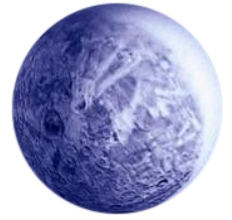
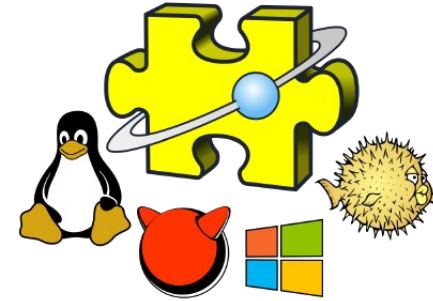


SIBERIAN
FEDERAL
UNIVERSITY

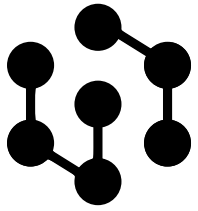
LDNova



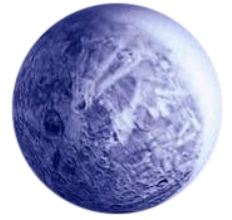
Применение Оберона для разработки программно-аппаратных комплексов

Иван Андреевич Денисов

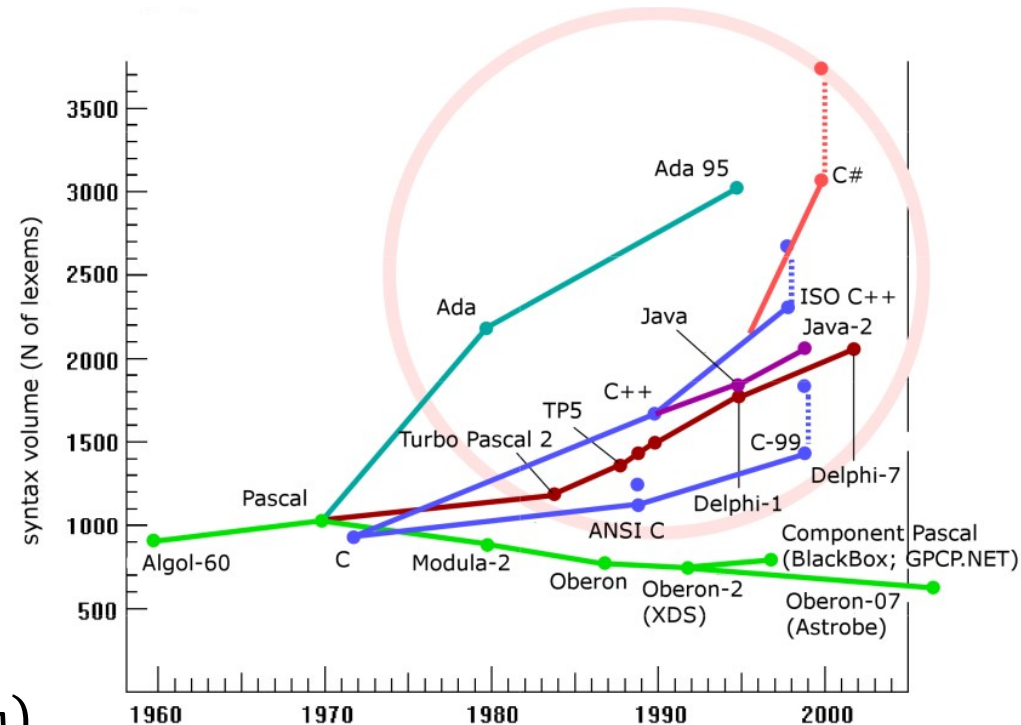
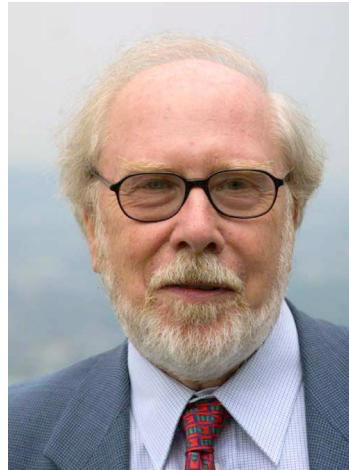
iadenisov@oberon.org



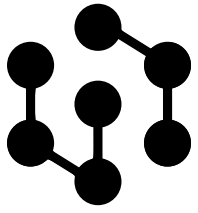
Оберон как академический путь избегания роста сложности в IT



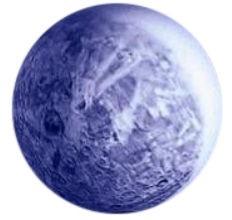
(1963) Euler
(1966) Algol-W
(1968) PL360
(1970) Pascal
(1976) Modula
(1979) Modula-2
(1988) Oberon (с Юргом Гуткнехтом)
(1992) Oberon-2 (с Х.П. Мёссенбёком)
(1997-2001) Component Pascal (аспиранты)
(1994/2015) Lola (для FPGA)
(1999) Oberon-SA (для Strong ARM)
(2007/2013/2016) Revised Oberon (Oberon-07)



ОБЕРОН
простота в основе
надёжности



Моё знакомство с миром программ для встроенных систем

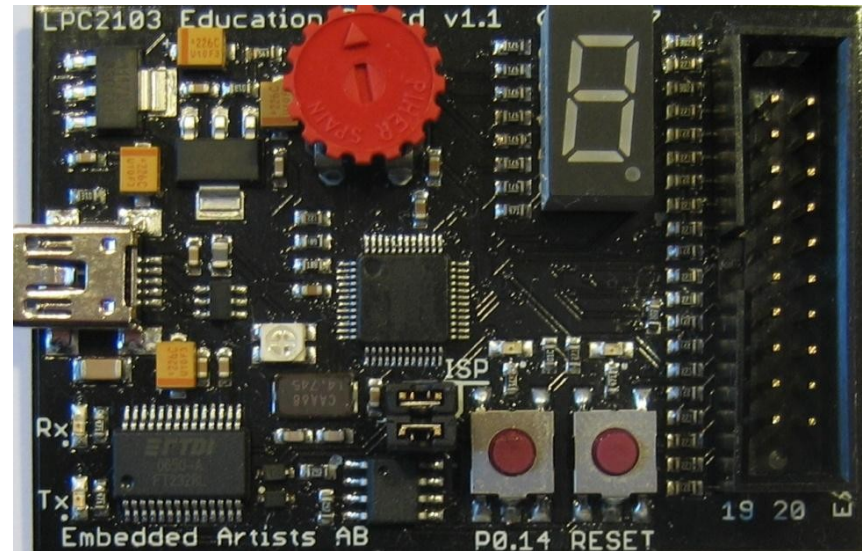
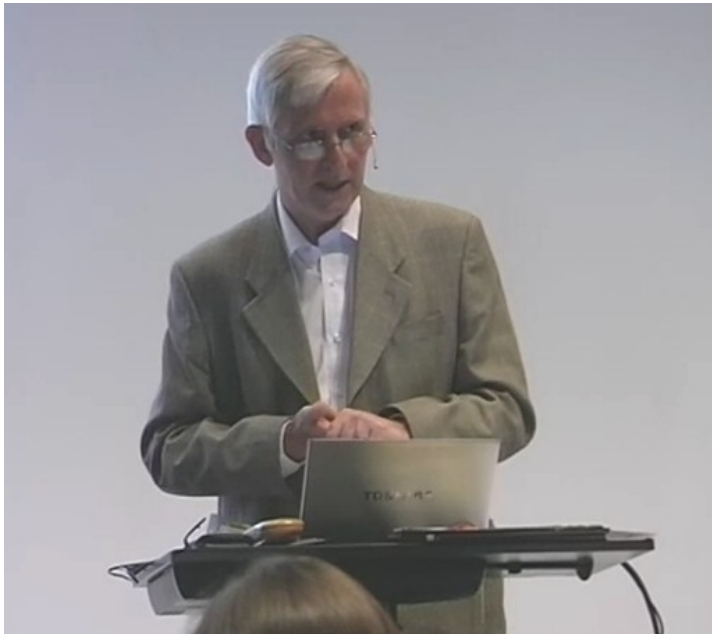


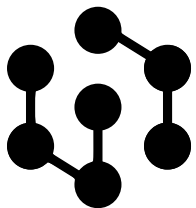
Oberon Day 2011 (27 мая, ETHZ, г. Цюрих, Швейцария)

