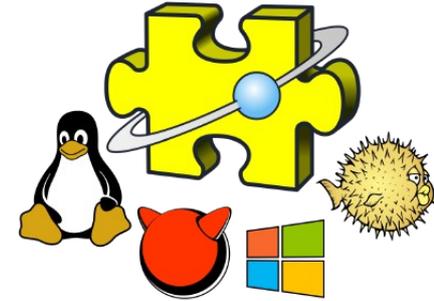


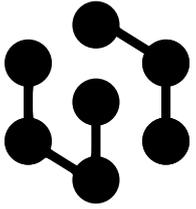
SIBERIAN  
FEDERAL  
UNIVERSITY

LDNova



# Развитие и применения Оберон-инструментария

Иван Андреевич Денисов  
d\_ivan@mail.ru



# 255 модулей и новые целевые микроконтроллеры в O7



<https://github.com/aixp/O7>

Commits on Oct 31, 2021

**STM32L1Flash**  
aixp committed on 31 Oct 2021 7e159b8

Commits on Oct 6, 2021

**STM32L1Flash**  
aixp committed on 6 Oct 2021 3f7c2d5

Commits on Sep 22, 2021

**STM32L1**  
aixp committed on 22 Sep 2021 3115a2e

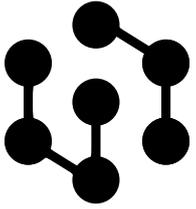
**STM32L1**  
aixp committed on 22 Sep 2021 e4dc04a

Commits on Aug 8, 2021

**STM32F4 unification: STM32F4System implementation**  
aixp committed on 9 Aug 2021 96caccd

**DBGMCU registers added**  
aixp committed on 8 Aug 2021 e7111c0

**ARMv7E-M: number of imported modules limit increased from 15 to 255**  
aixp committed on 8 Aug 2021 e64c8ff



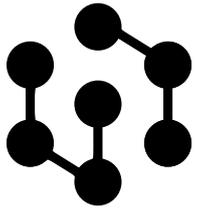
# BlackBox Haiku

<https://github.com/X547/BlackBox-Haiku>



The screenshot displays the Haiku desktop environment with several windows open:

- Log:** A window showing compilation logs for various system components like DevLinkerBin, DevLinkChk, DevInspector, etc.
- Tool (Test)PlotTests:** A window showing the configuration for the TestPlotTests tool, including the SetBasePath and a list of test modules.
- Trap:** A window displaying a memory dump or stack trace, listing various system objects and their addresses.
- BlackBox:** A window showing a file browser view of the BlackBox directory, listing files like Dev, Escher, Form, Gui, Host, Obx, Raster, Std, System, Test, Text, Zlib, and various .odc files.
- Plot:** A window showing a graph with four sine waves in red, green, blue, and purple, plotted on a coordinate system.
- HAIKU logo - black on transparent - big.png:** A window showing the Haiku logo, which consists of the word "HAIKU" in a stylized font with a green leaf, an orange fish, and a yellow leaf.
- Find / Replace:** A window with input fields for finding and replacing text, and checkboxes for "Word Begins With", "Word Ends With", "Ignore Case", and "Reverse Direction".



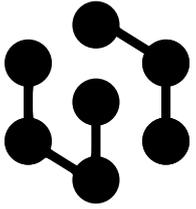
# Развитие Блэббокса



- Интеграция плиточного интерфейса
- Создание плиток меню, панелей и статусной строки
- Перестройка хост-концепции Блэббокса для кросс-разработки
- Поддержка растровых изображений на уровне райдера графического порта



