НИУ «МЭИ»
представляет учебный курс:

Проектирование цифровых систем управления на базе отечественного микроконтроллера **НИИЭТ К1921ВК01Т**

Москва 2019

Лекция 1.2

- Обзор существующих инструментов разработки программного обеспечения для К1921ВК01Т демонстрация их работы, выделение достоинств и недостатков:
  - IAR Embedded Workbench for ARM
  - Keil uVision
  - VectorIDE
  - CodeMaster++[ARM]
- Практическая работа по подключению к К1921ВК01Т из различных сред разработки, их конфигурирование.
- Запуск простейшей программы на К1921ВК01Т.

Какую среду разработки и компиляторы выбрать для программирования на языке C/C++?

IAR EWARM

Keil uVision

Популярные коммерческие среды разработки с собственными компиляторами. Цена на 2016г – около 300 т.р. за рабочее место

CodeMaster++[ARM]

Отечественная среда разработки (ООО "Фитон"). Средства редактирования кода недостаточно развиты.

QT Creator+ GCC+ OpenOCD

Eclipse+ GCC+ OpenOCD

Инструментарий с открытым исходным кодом. По удобству превосходит платные аналоги, однако имеет высокий порог вхождения (сложно настроить и разобраться)

Как это делается для популярных сред разработки?

IAR EWARM, Keil uVision – имеют возможность написать свой программатор. **Программатор реализован** – доступен на интернет-форуме НИИЭТ.

CodeMaster++[ARM] – имеет встроенную поддержку K1921BK01T, включая программатор.

Eclipse/QT Creator+GCC+OpenOCD – для поддержки своего алгоритма программирования flash необходимо доработать OpenOCD. **Программатор реализован** – доступен на интернет-форуме НИИЭТ.

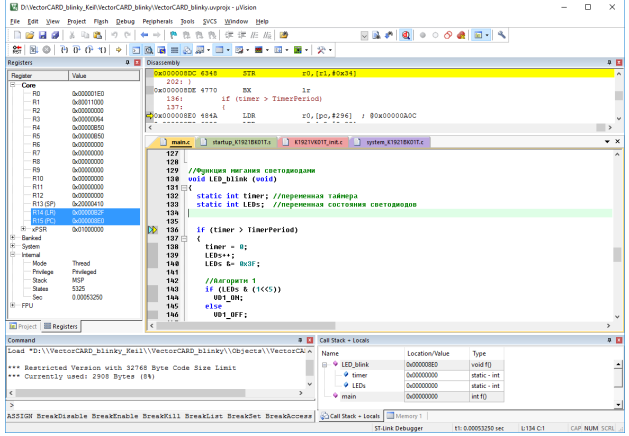
Открытая бесплатная среда разработки для K1921BK01T

ООО «НПФ ВЕКТОР» создала и выложила в открытый доступ готовую сборку открытого инструментария для разработки под K1921BK01T **Vector IDE**.

Vector IDE состоит из:

- **Eclipse (редактирование кода)**
- **GCC для ARM (набор свободных компиляторов)**
- **OpenOCD (драйвер программатора и отладчика)**
- **GNU ARM plugin (плагин для соединения всего вместе)**
- **Конфигурационные файлы (настройка конфигурации для K1921BK01T)**
- **Примеры (Мигание светодиодом, запуск таймера, ШИМ и прочее для K1921BK01T)**

Можно скачать на сайте http://motorcontrol.ru/



forum.niiet.ru/viewtopic.php?f=37&t=486

Mail.Ru | Хабрахабр | translate.google.ru | maps.yandex.ru | Server | SaveFrom.net | Noisli - background n | http://www.tickets-sa | Картинки Google | xkcd: что если? | http://www.factroom | dailytechinfo.org

Быстрый старт с МК К1921ВК01Т

Модераторы: pip, _sva_

Ответить | Поиск в теме... | 44 сообщения 1 2 3 4 5

Быстрый старт с МК К1921ВК01Т

dav » 14 дек 2015, 13:02

В данном руководстве представлен минимальный объем информации для начала работы с микроконтроллером К1921ВК01Т.
Руководство содержит информацию о разработке программ для К1921ВК01Т в IAR, Keil, CodeMaster++[ARM], краткая информация по работе с микроконтроллером (рассмотрена работа с альтернативными функциями портов), приводится схема включения микроконтроллера, а также некоторые фрагменты кода.