Chris Burrows, ARM Embedded Development Using Oberon-07

[PDF](#) [VIDEO](#)

<https://video.ethz.ch/conferences/2011/oberon.html>

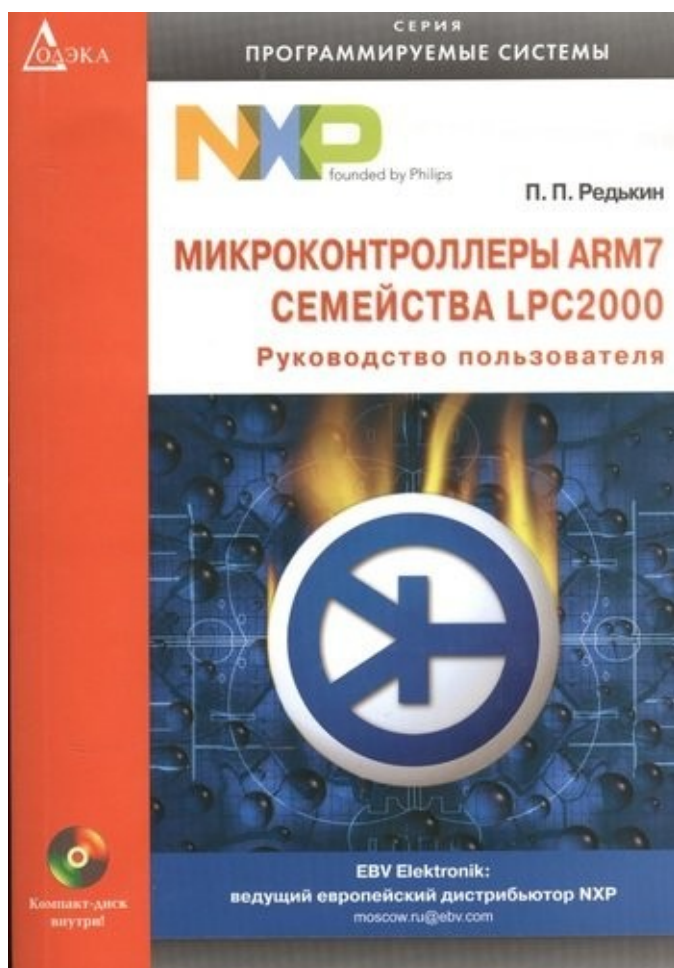




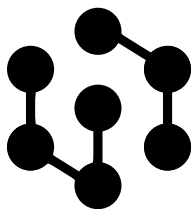
Программы LPC2103 в Astrobe



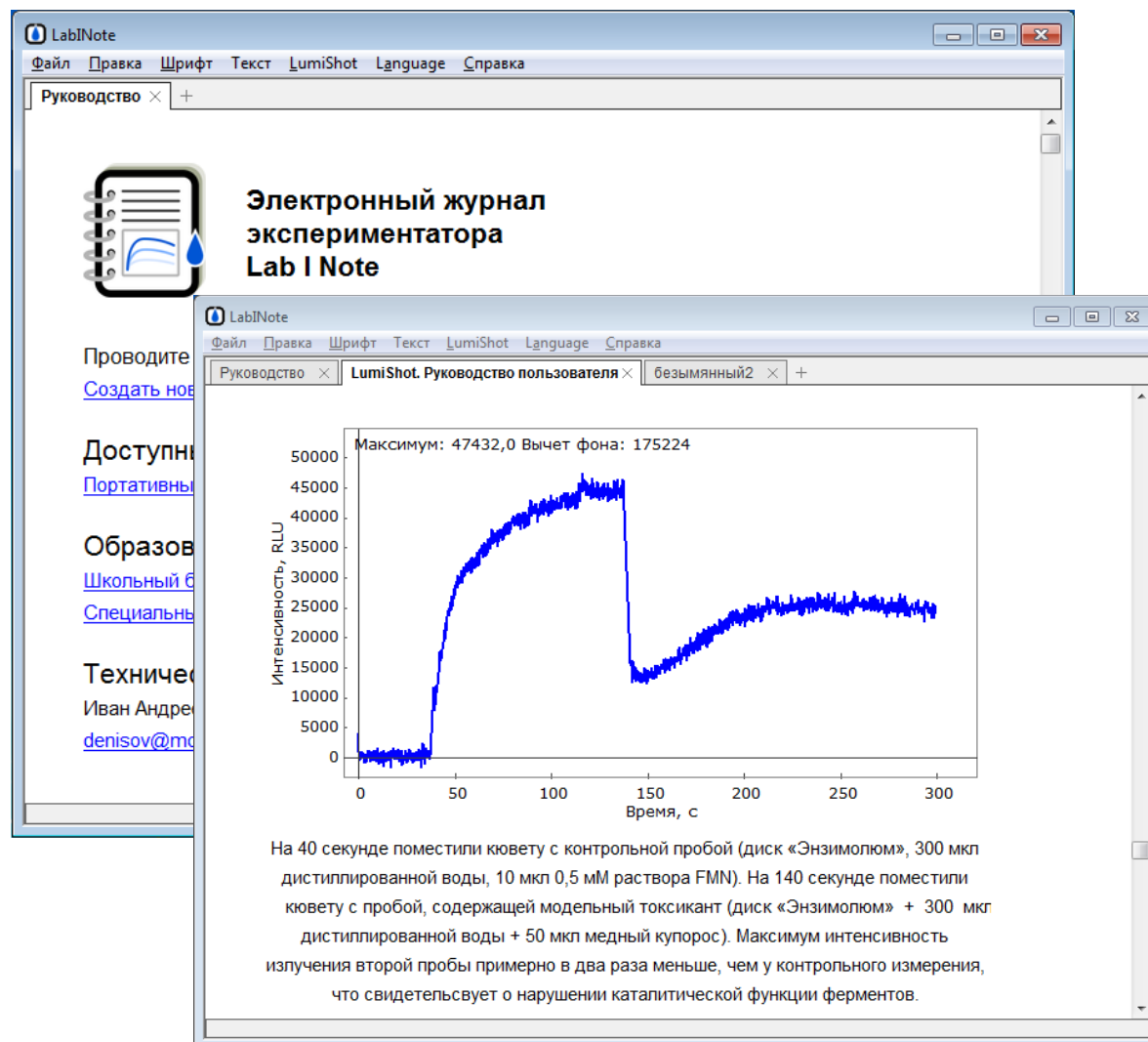
<https://www.astrobe.com>

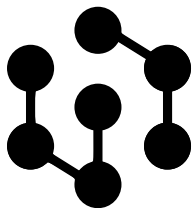


```
Astrobe for LPC2000 Professional Edition
File Edit Search View Project Run Tools Help
Procedures Imports Pressure.mod*
Init
Run
1 MODULE Pressure;
2 IMPORT SYSTEM, LPC, ADC, Main, Out, Timer;
3
4 PROCEDURE Init;
5 VAR s: SET;
6 BEGIN
7   (* Configure pins connected to potentiometer as ADC inputs *)
8   SYSTEM.GET(LPC.PINSEL0, s);
9   s := s + {21,20}; (* P0.10, AD0.3, PINSEL0 21:20 = 11 *)
10  s := s + {23,22}; (* P0.11, AD0.4, PINSEL0 23:22 = 11 *)
11  s := s + {25,24}; (* P0.12, AD0.5, PINSEL0 25:24 = 11 *)
12  SYSTEM.PUT(LPC.PINSEL0, s);
13  SYSTEM.GET(LPC.PINSEL1, s);
14  s := s + {13,12}; (* P0.22, AD0.0, PINSEL1 13:12 = 11 *)
15  s := s + {15,14}; (* P0.23, AD0.1, PINSEL1 15:14 = 11 *)
16  s := s + {17,16}; (* P0.24, AD0.2, PINSEL1 17:16 = 11 *)
17  s := s + {19,18}; (* P0.25, AD0.6, PINSEL1 19:18 = 11 *)
18  s := s + {21,20}; (* P0.26, AD0.7, PINSEL1 21:20 = 11 *)
19  SYSTEM.PUT(LPC.PINSEL1, s);
20  ADC.PowerUp
21 END Init;
22
23 PROCEDURE Run;
24 VAR data: ARRAY 8 OF INTEGER; i: INTEGER;
25 BEGIN
26   WHILE TRUE DO
27     FOR i := 0 TO 7 DO ADC.Read(i, data[i]) END;
28     FOR i := 0 TO 7 DO Out.Int(data[i], 5) END;
29     Out.Ln;
30     Timer.MSecDelay(1000)
31   END
32 END Run;
33 BEGIN
34   Init; Run
35 END Pressure.
```