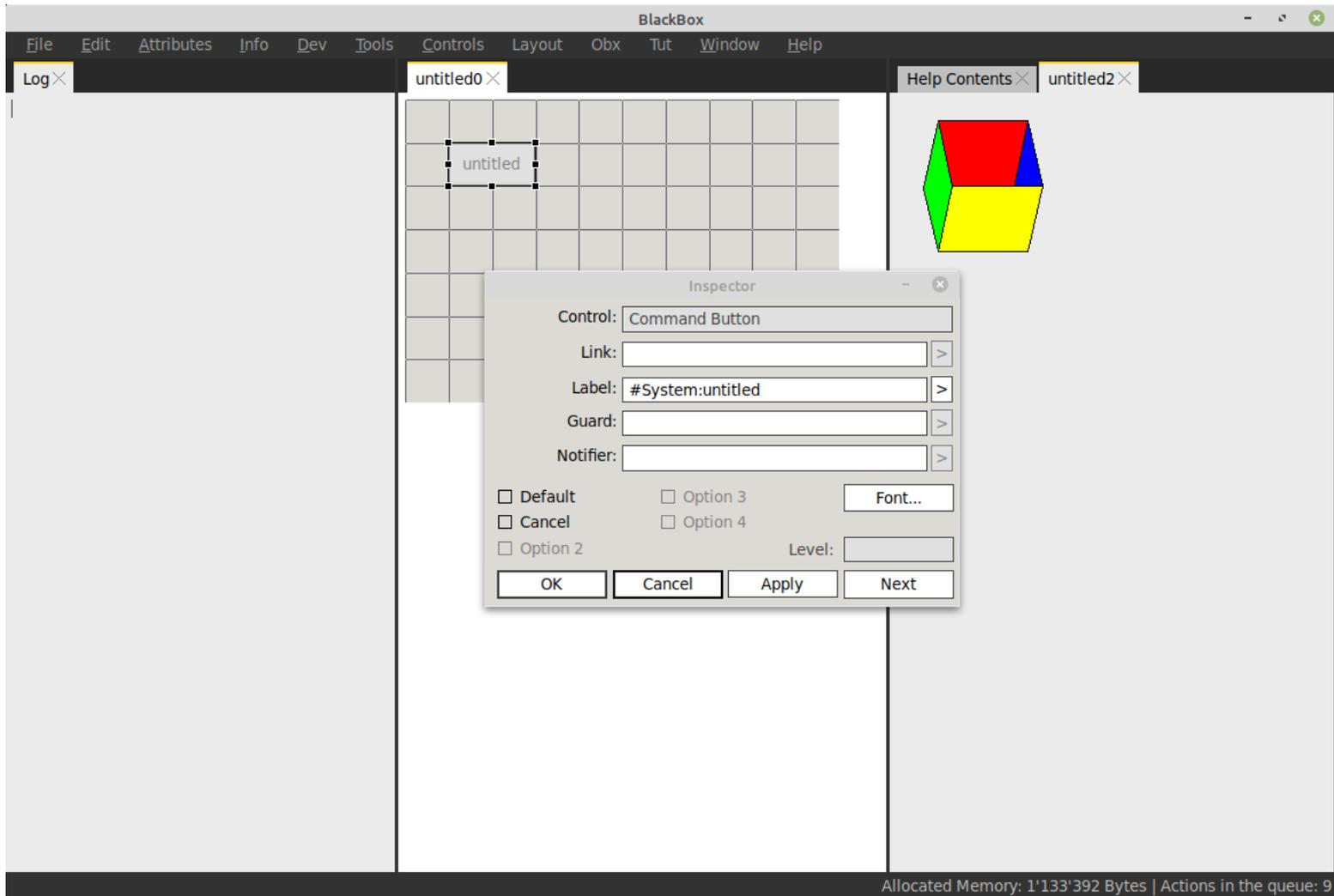
LDNova

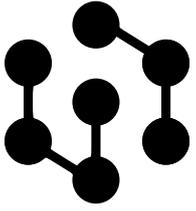


# Блэкбокс 2.0 (альфа версия)



<https://blackbox.oberon.org>

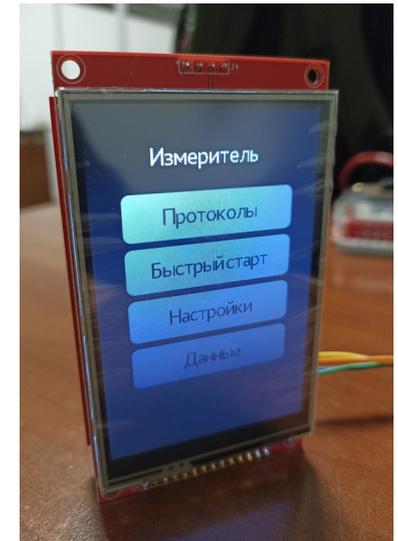


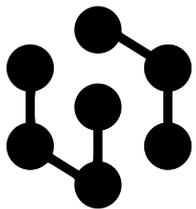


# Среда разработки для микроконтроллеров

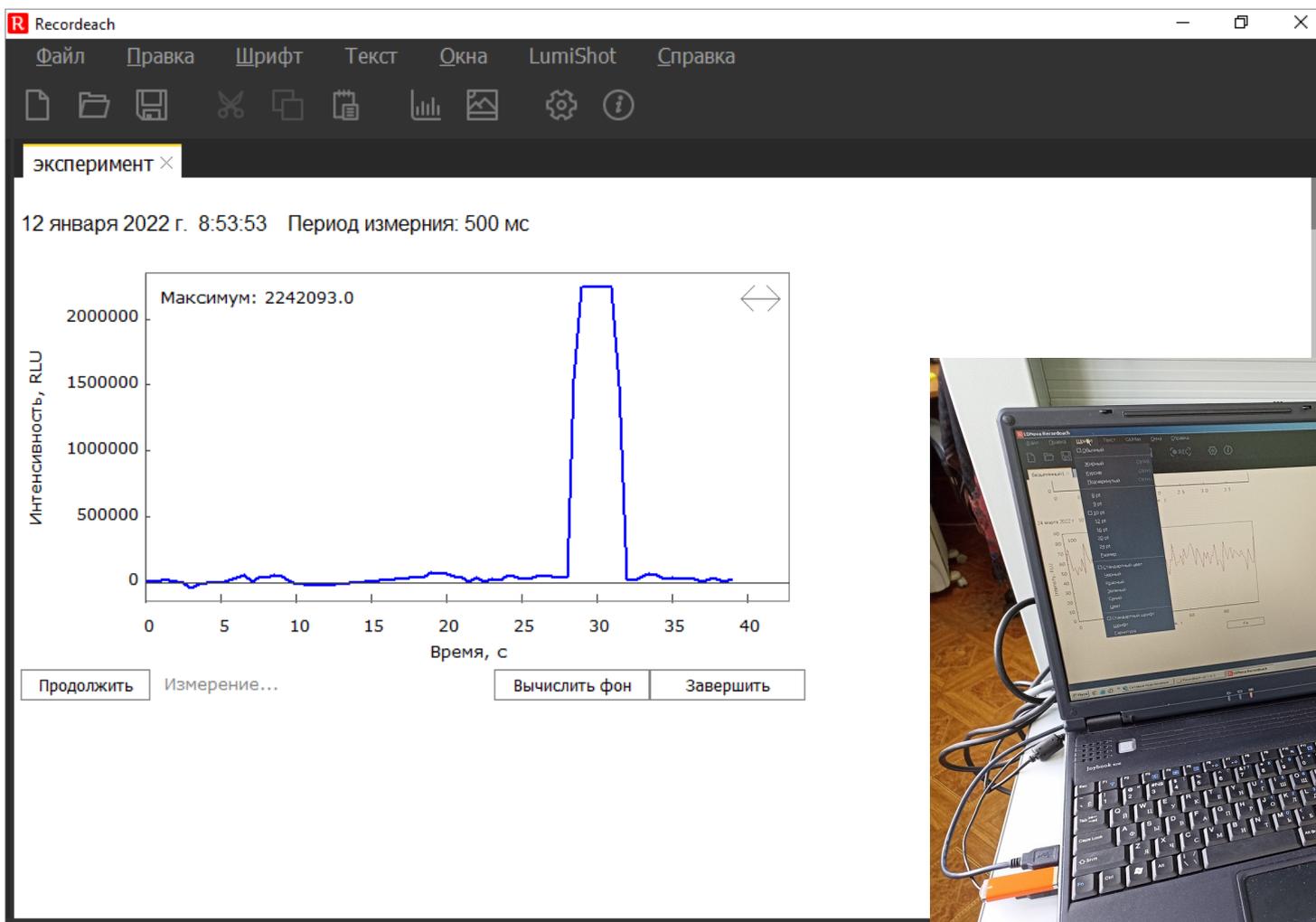


[recordino.ldnova.com](http://recordino.ldnova.com)

