The Universal ALICE Project

With the goal of "Instrumentation for Every Student" in mind this project is intended to lower the barriers to providing a hardware and software platform for teaching electronic circuits to engineering students. Every student should have access to affordable hardware and software to measure their experiments.

This project offers a powerful free open source user interface that is compatible with hardware produced in large quantities at low cost (ranging from $5 to $20) from multiple suppliers. Those of you who are perhaps familiar with the ADALM1000 (M1k) and the ALICE software will recognize the look and feel of ALICE Universal. ALICE Universal takes the rich suite of virtual instruments available in ALICE for M1k and brings them to a wide range of hardware agnostic options at very low cost.

The main Universal ALICE GUI is written in Python (3.x) and depends only on the more or less generic standardly available Python plugin libraries (such as Tcl/Tk, numpy, pyplot etc.). The hardware specific interface level is also generally implemented in Python plus any plugin libraries (such as pyserial, Pyusb, pyaudio etc.).

As Ken may have mentioned, I have been recently focusing on micro-controller boards supported in the Arduino ecosystem mainly for the wide cross computer operating system / hardware platform support. These two links are to the project GitHub repositories:

https://github.com/damercer/Universal-ALICE

https://github.com/damercer/Arduino-Scopes

Multiple Arduino compatible micro controller boards have been evaluated and each has its own pluses and minuses. These boards are little more than breakout carrier boards for the MCU chip. They contain the minimum support circuitry needed to interface to a host PC through a USB port. Often a sub-set of the MCU pins are broken out to through hole pins for User access. The attached Hardware-options-list.pdf file contains more detailed information for context on some of the boards evaluated so far.

Most all hardware platforms in the Arduino world have built in ADCs of some speed / resolution so any / all can provide a basic analog input “oscilloscope” type instrument. They also generally provide at least one fixed DC power supply voltage available for the user. Digital inputs and outputs are available with one or more that can be programed as square wave PWM outputs. Very few if any have internal DAC analog output(s) that could provide an analog waveform generator output (AWG) function.

As for use as a data acquisition platform, these MCU boards provide access to the raw ADC inputs and if available the raw DAC outputs of the MCU. These analog pins are limited to the power supply voltage of the chip, generally 3.3V. The onboard memory can serve as sample memory buffers to hold captured data to be sent back to the host PC by way of the USB serial port. Unlike more complete (more costly) solutions these MCU breakout boards do not include any of the analog signal processing chain often built into more advanced data acquisition hardware. However, many introductory circuits and electronics lab experiment use cases can be configured with some additional simple circuitry to interface with the raw ADC input and/or raw DAC output directly as is.

Current thinking is that the Seeed Studio XIAO SAMD21 micro-controller board ($5.40) might perhaps be the easiest to deploy in introductory circuits and electronics lab classes. The single ADC in the ATSAMD21 MCU is multiplexed to multiple input pins with a single DAC output pin. My reasoning is probably best explained in the attached XIAO-Scope-hardware.pdf. While not the fastest of the bunch (48 MHz ARM M0 core) it is the simplest to use and adequate for the job as I think will be apparent after looking through the documents.

The firmware is contained in a USB flash file (.uf2) that can be simply copied on to the USB drive that appears when the board is placed into the UF2 Boot loader. Arduino does this automatically as part of the upload process. Doing this outside the Arduino IDE for the Seeed Studio XIAO board is not so simple from the hardware because it has no push button to reset the board. You have to use a wire to short two tiny pads on the board. However, there is a software way to do it and a scheme has been implemented in the Python code to force an un-programmed board into the boot loader and then (in Windows at least for now) copy the .uf2 file on to the USB (E:) drive. This gets around the whole need to install and set up the Arduino IDE. A working Arduino sketch for the board firmware and a hardware interface file for Universal ALICE is available.

That being said, there is still potential educational value in having students installing and learning how to use Arduino. They will probably be using Arduino in an embedded programming micro-controller course eventually anyway. So an Arduino setup procedure document for this (XIAO) hardware has been put together as part of the total support package provided around the project. Other versions for hardware from other manufacturers can be easily spun from there.

Also within the Arduino ecosystem, the Raspberry Pi Pico board is an interesting option based on price ($4.00), availability (large quantities from multiple distributors) and hardware capabilities (133 MHz dual ARM M0 core). The pre-compiled firmware, in a USB flash file (.uf2), is simple to download to the board by simply powering up the board while holding down the reset button, and drag and drop the file onto the USB disk that pops up. Users do not need to have the Arduino IDE installed. Pico is pinned such that the single ADC can be multiplexed to 3 analog input pins but has no on chip DAC. There are 23 general purpose digital I/O pins which can be used with serial SPI DACs or external R/2R resistor networks (8 bits each) to make 2 DACs. The Hardware Outline document shows the basic parts list and cost for building out a personal instrument based on the Pico. A working Arduino sketch for the board firmware and a hardware interface file for Universal ALICE is available.

There are now about a dozen different hardware platforms supported from the likes of Adafruit, Raspberry Pi, Seeed Studio and SparkFun. Most are either ATSAMD21/51 based or RP-2040 based. The US based suppliers like Adafruit and SparkFun have been added for those potential buyers who have strong "Buy American" biases. Cost is higher than the China suppliers but still in the $20 range or less.

Getting Started With Seeed Studio XIAO:

Download and extract the zip file to a folder:

https://github.com/damercer/Arduino-Scopes/blob/main/XIAO_Scope_files.zip

Go to that folder to run ALICE from there.

It is assumed that you already have Python installed along with numpy/matplot and pyserial...

(otherwise you need to add PySerial by typing "pip3 install **pyserial**" in a command window)

To run ALICE from the source you sometimes just double click on the source file if the right version of Python is in the search path. If you by chance have multiple versions of Python on your computer you will have to point to the specific version. Put the full path in a .cmd batch file and just run that.

I have added the ability for ALICE (when running the hardware specific file for this board) to detect that a brand new board that is plugged into the USB port has not been programed yet and it will ask you if you want to do that. It will then place the board in the boot loader (Yellow LED will be "breathing") and copy the supplied .uf2 firmware file to the board. The board will then quickly reset (just green power LED lit) and ALICE will fully open. The Red Recon button in the upper right can now be pressed to connect to the hardware and you are ready to go.

All the files need to be in the same (write enabled) folder. The last two aren't specifically needed but initialize and configure things from the defaults into a state closer to what you need to just turn on the AWG and hit Run.

Extra hardware information stuff.

Unfortunately the Infineon PSoC 5 based development software works, as far as I can determine, only on Windows. Once a kit board has been programed (on Windows) it can be used with computers with other operating systems such as Linux.