ПАК «Энзимолюм»





Программирование на кириллице



Кириллица в сборке Центра



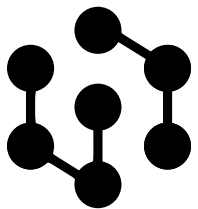
- На кириллице разработаны основные подсистемы **Люми** и **Передача**
- Кириллица синтаксически отделяет компоненты каркаса и компоненты проекта

```
PROCEDURE (соединение: Соединение) Отключить*, NEW;  
BEGIN  
    соединение.отключить := TRUE  
END Отключить;  
  
PROCEDURE (соединение: Соединение) Разорвать*, NEW;  
BEGIN  
    IF соединение.v24 # NIL THEN  
        соединение.установлено := FALSE;  
        активныеСоединения[соединение.порт] := FALSE;  
        соединение.ПриРазрыве;  
        CommV24.Close(соединение.v24);  
        соединение.v24 := NIL  
    END  
END Разорвать;  
  
PROCEDURE (соединение: Соединение) FINALIZE-;  
BEGIN  
    соединение.Разорвать  
END FINALIZE;
```

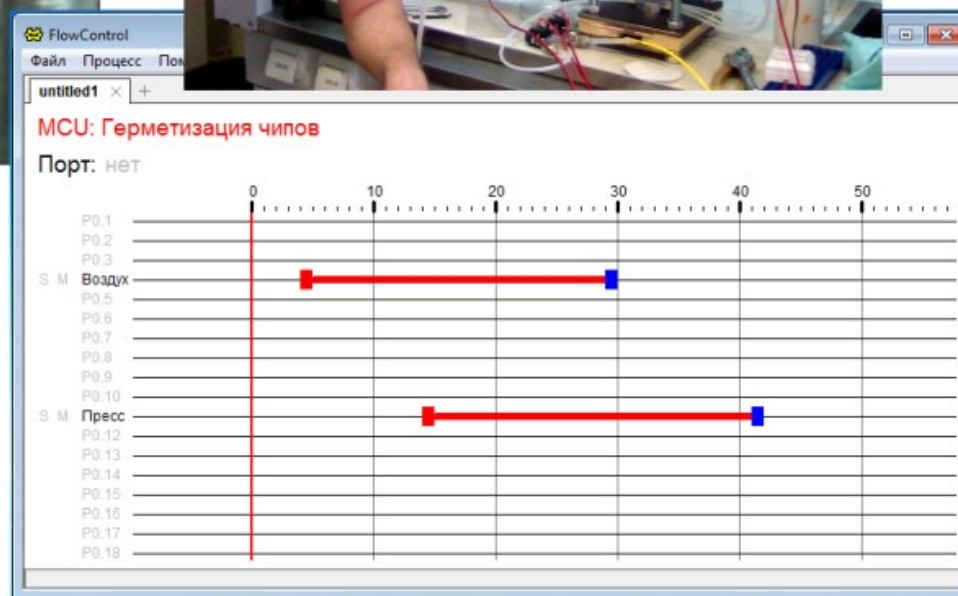
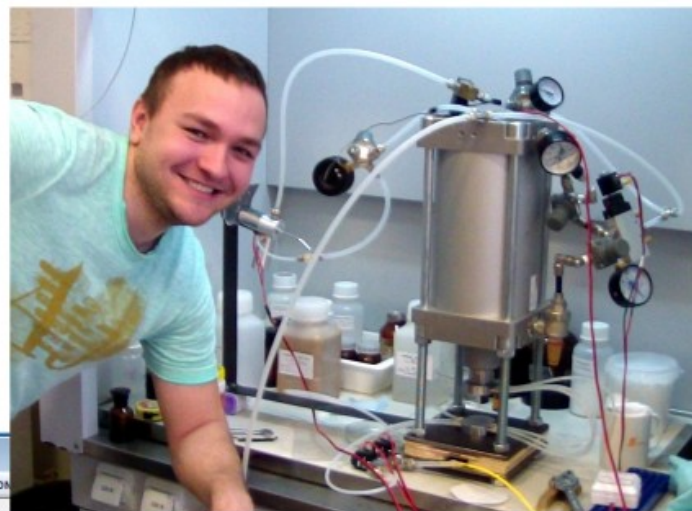
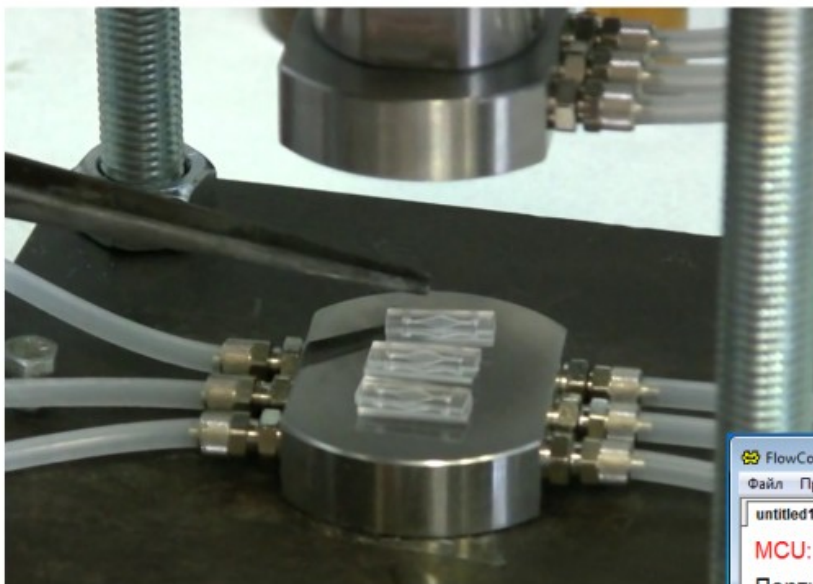
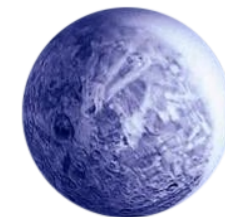
19.09.2015

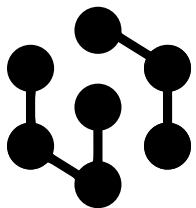
Оберон День в России, ИЯИ РАН, Москва

10



ПАК «Установка герметизации»





