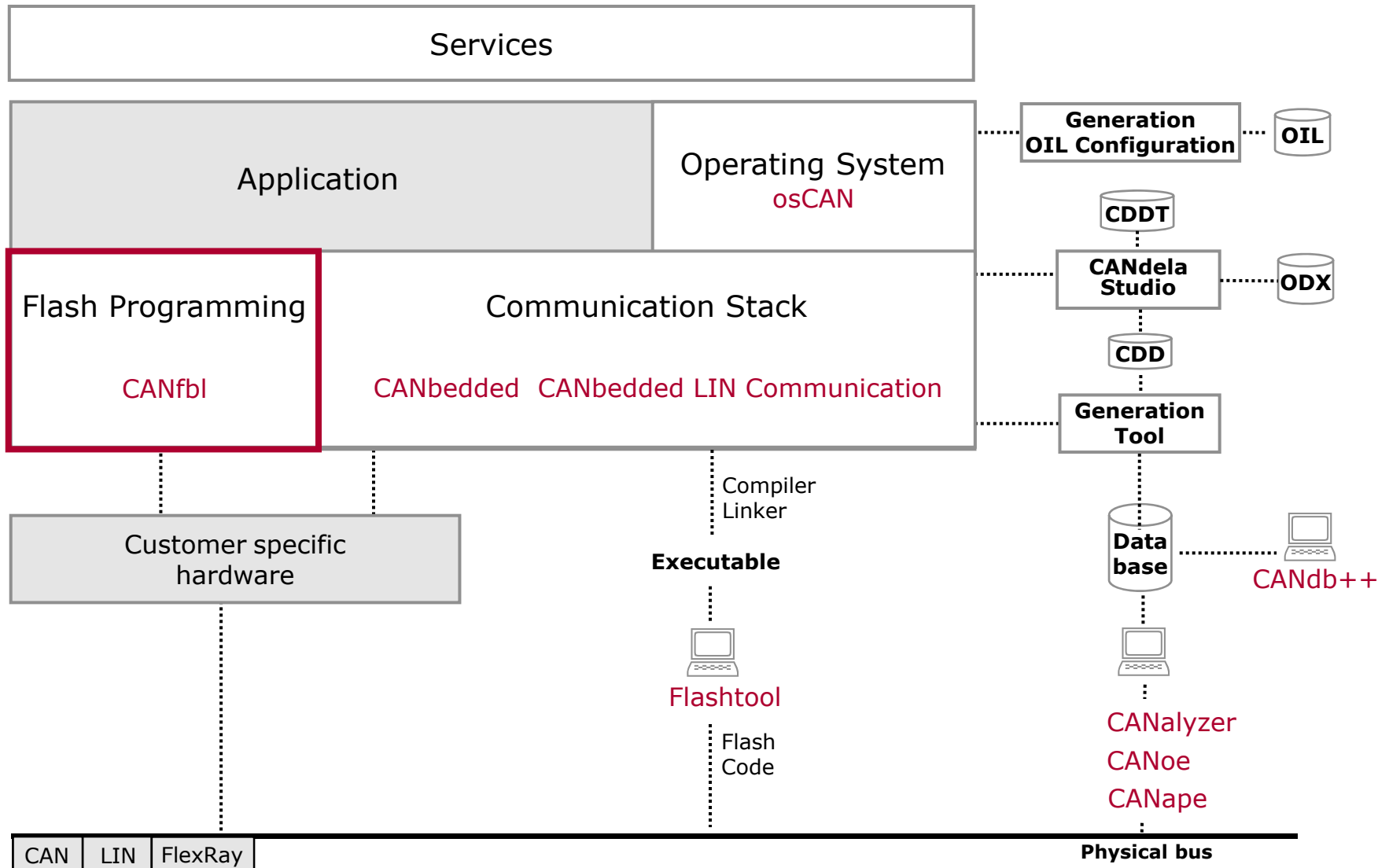


## Vector Embedded Solutions

### Embedded Software for Flash Bootloader

# Vector's Embedded Software Components



# Flash Bootloader Introduction

## Overview

ECUs with Bootloader

vFlash



Bus System

### Usage of Flash

- ❑ Development
- ❑ EOL programming
- ❑ After-Sales Programming

# Flash Bootloader Introduction

Very hardware dependent

Currently more than **40** supported hardware platforms



New hardware platforms will be supported on request

# Flash Bootloader Introduction

OEM specific

Vector knows what the OEMs want and provides it

ECUs with Bootloader



- ▶ Flash programming according to ISO 14230 („K-Line“)
- ▶ Flash programming via CAN according to ISO 15765-3 (WD) Protocols: KWP-2000 on CAN, ISO 14229 (UDS)
- ▶ Flash programming according to ASAM Protocols: CCP („CAN Calibration Protocol“), XCP („Universal Measurement and Calibration Protocol“)

# Flash Bootloader Introduction

Support for multiple bus systems

Vector offers flash solutions for usual bus systems in the automotive environment



Bus System CAN LIN FlexRay MOST ETHERNET

# Flash Bootloader Introduction

## Provision Of Source Code

```
include <sys/types.h>
#include <sys/stat.h>
#include <stdio.h>

#ifndef S_ISREG
#define S_IFREG __S_IFREG
#endif
#ifndef S_IFMT
#define S_ISREG(m) (((m) & S_IFMT) == S_IFREG)
#endif /* not S_ISREG */

#ifndef O_BINARY
#ifdef __O_BINARY
#define O_BINARY __O_BINARY
#else
#define O_BINARY 0
#endif
#endif

#include "filemap.h"
int filemap(const char *name,
            void (*processor)(const void *, size_t, const char *, void *arg),
            void *arg)
{
    size_t nbytes;
    int fd;
    int n;
    struct stat sb;
    void *p;

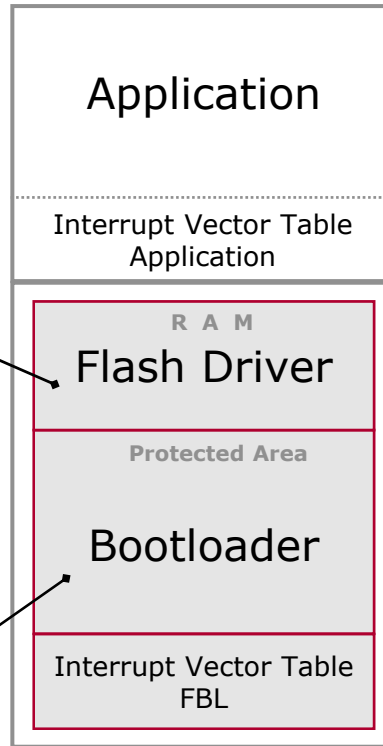
    fd = open(name, O_RDONLY|O_BINARY);
    if (fd < 0) {
        perror(name);
        return 0;
    }
    if (fstat(fd, &sb) < 0) {
        perror(name);
        return 0;
    }
    if (!S_ISREG(sb.st_mode)) {
        fprintf(stderr, "%s: not a regular file\n", name);
        return 0;
    }
}
```



Vector Flash Bootloader Software  
is Source Code.