Также во вложении содержатся архивы для неофициальной поддержки нашего МК в IAR, Keil.
Так для интеграции поддержки К1921ВК01Т в IAR достаточно разархивировать содержимое архива "IAR_1921ВК01Т" в каталог с установленным IAR, при этом добавятся необходимые файлы каталога arm
Для интеграции поддержки К1921ВК01Т в Keil необходимо разархивировать содержимое архива "Keil_1921ВК01Т" в каталог с установленным Keil (добавляются необходимые файлы каталога ARM)

ВЛОЖЕНИЯ

Быстрый_старт_К1921ВК01Т.pdf
(2.46 МБ) 356 скачиваний

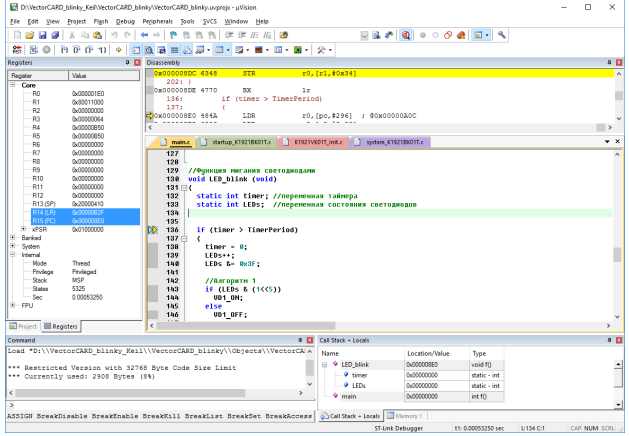
Keil_К1921ВК01Т.zip
(3.4 МБ) 244 скачивания

IAR_К1921ВК01Т.zip
(802.65 КБ) 184 скачивания

dav
Сообщения: 43
Зарегистрирован: 14 дек 2015, 09:21
Предприятие: АО НИИЭТ
Откуда: АО НИИЭТ, Воронеж
Контактная информация:

Скачать из презентации:

Быстрый_старт_К1921ВК01Т.pdf | Keil_К1921ВК01Т.zip | VectorCARD_blinky_Keil.zip



Обновление!!!

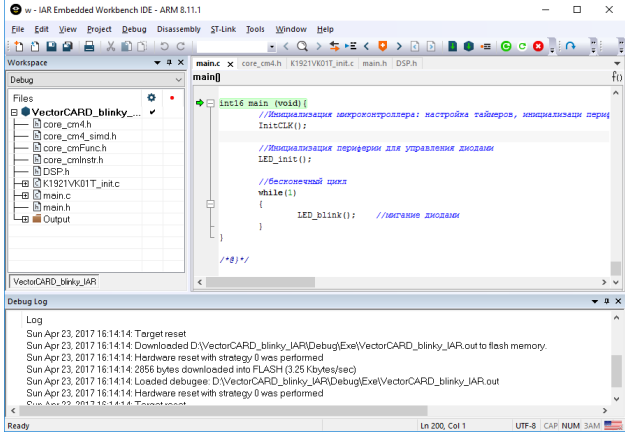
Теперь последняя версия SDK живет здесь:

https://bitbucket.org/niietcm4/k1921vkx_sdk/src/master/

Скачать из презентации:

NIIET.K1921VK01T_DFP.2.0.2.zip

IAR SYSTEMS



forum.niiet.ru/viewtopic.php?f=37&t=486

Mail.Ru Хабрахабр translate.google.ru maps.yandex.ru Server SaveFrom.net Noisli - background n http://www.tickets-sa Картинки Google xkcd: что если? http://www.factroom dailytechinfo.org

Быстрый старт с МК К1921ВК01Т

Модераторы: pip, _sva_

Ответить | Поиск в теме... | 44 сообщения 1 2 3 4 5

Быстрый старт с МК К1921ВК01Т

dav » 14 дек 2015, 13:02

В данном руководстве представлен минимальный объем информации для начала работы с микроконтроллером К1921ВК01Т.
Руководство содержит информацию о разработке программ для К1921ВК01Т в IAR, Keil, CodeMaster++[ARM], краткая информация по работе с микроконтроллером (рассмотрена работа с альтернативными функциями портов), приводится схема включения микроконтроллера, а также некоторые фрагменты кода.

