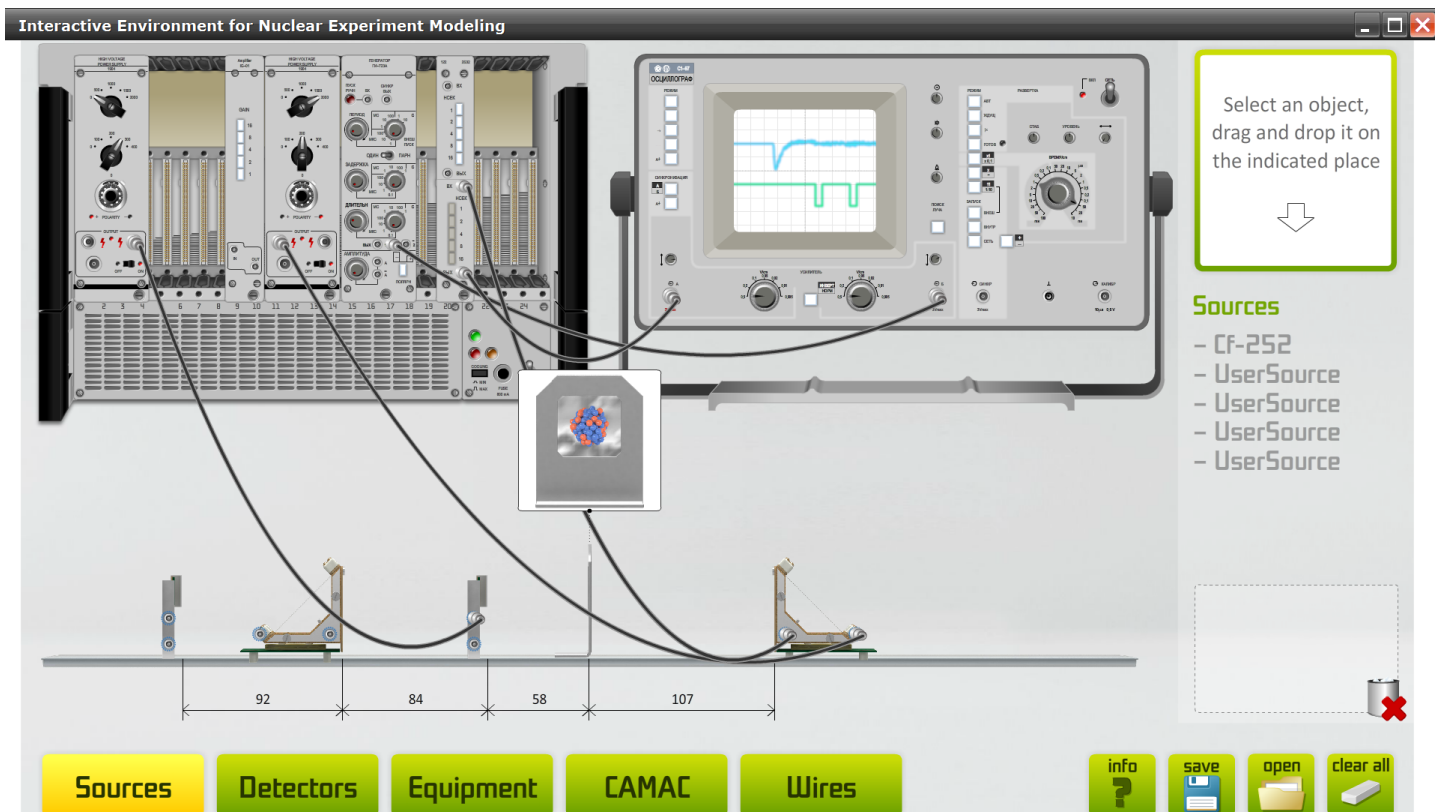


MANUAL

for the Interactive Environment for Nuclear Experiment Modeling

1. About the program

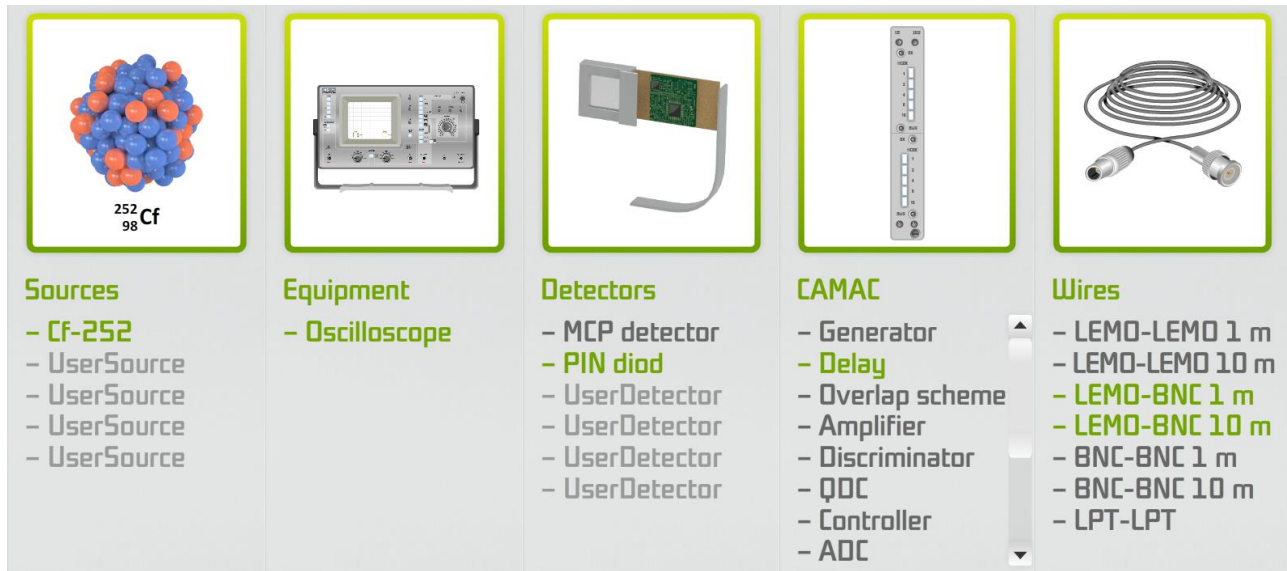
The Interactive Environment for Nuclear Experiment Modeling (further “Setup Builder”) is the interactive software with libraries of physical equipment and materials necessary for physical experiment in nuclear physics. This Setup Builder gives an opportunity for students to model and carry out their own nuclear experiments. The Setup Builder can also be used for graduates and postgraduate students as an instrument of their research works on the development of new elements of equipment for nuclear physics experiment.



General view of the Setup Builder

The Setup builder includes the set of the following libraries:

- sources (radioactive sources),
- detectors (detectors of radiation)
- equipment (an oscilloscope, a computer etc.),
- CAMAC blocks,
- wires.



The set of libraries of the Setup Builder

The user can assemble a setup choosing the elements of equipment and arranging them in the working field. The places where you can arrange chosen elements of the equipment are highlighted by the green outlining. The user also can add his own elements of the equipment which were programmed by him.

2. User options

The main operations in the Setup Builder are made using the LEFT mouse button:

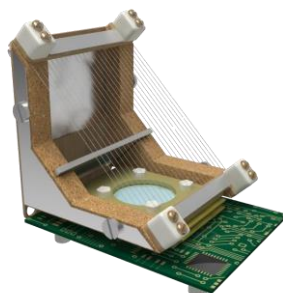
- dragging and dropping of the elements (sources, detectors, equipment, CAMAC blocks etc.);
- connecting of input/output sockets of elements with wires;
- changing parameters of the elements (controls, switchers etc.).

The additional options are made using the RIGHT mouse button:

- zoom of CAMAC blocks;
- zoom of the oscilloscope controls;
- zoom of the oscilloscope screen.

