1.1 Introduction

Neurotechnology's Ultrasonic Array Controller UAC128 was designed to be used in continuous-wave phased array applications. It is specifically suited for ultrasonic particle manipulation experiments/applications. UAC128 has a total of 128 channels that can operate over a wide frequency range: 25 kHz-1.25 Mhz. Sine-PWM is used for generating sine-like waveforms in the frequency range below 250 kHz, while simple PWM is used for frequencies above 250 kHz. The amplitude and phase of each channel can be controlled individually allowing implementation of many algorithms for ultrasonic field control with phased array systems.

1.2 Functional Diagram

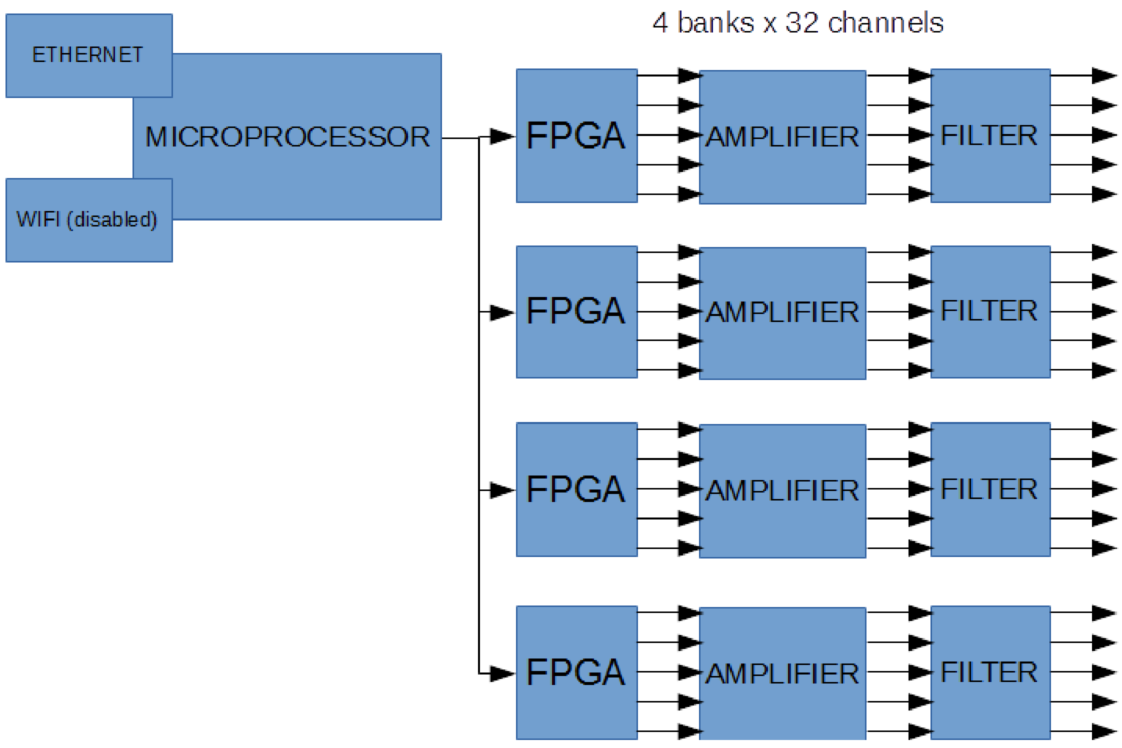

Figure 1: Functional diagram of ultrasonic array controller UAC128.

The functional diagram of UAC128 is shown in Figure 1. The ethernet connection is used to communicate with the UAC128 using a TCP/IP protocol. A wifi communication option is also possible but is disabled by default. A server application is running on the on-board microprocessor which receives the data and distributes it to the FPGA boards. Each FPGA board is responsible for generating waveforms for 32 channels. The 4 FPGA boards are synchronized so that generated waveforms from different boards are in-sync. The tri-level sine-PWM/PWM signals are then amplified to ± 24 V voltage. The amplified signals are then fed to a replaceable low-pass filters in order to reduce higher order harmonics in the waveforms and produce sine-like waveforms. The filters also perform a current-limiter function, that limits the maximum power delivered to load. Different filters should be used for different loads/transducers.

