mesytec MDLL is a readout system for multi wire proportional counters with two delay lines for X and Y coordinate. The position is determined from the delay of the signals from the two ends of a delay line.

Features MDLL:

- X and Y position calculation
- Anode and Cathode amplitude is converted and available for threshold adjustment.
- Anode signal is optional
- Register adjustable to detector delays from 10 ns to 1000 ns
- List mode with time stamped events (100 ns resolution)
- NIM form factor
- Can be time synchronized with several MDLL modules and mesytec MCPDs (neutron systems)
- Very low noise adjustable input amplifier. Allows flexible input range: +-5 V to +- 15mV maximum range.
- Accepts preshaped signals or raw signals from detector.
- Timing resolution for delay line down to 65 ps rms
- Ethernet (100Mbit) connectivity to readout PC.
- New Ethernet features: gets IP address by DHCP, can be accessed by fixed host name ("MDLL-0012") where 0012 are the last numbers of serial number.
- Additional interface to mesytec open source data acquisition and analysis program "mvme" allowing to use the powerful features of the new digital signal processing.

mesytec GmbH & Co. KG
Wernher-von-Braun-Str. 1, D-85640 Putzbrunn, Germany
phone: +49 – 89 / 456007-30 fax: +49 – 89 / 456007-39
info@mesytec.com – www.mesytec.com

Technical Data

Input stage

- 5 inputs:
  X0, X1 for first coordinate delay line,
  Y0, Y1 for second one
  Anode input for common signal.

- Accepts negative or positve input. Maximum amplitude is coded by an internal gain jumper module. Default is 2V. The range can be further set by internal gain stage with a gain of up to 20. So gainjumper of 2V allows to set maximum range from 2V down to 100mV.
  Available gain jumpers: 0.3...5V.

Input filter

Time constants: are set by register.

The Discriminator is a software implementation and usually needs only a detector specific threshold.

Power consumption NIM module:

(max 7 W)

+6 V 1 A (at power up max 2 A)
–6 V –200 mA

Adjustment of MDLL module for MWPC

Preparations

Connect detector to the MDLL module (inputs X0, X1, Anode, Y0, Y1). All signals negative, Amplitude typically –1 V, rise time 40 % (= fraction) to maximum 10 ns to 20 ns.
Set CFD-delay via jumpers for 5 inputs.
Install the Detector and add a source to fill all the active area of the detector.
Apply typical voltages to the detector.

The input LEDs should light up.

Basic adjustments

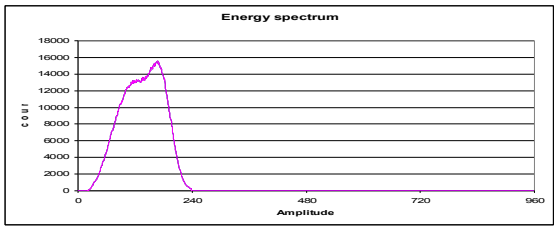
Set parameters to adjustment defaults:
1. Open **Set MDLL** window
2. Set low thresholds for discriminators X, Y, and Anode to value 20
3. Set X,Y spread to 40 and X, Y offset to 100
4. Open timing and Energy restrictions
   Energy: (minimum = 10, maximum = 240),
   Timing: low 100, high 1000
5. Click Spectrum „E, X, Y“
6. Click „apply“

Start Data acquisition. → event counter should increment.

After activating the set command the spectra should be cleared with the „clear“ button to see the new data.

Adjust Amplitude

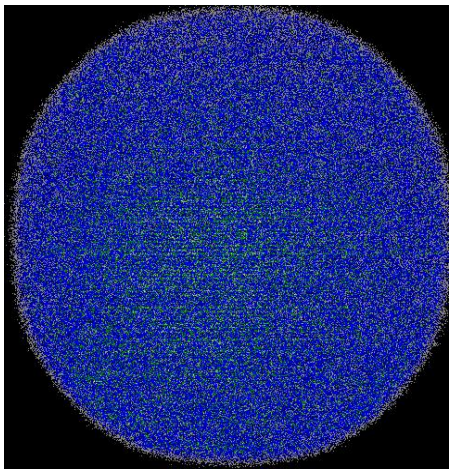
1. Spectrum: click „Energy“
2. Adjust Detector Voltage for correct output amplitude. The amplitude range can be from 0 to 240, the spectrum should not be cut off significantly at the upper edge (240)



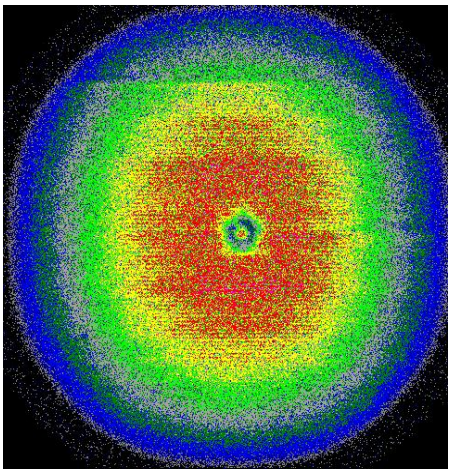
Adjust timing range and offset (spectrum spread)

1. Open 2-D (XY) view
2. Vary the spectrum spread X, Y (reasonable values 20...60)
3. and shift X, Y (range 50...150) to fill all the 2-d spectrum

Well adjusted 2-D views



counts



amplitude

Add restrictions to the data (data filter)

The restrictions help to reduce the background. They are not essentially needed for first tests. If you change any other parameter after the restrictions are set, the efficiency may decrease or all data may be filtered off. So when changing other parameters, the restrictions should be removed before (set timing and energy window to adjustment default).