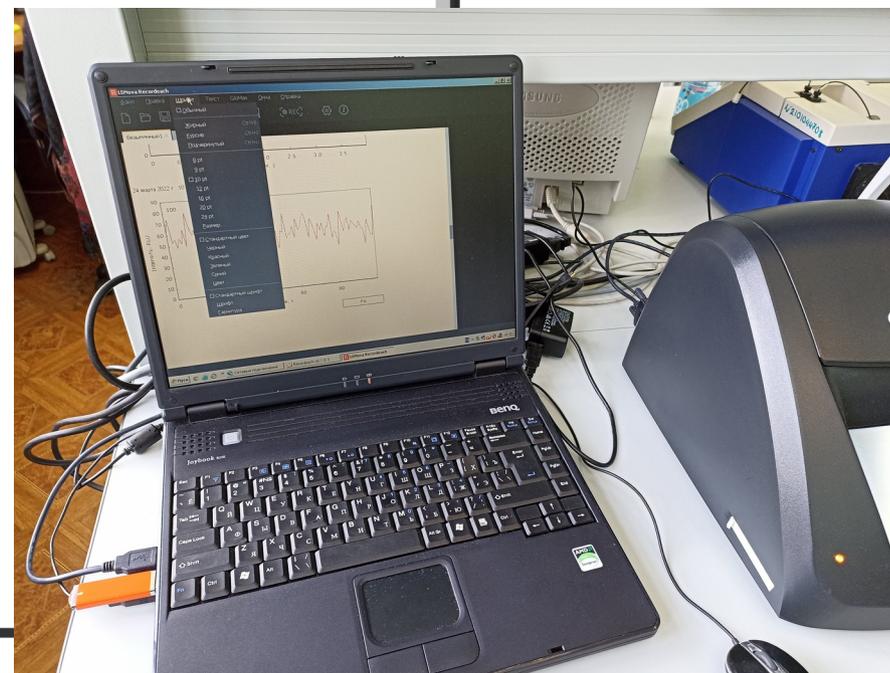


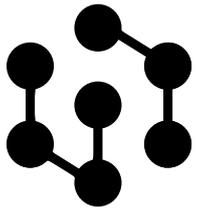


# Приложение для сбора данных с измерительных устройств



[www.ldnova.com](http://www.ldnova.com)





# Сотрудничество с Мобатек



## Mathieu Westerweele

Managing Director  
Co-founder of Mobatec  
Software Developer  
Process Modelling Expert