1.3 Back Cover Layout

Electrical connections and replaceable filters are at the back of UAC128, see Figure 2. The purpose of different components in the back cover of UAC128 is detailed in Table 1 with reference to the numbering in Figure 2.

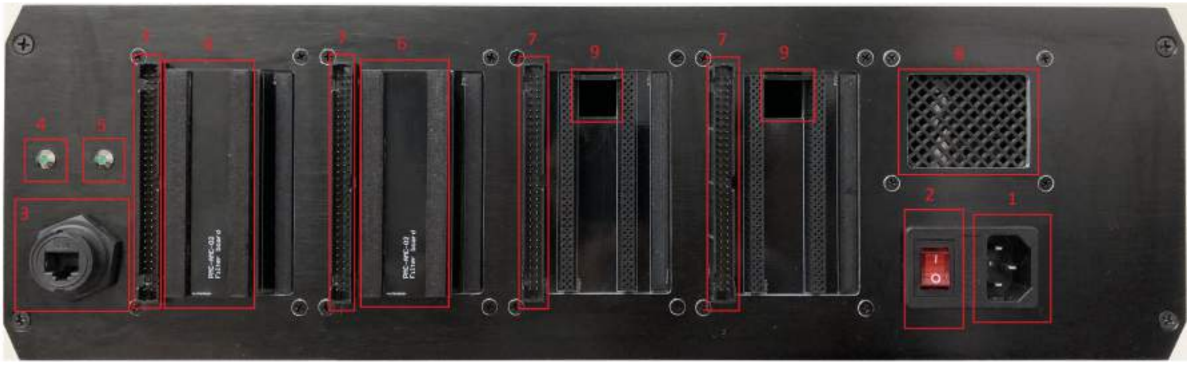

Figure 2: The back view of UAC128.

1	Standard power socket for 230 V voltage with earth connection. Controller's case is connected to earth for safety. Signal ground is also connected to earth. Please connect the controller to the power outlet that has an earth connection.
2	Power switch. The switch should light-up when turned on and the fans should start.
3	Ethernet socket. Use it to connect controller to the network or directly to the computer.
4	Indicator light for server application. Some time after the controller is powered it lights up indicating the server application has started. The controller is then ready to receive the data.
5	Indicator light for data transmission. It indicates when the data is beeing received by the server application and sent to FPGA boards.
6	Filter boards. They contain low-pass RC filters that reduce high frequency components in the signals and also perform current-limiting function. The filter boards should be inserted in the direction so that the label "Filter board" reads upwards.
7	Output sockets. Standard 64pin IDC scokets with 32 pins dedicating for signals and 32 pins for ground (internally connected). Use 64-wire ribbon cables to connect controller to transducers. The ribbon cable should not be larger than 1m for low frequencies and even shorter for high frequencies. The first channel corresponds to the buttom pin on the leftmost output bank when viewing the from the back as in Figure 2. For more detailed description of output pins refer to the description of Figure 3.
8	Air intake. Do not block the air intake as enought air needs to flow in to cool down the amplifiers and output filters.
9	Air outtake holes. The air that cools down the amplifiers is forced out through the air outtake holes and is used to cool down the output filters. Do not cover the back of the controller or in any other ways obstruct the air in- and out-flow.

Table 1: The details on the back cover components of UAC128 with reference to the numbering given in Figure 2.

The IDC sockets for outputs of one bank is shown in more detail in Figure 3. The red lines indicate the signal carrying pins while the blue lines the ground pins that are also internally connected. The

Figure 3: The pin layout of IDC sockets.

signal pins indicated by number 10 are the output pins from the amplifier, and should connect to the input of the filters, the pins indicated by number 11 should connect to the output of the filters. The filter board should be inserted into these sockets in direction so that the "filter board" label reads upwards. The output socket accepts the standard IDC female plug mountable on a ribbon cable. The channel numbering is also indicated in Figure 3 with white numbers. The first output bank (leftmost when looking from back at the controller) has 1-32 channels, the second output bank: 33-64, the third: 65-96, the fourth 97-128, the channel number increasing in the upward direction for all the banks.