Sum timing restrictions

1. Stop data acquisition
2. Open set MDLL window
3. Click “E, Xtsum, Ytsum”
4. Clear spectra and start data acquisition

The position X and position Y spectra now show the timing sum of the X-coordinate delay line, and the Y delay line. There should be a sharp peak of about 10 bins FWHM at about position 400 to 500. The peak width is correlated to the integral linearity achievable in the position resolution.

Now you can set the timing sum lower and upper limits for X and Y coordinate via „Set MDLL“ window. You can immediately see the effect in the spectrum.

Energy restrictions

If the MWPC provides a significant amplitude resolution, the energy restriction may make sense. For neutron detectors. A lower cut off may improve the position resolution with cost of efficiency.

Overview, serviceable elements inside the box

→ gain jumper for raw range and input polarity.

Interfacing to mesytec “mvme”

The new MDLL V20 has a completely different internal processing than the previous module.
It is implemented as a 6 channel variable gain preamplifier with continuous gain of 1 up to 20. The amplifiers are followed by filters and 6 sampling ADCs. Signals are processed in an FPGA.
The setup is identical to the mesytec module MDPP-16 with “QDC” software, but limited to 6 channels. For details have a look at the MDPP-16 manuals.
It can be connected to the free and open source “mvme” data acquisition and analysis software, running on LINUX and windows platforms.
When connected by “mvme” the MDLL module behaves like mesytec MVLC VME-crate controller with a connected MDPP-16 module. The emulated VME base address is then 0xFFFF0000.

When connected to mvme the MDLL provides 3 values per channel: Amplitude with a long integration, Amplitude with a short integration, the time between triggering Anode and the four cathode inputs.
Default settings are:

Gain Jumper: 3000 = 3V
Input Amplitude: 1000 = 1V
Pulser width 50ns
Timing resolution 24ps
Short integration 7 ==12.5 x 7ns =87.5ns
Long integration 7 #not used, but can be used for pulse shape discrimination

The time is used for X-Y position calculation. The amplitudes can be used to adjust the input range, and optionally for pulse shape discrimination of input pulses. This may help to suppress gamma hits from neutron hits or particle hits from micro sparcs.

Starting with mvme:
MDLL should be on an Ethernet network with DHCP service.
When connecting the link/rx LEDs blink until the module receives its IP addres fro DHCP server.

In mvme main window:
* → File → Open Workspace
select unpacked delivered workspace directory

* Communication: → settings:
controller type MVLC_ETH
Hostname: MDLL-0012 (for MDLL with end serial number 0012)
leave advanced features at default.

* press → start
MDLL channel LEDs should light
On Analysis page, the spectra should show a peak
On the right bottom side of Analysis window the calculated positions are shown.

Channel numbers in MVME translate the following way

chan 0	Anode
chan 1	Aux (usually not used)
chan 2	Cathode0a
chan 3	Cathode0b

chan 4	Cathode1a
chan 5	Cathode1b

Ethernet DHCP setup

When using the new MDLL_v2 via Ethernet some network setup has to be done.

The easiest way to get a working setup is if you are running a DHCP server on your network. The MDLL_v2 will request an IPv4-Address and a hostname via DHCP after powerup.
The hostname is mdll-NNNN where NNNN is the serial number shown on the back side of the module, near the ethernet port. After the DHCP phase the MDLL should be reachable via its host-name:

```
ping mdll-0010
```

for MDLL_v2 with serial number 0010.

Ethernet ARP setup

In case DHCP with hostname assignment should not or cannot be used an alternative approach is to manually associate the MAC-address of the MDLL_v2 with an IP-address.

The MAC address is printed on the back side of the module. The MAC-address has the form 04:85:46:d3:NN:NN where NN:NN is the serial number of the MDLL_v2 in decimal. For MDLL-0012 the full MAC address is 04:85:46:d3:00:12.

With the MAC-address at hand we can now create an IPv4-address to MAC-address mapping in the operating systems ARP table.

This step is specific to the operating system and will require root/admin permissions. The below examples associate the IP-address 192.168.100.42 with the MDLLs MAC-address. You have to change the IP-address to match your local network setup, otherwise the operating system does not know how to reach the controller.

Creating the ARP entry under GNU/Linux:

With root permissions an ARP entry can be addded this way:

```
arp -s 192.168.100.42 04:85:46:d3:00:12
```

To make the entry permanent (at least on debian and ubuntu systems) the file /etc/ethers can be used. Add a line like this to the file:

```
04:85:46:d3:00:12 192.168.100.42
```

This will take effect on the next reboot (or when restarting the networking services).

Creating the ARP entry under windows:

Open a cmd.exe prompt with administrator permissions and use the following command to create the ARP entry:

```
arp -s 192.168.100.42 04-85-46-d3-00-12
```

To verify that the connection is working you can try to ping the controller:

```
ping 192.168.100.42
```

If everything is setup correctly the controller should answer the ping requests.