Компилятор O7

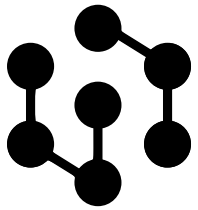
создавался
для критериев
надежности БЛА



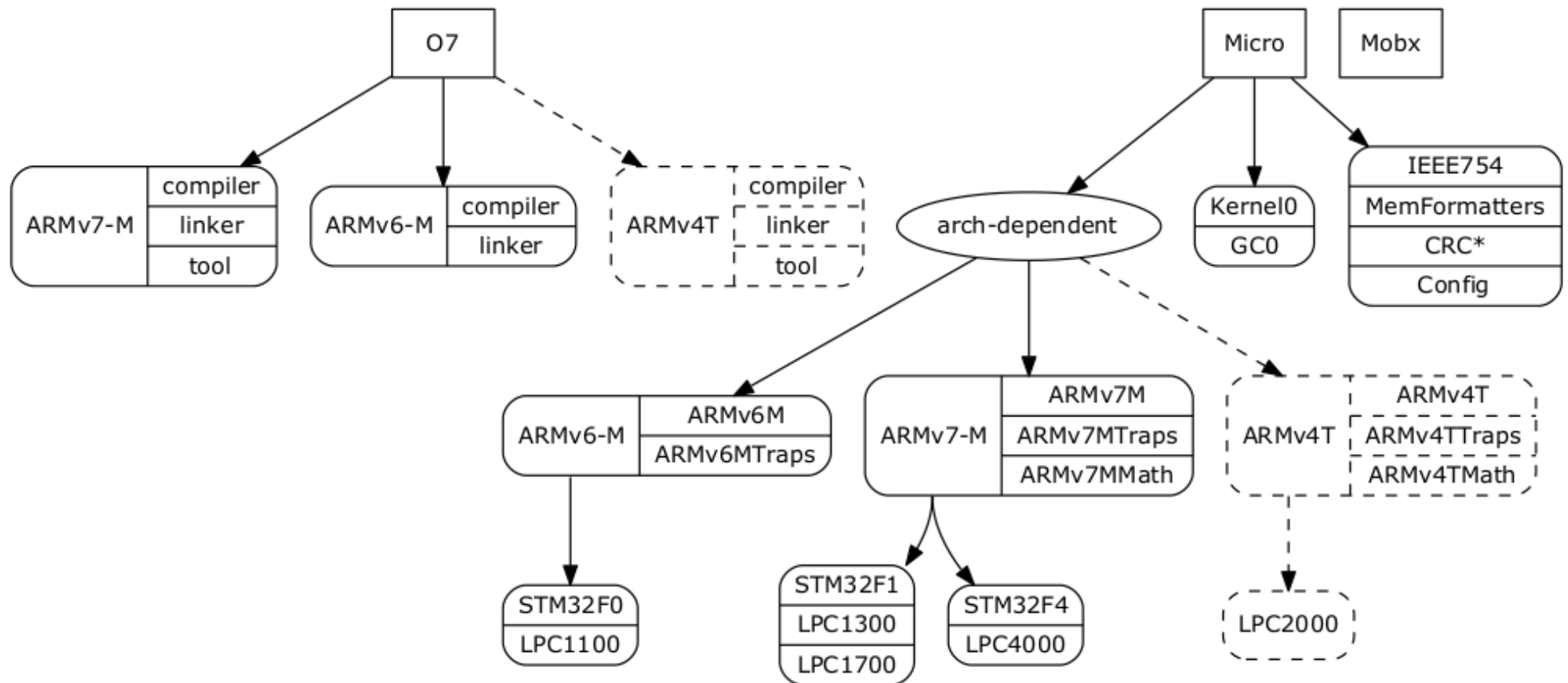
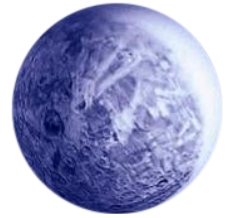
Александр Ширяев

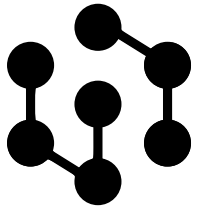
День Оберона 2017 Перспективы

- Поддержка других архитектур
ARMv4T ([NXP LPC2000](#))
[RISC-V](#) ([SiFive FE310](#))
- Поддержка микроконтроллеров «[Миландр](#)»
ARM Cortex-M
ARMv6-M
- Программный FPU
- Написание документации

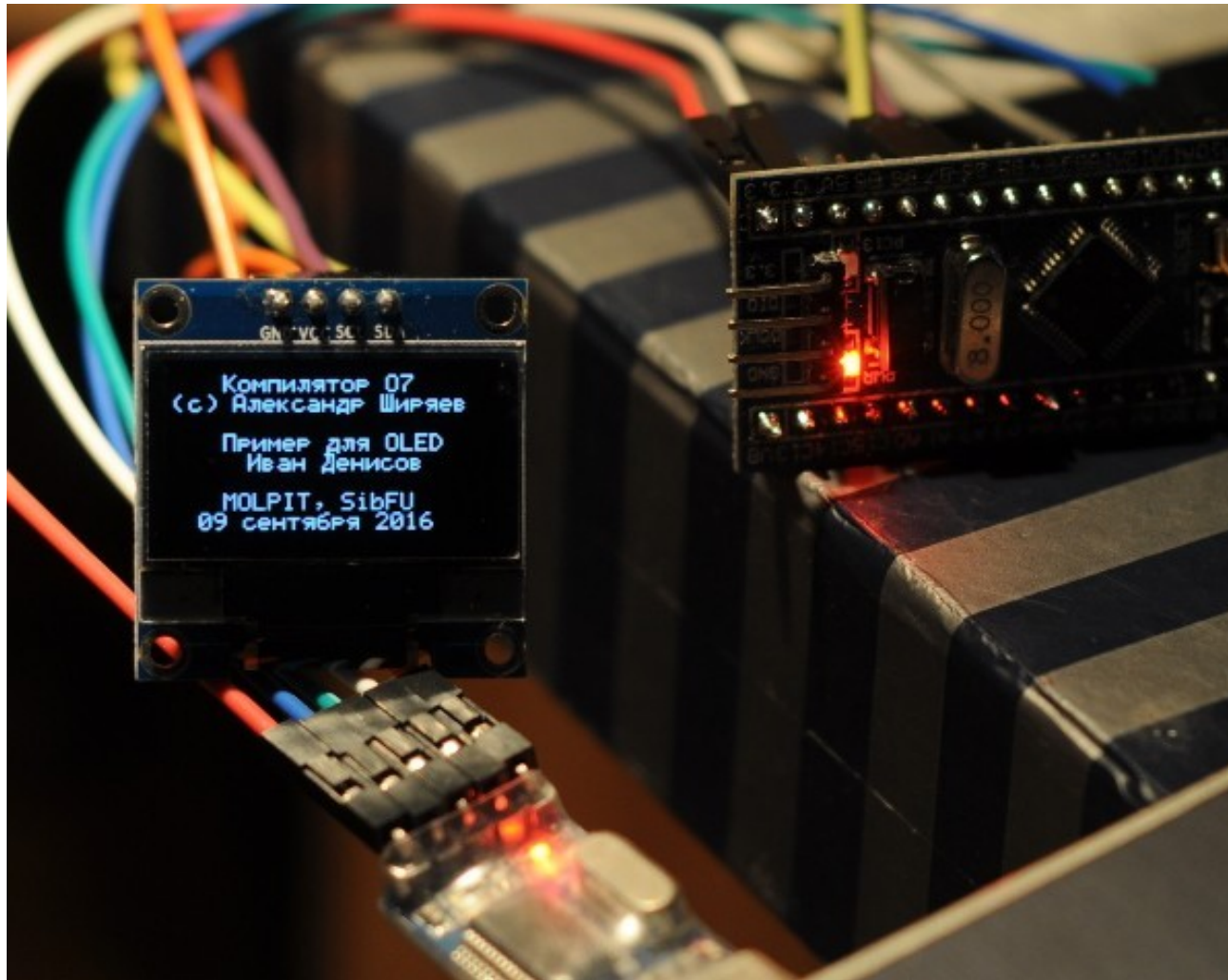
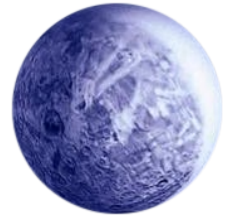