2 Specifications

2.1 High and Low Frequency Regimes

The UAC128 has two different operational regimes. In a low frequency regime (<250 kHz), it generates sine-PWM waveforms, where each half-cycle of a sine waveform is formed out of train of pulses of varying width. The sine-PWM waveforms are then fed into the RC low-pass filters and sine-like waveforms are produced. In a high frequency regime (≥250 kHz), it generates only a single pulse for each half-cycle of a sine wave. Hence, the sine waveform has larger distortions after passing through the low pass filter than in low frequency regime. This is done so that high phase and amplitude control resolution is maintained at high frequencies.

2.2 Frequency Control

Frequency value f is determined by an multiplier α: $f = \frac{50MHz}{\alpha}$. The multiplier α has a range of 40-2047, resulting in a frequency range of 24.426 kHz – 1.25 MHz. As can be seen from the expression, the resolution is varying, with highest resolution (12 Hz) at lowest frequencies and lowest resolution (30 kHz) at highest frequencies.

2.3 Phase Control

The individual phase delay can be applied to each channel with reference to the first channel. The phase delay can be adjusted in the range of 0-360. The resolution depends on operational frequency as presented in Table 2.

25 kHz	255 steps (8bit)
40 kHz	255 steps (8bit)
100 kHz	255 steps (8bit)
200 kHz	200 steps
250 kHz	200 steps
500 kHz	100 steps
1000 kHz	50 steps

Table 2: Phase resolution of UAC128 at different frequencies.

2.4 Amplitude Control

Amplitude of each channel is controlled by adjusting the width of PWM pulses. From the software side the amplitude is adjusted in the range of 0-1, the actual voltage that will appear accross the load will depend on the load impedance and output filter used. The resolution depends on operational frequency as indicated in Table 3.

25 kHz	224 steps
40 kHz	140 steps
100 kHz	56 steps
200 kHz	28 steps
250 kHz	100 steps
500 kHz	50 steps
1000 kHz	25 steps

Table 3: Amplitude resolution of UAC128 at different frequencies.

2.5 Load and Output Power

The UAC128 is designed to drive capacitive, high impedance loads such as piezoelectric ultrasonic transducers. In the low frequency regime, controller is capable of driving loads with impedance 300-1000 Ω and delivering around 0.3 W per channel. In the high frequency regime, the load impedance should be >1200 Ω and the maximum power delivered to the load is around 0.1 W.

Specific filter boards will be supplied to you for your target loads. The filters reduce higher order harmonics in PWM signal in order to produce sine-like waveforms and also limit the current delivered to the load in order to prevent the amplifiers from overheating.

The UAC128 has been tested to work well with common air ultrasonic transducers, such as Prowave 250ST180, 400ST160, 400ST120, 330ST160 and Murata MA40S4S. The impedance of these transducers at resonance frequency is in the range of 300-700 Ω and the maximum power is around 0.25 W.

Note that the user is responsible to check that the voltage appearing accross the load does not exceed the safe operation limits of the load. Otherwise the load can get destroyed.

2.6 Control Speed

The total amount of time taken to send the control parameters to the UAC128 using the interface library provided is less than 3 ms with direct ethernet connection to the controller. Under normal network traffic, similar speeds should be achieved if controller and computer are connected to a network. 2.5ms control speed was measured using computer with Intel Core i7-3770 processor and running Windows 8.

3 Set Up Guide

This section provides a general step-by-step guide for setting up a controller.

1. Connect the power cable with earth connection to the controller and plug it to the mains outlet. Make sure that the mains outlet has the earth connection.
2. Connect the ethernet cable to the controller. Connect the other end of the cable either to a network or directly to a computer.
3. Check if the correct filter boards have been inserted at the back of the controller.
4. Connect the outputs of the controller to transducers using ribbon cables. Check that signal/-ground connections are not mixed-up. See Figure 3 for details.
5. Set-up your connection settings. See Section 4.1 if using network connection or Section 4.2 if connecting controller directly to a computer.
6. Turn the power switch on. The cooling fans should start immediatelly, the server-on LED should also light-up after some time, which depends on the connection option. The controller is then ready to use.
7. Send control data to controller using the provided interface library (*NeurotechnologyArrayController.dll* and *NeurotechnologyArrayController.h*). C/C++ and Matlab examples on how to use the library and send data to controller are provided. See Section 5 for more details.