## Skills

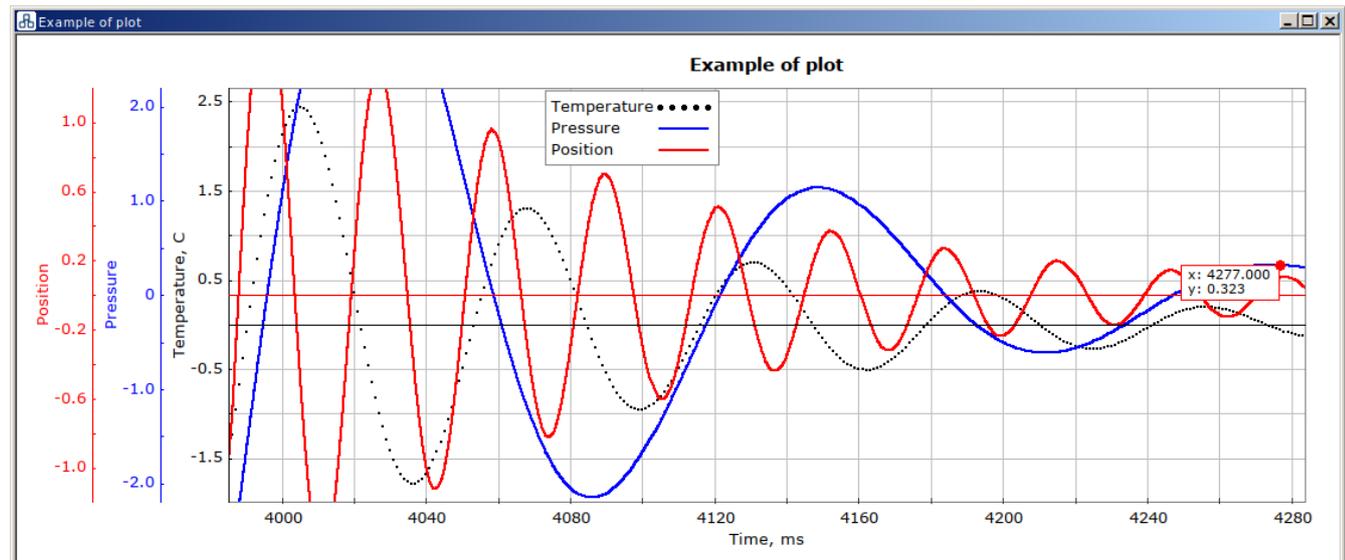
Modelling

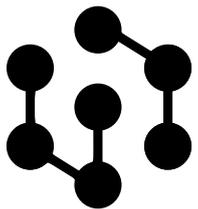
Simulation

Management

Programming

[www.mobatec.nl](http://www.mobatec.nl)





# Блэкбокс в инженерной науке

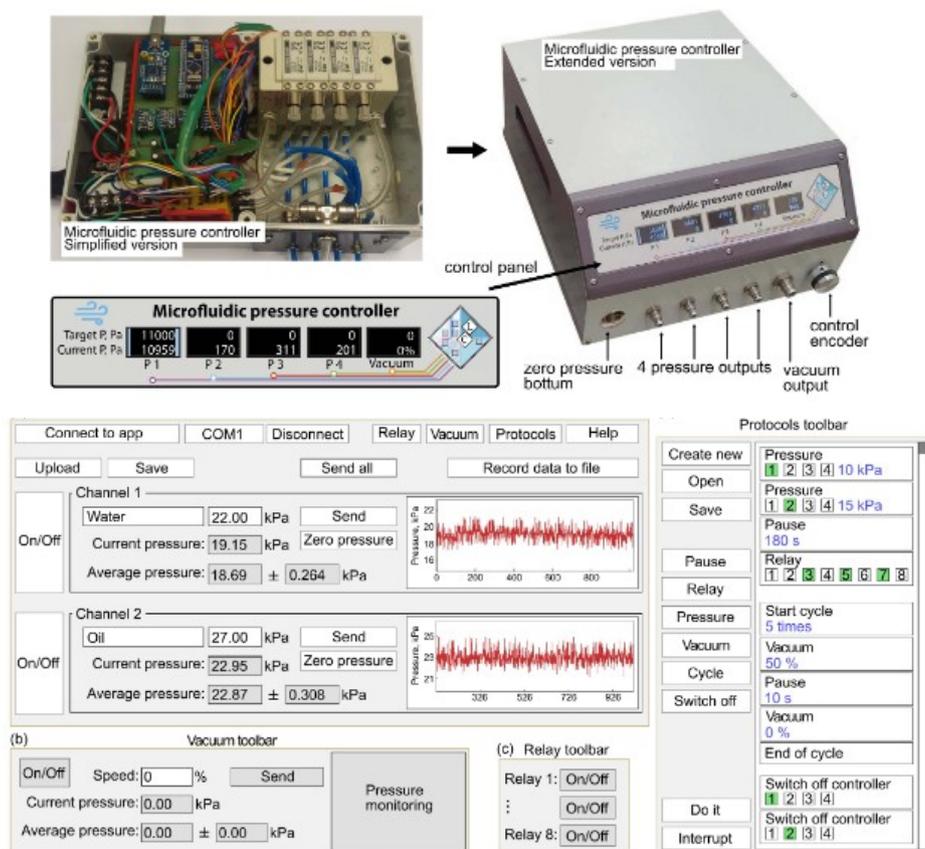
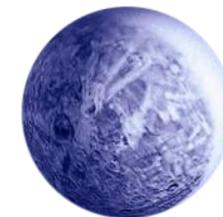
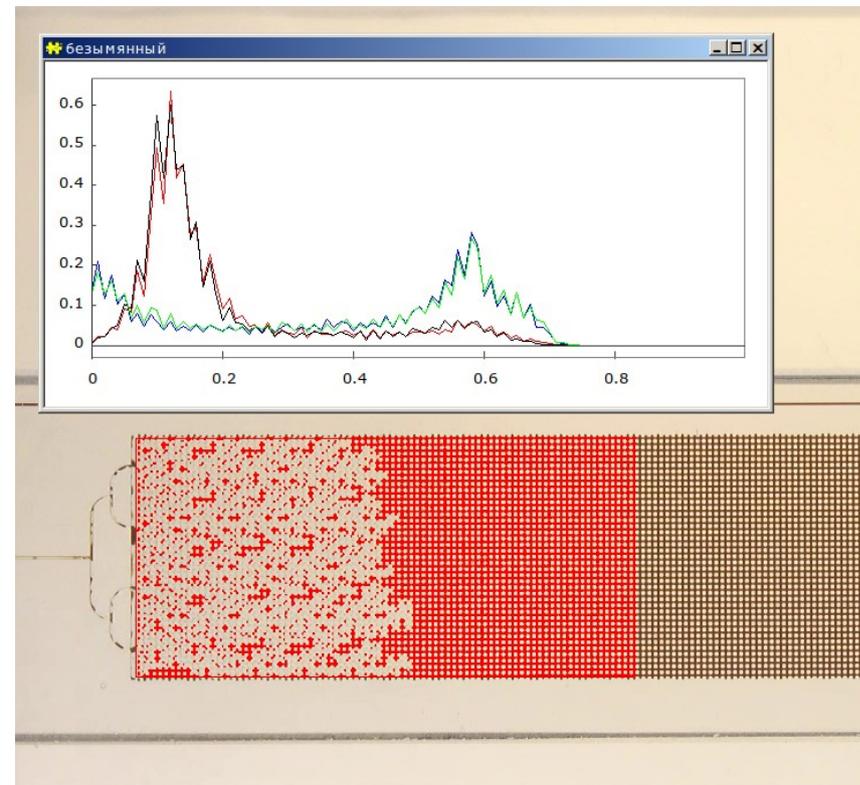
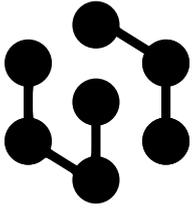