Также во вложении содержатся архивы для неофициальной поддержки нашего МК в IAR, Keil.
Так для интеграции поддержки К1921ВК01Т в IAR достаточно разархивировать содержимое архива "IAR_1921ВК01Т" в каталог с установленным IAR, при этом добавятся необходимые файлы каталога arm
Для интеграции поддержки К1921ВК01Т в Keil необходимо разархивировать содержимое архива "Keil_1921ВК01Т" в каталог с установленным Keil (добавляются необходимые файлы каталога ARM)

ВЛОЖЕНИЯ

Быстрый_старт_К1921ВК01Т.pdf
(2.46 МБ) 356 скачиваний

Keil_К1921ВК01Т.zip
(3.4 МБ) 244 скачивания

IAR_К1921ВК01Т.zip
(802.65 КБ) 184 скачивания

dav
Сообщения: 43
Зарегистрирован: 14 дек 2015, 09:21
Предприятие: АО НИИЭТ
Откуда: АО НИИЭТ, Воронеж
Контактная информация:

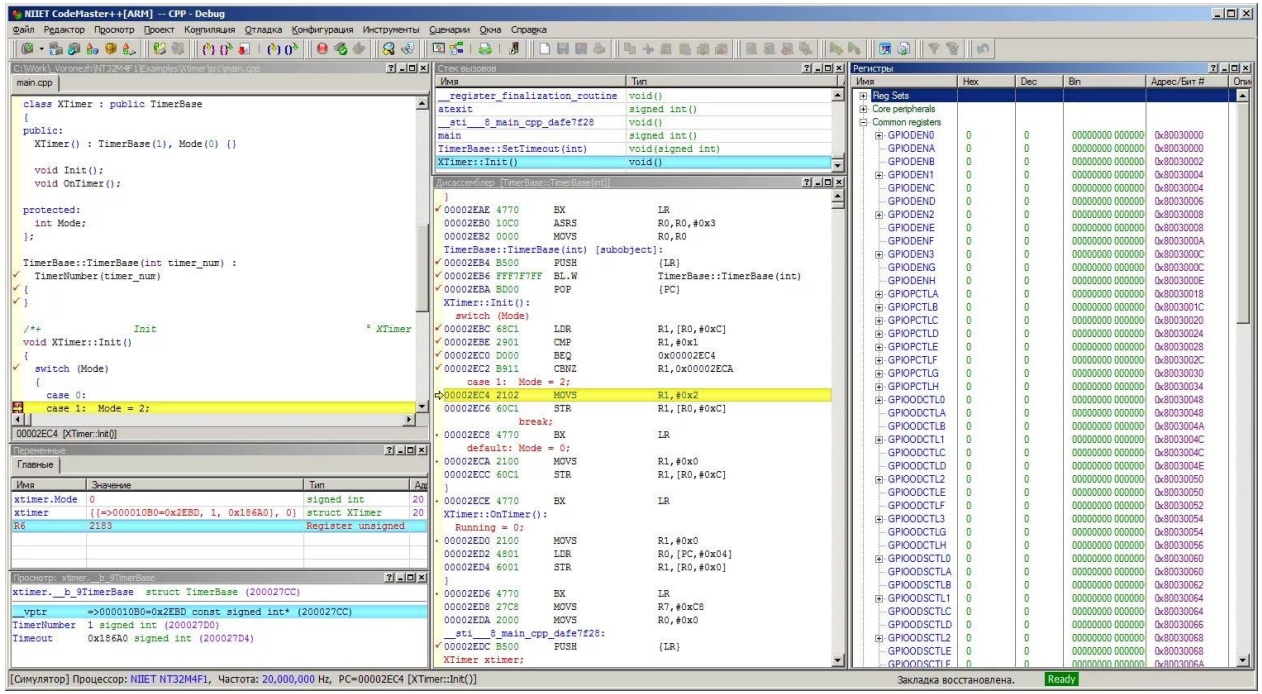
Скачать из презентации:

Быстрый_старт_К1921ВК01Т.pdf IAR_К1921ВК01Т.zip VectorCARD_blinky_IAR.zip

НИИЭТ
ИЗДЕЛИЯ МИКРОЭЛЕКТРОНИКИ
РАЗРАБОТКА ПРОИЗВОДСТВО ИСПЫТАНИЯ

Phyton™
Универсальный программатор, программирование и отладка микроконтроллеров

CodeMaster++[ARM]



Сертифицированная отечественная официальная среда разработки от ОА «НИИЭТ» (разработана Phyton)

Сравнение сред разработки для K1921BK01T (по 10-ти бальной шкале)

	Keil uVision	IAR EVARM	CodeMaster++	VectorIDE
Цена на 1 раб. место	~250 т.р.	~250 т.р.	???	Бесплатн.
Удобство редактирования кода	7	5	3	9
Стабильность работы	8	9	5	6
Простота создания и прошивки первой программы	5	5	9	7
Скорость работы скомпилированного кода	10	10	5	8
Просмотр переменных в реальном времени	есть	нет	нет	нет
Сертификация средств разработки	нет	нет	есть	нет
Выбор JTAG	Любой для ARM	Любой для ARM	Только JEM-ARM	J-link и ST-link +другие