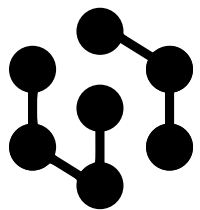
Компилятор O7



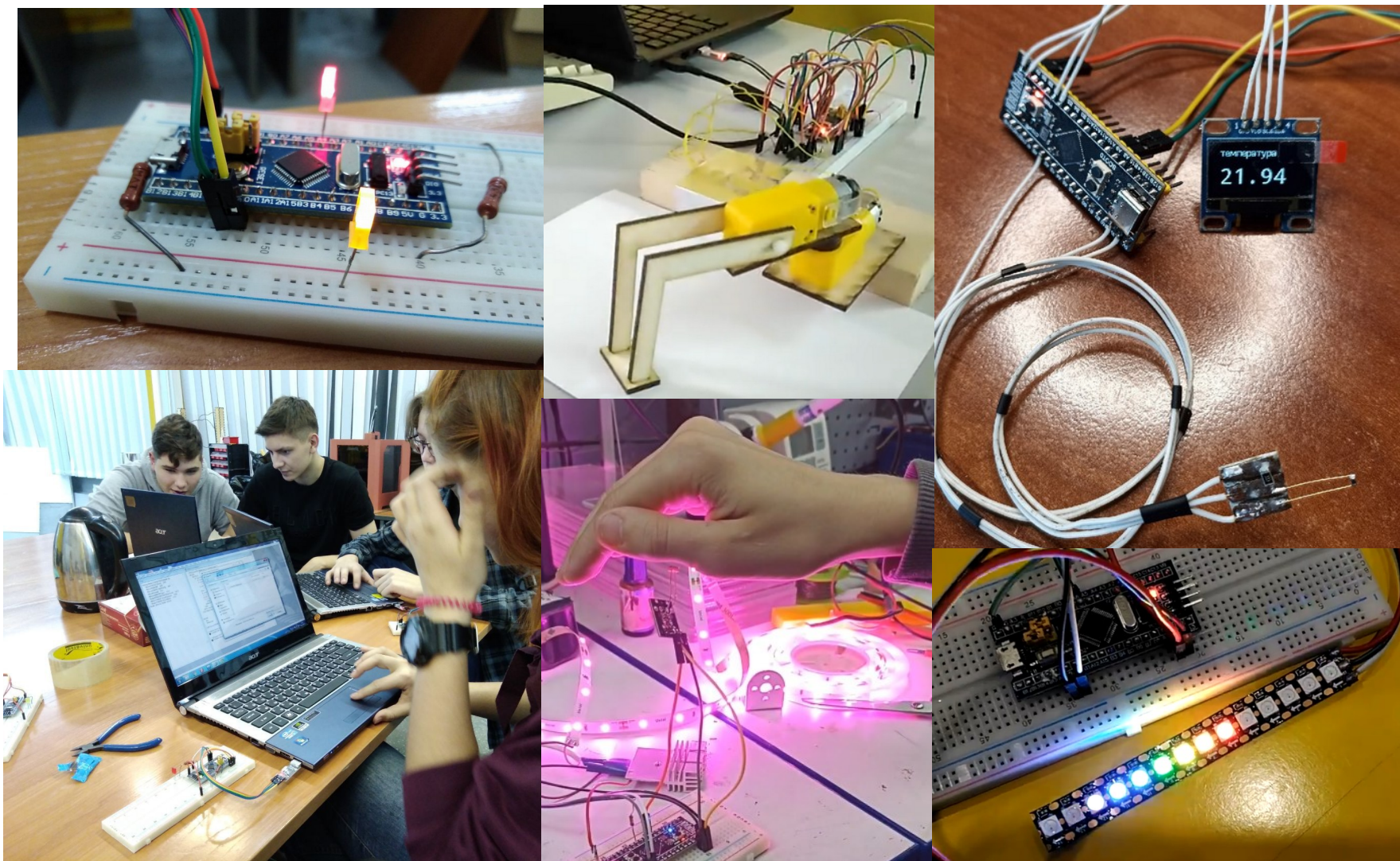
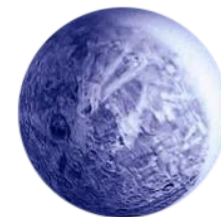


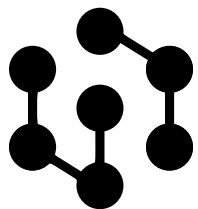
Компилятор O7 + мои библиотеки



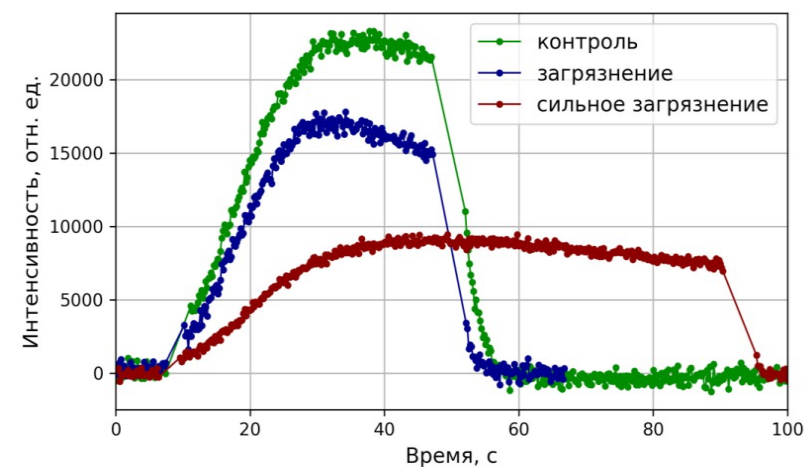
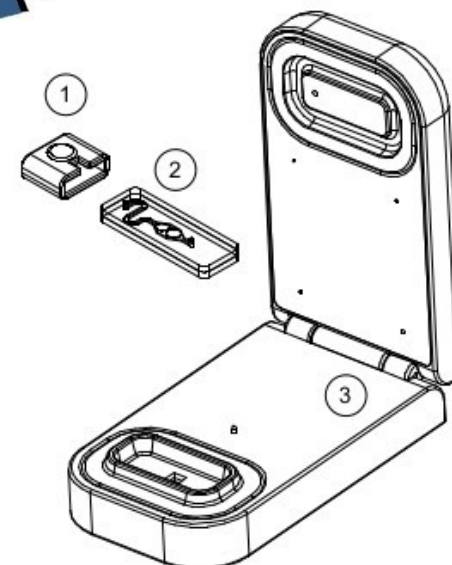
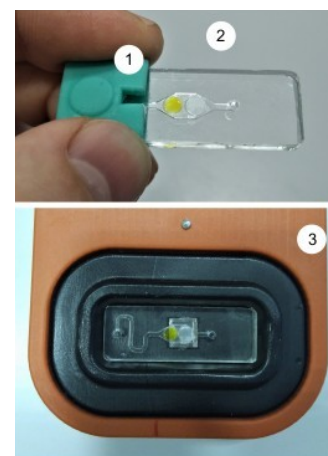
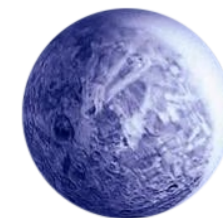


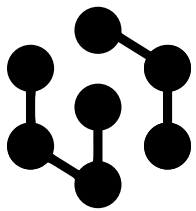
Компилятор 07 в школе





ПАК «Люминометр»





ПАК «Микрофлюидный контроллер давлений»

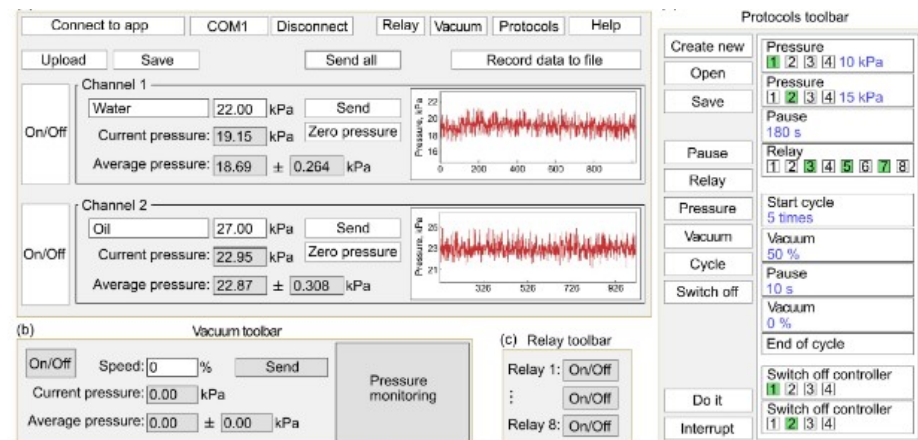
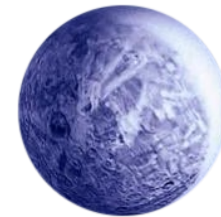


Fig. 3. Software interface of the microfluidic pressure controller. (a) Main window of the interface that allows to control and monitor pressures in each channel. (b) Toolbar for controlling the integrated vacuum pump. (c) Toolbar for controlling external devices via relay outputs. (d) Toolbar for defining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

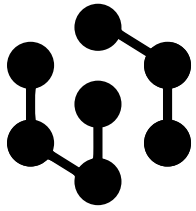
<https://github.com/microfluidic-pressure-controller>

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.

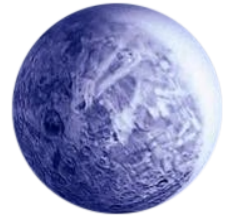
<https://ieeexplore.ieee.org/document/9732441>



<https://youtu.be/IE8-MaJ1fgQ>

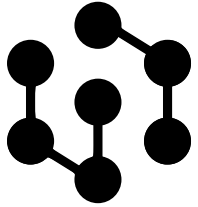


Создание ООО «ЦСТ» для коммерциализации ПАК

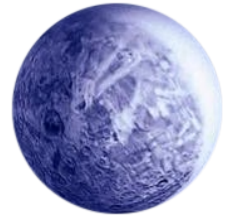


- Доработка программной платформы Блэкбокс 2.0
- Создание среды разработки Рекордино

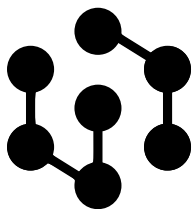
LDNova



Набор технологических решений



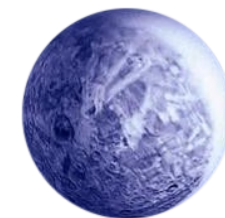
- Кросс-платформенный Блэкбокс
язык Компонентный Паскаль
(Блэкбокс Оберон)
- Компилятор O7
язык Оберон в последней редакции Н.Вирта
(Оберон-07)
- Слияние двух решений в среде Рекордино
разработка в **рекордные** сроки!