Fig. 3. Software interface of the microfluidic pressure controller. (a) Main window of the interface that allows to control and monitor pressures in each channel. (b) Toolbar for controlling the integrated vacuum pump. (c) Toolbar for controlling external devices via relay outputs. (d) Toolbar for defining operation sequences.



Filatov N.A., Denisov I.A., Evstrapov A.A., Bukatin A.S. Open-Source Pressure Controller Based on Compact Electro-Pneumatic Regulators for Droplet Microfluidics Applications // IEEE Trans. Instrum. Meas. 2022. Vol. 71. P. 1–10.

Pryazhnikov M.I., Minakov A.V., Pryazhnikov A.I., Denisov I.A., Yakimov A.S. Microfluidic Study of the Effect of Nanosuspensions on Enhanced Oil Recovery // Nanomaterials, 2022. P. 15.



# Оберон Онлайн

<https://visual.sfu-kras.ru>



Visual Главная О проекте Регистрация Войти English

## Сердце

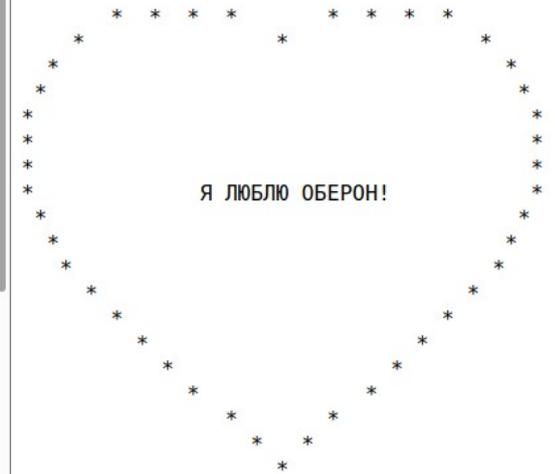
Псевдографика

Модель опубликована

Исходный код **Init**

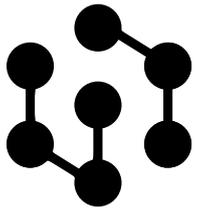
Оберон & OberonJS Результат

```
1 MODULE Heart;
2 IMPORT Log;
3 VAR i: INTEGER;
4
5 PROCEDURE Print(spc1,chr1,spc2,chr2,spc3:INTEGER);
6 BEGIN
7 FOR i:=1 TO spc1 DO Log.String(" ");END;
8 FOR i:=1 TO chr1 DO Log.String("**");END;
9 FOR i:=1 TO spc2 DO Log.String(" ");END;
10 FOR i:=1 TO chr2 DO Log.String("**");END;
11 FOR i:=1 TO spc3 DO Log.String(" ");END;
12 Log.Ln;
13 END Print;
14
15 PROCEDURE Print2(spc1,chr1,spc2, chr2,spc3:INTEGER );
16 BEGIN
17 FOR i:=1 TO spc1 DO Log.String(" ");END;
18 FOR i:=1 TO chr1 DO Log.String("**");END;
19 FOR i:=1 TO spc2 DO Log.String(" ");END;
20 FOR i:=1 TO chr2 DO Log.String("**");END;
21 FOR i:=1 TO spc3 DO Log.String(" ");END;
22 END Print2;
23
24 BEGIN
25 Log.Ln;
26 Print2(7,1,2,1,2); Print2(0,1,2,1,1); Print2(6,1,2,1,2); Print(0,1,2,1,1); Print2(4,1,15,1,0); Print2(0,0,15,1,4);
```



Исполнено за 0.059 с.