Vector IDE

Debug - MK40_1_ETV_TD/vsrc/V_adc.c - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access | C/C++ | Debug

Project Explorer

- Binaries
- Includes
- asm
- cmd
- Debug
- include
- lib
- src
- vinclude
- vsrc
  - CANBlockTransferInterface.c
  - CANOpen_drv.c
  - CANPDOCallback.c
  - clarke.c
  - co_ODvars.c
  - cood1.c
  - DRV_INTERFACE.c
  - filter.c
  - ipark.c
  - kalm_pos.c
  - LookUpXY.c
  - main.c
  - mbod.c
  - park.c
  - SM_alarm.c
  - SM_cmd_logic.c
  - SM_ctrl.c
  - SM_NET.c
  - SM_protect.c
  - SM_SYS.c
  - smopos_const.c
  - smopos.c

Debug

- Thread #1 (Suspended : Breakpoint)
  - AdcDrv_ms_calc() at V_adc.c:1 014 0x1651c
  - SM_SYS_ms_Calc() at SM_SYS.c:328 0xb340
  - TIM1_IRQHandler() at main.c:233 0x19768
  - 0xfffffe8
- openocd.exe

main.c | build.h | V_PWM_6_12.c | V_adc.c

```
999
1000 //SSP_TxD(0) A[1]
1001 #define ADC_MPY_2_ON GPIO_A->LB_MASKED[(1<<1)]=(1<<1)
1002 #define ADC_MPY_2_OFF GPIO_A->LB_MASKED[(1<<1)]=0
1003 #define ADC_MPY_2_STATE GPIO_A->DATA.bit.GPIO1
1004
1005
1006
1007 //! \memberof TAdcDrv
1008 void AdcDrv_ms_calc(TAdcDrv *p) {
1009
1010
1011     p->ms_calc_delim++;
1012     if (p->ms_calc_delim<20)
1013         return;
1014     p->ms_calc_delim=0;
1015
1016     if (ADC_MPY_1_STATE==1) {//мультиплексор на плате МП1=1
1017         if (ADC_MPY_2_STATE==1) { //мультиплексор на плате МП2=1
1018
1019         } else {//МП1=1, МП2=0
1020             p->TempDrive_temp = (Uint16) (ADC->ADCDCVAL.ADCDCVAL9.bit.VAL<<4); //температура двигател
1021         }
1022     } else {//МП1=0
1023         if (ADC_MPY_2_STATE==1) {//МП1=0, МП2=1
1024             p->Temp1_temp = (Uint16) (ADC->ADCDCVAL.ADCDCVAL9.bit.VAL<<4); //температура радиатора
```

Registers

Name	Value
General Registers	
r0	0x20001228
r1	0x0
r2	0x14
r3	0x80015000
r4	0x200008b8
r5	0x20000a10
r6	0x20002910
r7	0x1
r8	0x20002f28
r9	0x20000f18
r10	0x168
r11	0x14
r12	0x76c
sp	0x2002fee8
lr	0xb341
pc	0x1651c
xPSR	0x21000014
d0	0x0
d1	0x0
d2	0x0
d3	0x0
d4	0x0

Problems | Expressions | Console | Disassembly | Search | Memory | Breakpoints | Progress

Expression	Type	Value
p->ms_calc_delim	int16	19
Add new expression		

Writable | Smart Insert | 1015 : 1

CodeMaster++[ARM]

NIIET CodeMaster++[ARM] -- CPP - Debug

Файл Редактор Просмотр Проект Компиляция Отладка Конфигурация Инструменты Сценарии Окна Справка

C:\Work_Voronezh\NT32M4F1\Examples\Xtimer\src\main.cpp

main.cpp

```
class XTimer : public TimerBase
{
public:
  XTimer() : TimerBase(1), Mode(0) {}

  void Init();
  void OnTimer();

protected:
  int Mode;
};

TimerBase::TimerBase(int timer_num) :
  TimerNumber(timer_num)
{
}

/*+                Init                                  * XTimer
void XTimer::Init()
{
  switch (Mode)
  {
    case 0:
    case 1:  Mode = 2;
```

00002EC4 [XTimer::Init()]

Стек вызовов

Имя	Тип
__register_finalization_routine	void()
atexit	signed int()
__sti___8_main_cpp_dafe7f28	void()
main	signed int()
TimerBase::SetTimeout(int)	void(signed int)
XTimer::Init()	void()