Рекордино

среда разработки для микроконтроллеров

<https://recordino.ldnova.com>



LDNova Recordino

Файл Правка Шрифт Инфо Разработка Инструменты Диалоги Текст Обх Tut Экстра O7 Окна Справка

ARMv7 Target: STM32F401CC
Top module: DetHTU21F4Demo

Рабочий журнал

compiling DetHTU21F4Demo 775 756
compiling DetHTU21F4Demo 775 756

linking DetHTU21F4Demo Ok
256 MicroSTM32F4
260 MicroARMv7M
306 MicroKernel0
992 MicroARMv7MTraps
1288 MicroSTM32F4System
2020 MicroSTM32F4Pins
2296 MicroSTM32F4TPorts
3976 MobxARMv7MSTM32SysTick0
4086 MicroI2CBus
4090 DetHTU21
4626 MicroMemFormatters
5892 MicroSTM32F4I2C1Bus
8446 DetHTU21F4Demo
ROM: 18808 B; RAM: 1788 B

HTU21F4Demo

```
PROCEDURE Loop;  
VAR x, adc: SET; i: INTEGER;  
BEGIN  
  sec := 0;  
  msec := 0;  
  REPEAT  
    Tps.Receive(p0);  
    IF SysTick0.OnTimer() THEN  
      IF msec >= 9999 THEN  
        msec := 0;  
        INC(sec);  
        IF sec = 6 THEN OnUPS ELSIF sec = 7 THEN OffUPS END;  
      ELSE  
        INC(msec)  
      END;  
  
      (* fan speed *)  
      IF fanIsOn THEN  
        IF msec MOD 10 = 0 THEN  
          SYSTEM.PUT(MCU.GPIOBBSRR, {13+16});  
        ELSIF msec MOD 10 = fanSpeed THEN  
          SYSTEM.PUT(MCU.GPIOBBSRR, {13});  
        END;  
      END;  
  
      (* status led *)  
      IF msec = 0 THEN  
        SYSTEM.PUT(MCU.GPIOCBSRR, {13+16});  
      ELSIF msec = 5000 THEN  
        SYSTEM.PUT(MCU.GPIOCBSRR, {13});  
      END;  
  
    END;  
    I2C1.Process;  
    ARMv7M.WFI  
  UNTIL FALSE  
END Loop;  
  
PROCEDURE Receive (id: CHAR; a: ARRAY OF CHAR; len: INTEGER);  
VAR ok: BOOLEAN; type: BYTE; row, col, char, i, width: INTEGER;
```

recordino UPScontrol

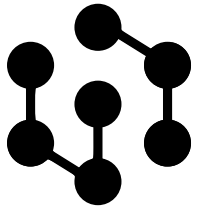
- Det
- Micro
- O7
- default.syntax
- DetHTU21F4Demo.bin
- DetHTU21F4Demo.hex
- HTU21F4Demo.odc
- project.conf
- server
- startRecordino

project.conf (~/.rec

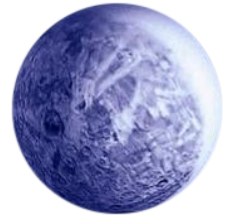
Файл Правка Вид Поиск Сервис

project.conf *

```
1 STM32F401CC  
2 7  
3 DetHTU21F4Demo  
4 "/dev/ttyUSB0"  
5 115200
```

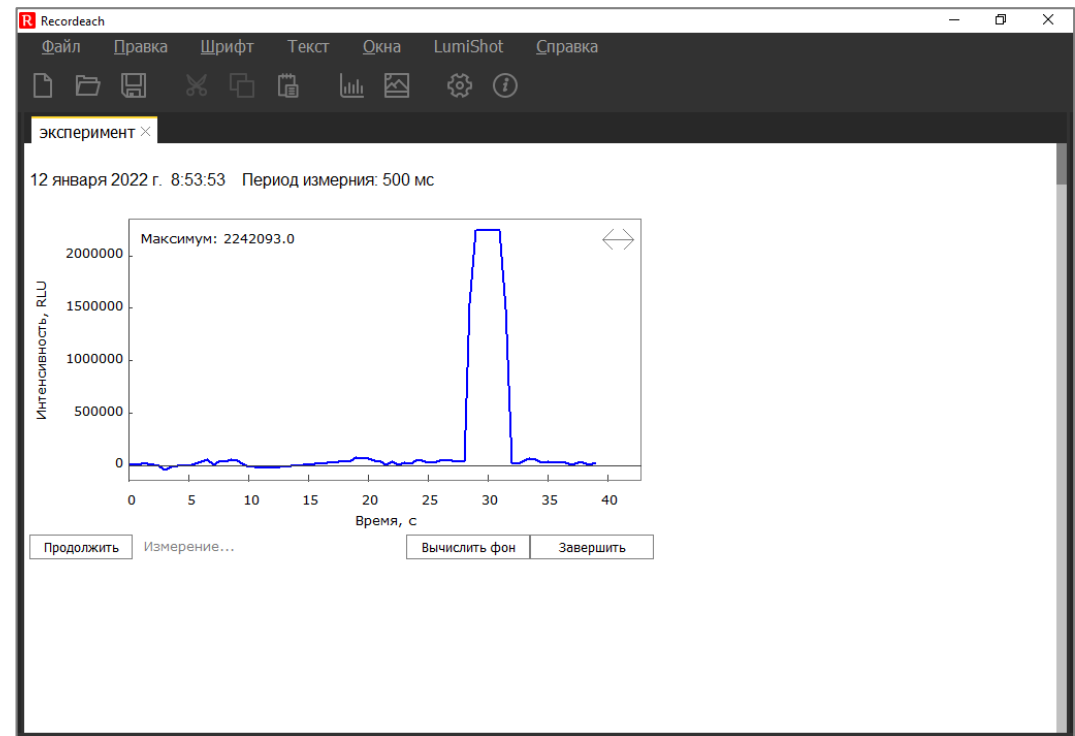


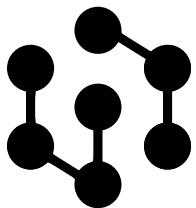
ПАК «Примус Рекордыч»



LDNova Primus

RecordEach





Управляем термостатом с ПК



IMPORT SYSTEM,

```
MCU := MicroSTM32F4,  
Sys := MicroSTM32F4System,  
ARMv7M := MicroARMv7M,  
Traps := MicroARMv7MTraps,  
TPs := MicroSTM32F4TPorts,  
Pins := MicroSTM32F4Pins,  
SysTick0 := MobsARMv7MSTM32SysTick0,  
MicroI2CBus,  
HTU21 := DetHTU21,  
I2C1 := MicroSTM32F4I2C1Bus;
```

```
PROCEDURE InitUart;  
VAR par: TPs.InitPar1;  
BEGIN
```

```
    par.n := TPs.USART1;  
    par.RXPinPort := Pins.A; par.RXPinN := 10; par.RXPinAF := Pins.AF7;  
    par.TXPinPort := Pins.A; par.TXPinN := 9; par.TXPinAF := Pins.AF7;  
    par.UCLK := Sys.PCLK2;  
    par.baud := 19200;  
    par.parity := TPs.parityNone;  
    par.receive := Receive;  
    TPs.Init(p0, par);
```

```
END InitUart;
```

```
PROCEDURE Setup;
```

```
    VAR ok: BOOLEAN; bus: MicroI2CBus.Bus;
```

```
    BEGIN
```

```
        InitUart;
```

```
        TPs.Send(p0, idInit, 0, 0, ok); (* heating UART *)
```

```
        IF Traps.trapFlag THEN
```

```
            Traps.ClearTrapFlag; (* send trap information *)
```

```
            TPs.Send(p0, idTrap, SYSTEM.ADR(Traps.trap), 9, ok)
```

```
        END;
```

```
        SysTick0.Init(Sys.HCLK, 1000);
```

```
        I2C1.Init(I2C1Freq);
```

```
        I2C1.GetBus(bus);
```

```
        HTU21.SetupBus(bus);
```

```
        HTU21.SetupHumCallback(CallbackHum);
```

```
        HTU21.SetupTempCallback(CallbackTemp);
```

```
        HTU21.SetupErrCallback(CallbackErr);
```