Working with equipment you should remember some rules of work. For example you can't put or take out CAMAC blocks from the CAMAC crate when it is switched on, because it will lead to the block burn. Working with detectors you should take into account their operating voltage range. If you supply a detector with the wrong voltage you won't get a proper signal or you can spoil a detector. For example the Setup Builder has the following operating voltage ranges for detectors:

- a detector based on the MCP has the operating voltage range of $[-2200, -2000]$ V;



A detector based on the microchannel plates

- a PIN diode has the operating voltage range of $[+60, +300]$ V.

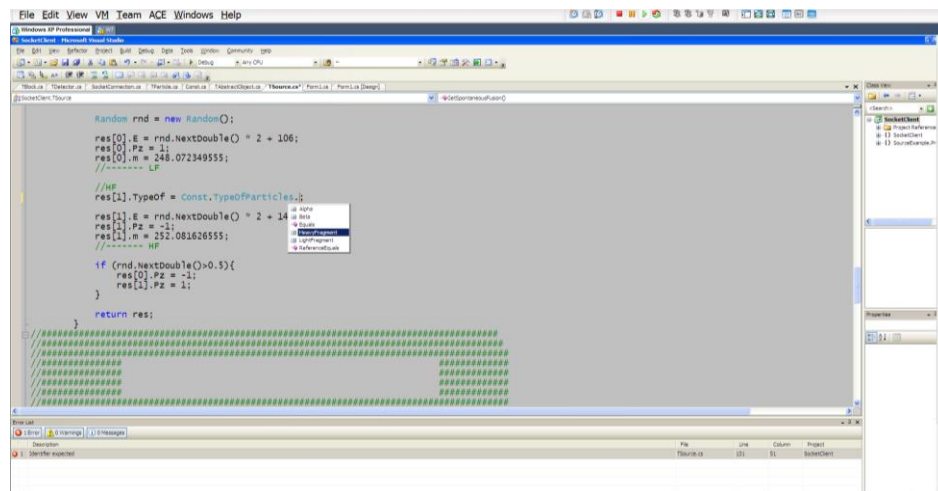


A pin diode

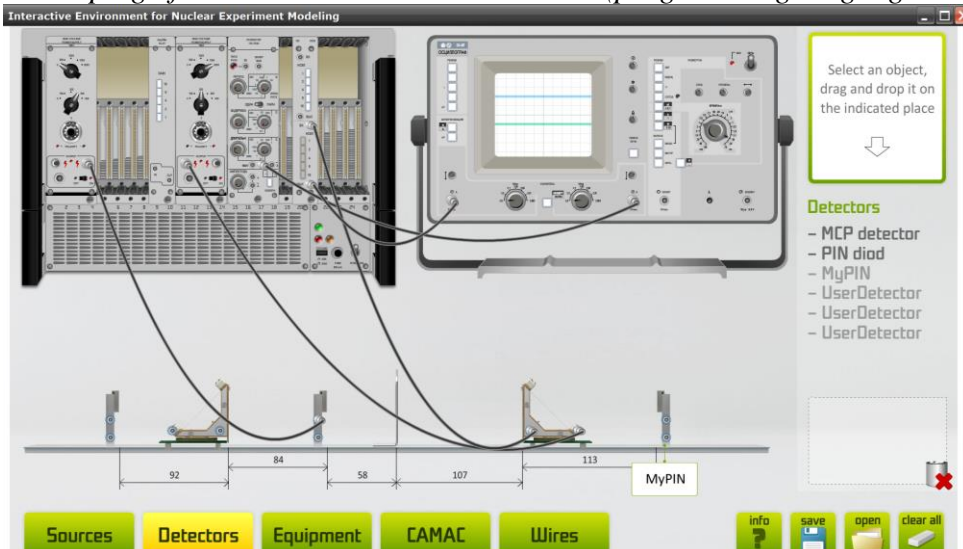
3. Programmable modules

As it was mentioned before the Setup Builder gives an opportunity to create your own programmable modules which realize the user algorithms. The programmable modules can be represented by the following types of equipment and materials:

- sources,
- detectors,
- CAMAC blocks.