Дисассемблер [TimerBase::TimerBase(int)]

```
 }
00002EAE 4770      BX              LR
00002EB0 10C0      ASRS            R0,R0,#0x3
00002EB2 0000      MOVS            R0,R0
TimerBase::TimerBase(int) [subobject]:
00002EB4 B500      PUSH            {LR}
00002EB6 FFF7F7FF  BL.W            TimerBase::TimerBase(int)
00002EBA BD00      POP             {PC}
XTimer::Init():
  switch (Mode)
00002EBC 68C1      LDR             R1,[R0,#0xC]
00002EBE 2901      CMP             R1,#0x1
00002EC0 D000      BEQ             0x00002EC4
00002EC2 B911      CBNZ            R1,0x00002ECA
    case 1:  Mode = 2;
00002EC4 2102      MOVS            R1,#0x2
00002EC6 60C1      STR             R1,[R0,#0xC]
            break;
00002EC8 4770      BX              LR
    default: Mode = 0;
00002ECA 2100      MOVS            R1,#0x0
00002ECC 60C1      STR             R1,[R0,#0xC]
 }
00002ECE 4770      BX              LR
XTimer::OnTimer():
  Running = 0;
00002ED0 2100      MOVS            R1,#0x0
00002ED2 4801      LDR             R0,[PC,#0x04]
00002ED4 6001      STR             R1,[R0,#0x0]
 }
00002ED6 4770      BX              LR
00002ED8 27C8      MOVS            R7,#0xC8
00002EDA 2000      MOVS            R0,#0x0
__sti___8_main_cpp_dafe7f28:
00002EDC B500      PUSH            {LR}
XTimer xtimer;
```

Переменные

Главные

Имя	Значение	Тип	Адр
xtimer.Mode	0	signed int	20
xtimer	{{=>000010B0=0x2EBD, 1, 0x186A0}, 0}	struct XTimer	20
R6	2183	Register unsigned	

Просмотр: xtimer.__b_9TimerBase

```
xtimer.__b_9TimerBase  struct TimerBase (200027CC)

__vptr        =>000010B0=0x2EBD const signed int* (200027CC)
TimerNumber   1 signed int (200027D0)
Timeout       0x186A0 signed int (200027D4)
```

Регистры

Имя	Hex	Dec	Bin	Адрес/Бит #	Опис
Reg Sets					
Core peripherals					
Common registers					
GPIODEN0	0	0	00000000 000000	0x80030000	
GPIODENA	0	0	00000000 000000	0x80030000	
GPIODENB	0	0	00000000 000000	0x80030002	
GPIODEN1	0	0	00000000 000000	0x80030004	
GPIODENC	0	0	00000000 000000	0x80030004	
GPIODEND	0	0	00000000 000000	0x80030006	
GPIODEN2	0	0	00000000 000000	0x80030008	
GPIODENE	0	0	00000000 000000	0x80030008	
GPIODENF	0	0	00000000 000000	0x8003000A	
GPIODEN3	0	0	00000000 000000	0x8003000C	
GPIODENG	0	0	00000000 000000	0x8003000C	
GPIODENH	0	0	00000000 000000	0x8003000E	
GPIOPCTLA	0	0	00000000 000000	0x80030018	
GPIOPCTLB	0	0	00000000 000000	0x8003001C	
GPIOPCTLC	0	0	00000000 000000	0x80030020	
GPIOPCTLD	0	0	00000000 000000	0x80030024	
GPIOPCTLE	0	0	00000000 000000	0x80030028	
GPIOPCTLF	0	0	00000000 000000	0x8003002C	
GPIOPCTLG	0	0	00000000 000000	0x80030030	
GPIOPCTLH	0	0	00000000 000000	0x80030034	
GPIOODCTL0	0	0	00000000 000000	0x80030048	
GPIOODCTLA	0	0	00000000 000000	0x80030048	
GPIOODCTLB	0	0	00000000 000000	0x8003004A	
GPIOODCTL1	0	0	00000000 000000	0x8003004C	
GPIOODCTLC	0	0	00000000 000000	0x8003004C	
GPIOODCTLD	0	0	00000000 000000	0x8003004E	
GPIOODCTL2	0	0	00000000 000000	0x80030050	
GPIOODCTLE	0	0	00000000 000000	0x80030050	
GPIOODCTLF	0	0	00000000 000000	0x80030052	
GPIOODCTL3	0	0	00000000 000000	0x80030054	
GPIOODCTLG	0	0	00000000 000000	0x80030054	
GPIOODCTLH	0	0	00000000 000000	0x80030056	
GPIOODSCTL0	0	0	00000000 000000	0x80030060	
GPIOODSCTLA	0	0	00000000 000000	0x80030060	
GPIOODSCTLB	0	0	00000000 000000	0x80030062	
GPIOODSCTL1	0	0	00000000 000000	0x80030064	
GPIOODSCTLC	0	0	00000000 000000	0x80030064	
GPIOODSCTLD	0	0	00000000 000000	0x80030066	
GPIOODSCTL2	0	0	00000000 000000	0x80030068	
GPIOODSCTLE	0	0	00000000 000000	0x80030068	
GPIOODSCTLF	0	0	00000000 000000	0x8003006A	