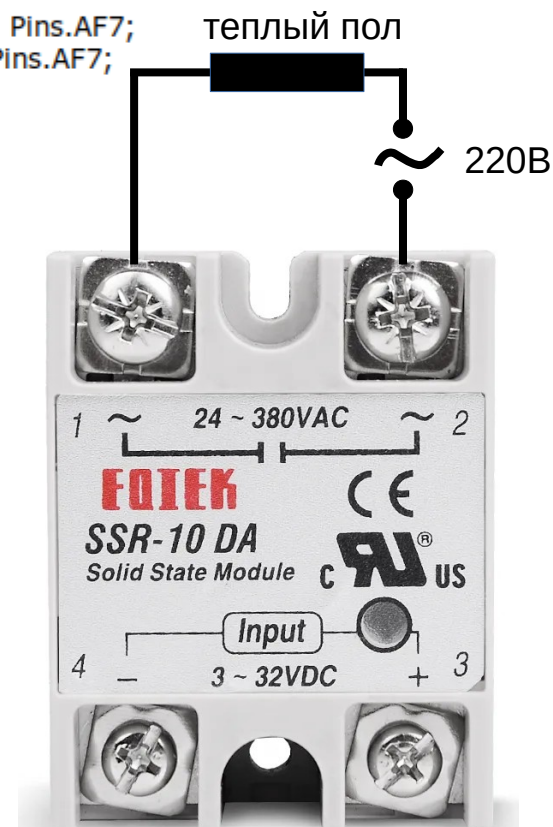
```
        HTU21.SoftReset;
```

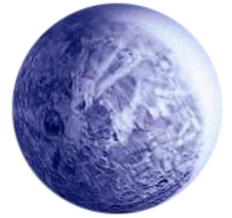
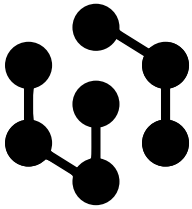
```
(* HEATER *)
```

```
Pins.Configure(Pins.A, 1, Pins.output, Pins.pushPull, Pins.medium, Pins.noPull, Pins.AF0);
```

```
Pins.Configure(Pins.A, 2, Pins.output, Pins.pushPull, Pins.medium, Pins.noPull, Pins.AF0);
```

```
Pins.Configure(Pins.A, 3, Pins.output, Pins.pushPull, Pins.medium, Pins.noPull, Pins.AF0);
```





CONST

I2C1Freq = 100000;

idInit = 0CBX;
idTrap = 0DEX;

idSoftReset = 020X;
idResetErr = 021X;

idHumAsk = 031X;
idHumRead = 032X;
idHumErr = 033X;

idTempAsk = 041X;
idTempRead = 042X;
idTempErr = 043X;

idOnHeater = 050X;
idOffHeater = 051X;
idOnFan = 052X;
idOffFan = 053X;
idSetFanSpeed = 054X;

idGetLineState = 055X;

```
PROCEDURE Receive (id: CHAR; a: ARRAY OF CHAR; len: INTEGER);
  VAR ok: BOOLEAN; row, col, char, i, width: INTEGER; x: SET; b: ARRAY 2 OF CHAR;
BEGIN
  IF id = idSoftReset THEN
    I2C1.Init(I2C1Freq);
    HTU21.SoftReset;
  ELSIF id = idHumAsk THEN
    HTU21.AskHum
  ELSIF id = idHumRead THEN
    HTU21.ReadHum
  ELSIF id = idTempAsk THEN
    HTU21.AskTemp
  ELSIF id = idTempRead THEN
    HTU21.ReadTemp
  ELSIF id = idSetFanSpeed THEN
    SYSTEM.GET(SYSTEM.ADR (a), fanSpeed)
  ELSIF id = idOnHeater THEN
    OnHeater
  ELSIF id = idOffHeater THEN
    OffHeater
  ELSIF id = idOnFan THEN
    OnFan
  ELSIF id = idOffFan THEN
    OffFan
  ELSIF id = idGetLineState THEN
    SYSTEM.GET(MCU.GPIOBIDR, x);
    IF 12 IN x THEN
      b[0] := 077X;
    ELSE
      b[0] := 055X;
    END;
    TPs.Send(p0, idGetLineState, SYSTEM.ADR(b), 1, ok)
  ELSE
    TPs.Send(p0, id, SYSTEM.ADR(a), len, ok)
  END
END Receive;
```

```
PROCEDURE OnHeater;
BEGIN
  SYSTEM.PUT(MCU.GPIOABSRR, {1+16..3+16});
END OnHeater;
```

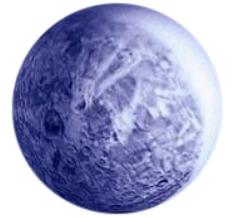
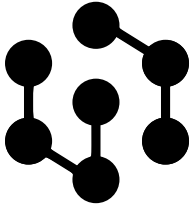
```
PROCEDURE OffHeater;
BEGIN
  SYSTEM.PUT(MCU.GPIOABSRR, {1..3});
END OffHeater;
```

```
PROCEDURE OnFan;
BEGIN
  SYSTEM.PUT(MCU.GPIOABSRR, {6+16});
  fanIsOn := TRUE;
END OnFan;
```

```
PROCEDURE OffFan;
BEGIN
  SYSTEM.PUT(MCU.GPIOABSRR, {6});
  fanIsOn := FALSE;
END OffFan;
```

```
PROCEDURE OnUPS;
BEGIN
  SYSTEM.PUT(MCU.GPIOABSRR, {7+16});
END OnUPS;
```

```
PROCEDURE OffUPS;
BEGIN
  SYSTEM.PUT(MCU.GPIOABSRR, {7});
END OffUPS;
```



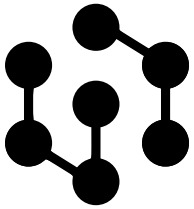
Главная петля программы

```
PROCEDURE Loop;  
VAR x, adc: SET; i: INTEGER;  
BEGIN  
  sec := 0;  
  msec := 0;  
  REPEAT  
    TPs.Receive(p0);  
    IF SysTick0.OnTimer() THEN  
      IF msec >= 999 THEN  
        msec := 0;  
        INC(sec);  
        IF sec = 6 THEN OnUPS ELSIF sec = 7 THEN OffUPS END;  
      ELSE  
        INC(msec)  
      END;  
      (* status led *)  
      IF msec = 0 THEN  
        SYSTEM.PUT(MCU.GPIOCBSRR, {13+16});  
      ELSIF msec = 5000 THEN  
        SYSTEM.PUT(MCU.GPIOCBSRR, {13});  
      END;  
    END;  
    I2C1.Process;  
    ARMv7M.WFI  
  UNTIL FALSE  
END Loop;
```

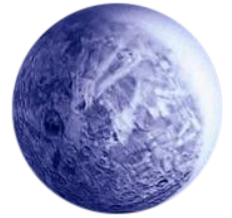
приём данных по UART

обработка I2C шины для общения с датчиком температуры и влажности

перейти в режим энергосбережения до срабатывания следующих прерываний



Консольная программа для ПК



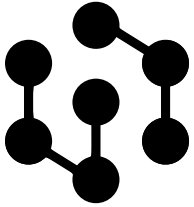
DevLinker1.LinkElfExe Linux server := Kernel\$+ Utf LinKernel Files LinEnv LinFiles StdLoader DetLoader

```
MODULE DetLoader;
  IMPORT Kernel;

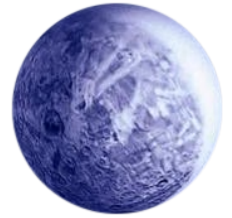
  CONST init = 'DetInit';

  PROCEDURE Load (IN name: ARRAY OF CHAR);
    VAR importing, imported, err, object: ARRAY 256 OF CHAR; m: Kernel.Module; res: INTEGER;
  BEGIN
    m := Kernel.ThisLoadedMod(SHORT(name));
    IF m = NIL THEN
      Kernel.LoadMod(name);
      Kernel.GetLoaderResult(res, importing, imported, object);
      IF res # 0 THEN
        CASE res OF Kernel.done: err := ': no error'
        | Kernel.fileNotFound: err := ': file not found!'
        | Kernel.syntaxError: err := ': syntax error'
        | Kernel.objNotFound: err := ': object not found'
        | Kernel.illegalFPrint: err := ': illegal foot print'
        | Kernel.cyclicImport: err := ': cyclic import'
        | Kernel.noMem: err := ': no memory'
        | Kernel.commNotFound: err := ': comm not found'
        | Kernel.commSyntaxError: err := ': comm syntax error'
        | Kernel.moduleNotFound: err := ': module not found'
        ELSE err := ': host loader error' END;
        Kernel.FatalError(res, '['+name+'] MODULE ' + importing + " IMPORT " + imported + err)
      END
    ELSE Kernel.FatalError(1, err) END
  END Load;

BEGIN
  Load(init)
END DetLoader.
```



