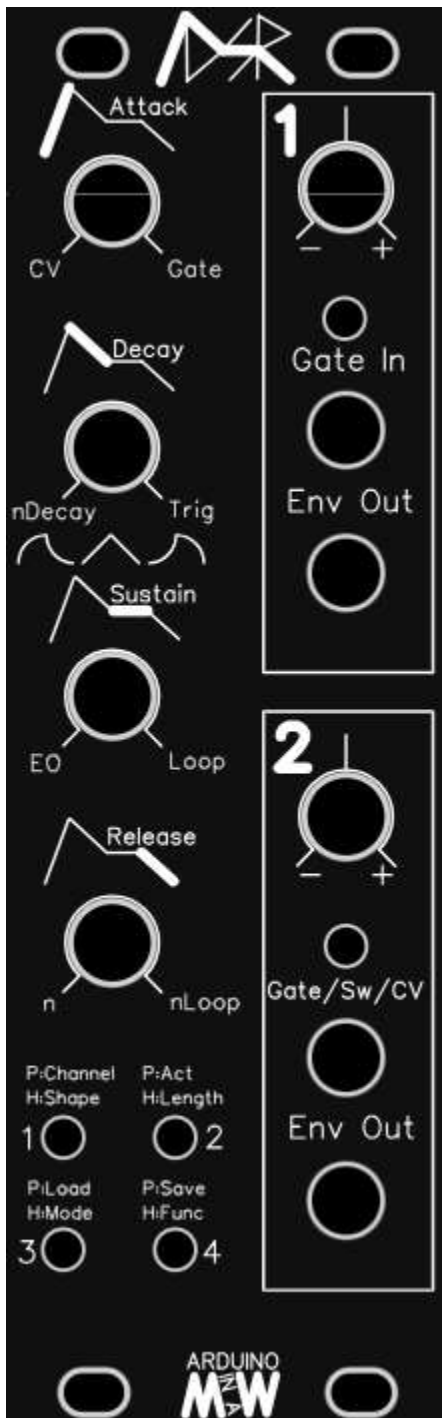


Users manual for the MiaW ADSR

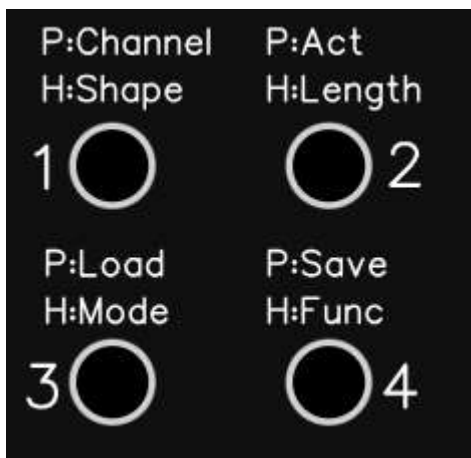


The MiaW ADSR is a two channel Envelope generator built around an Arduino. The schematics and code is open source for personal use and can be downloaded and modified from the github page of the product. It can also be bought as a

PCB+Panel set or as a full kit from this store:

As said this is a two channel ADSR, but there are only one set of controls for the envelope curve. Because of this there are a few nifty features that makes it possible to control two ADSR with one set of controls.

The menu system



The menu system consists of four buttons on the bottom of the module. These four buttons has two functions each. One function can be activated by a quick press of the button (shown with a P), and the second function, often with more functionality can be used to hold the button and turning the ADSR-knobs (shown with a H).

An important aspect of the menu and changing functions, and channels is that every time a channel is switched or a hold menu is entered the knobs needs to be "caught" meaning you have to turn the knob both higher and lower than the value it has in its current function. So if ADSR Ch1 has an Attack of 10 (from 0-1023) and ADSR Ch2 has an attack of 500 and you switch

from Ch1 to Ch2 then you need to turn the Attack knob from 10 (which was the value you had on Ch1) up to 501 before the knob will do any difference to the curve. This is ofcourse so that values wont change the second you enter a menu or change a channel.

Button 1 Press: Channel 1/2 Change, by quickly pressing the first button you will change between the two channels. All the knobs will light up in green for channel 1 and red for channel 2.

Button 1 Hold: Change slope of the curve. In this mode you can change the behaviour of the slope of each curve. This is individual for each part of the envelope, so turning the Attack knob will change the curve of the attack phase, and the same for the other knobs. In the middle the knob will be yellow and the curve will be linear. Turn the knob to the left it will turn green and it will be fast start, slow finish exponential curve, similar to how a capacitor driven envelope would work. Turning the pot to the right will instead turn the knob red and have a slow start and fast finish exponential curve. There is this small drawing on the panel for you to remember which way to turn to get the different slopes.