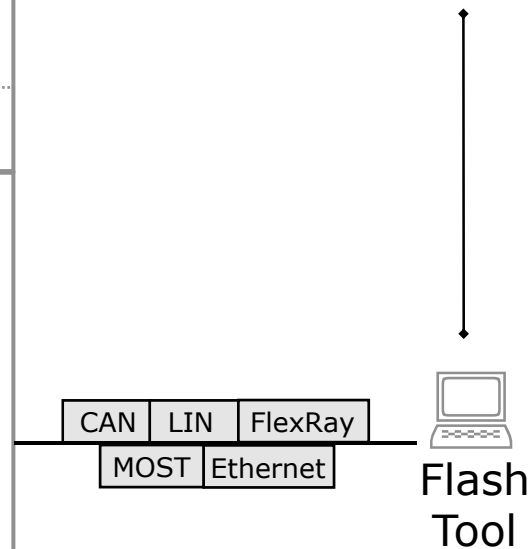
### FBL memory architecture

The **Flash Driver** erases the Flash and writes the new Data into the flash. It contains the derivative-specific routines.



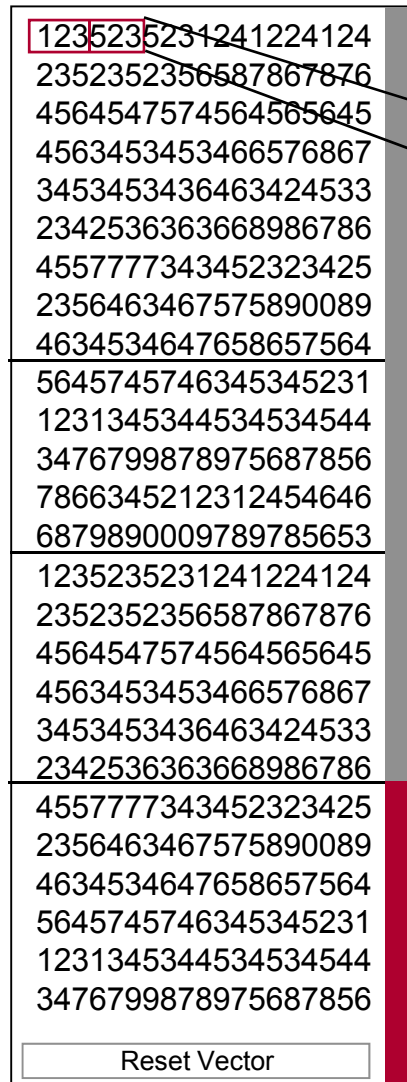
The **Bootloader** contains OEM-specific algorithm, basic bus communication, a Transport Protocol and Diagnostics (KWP2000, UDS), all code optimized to use minimum memory.

The **Flashtool** is an easy to use PC tool and controls the download of your application (as executable) via a PC card, e.g. CANcardXL.