Пример модуля инициализации для консольного приложения Linux



```
MODULE DetInit;  (** the first dynamically-loaded module *)

IMPORT
  SYSTEM, Kernel, Services, Log, Dialog, Converters, Loop, LinConsole, LinV24,
  GLib := LinGtk2GLib, Gtk := LinGtk2Gtk, StdInterpreter, LinDates, LinLang;

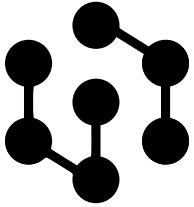
PROCEDURE [ccall] Fake (_: GLib.gpointer): GLib.gboolean;
BEGIN RETURN 1
END Fake;

PROCEDURE FetchEvent;
  VAR res_: INTEGER;
BEGIN res_ := Gtk.gtk_main_iteration();
END FetchEvent;

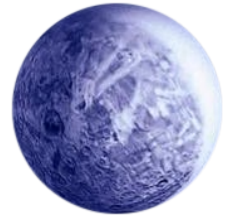
PROCEDURE ExitCondition(): BOOLEAN;
BEGIN RETURN Dialog.exit
END ExitCondition;

PROCEDURE Init;
  VAR res_: INTEGER; m: Kernel.Module;
BEGIN
  Dialog.Call("DetHTU21DemoConsole.Start", "", res_);
  res_ := GLib.g_timeout_add(Dialog.idlePeriod, Fake, 0);
  Dialog.RequestExit(~Dialog.exitWithoutWindows);
  Loop.quantum := Dialog.idlePeriod;
  Loop.Start(FetchEvent, NIL, ExitCondition);
END Init;

BEGIN
  Init;
  Kernel.Quit(0)
END DetInit.
```



Консольная программа для ПК



```
MODULE DetHTU21DemoConsole;
```

```
IMPORT SYSTEM, Dates, TP := MicroTPorts, Strings, Services, Log, ConsLog, Dialog;
```

```
TYPE
```

```
  MessagesHandler = POINTER TO RECORD (TP.MessagesHandler) END;
```

```
CONST
```

```
  idInit = 0CBX;  
  idTrap = 0DEX;
```

```
  idSoftReset = 020X;  
  idResetErr = 021X;
```

```
  idHumAsk = 031X;  
  idHumRead = 032X;  
  idHumErr = 033X;
```

```
  idTempAsk = 041X;  
  idTempRead = 042X;  
  idTempErr = 043X;
```

```
  idOnHeater = 050X;  
  idOffHeater = 051X;  
  idOnFan = 052X;  
  idOffFan = 053X;  
  idSetFanSpeed = 054X;
```

```
  idGetLineState = 055X;
```

```
VAR
```

```
  p: TP.Port;  
  mh: MessagesHandler;
```

```
PROCEDURE Open*(port: ARRAY OF CHAR; speed: INTEGER);  
BEGIN
```

```
  IF p = NIL THEN  
    p := TP.NewPort(port, speed, {}, mh);  
    Log.Bool(p # NIL); Log.Ln;
```

```
  END
```

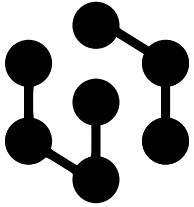
```
END Open;
```

```
PROCEDURE Start*;
```

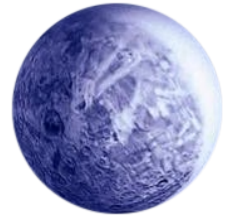
```
BEGIN
```

```
  Open('/dev/ttyUSB0', 19200);  
  OffHeater;  
  OffFan;  
  stop := FALSE;  
  Services.DoLater(aAsk, Services.Ticks() + 500)
```

```
END Start;
```



Запрос измерений с устройства

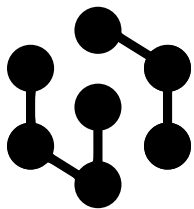


```
PROCEDURE AskTemp*;  
  VAR a: ARRAY 1 OF SHORTCHAR;  
BEGIN  
  TP.SendMessage(p, IdTempAsk, a, 1)  
END AskTemp;
```

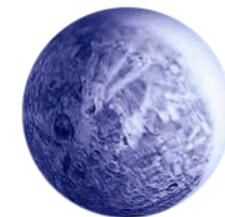
```
PROCEDURE ReadTemp*;  
  VAR a: ARRAY 1 OF SHORTCHAR;  
BEGIN  
  TP.SendMessage(p, IdTempRead, a, 1)  
END ReadTemp;
```

```
PROCEDURE (a: ActionAsk) Do;  
BEGIN  
  IF p # NIL THEN  
    AskTemp;  
    IF ~ stop THEN  
      Services.DoLater(aRead, Services.Ticks() + 1000)  
    END;  
  ELSE  
    Log.String("no connection"); Log.Ln;  
    Dialog.RequestExit(TRUE);  
  END  
END Do;
```

```
PROCEDURE (a: ActionRead) Do;  
BEGIN  
  IF p # NIL THEN  
    ReadTemp;  
    IF ~ stop THEN  
      Services.DoLater(aAsk, Services.Ticks() + 9000)  
    END;  
  ELSE  
    Log.String("no connection"); Log.Ln;  
    Dialog.RequestExit(TRUE);  
  END  
END Do;
```



Обработка сообщений устройства и принятие решения



```
PROCEDURE OnHeater*;  
VAR a: ARRAY 1 OF SHORTCHAR;  
BEGIN  
  TP.SendMessage(p, idOnHeater, a, 1);  
  heaterStatus := "1";  
END OnHeater;
```

```
PROCEDURE OffHeater*;  
VAR a: ARRAY 1 OF SHORTCHAR;  
BEGIN  
  TP.SendMessage(p, idOffHeater, a, 1);  
  heaterStatus := "0";  
END OffHeater;
```

```
PROCEDURE (h: MessagesHandler) MessageReceived (id: SHORTCHAR;  
  IN a: ARRAY OF SHORTCHAR; len: INTEGER);  
  VAR pc, pos, code, val: INTEGER; s: ARRAY 256 OF CHAR;  
  b1, b2, b3: SHORTCHAR; date: ARRAY 64 OF CHAR;  
  
  BEGIN  
    IF id = idTempRead THEN  
  
      SYSTEM.GET(SYSTEM.ADR (a), val);  
  
      tempD := val / 100;  
      MakeDate(date);  
  
      Log.String(date + "; " + heaterStatus + "; "); Log.Real(tempD); Log.String(";"); Log.Ln;  
  
      IF (tempD < 10) & (heaterStatus = "0") THEN  
        OnHeater  
      END;  
  
      IF (tempD > 12) & (heaterStatus = "1") THEN  
        OffHeater  
      END;  
  
    ELSIF ...
```



```
2022;11;26;20:17:55; 1; 10.71;  
2022;11;26;20:18:05; 1; 10.94;  
2022;11;26;20:18:15; 1; 11.15;  
2022;11;26;20:18:25; 1; 11.35;  
2022;11;26;20:18:35; 1; 11.59;  
2022;11;26;20:18:45; 1; 11.8;  
2022;11;26;20:18:55; 1; 12.03;  
2022;11;26;20:19:05; 0; 12.22;  
2022;11;26;20:19:15; 0; 12.36;  
2022;11;26;20:19:25; 0; 12.43;  
2022;11;26;20:19:35; 0; 12.48;  
2022;11;26;20:19:45; 0; 12.44;  
2022;11;26;20:19:55; 0; 12.39;  
2022;11;26;20:20:05; 0; 12.3;  
2022;11;26;20:20:15; 0; 12.18;
```