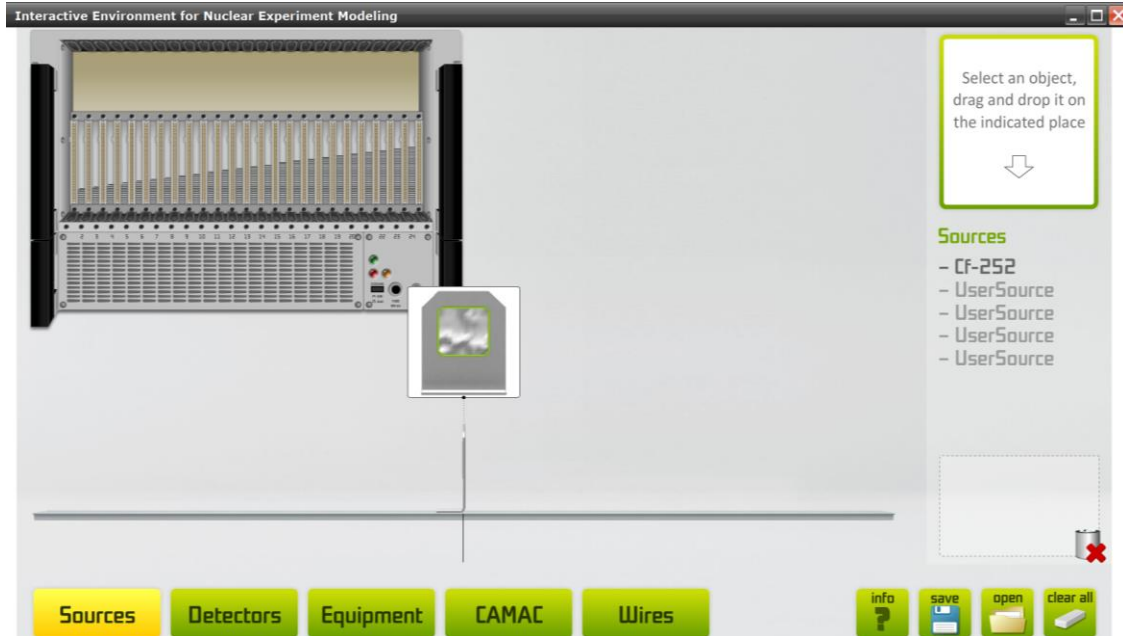
Developing of the user source in Visual Studio (programming language C#)



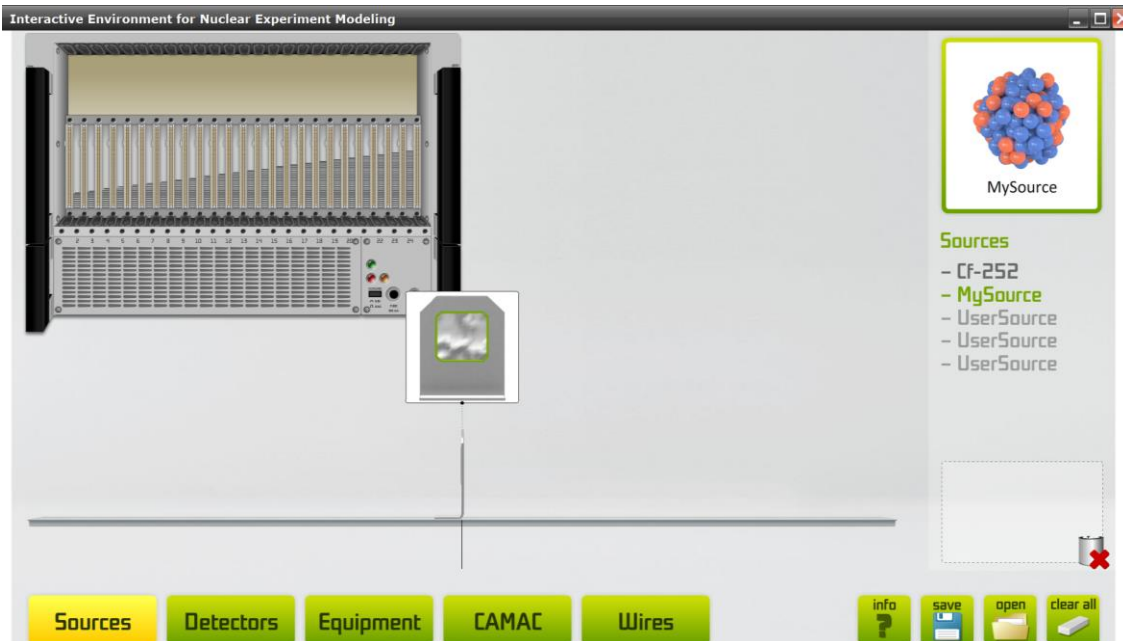
The user detector (MyPIN) integrated into the Setup Builder