# Overview

## Memory Units



**Smallest programmable flash unit (page).**  
Memory-Alignments must be considered!



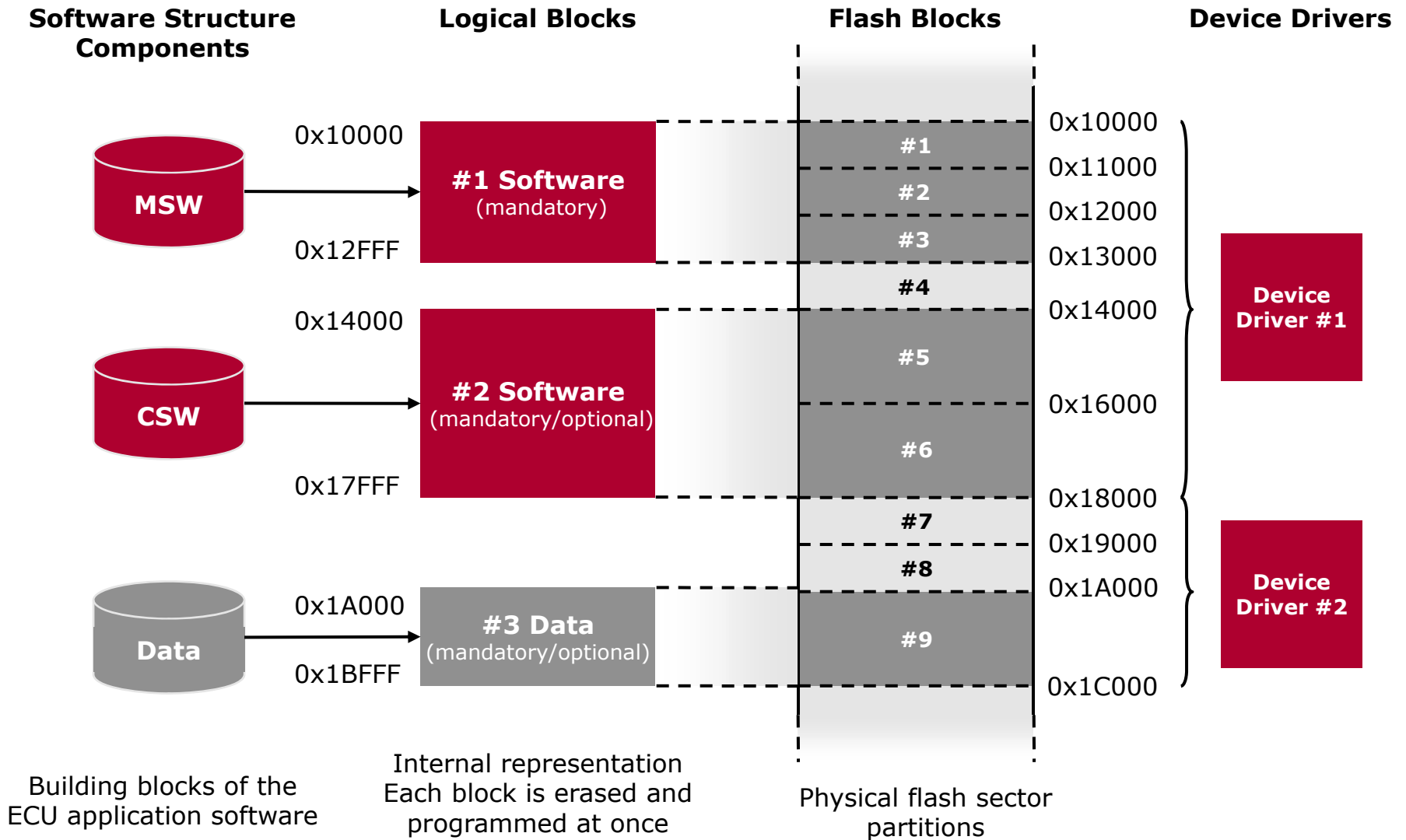
**Eraseable flash blocks (sector)**

**Do not write twice to the same location without previous erase operation!!**

**Protected sector in flash memory for bootloader**

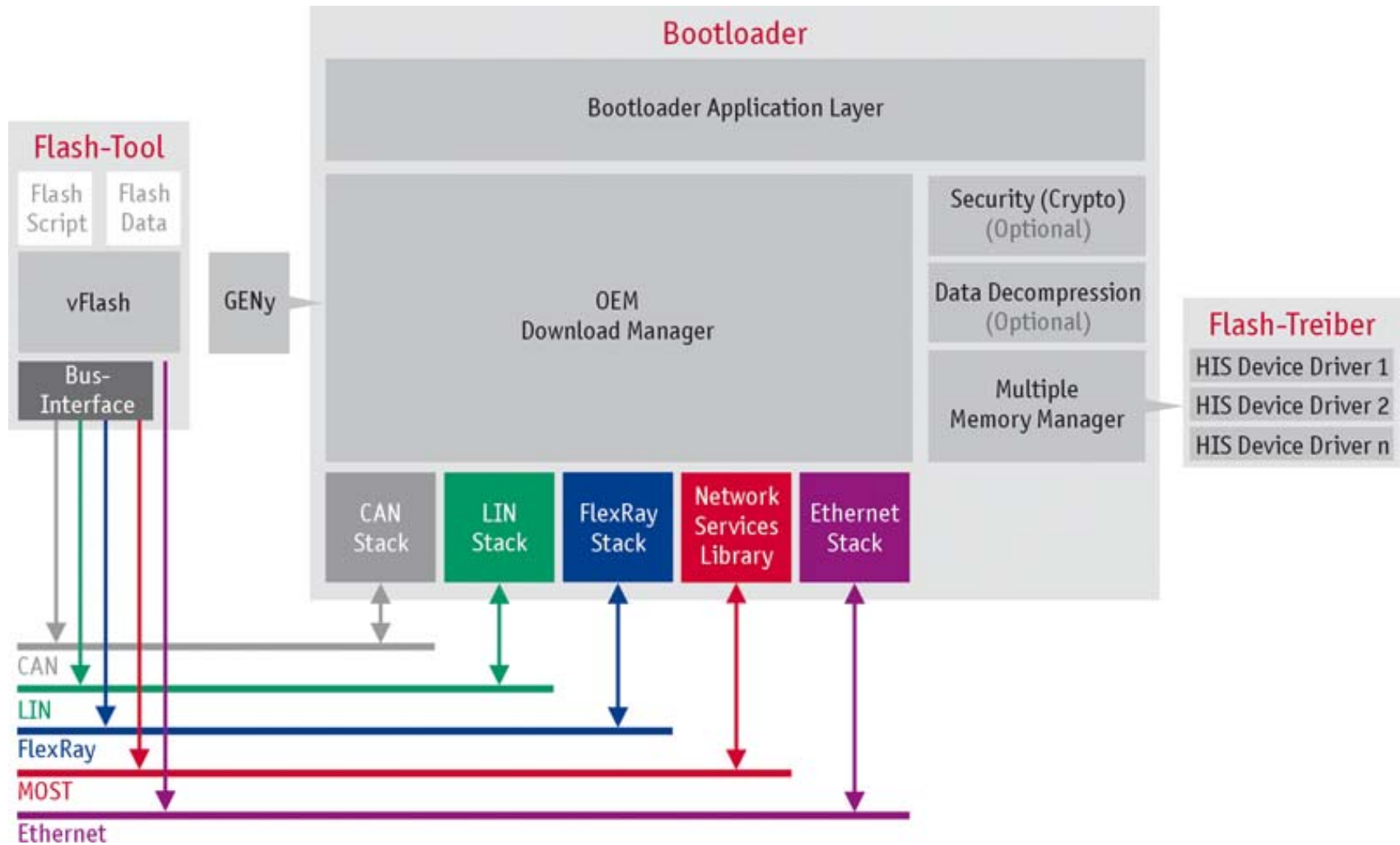
# Overview

## Memory Partition

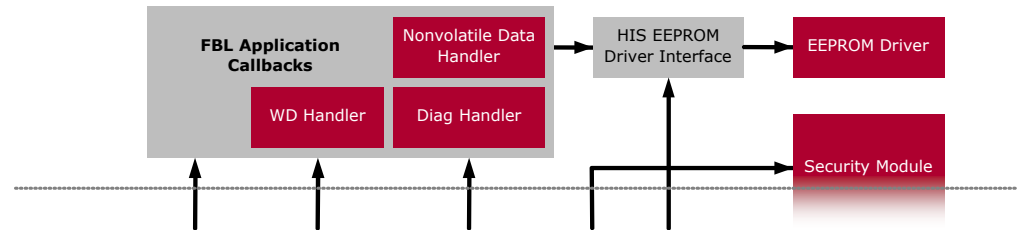


# Overview

## Bootloader building blocks



- ▶ Flash Bootloader offers great flexibility
  - ▶ GENy configuration
    - > Memory mapping
    - > Communication settings
    - > ...
  - ▶ User callbacks
    - > Watchdog / SBC handling
    - > Transceiver initialization
    - > Non volatile memory
  - ▶ Memory drivers
  - ▶ Bootloader AND Application have to be adapted!
    - > Does not run "out of the box"