проверка индексов в массивах



# Оберон Онлайн



## WebDecoder

Модель опубликована

### Исходный код Init

```

134 WHILE j # 256 DO revCode[j] := -1; INC(j) END;
135 code[0] := "."; revCode[ORD(".")] := 0; code[1] := ","; revCode[ORD(",")] := 1;
136 i := 2; j := ORD("0");
137 WHILE j <= ORD("9") DO code[i] := CHR(j); revCode[j] := i; INC(i); INC(j) END;
138 j := ORD("A");
139 WHILE j <= ORD("Z") DO code[i] := CHR(j); revCode[j] := i; INC(i); INC(j) END;
140 j := ORD("a");
141 WHILE j <= ORD("z") DO code[i] := CHR(j); revCode[j] := i; INC(i); INC(j) END;
142 ASSERT(i = 64)
143 END InitCodes;
144
145 BEGIN
146   InitCodes
147   (* Test part: *)
148   ;
149   str := "StdCoder.Decode IJ4Sot0RsF5.,0 ov5R..X....F00,NHH9PTvQ176H8RdtQ,M
150   R9vM,dP9vR,tQdPRBFN1,Qi....
151   --- end of encoding ---";
152   FOR n := 0 TO LEN(str) - 1 DO bytestr[n] := ORD(str[n]) END;
153   IF Decode(bytestr, n, name) THEN Log.String("Done!");
154   Log.Ln; Log.String("size of file = "); Log.Int(n); Log.Ln;
155   Log.String("Name = "); Log.String(name); Log.Ln
156 ELSE
157   Log.String("Error")
158 END
159 END WebDecoder.

```

Компилировать    Запустить    Исполнено за 0.014 с.

проверка индексов в массивах

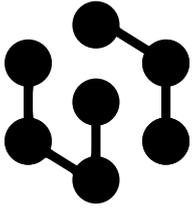
Оберон & OberonJS    Результат   

Done!  
size of file = 35  
Name = Text.txt

Олег Чередниченко  
••• печатает

Всё получилось, Иван 23:20

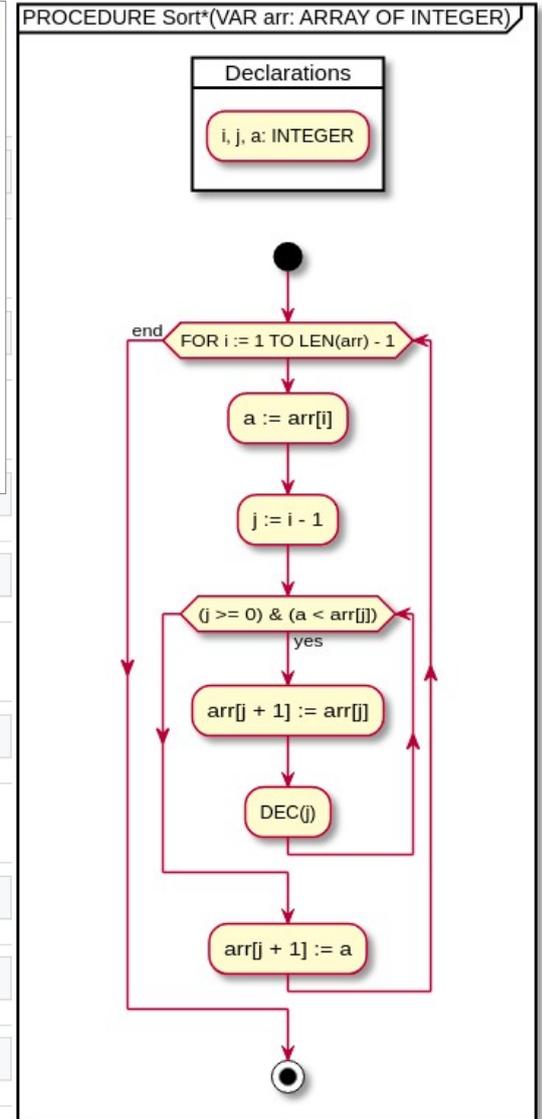
Онлайн-Оберон - инструмент что надо! 23:20