3.1. Technology of using of the programmable modules

Loading the Setup Builder for the first time in the source library you can see just one virtual radioactive source – $^{252}_{98}\text{Cf}$. When you start the program with the algorithm of your own module (here, the user source) it appears in the source library as a new menu item, becomes active and ready for use.



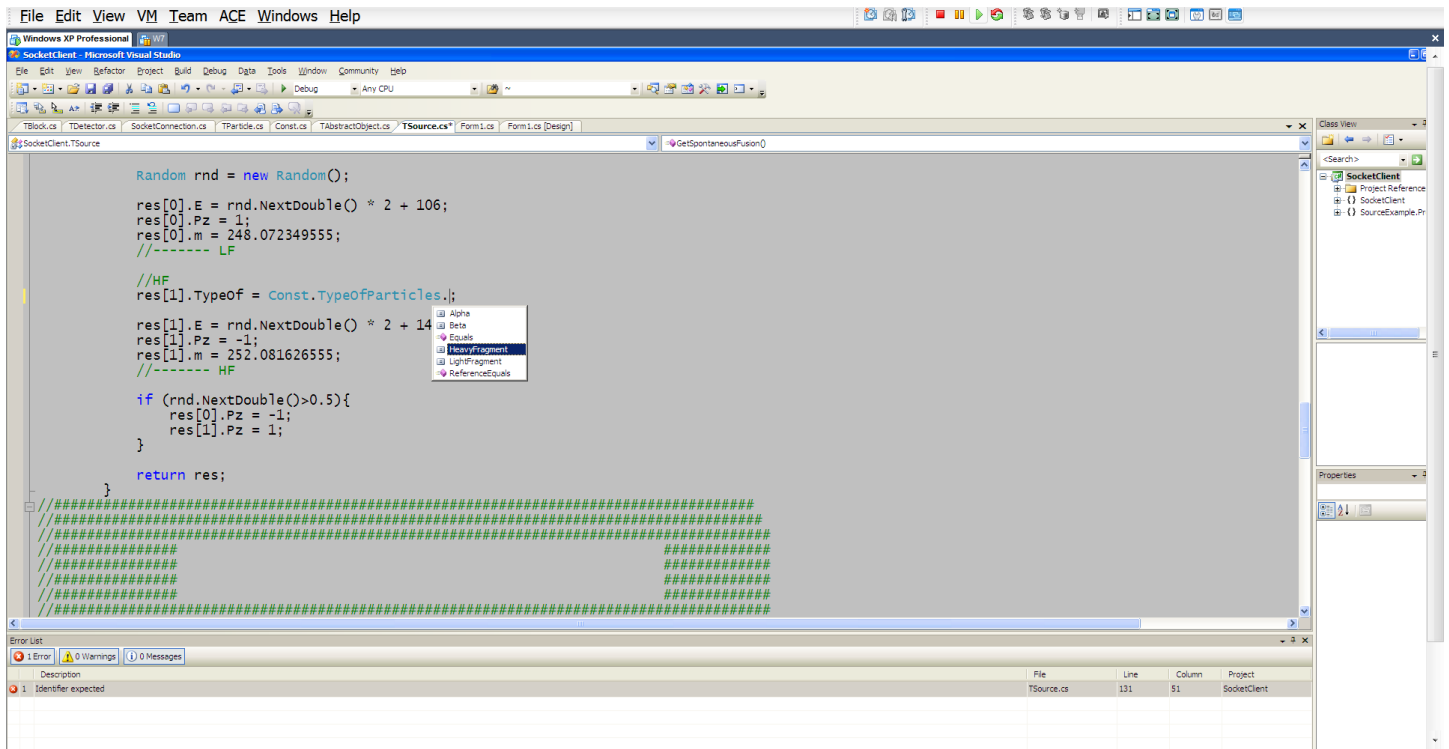
The source library without the user source in the menu