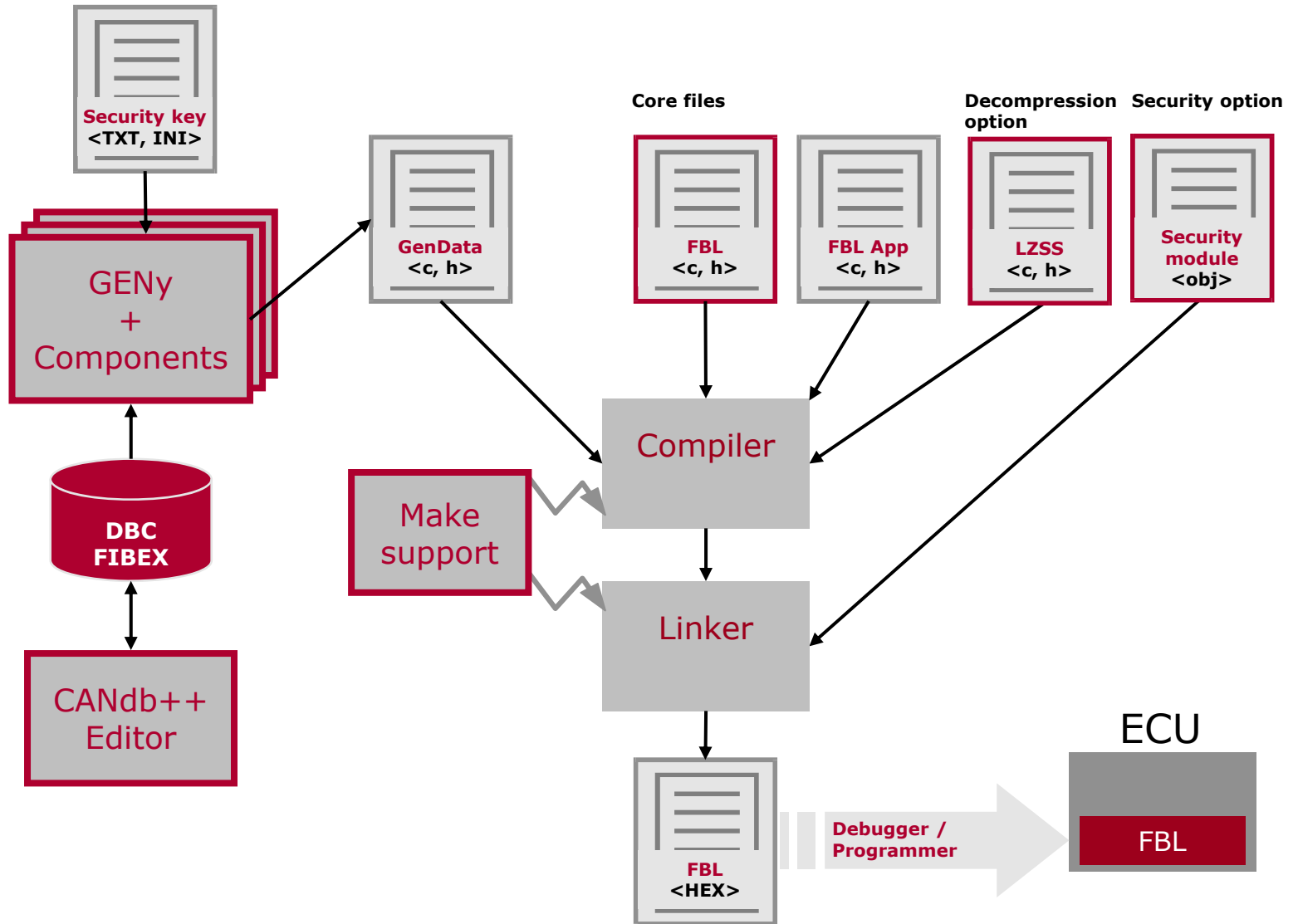


- ▶ Multiple memory devices
  - ▶ Internal/external flash
  - ▶ EEPROM
  - ▶ Dual processor ECUs
- ▶ Multiple identity
- ▶ Security
- ▶ Data processing
  - ▶ Encryption
  - ▶ Compression
- ▶ Bus systems
  - ▶ CAN
  - ▶ LIN
  - ▶ FlexRay
  - ▶ Ethernet, MOST

- ▶ Development process is supported by Vector toolchain
  - ▶ Generation tool
    - > GENy – Easy configuration of FBL through GUI
  - ▶ Development support tools
    - > Makesupport
    - > Hexview
  - ▶ Flash tool
    - > vFlash
- ▶ Vector Support