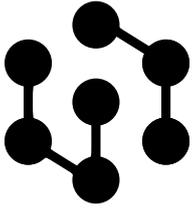


master

- Commits on Apr 4, 2022
  - Исправлена генерация присваивания односимвольной строки, за**  
ComdivByZero committed 19 days ago
- Commits on Feb 24, 2022
  - Корректировка changelog для android-сборщика**  
ComdivByZero committed on 24 Feb
- Commits on Feb 23, 2022
  - Устранено предупреждение о неиспользовании**  
ComdivByZero committed on 23 Feb
  - Обозначены версии транслятора 0.0.9 и библиотеки 0.0.8**  
ComdivByZero committed on 23 Feb
- Commits on Feb 19, 2022
  - Исправлено определение статической инициализированности с необходимос...**  
ComdivByZero committed on 19 Feb
- Commits on Feb 18, 2022
  - Считывание языка для пользовательского интерфейса в Windows**  
ComdivByZero committed on 19 Feb
  - Указание Utf8 для ввода и вывода в Windows**  
ComdivByZero committed on 19 Feb
  - Частичная обёртка для windows.h**  
ComdivByZero committed on 19 Feb

```
PROCEDURE Sort*(VAR arr: ARRAY OF INTEGER);
VAR i, j, a: INTEGER;
BEGIN
  FOR i := 1 TO LEN(arr) - 1 DO
    a := arr[i];
    j := i - 1;
    WHILE (j >= 0) & (a < arr[j]) DO
      arr[j + 1] := arr[j];
      DEC(j)
    END;
    arr[j + 1] := a
  END
END Sort;
```





# Ofront+

<https://zx.oberon.org/ofrontplus>



- Интеграция с FreeOberon
- Улучшается совместимость с КП, с O7
- Сделана автосборка через системы непрерывной интеграции
- Набор раскрыточных файлов в отдельном хранилище

Commits on Apr 12, 2022

- \* GetStartDir renamed GetAppDir, GetStartDirW renamed GetAppDirW**  
kekcleader committed 10 days ago ✓ 8cf9f30
- + Platform.GetStartDirW on Windows**  
kekcleader committed 10 days ago 7efa9f4

Commits on Apr 11, 2022

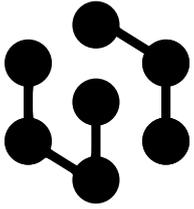
- + CmdArgs.GetEnvN on Windows; \* GetEnvCount renamed EnvCount**  
kekcleader committed 11 days ago ✓ cbc9254

Commits on Apr 5, 2022

- \* Cyclic type definition.**  
Oleg-N-Cher committed 18 days ago ✓ ea55959
- \* Cyclic type definition.**  
Oleg-N-Cher committed 18 days ago 6246c7d

Commits on Apr 4, 2022

- \* Cyclic type definition.**  
Oleg-N-Cher committed 19 days ago ✓ e50fc02



# Eigen Compiler Suite



<https://ecs.openbrace.org>

**EIGEN COMPILER SUITE**

[Overview](#) [Activity](#) [Roadmap](#) [Issues](#) [News](#) [Forums](#)

## News

### Version 0.0.36 available

Added by [Florian Negele](#) 22 days ago

Development Version 0.0.36 of the Eigen Compiler Suite is available for [evaluation](#). It fixes various issues and includes the following improvements.

**Oberon:**

- Added basic exception handling facility
- Added allocation failure exceptions
- Added compatible set types
- Added typed set constructors

### Version 0.0.35 available

Added by [Florian Negele](#) 2 months ago

Development Version 0.0.35 of the Eigen Compiler Suite is available for [evaluation](#). It fixes various

**General:**

- Added utility for viewing binary files

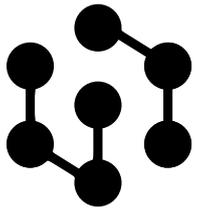
**Oberon:**

- Added generic set types
- Added generic utility functions
- Added generic array operations
- Added support for structured result types
- Added note about implementation limitations
- Fixed variable record parameter compatibility
- Improved diagnostics for generic modules
- Improved temporary variable allocation

## Chapter 7 Oberon

Oberon is a general-purpose programming language based on Pascal and Modula-2. It supports type extension with type-bound procedures which makes it an object-oriented language. This chapter describes Oberon and its implementation by the Eigen Compiler Suite.

- 7.1 [Introduction](#)
- 7.2 [Implementation-Defined Behavior](#)
  - 7.2.1 [Vocabulary and Representation](#)
  - 7.2.2 [Declarations and Scope Rules](#)
  - 7.2.3 [Type Declarations](#)
  - 7.2.4 [Variable Declarations](#)
  - 7.2.5 [Expressions](#)
  - 7.2.6 [Procedure Declarations](#)
  - 7.2.7 [Predeclared Procedures](#)
  - 7.2.8 [Modules](#)
  - 7.2.9 [The Module SYSTEM](#)
  - 7.2.10 [The Oberon Environment](#)
- 7.3 [The Oberon Library](#)




## Herschel

Component Pascal x64 compiler for BlackBox Component Builder.