The source library with the user source in the menu

The same way you can add other types of the programmable modules such as user detectors and user CAMAC blocks.

Data exchange is based on the client-server technology using network sockets (https://en.wikipedia.org/wiki/Network_socket).



Development of the user source in Visual Studio (programming language C#)

In the general case the **algorithm of the client work** is following:

- the client gets data from the server;
- the client processes the received data;
- the client sends the processed data to the server.

Examples with the source code (C#, Delphi) are in the folder:

Path_to_virtual_lab\data\lab\Constructor_UserBlocks\

Input /output data for the programmable modules are:

- parameter delimiter of the element – \$
- end-of-data indicator – #
- unit separator – &

3.3. Source:

Input – no data.

Output data in the string format:

- for one particle:

Name:CfPavel\$N:0\$T:0\$TypeOf:Alpha\$E:6.75172073475631\$Px:0\$Py:0\$Pz:1\$m:4.001506179125#

- for several particles:

Name:CfPavel\$N:0\$T:0\$TypeOf:LF\$E:106.894644977941\$Px:0\$Py:0\$Pz:1\$m:248.072349555&Name:CfPavel\$N:0\$T:0\$TypeOf:HF\$E:142.027697643744\$Px:0\$Py:0\$Pz:-1\$m:252.081626555#

Name:CfPavel – the name of a source.

Then the **information about current emitted particles** follows:

N:0 – a number of the event (it is not used yet).

T:0 – event time (it is not used yet).

TypeOf:LF – the particle type. There are possible variants: *Alpha*, *LF*, *HF* and types created by the user.

E:106.894644977941 – the particle energy in MeV.
Px:0\$Py:0\$Pz:1 – directions of the particle escape.
Pz:1 (escape to the right), *Pz:-1* (escape to the left).
Px and *Py* are programmed for future but are not used yet.
m:248.072349555 – the particle mass in amu.

3.4. Detector:

Input – one or several particles in the string format:

Name:CfPavel\$N:0\$T:0\$TypeOf:LF\$E:106.894644977941\$Px:0\$Py:0\$Pz:1\$m:248.072349555&Name:CfPavel\$N:0\$T:0\$TypeOf:HF\$E:142.027697643744\$Px:0\$Py:0\$Pz:-1\$m:252.081626555#

Name:CfPavel – the name of a detector.

Then the **information about current emitted particles** follows:

N:0 – a number of the event (it is not used yet).
T:0 – event time (it is not used yet).
TypeOf:LF – the particle type. There are possible variants: *Alpha*, *LF*, *HF* and types created by the user.
E:106.894644977941 – the particle energy in MeV.
Px:0\$Py:0\$Pz:1 – directions of the particle escape.
Pz:1 (escape to the right), *Pz:-1* (escape to the left).
Px and *Py* are programmed for future but are not used yet.
m:248.072349555 – the particle mass in amu.

Output data in the string format:

– Signal:
0\$0\$1\$10\$11\$10\$11\$0\$.....#

Where numbers represent the array which describes the signal parameters (T_i – time, A_i – amplitude).

3.5. CAMAC block:

Input in the string format:

– Signal
0\$0\$1\$10\$11\$10\$11\$0\$.....#

Where numbers represent the array which describes the signal parameters (T_i – time, A_i – amplitude).

Output data in the string format:

– Signal:
0\$0\$1\$10\$11\$10\$11\$0\$.....#

Where numbers represent the array which describes the signal parameters (T_i – time, A_i – amplitude).