[Симулятор] Процессор: NIIET NT32M4F1, Частота: 20,000,000 Hz, PC=00002EC4 [XTimer::Init()] Закладка восстановлена. Ready

Keil uVision

D:\VectorCARD_blinky_Keil\VectorCARD_blinky\VectorCARD_blinky.uvprojx - µVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers

Register	Value
Core	
R0	0x000001E0
R1	0x80011000
R2	0x00000000
R3	0x00000064
R4	0x00000B50
R5	0x00000B50
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20000410
R14 (LR)	0x00000B2F
R15 (PC)	0x000008E0
xPSR	0x01000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	5325
Sec	0.00053250
FPU	

Project | Registers

Disassembly

```
0x000008DC 6348      STR          r0,[r1,#0x34]
  202: }
0x000008DE 4770      BX           lr
  136:         if (timer > TimerPeriod)
  137:         {
0x000008E0 484A      LDR          r0,[pc,#296]  ; @0x00000A0C
```

main.c | startup_K1921BK01T.s | K1921VK01T_init.c | system_K1921BK01T.c

```
127
128
129  //Функция мигания светодиодами
130  void LED_blink (void)
131  {
132    static int timer; //переменная таймера
133    static int LEDs;  //переменная состояния светодиодов
134
135
136    if (timer > TimerPeriod)
137    {
138      timer = 0;
139      LEDs++;
140      LEDs &= 0x3F;
141
142      //Алгоритм 1
143      if (LEDs & (1<<5))
144        VD1_ON;
145      else
146        VD1_OFF;
```

Command

```
Load "D:\\VectorCARD_blinky_Keil\\VectorCARD_blinky\\Objects\\VectorCA

*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 2908 Bytes (8%)
```

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess

Call Stack + Locals

Name	Location/Value	Type
LED_blink	0x000008E0	void f()
timer	0x00000000	static - int
LEDs	0x00000000	static - int
main	0x00000000	int f()

Call Stack + Locals | Memory 1

ST-Link Debugger | t1: 0.00053250 sec | L:134 C:1 | CAP NUM SCRL

IAR EVARM

w - IAR Embedded Workbench IDE - ARM 8.11.1

File Edit View Project Debug Disassembly ST-Link Tools Window Help

Workspace

Debug

Files

VectorCARD_blinky_...
- core_cm4.h
- core_cm4_simd.h
- core_cmFunc.h
- core_cmInstr.h
- DSP.h
- K1921VK01T_init.c
- main.c
- main.h
- Output

VectorCARD_blinky_IAR

main.c | core_cm4.h | K1921VK01T_init.c | main.h | DSP.h

main()

```
int16 main (void){
        //Инициализация микроконтроллера: настройка таймеров, инициализаци перие
        InitCLK();

        //Инициализация периферии для управления диодами
        LED_init();

        //бесконечный цикл
        while(1)
        {
                LED_blink();    //мигание диодами
        }
}

/*@}*/
```