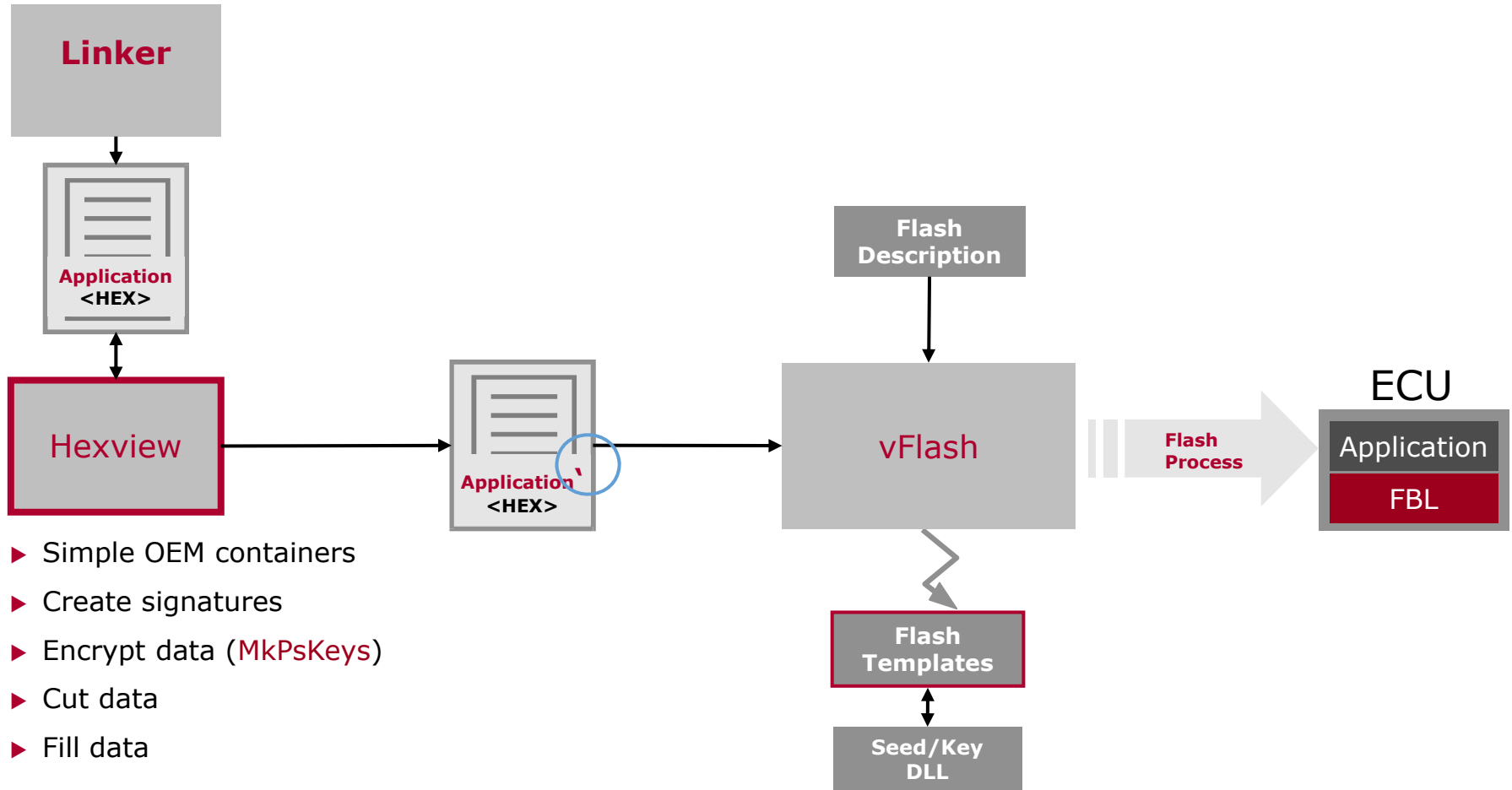
# Vector – Toolchain

## Bootloader - Build



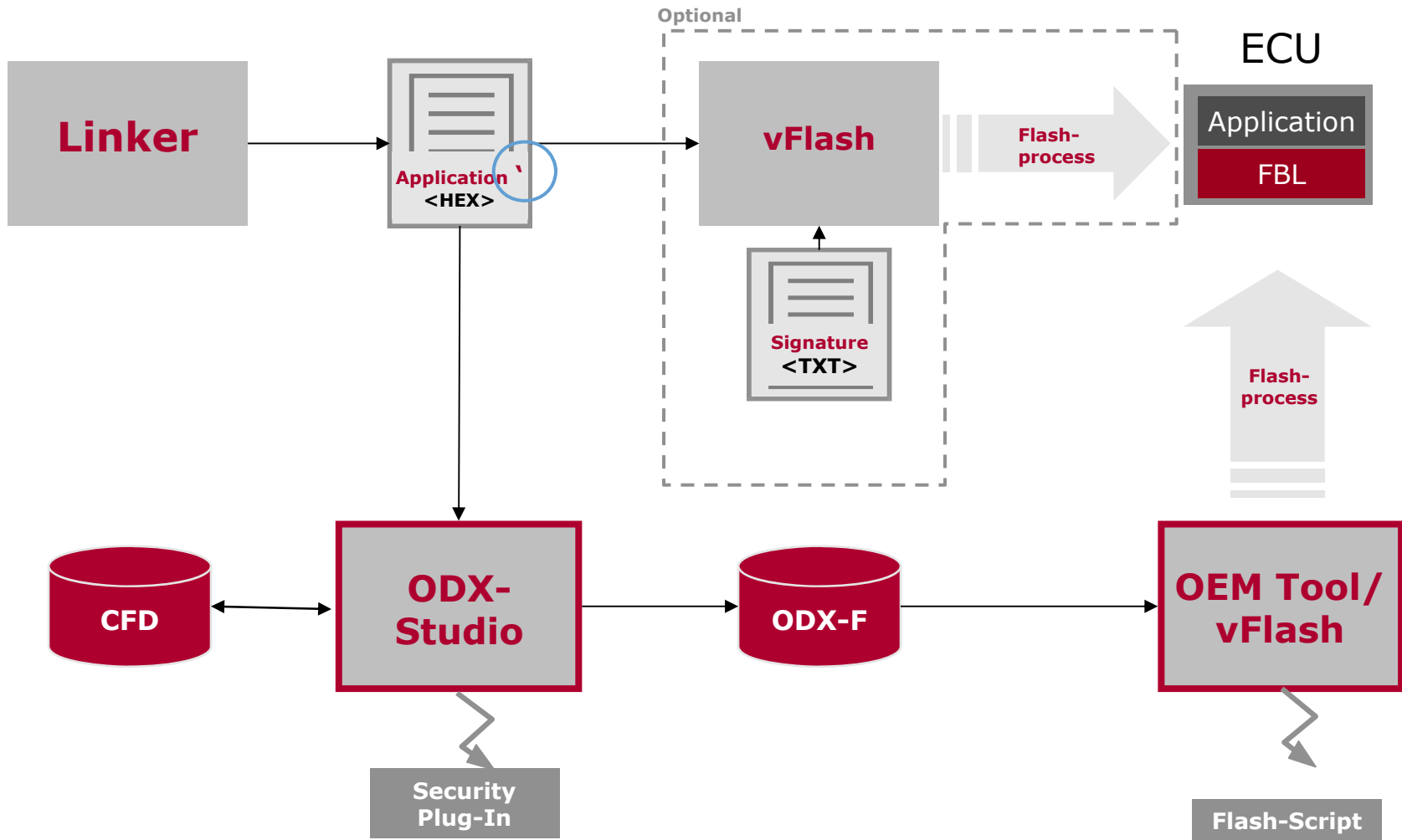
# Vector - Toolchain

## Download preparation (development phase)

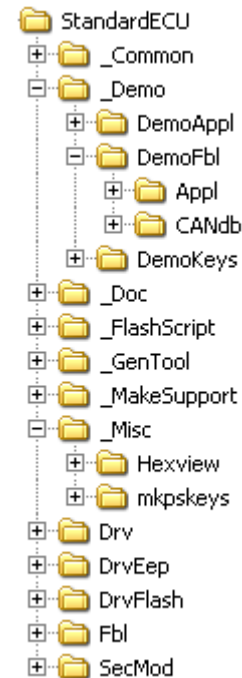


# Vector – Toolchain

## ODX and Flash Download (production phase)



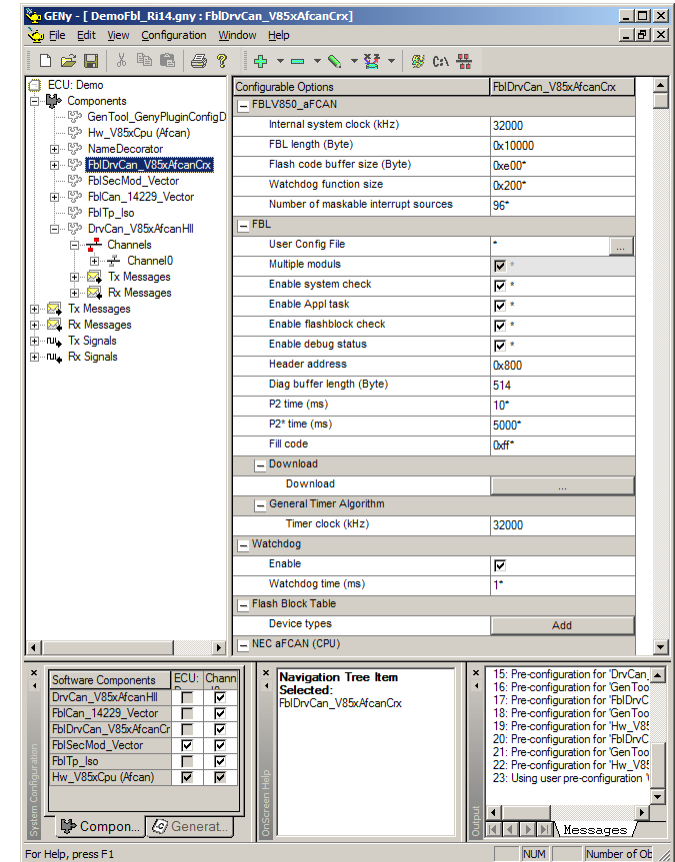
- ▶ Flash Bootloader core files
- ▶ User callback templates
- ▶ Flash driver
- ▶ CAN driver
- ▶ Dummy EEPROM driver (RAM buffer)
- ▶ Security module (object code)
- ▶ Generation tool GENy
- ▶ Makesupport
- ▶ Flash script
- ▶ Demo FBL
- ▶ Documentation
- ▶ Development support tools



# Toolchain

## Generation Tool – GENy

- ▶ Comfortable Bootloader configuration
- ▶ Graphical interface
- ▶ Component based
- ▶ Structured by module
  - ▶ Hardware
  - ▶ Security module
  - ▶ OEM specific
  - ▶ Transport protocol
  - ▶ CAN driver
- ▶ Preconfigured for target derivative



# Toolchain

## Generation Tool – GENy

- ▶ Hardware configuration
  - ▶ Timer clock
  - ▶ Watchdog trigger
  - ▶ Device types
  - ▶ ...
- ▶ General FBL configuration
  - ▶ Diagnostic buffer size
  - ▶ Communication timeouts
  - ▶ ...