It should be noted that there is a setting in the code where you can set how extreme you want the edges to be, and also an advanced mode where you can set a floating value between the max and min values. In the basic mode you will just have the three most optimal curves to choose from.

Button 2 Press: Act. This affects how the ADSRs will act to the input. There are four modes: Gate, Trig, Loop and nLoop. You know which acting you have on your channel as one of the four knobs will light up and you see the name of the acting on the right side of the knob.

Gate is “normal” ADSR functionality, where the sustain is held as long as the gate is held.

Trig is a “one shot” of the Envelope, so a short ping will play the whole Envelope, with sustain as just a waypoint.

Loop will basically loop the envelope again with the sustain as a waypoint.

nLoop will also loop either as long as the gate is pressed, or the amount of loops defined by the release knob I Button 4 Hold menu.

Button 2 Hold: Curve length. The same as with the slope this setting is individual for each of the envelope parts. This setting decides how long the curve can be when you turn the knob all the way to the right. Here you have three settings to choose from fast (~1s), medium (~5s), and slow (~50s). These three values can also be set in the code if you want to have even faster or slower envelopes.

Button 3 Press: Load. This loads the saved settings from memory, this also happens everytime

you start the module.

Button 3 Hold: Module Mode. This setting sets the mode for the whole module. More information coming.

Button 4 Press: This will **save all settings** of the module, so that next time you start it up it will remember how all settings were and set it up as such. This uses the Eeprom of the Arduino, and this can be used around 100 000 times before the memory will start to deteriorate, at which point there is another setting to move the memory blocks used for the save function.

Button 4 Hold: Functions, this menu has "the rest" of the settings of the module. In this menu each knob changes a different function according to the text to the left of the knob:

- **Attack knob sets the CV input** - When holding button 4 and turning this knob it goes from all leds off to first light up Attack green, then D,S,R respectively, and then red A,D,S,R respectively. When all leds are off, CV is off, and input 2 is acting as a gate for Envelope 2. When any of the leds are lit up green it means that a CV signal on input 2 will change the value of that knob on Channel 1 adding the CV value from the input. When red it will instead subtract the CV input value from the knob. **CV input will disable the 2nd input**, however the 2nd envelope will still function with for example End Of or other module mode.

- **Decay knob sets the nDecay for the nLoop** - When in nLoop setting the nDecay will lower the amplitude for every loop until the amplitude is 0. It will release as soon as the nLoop is done or the gate is released.

- **Sustain knob sets the EO (End Of)** - End Of Attack/Decay/Sustain/Release, when turning this knob you choose which stage will trigger the End Of, that in turn will trigger envelope 2. When all leds are turned off EO is turned off, and when any of the leds are turned green, that is the stage the End Of will trigger from when finished. This can be used for sending a short trigger or gate using the second output or for starting a nLoop or second envelope later in the first envelope. It can also be used as a HADSR where you can either have a delayed ADSR by using the second envelope and the first for the delay, or even hold if you have End Of Sustain, thus the second envelope will trigger once the sustain is released. **End of will disable the 2nd input.**

- **Release knob sets n for the nLoop** - if no led is lit then nLoop will loop as long as the input gate is held. But if you turn the knob the leds will light up in a binary counting way, where the Green Release led is worth 1, Sustain led is worth 2, Decay 4 and Attack 8. When the Release led turns orange you have counted up to 16 which is maximum at this time. Any value except 0 will loop that many times until it stops.

Gate and Envelope outputs



On the right of the module there are two sections labeled 1 & 2. These are the sections for each ADSR input gate/trig and Envelope out.

The knob is an intenuverter and can either amplify or attenuate or invert the output signal. All the way to the right the envelope will be around 0-8V, all the way to the left the envelope will be inverted and instead output between 0 and -8V. The closer you get to the middle the lower the signal out will be for really subtle envelopes. Using these in an opposite manner one of the more advanced modes can create some interesting intereactions between VCAs and VCFs for example.

The button is a manual trig if you want to manually play the ADSR or test functionality.

Gate in, is ofcourse where you add either a gate or a trigger from another module to control the envelope from a sequencer, clock divider or other module.

Env Out, is what you connect to a CV input of the module you would like the ADSR to control. It can be something classic like a VCF or a VCA, or something more unconventional like any other module with a CV input... ;) Note that you can take the output of channel 2 and input into the input of channel 2 if you are in CV mode and using EO or a different module mode to trigger envelope 2.

While it may seem like a complex module when it even needs a manual for something as trivial as an ADSR once you start working with it it is very easy to understand even this "complex"

menu system, which I think for being a menusystem is very easy to understand ;) (I am ofcourse biased)...

Hope you will enjoy this module and that you will do some great patches with it.