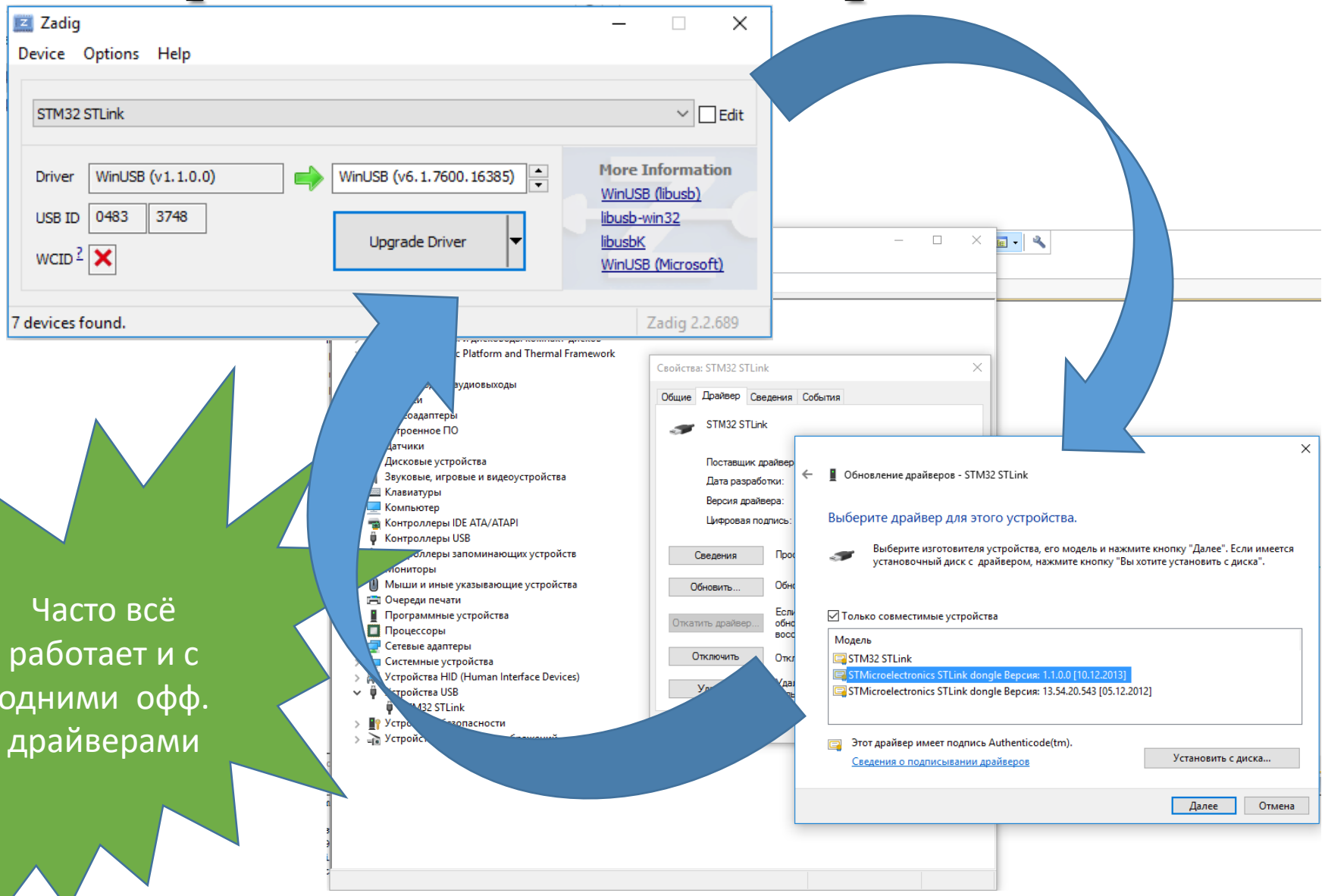
Debug Log

Log

Sun Apr 23, 2017 16:14:14: Target reset
Sun Apr 23, 2017 16:14:14: Downloaded D:\VectorCARD_blinky_IAR\Debug\Exe\VectorCARD_blinky_IAR.out to flash memory.
Sun Apr 23, 2017 16:14:14: Hardware reset with strategy 0 was performed
Sun Apr 23, 2017 16:14:14: 2856 bytes downloaded into FLASH (3.25 Kbytes/sec)
Sun Apr 23, 2017 16:14:14: Loaded debugee: D:\VectorCARD_blinky_IAR\Debug\Exe\VectorCARD_blinky_IAR.out
Sun Apr 23, 2017 16:14:14: Hardware reset with strategy 0 was performed

Ready | Ln 200, Col 1 | UTF-8 | CAP NUM 3AM

Смена драйверов JTAG между официальными и открытыми



Если вдруг микроконтроллер перестал прошиваться

Такое может случиться, если прошита неправильная программа:

- неправильно настраивается тактирование и ядро вообще перестает работать
- Переопределяются ножки, использующие JTAG

Что делать?