Configurable Options	FblDrvCan_V85xAfcAnCrX
[- FBLV850_aFCAN	
Internal system clock (kHz)	32000
FBL length (Byte)	0x10000
Flash code buffer size (Byte)	0xe00*
Watchdog function size	0x200*
Number of maskable interrupt sources	96*
[- FBL	
User Config File	* <input type="text" value="..."/>
Multiple moduls	<input checked="" type="checkbox"/> *
Enable system check	<input checked="" type="checkbox"/> *
Enable Appl task	<input checked="" type="checkbox"/> *
Enable flashblock check	<input checked="" type="checkbox"/> *
Enable debug status	<input checked="" type="checkbox"/> *
Header address	0x800
Diag buffer length (Byte)	514
P2 time (ms)	10*
P2* time (ms)	5000*
Fill code	0xff*
[- Download	
Download	<input type="text" value="..."/>
[- General Timer Algorithm	
Timer clock (kHz)	32000
[- Watchdog	
Enable	<input checked="" type="checkbox"/>
Watchdog time (ms)	1*
[- Flash Block Table	
Device types	<input type="text" value="Add"/>
[- NEC aFCAN (CPU)	

- ▶ Flash Block Table
  - ▶ Physical flash layout
  - ▶ Affects erase granularity
- ▶ Logical Block Table
  - ▶ Logical software partitioning
  - ▶ Mapping on flash block table

	StartAddress	EndAddress	Device Type	Description
0	0x0*	0x3fff	Protected	*
1	0x4000	0x7fff	Protected	*
2	0x8000	0xbfff	Protected	*
3	0xc000	0xffff	Protected	*
4	0x10000	0x13fff	Flash	*
5	0x14000	0x17fff	Flash	*
6	0x18000	0x1bfff	Flash	*
7	0x1c000	0x1ffff	Flash	*
8	0x20000	0x23fff	Flash	*
9	0x24000	0x27fff	Flash	*
10	0x28000	0x2bfff	Flash	*
11	0x2c000	0x2ffff	Flash	*
12	0x30000	0x33fff	Flash	*
13	0x34000	0x37fff	Flash	*
14	0x38000	0x3bfff	Flash	*
15	0x3c000	0x3ffff	Flash	*
16	0x40000	0x43fff	Flash	*

	Block Index	Disposability	StartAddress	EndAddress	Max.	Description
0	0x9a	mandatory	0x10000	0x3ffff	0x0*	*
1	0x9a	mandatory	0x40000	0x5ffff	0x0*	*
2	0x9b	mandatory	0x60000	0x7fff	0x0*	*

- ▶ Security module configuration
  - ▶ Seed/Key constant
  - ▶ Security class
  - ▶ Authenticity key for signature calculation

Configurable Options	FblSecMod_Vector
[-] FblSecMod_Vector	
Constant for key	0xffffffff*
Timeout for key (ms)	100*
Call cycle	10*
Memory access mode	Memory Access
Operation mode of security module	Production
CRC	Size optimized
Enable CRC Total	<input checked="" type="checkbox"/>
Add segment address and length to	<input checked="" type="checkbox"/>
[-] File Selection	
Select File Type for Key	Authenticity File
Path for authenticity key file	D:\usr\CANDISTRIB\VolvoAB\Gene...
Select security class	Class CCC

# Toolchain

## Generation Tool – GENy

- ▶ Transport protocol configuration
  - ▶ Call cycle
  - ▶ Timing
  - ▶ Flow control
  - ▶ Diagnostic callbacks

Configurable Options	FblTp_Iso
[-] Advanced settings	
Iso-compliance for 15765-2.2	<input checked="" type="checkbox"/> *
Variable Tx DLC	<input checked="" type="checkbox"/> *
Rx DLC check:	No Rx DLC check ▾
Use internal memcpy	<input checked="" type="checkbox"/>
Use CAN confirmation sync.	<input checked="" type="checkbox"/> *
[-] Timing	
TP call cycle (ms):	1*
[-] Tx Timing	
Confirmation timeout or CAN-Tx retry	70*
Tx Timeout FC (ms)	250*
Tx Transmit CF (ms)	10*
[-] Tx FlowControl	
STmin (ms)	0*
Blocksize	0*
[-] Rx Timing	
Rx Timeout CF (ms)	250*
[-] Callbacks	
TpErrorIndication	FblDiagErrorIndication*
TpConfirm	FblDiagTpConfirmation*
TpDataIndication	FblDataInd*
Data buffer overrun indication	FblDataIndBO*
TpCopyFromCAN	(void)MEMCPY((void *)dst, 0, MEMFROMBY_MEMORY)
TpCopyToCAN	(void)MEMCPY((void *)dst, 0, MEMFROMBY_MEMORY)
TpNotifyTx	*
TpCanMessageTransmitted	*
TpCanMessageReceived	*
TpRxSF	FblDiagTpFirstSingleFrameIndi

# Toolchain

## Generation Tool – GENy

- ▶ CAN driver configuration
  - ▶ Bittiming
    - > Multiple init structures

The image shows two overlapping windows from the GENy tool. The foreground window is titled 'CAN bustiming register setup' and contains the following fields and controls:

- Bit timing section with a 'View mode' selector (1, 2).
- Clock (kHz): 82000
- Baudrate (kBaud): 500.0
- CnBRP register (hex): 0003
- CnBTR register (hex): 220B
- nominal bit timing (bus) graph showing a blue bar with a red and green segment.
- TSeg1 (time quanta): 12
- TSeg2 (time quanta): 3
- Time quantum (ns): 125
- Bit time (μs): 2
- Buttons: Calculate baudrate, Calculate bustiming register
- Dropdowns: Samples (1), Prescaler (4)
- Table of register values:

CnBRP	CnBTR	Sample	BTL cycles	SJW
0x0003	0x330A	75%	16	4
0x0003	0x2706	50%	16	3
0x0003	0x2607	56%	16	3
0x0003	0x2508	62%	16	3
0x0003	0x2409	68%	16	3
0x0003	0x230A	75%	16	3
0x0003	0x220B	81%	16	3
0x0003	0x1706	50%	16	2