4 Connectivity

The ethernet connection is used to send control data to UAC128. A Wi-Fi option is also possible, but is not enabled by default. Make a request when purchasing UAC128 for a Wi-Fi to be enabled if it is required in your application. The control data is sent to UAC128 using TCP/IP protocol.

4.1 Network Connection

By default the UAC128 is set-up to be connected to a network. The on-board microprocessor has DHCP service enabled and once the controller is powered it makes a request for an IP address. Ask your network administrator for an IP address or a hostname that was provided to the controller. You can then connect to the controller using either the hostname or the IP address with the provided interface library. You will find the MAC address and default hostname of UAC128 at the bottom of the case.

The ethernet connection should be supplied before the UAC128 is powered on. Once it is powered and IP address is succesfully obtained a server application starts, an indicator led lights up and UAC128 is ready to receive control data. It should take around 20 seconds, if it takes much longer check with you network administrator if IP address was assigned to the controller and internet is available and then re-power UAC128.

4.2 Direct Connection

The controller can also be connected directly to the computer through the ethernet connection. When UAC128 is powered it attempts to obtain an IP address from DHCP server and as DHCP server will not be available in this case, it will time out after about 1-2 minutes. It is then possible to connect to UAC128 using its static IP address. Its IP address is 192.168.100.9 and subnet mask:255.255.255.0. Change internet protocol (TCP/IPv4) settings accordingly, i.e. IP address to 192.168.100.1 and subnet mask to 255.255.255.0. Note that due to the need of DHCP request to time out, the start-up time will be longer using direct connection option.

5 Software

A DLL library (*NeurotechnologyArrayControllerClient.dll*) provides a communication interface with UAC128. It has three main functions *connectController*, *disconnectController* and *sendController*. *connectController* establishes connection with the controller and returns a TCP/IP socket handle. *sendController* disconnects from the controller. *sendController* is the function used to send control data to the controller. Control data includes frequency value, phase and amplitude arrays containing values for each channel. The library also has demo functions for ultrasonic field focusing, that calculate phase delays for focusing field to a certain point in space given some ultrasonic array configuration. They can be used to trap and move particle in ultrasonic particle manipulation experiments/applications. See the library's header file for a detailed description of the functions.

5.1 C Samples

A C/C++ sample (*NeurotechnologyArrayControllerDemo.cpp*) demonstrates how to use functions provided in the DLL library. It shows how to load the functions, connect/disconnect from controller and send control parameters. In the sample a particular transducer array is also defined, i.e. transducer positions in the array, their operational frequency and sound velocity in the medium. The array information is used to calculate phase delays for focusing energy to desired position. In this particular example, a focus position is moved forward and backward along the x direction in 0.25 mm steps. A particle could be trapped in the focus and hence moved forward and backward in a line. Change array information with your array parameters for this sample to work.

5.2 MATLAB Samples

Matlab samples include an *ArrayControllerDll* class, which is a wrapper class for the provided DLL library. The methods of the class correspond to the functions in the Dll library. This class is used in the provided Matlab samples. The DLL library and corresponding header file should be place in Matlab's path.

array_controller_demo script demonstrates how to use *ArrayControllerDll* class and change various control parameters of the controller, namely frequency, phase and amplitude. It is split into sections that show how to perform a frequency sweep, an amplitude sweep, turn each channel on/off and sweep phase of each channel one by one.

array_controller_gui is a GUI application which can be used to connect/disconnect from the controller and change phase/amplitude of each channel interactively.

array_controller_focusing _demo shows how to use focusing functions provided by the DLL library. A structure containing the information of particular transducer array is defined in the script, change the parameters that correspond to your particular array. In this particular example a focus is moved along x-direction in 0.25 mm steps. A particle could be trapped in this focus and moved along with it.