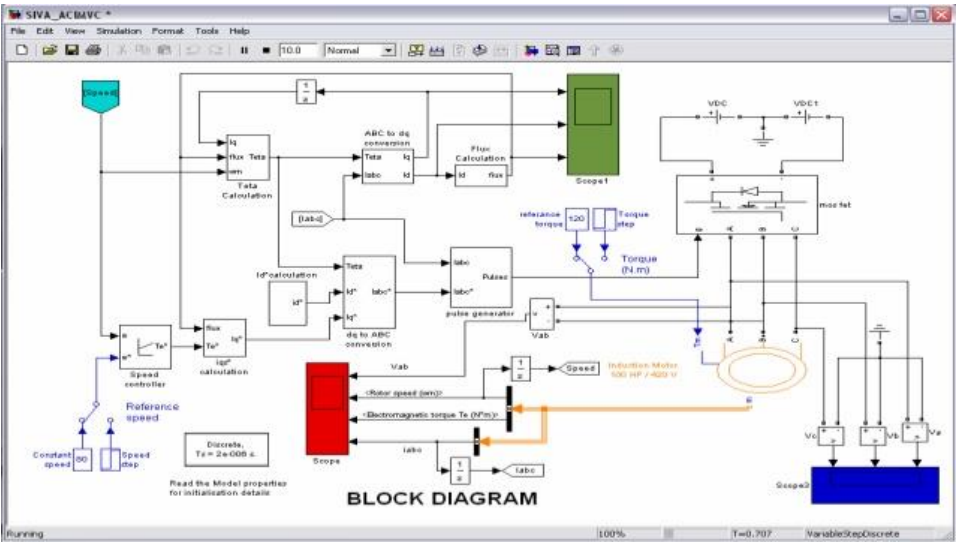
7.3 Алгоритм стирания всей флеш-памяти

1 Во время сброса микроконтроллера анализируется состояние вывода H2. Если вывод находится в состоянии логической единицы (подтянут к 3,3 В), то загрузочная и пользовательская флеш-памяти (включая защищенный и информационный блоки) переводятся в режим, в котором чтение запрещено (при чтении возвращаются нули).

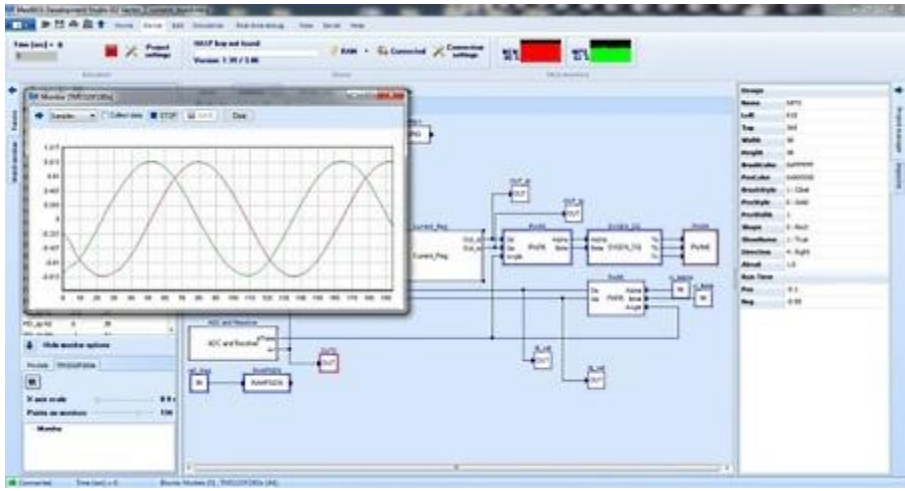
2 Далее по отладочному интерфейсу (SWD или JTAG) должна быть подана команда записи значения 00000001h в регистр FLASH_FULL_ERASE, после чего будет активировано полное стирание.

«Рисование» программы для микроконтроллера

MATLAB SIMULINK



НПФ МЕХАТРОНИКА-ПРО «MEXBIOS»



- Можно «рисовать» программу из готовых блоков вместо написания кода на Си
- Код на Си обычно все равно писать надо, но внутри блоков
- Подход уменьшает общую производительность и контроль над происходящим внутри МК
- Не всегда графическое представление программы удобнее, чем модульная структура программы на Си

Начало работы с К1921ВК01Т и поддержка

- **Примеры по «миганию светодиодом», запуску ШИМ есть в комплекте с Vector IDE (motorcontrol.ru/vector-ide)**

- **Примеры по работе с АЦП, UART, часами и так далее есть в репозитории Bitbucket НИИЭТ (bitbucket.org/niietcm4)**

- **Пример проекта для IAR, Keil есть на интернет-форуме НИИЭТ (forum.niiet.ru). Также там есть обсуждение открытых средств разработки.**

- **Техническая поддержка по микроконтроллеру производится на интернет-форуме НИИЭТ (forum.niiet.ru), по примерам и заголовочным файлам на ресурсе Bitbucket НИИЭТ (bitbucket.org/niietcm4), по среде Vector IDE и вопросам управления электродвигателем на интернет-форуме НПФ Вектор (motorcontrol.ru/forum/).**