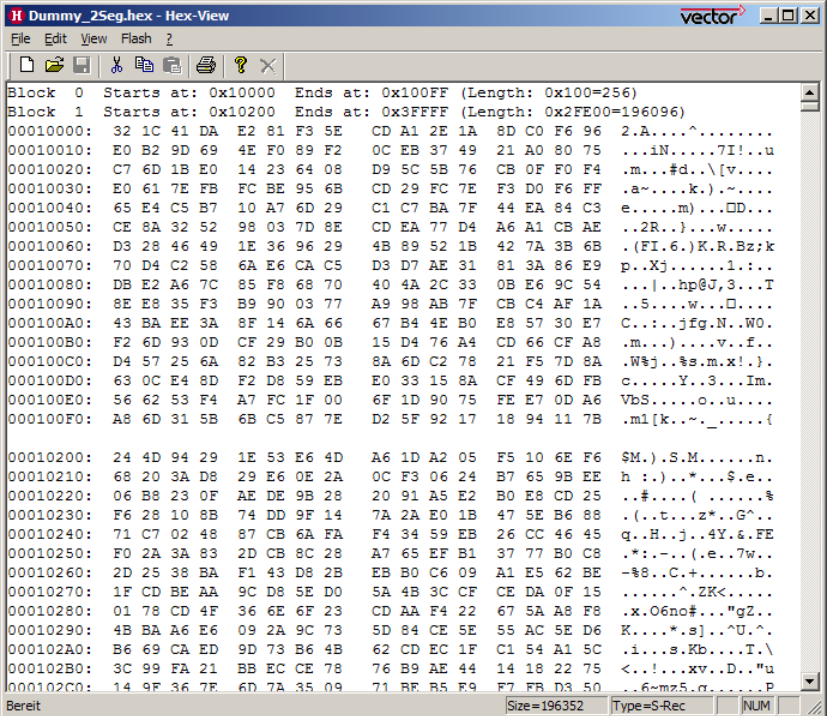
The background window is titled 'Configurable Options' and shows a list of initialization options for Channel0:

Option	Value
Interrupt Enable (CnE)	0x2700*
Bit Rate Prescaler (Cn BRP)	0x3*
Bit Rate (Cn BTR)	0x220b
Acceptance Filter Configuration	...
Bustiming Configuration	...
Interrupt Enable (CnE)	0x2700*
Bit Rate Prescaler (Cn BRP)	0x7
Bit Rate (Cn BTR)	0x220b
Acceptance Filter Configuration	...
Bustiming Configuration	...

# Toolchain

## Development support – Hexview

- ▶ Prepare application for flashing
- ▶ Binary and Hex file manipulation
  - ▶ Checksum and signature calculation
  - ▶ Alignment
  - ▶ Gap fill
  - ▶ ...
- ▶ All operations available through command line interface



```
Dummy_2Seg.hex - Hex-View
File Edit View Flash ?
[Icons]
Block 0 Starts at: 0x10000 Ends at: 0x100FF (Length: 0x100=256)
Block 1 Starts at: 0x10200 Ends at: 0x3FFFF (Length: 0x2FE00=196096)
00010000: 32 1C 41 DA E2 81 F3 5E CD A1 2E 1A 8D C0 F6 96 2.A....^.....
00010010: E0 B2 9D 69 4E F0 89 F2 0C EB 37 49 21 A0 80 75 ...iN....7I!..u
00010020: C7 6D 1B E0 14 23 64 08 D9 5C 5B 76 CB 0F F0 F4 .m...#d...[v...
00010030: E0 61 7E FB FC BE 95 6B CD 29 FC 7E F3 D0 F6 FF .a~....k.)~....
00010040: 65 E4 C5 B7 10 A7 6D 29 C1 C7 BA 7F 44 EA 84 C3 e.....m)...DD...
00010050: CE 8A 32 52 98 03 7D 8E CD EA 77 D4 A6 A1 CB AE ..2R.)...w....
00010060: D3 28 46 49 1E 36 96 29 4B 89 52 1B 42 7A 3B 6B .(FI.6.)K.R.Bz;k
00010070: 70 D4 C2 58 6A E6 CA C5 D3 D7 AE 31 81 3A 86 E9 p..Xj.....1...
00010080: DB E2 A6 7C 85 F8 68 70 40 4A 2C 33 0B E6 9C 54 ...|.hp@J,3...T
00010090: 8E E8 35 F3 B9 90 03 77 A9 98 AB 7F CB C4 AF 1A ..S.....w...D...
000100A0: 43 BA EE 3A 8F 14 6A 66 67 B4 4E B0 E8 57 30 E7 C...jfg.N..W0.
000100B0: F2 6D 93 0D CF 29 B0 0B 15 D4 76 A4 CD 66 CF A8 .m...)...v..f..
000100C0: D4 57 25 6A 82 B3 25 73 8A 6D C2 78 21 F5 7D 8A .W#j...%s.m.x!..}
000100D0: 63 0C E4 8D F2 D8 59 EB E0 33 15 8A CF 49 6D FB c.....Y..3...Im.
000100E0: 56 62 53 F4 A7 FC 1F 00 6F 1D 90 75 FE E7 0D A6 VbS.....o..u...
000100F0: A8 6D 31 5B 6B C5 87 7E D2 5F 92 17 18 94 11 7B .ml[k..~_.....{

00010200: 24 4D 94 29 1E 53 E6 4D A6 1D A2 05 F5 10 6E F6 $M.).S.M.....n.
00010210: 68 20 3A D8 29 E6 0E 2A 0C F3 06 24 B7 65 9B EE h :.)...*...$.e..
00010220: 06 B8 23 0F AE DE 9B 28 20 91 A5 E2 B0 E8 CD 25 ..#....( .....%
00010230: F6 28 10 8B 74 DD 9F 14 7A 2A E0 1B 47 5E B6 88 .(.t...z*..G^..
00010240: 71 C7 02 48 87 CB 6A FA F4 34 59 EB 26 CC 46 45 q..H..j...4Y.&.FE
00010250: F0 2A 3A 83 2D CB 8C 28 A7 65 EF B1 37 77 B0 C8 .*!-..(e..7w..
00010260: 2D 25 38 BA F1 43 D8 2B EB B0 C6 09 A1 E5 62 BE -%B..C+.....b.
00010270: 1F CD BE AA 9C D8 5E D0 5A 4B 3C CF CE DA 0F 15 .....^ZK<.....
00010280: 01 78 CD 4F 36 6E 6F 23 CD AA F4 22 67 5A A8 F8 .x.O6no#...gZ..
00010290: 4B BA A6 E6 09 2A 9C 73 5D 84 CE 5E 55 AC 5E D6 K....*s.j]..^U.^
000102A0: B6 69 CA ED 9D 73 B6 4B 62 CD EC 1F C1 54 A1 5C .i...s.Rb...T.\
000102B0: 3C 99 FA 21 BB EC CE 78 76 B9 AE 44 14 18 22 75 <.!...xv..D...u
000102C0: 14 9F 36 7E 6D 7A 35 09 71 BF B5 F9 F7 FR D3 50 ..6~mz5.g.....P

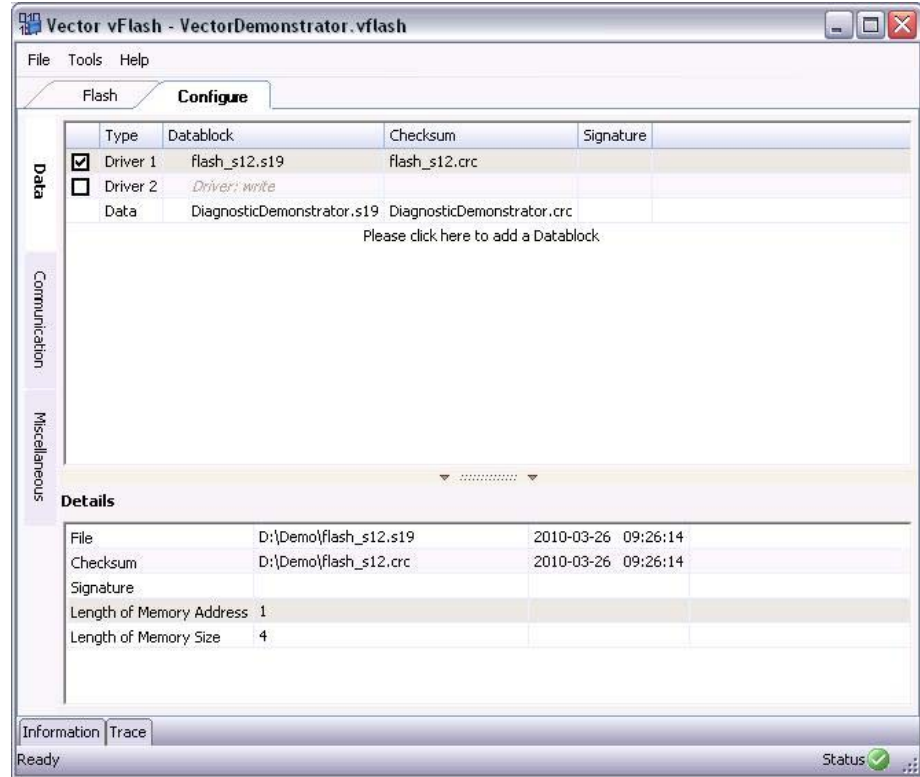
Bereit Size=196352 Type=S-Rec NUM
```