- Home
- About
- Donate
- Download



2021

## NEW, IS and SYSTEM.TYP

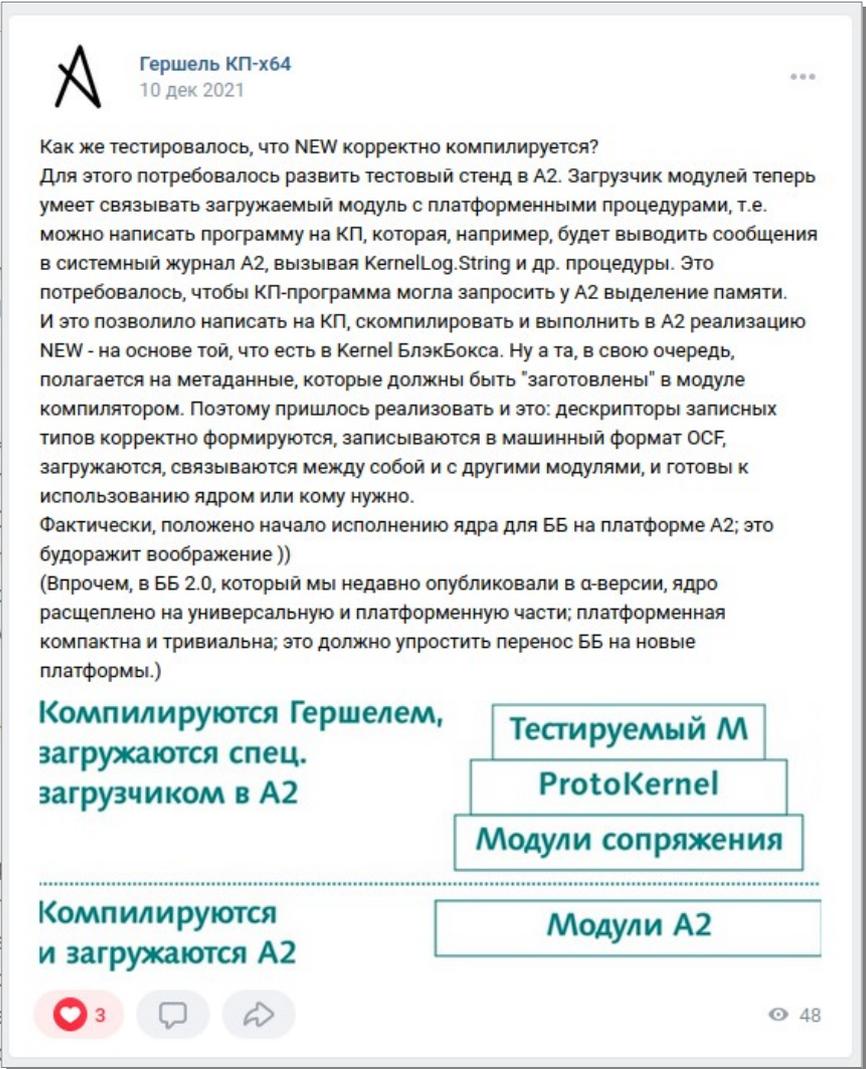
09 Dec 2021

As of December 2021, these features have been implemented in the new version of the compiler. NEW applied to record parameters; relational operator IS and function procedure SYSTEM.TYP applied to pointer and pointer types:

```
TYPE T = POINTER TO EXTENSIBLE RECORD END; Ext =
END; VAR p: T; x: LONGINT; BEGIN NEW(p); IF p IS Ext T
ELSE x := SYSTEM.TYP(T) END; x := SYSTEM.TYP(p); END
Of course, NEW is not implemented by the compiler, but in the
environment mandated by the Language Report. What the loader
transform NEW(p) into calls to a procedure of the following
form:
```

```
NEW = PROCEDURE (tag: ADDRESS): T;
where *tag *is the address of the type descriptor of the type
the compiler performs the following transformation:
```

```
NEW(p) => p := @Kernel.NewRec(SYSTEM.TYP(T))
The peculiarity is that subject module containing NEW(p) must
Kernel explicitly, and cannot import Kernel implicitly: in the
procedure implementing NEW(p) may be defined in an external
is done at module load time by the loader. So, the compiler
the linking loader to be able to perform the linking. This
changed in the updated OCF format, as has the format of the
and import fixups in the machine code and module metadata.
```



**Гершель КП-х64**  
10 дек 2021

Как же тестировалось, что NEW корректно компилируется?  
Для этого потребовалось развить тестовый стенд в A2. Загрузчик модулей теперь умеет связывать загружаемый модуль с платформенными процедурами, т.е. можно написать программу на КП, которая, например, будет выводить сообщения в системный журнал A2, вызывая KernelLog.String и др. процедуры. Это потребовалось, чтобы КП-программа могла запросить у A2 выделение памяти. И это позволило написать на КП, скомпилировать и выполнить в A2 реализацию NEW - на основе той, что есть в Kernel БлэкБокса. Ну а та, в свою очередь, полагается на метаданные, которые должны быть "заготовлены" в модуле компилятором. Поэтому пришлось реализовать и это: дескрипторы записных типов корректно формируются, записываются в машинный формат OCF, загружаются, связываются между собой и с другими модулями, и готовы к использованию ядром или кому нужно.  
Фактически, положено начало исполнению ядра для ББ на платформе A2; это будоражит воображение ))  
(Впрочем, в ББ 2.0, который мы недавно опубликовали в alpha-версии, ядро расщеплено на универсальную и платформенную части; платформенная компактна и тривиальна; это должно упростить перенос ББ на новые платформы.)

**Компилируются Гершелем, загружаются спец. загрузчиком в A2**

**Тестируемый M ProtoKernel Модули сопряжения**

**Компилируются и загружаются A2**

Модули A2

3 48