- ▶ Use case driven – different views for different use cases
  - > Native Flashing (Intel-Hex, Motorola-S, Binary), ODX-F, VBF
  - > Interactive *AND* batch processing
- ▶ Support different bus systems
  - > CAN
  - > DoIP
  - > Flexray
- ▶ Generic and data-driven
  - > Support of different OEM protocols and flash sequences
  - > User-oriented GUI (minimal interaction, clear progress overview)
  - > Easy start-up: Configuration is based on template (pre-defined OEM and/or Bootloader specifics)
  - > One flash procedure works for different data sources

# Toolchain

## vFlash - How to Configure

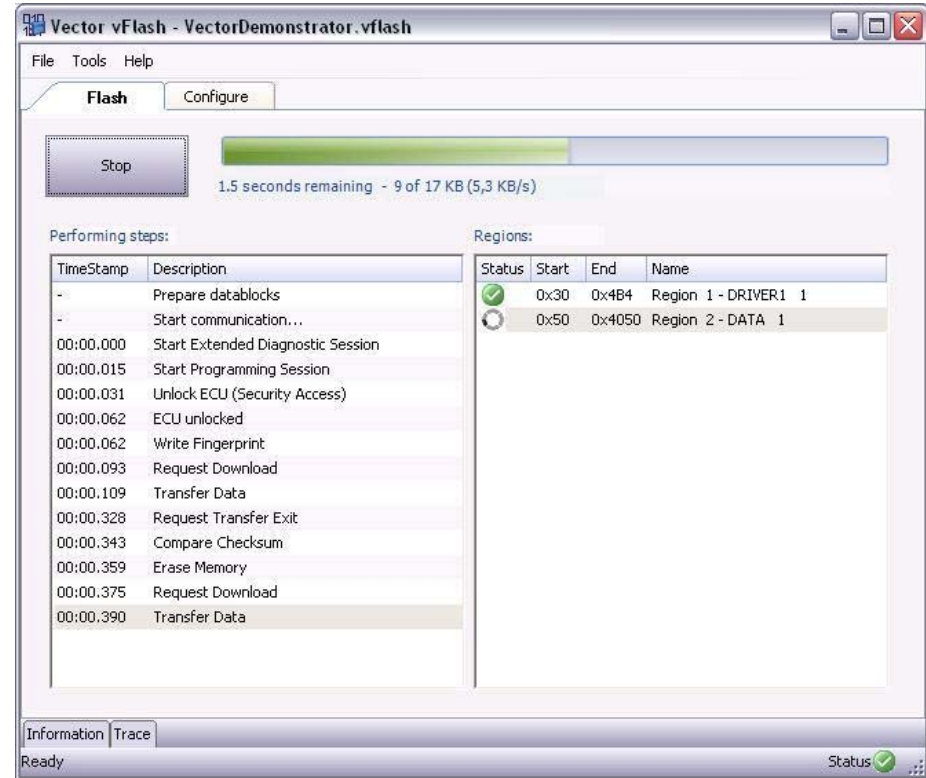
- ▶ Create new project based on flash template
  - > The template defines basic settings like flash procedure, OEM specifics, ...
- ▶ Set communication parameters and select Seed-Key-DLL
- ▶ Select data to be flashed
- ▶ Start flash
- ▶ Save project (optionally as Pack&Go)



# Toolchain

## vFlash – How to use

- ▶ Load existing project
  - ▶ Project file
    - > All required files are just referenced
  - ▶ Pack&Go
    - > All required files are packed together (Project, Seed&Key.dll, flashware, ...)
- ▶ Start flashing immediately



- ▶ Registration to the upcoming Webinars and the list of recorded Webinars:  
[http://www.vector.com/vi\\_webinars\\_en.html](http://www.vector.com/vi_webinars_en.html)
- ▶ The overview of Vector's training services:  
[http://www.vector.com/vi\\_training\\_en.html](http://www.vector.com/vi_training_en.html)
- ▶ Contact data for additional questions, product information or presentation:
  - ▶ [Armin.Happel@vector.com](mailto:Armin.Happel@vector.com)
  - ▶ +49 (0) 711 80670 364
  - ▶ [embedded@de.vector.com](mailto:embedded@de.vector.com)

Thank you for your attention.

For detailed information about Vector  
and our products please have a look at:

[www.vector-informatik.com](http://www.vector-informatik.com)

Author:

Armin Happel

[Armin.happel@vector.com](mailto:Armin.happel@vector.com)

Vector Informatik GmbH

Ingersheimer Str. 24

70499 Stuttgart