VECTOR

MICROSAR ComStackLib

VECTOR

Document Information

History

Author	Date	Version	Remarks
	- -		
	- -		
	- -		- -
	- -		
	- -		
	- -		
	- -		-

Reference Documents

No.	Source	Title	Version
[1]		-	

Caution

VECTOR

Contents

VECTOR

Illustrations

Tables

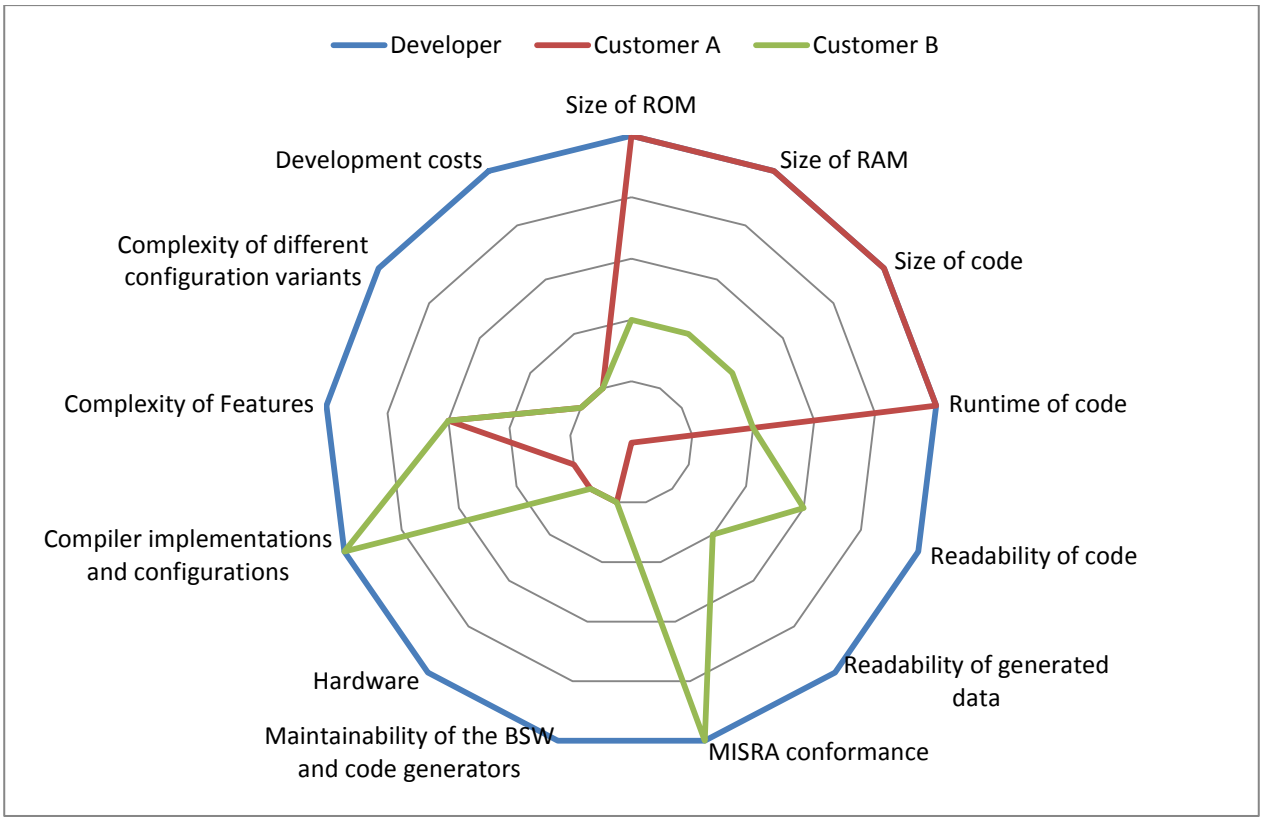
1 Component History

Component Version	New Features
	- - - -
	- - - - -
	- -

-

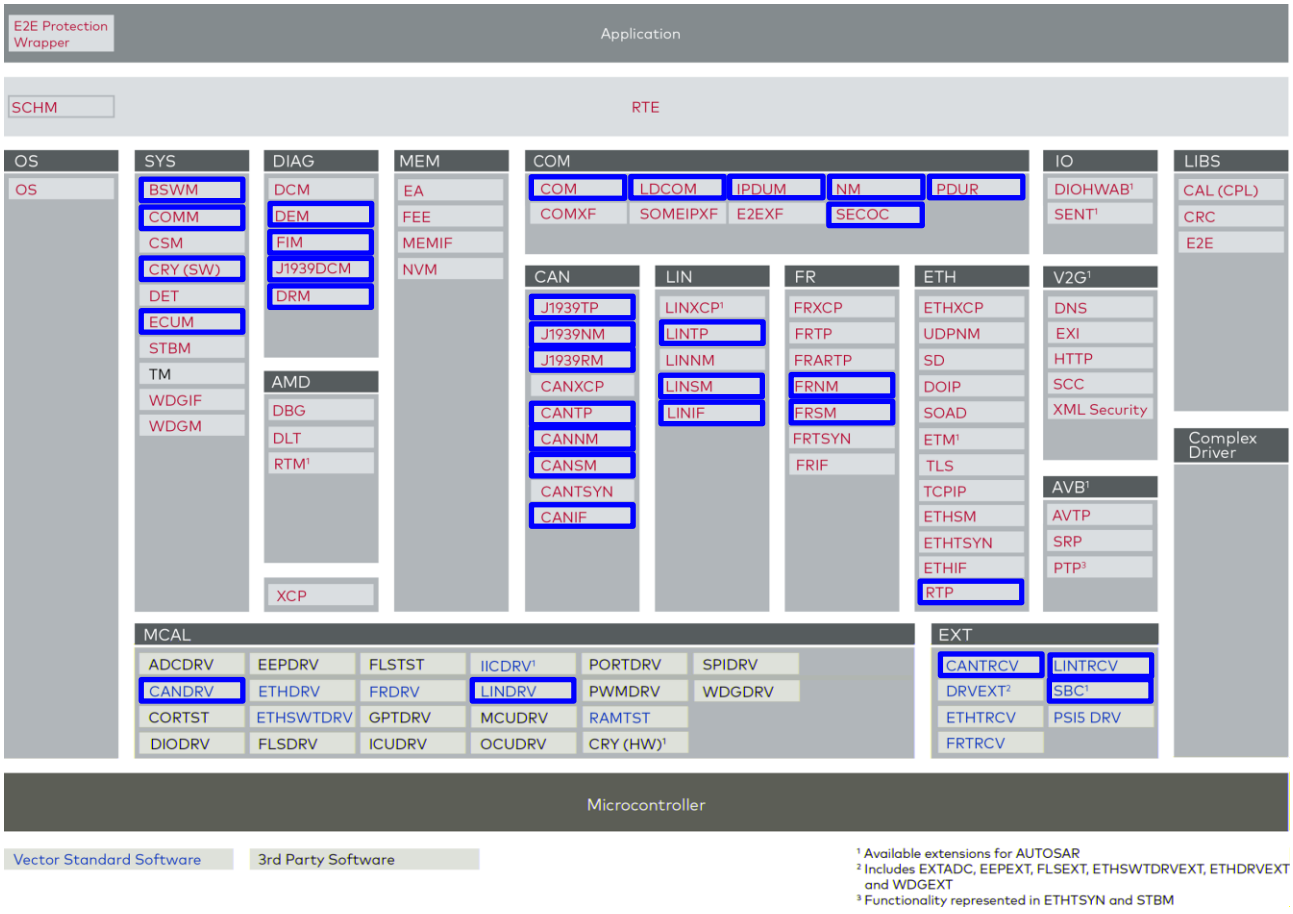
2 Introduction

Supported AUTOSAR Release*:	
Supported Configuration Variants:	- - -



-

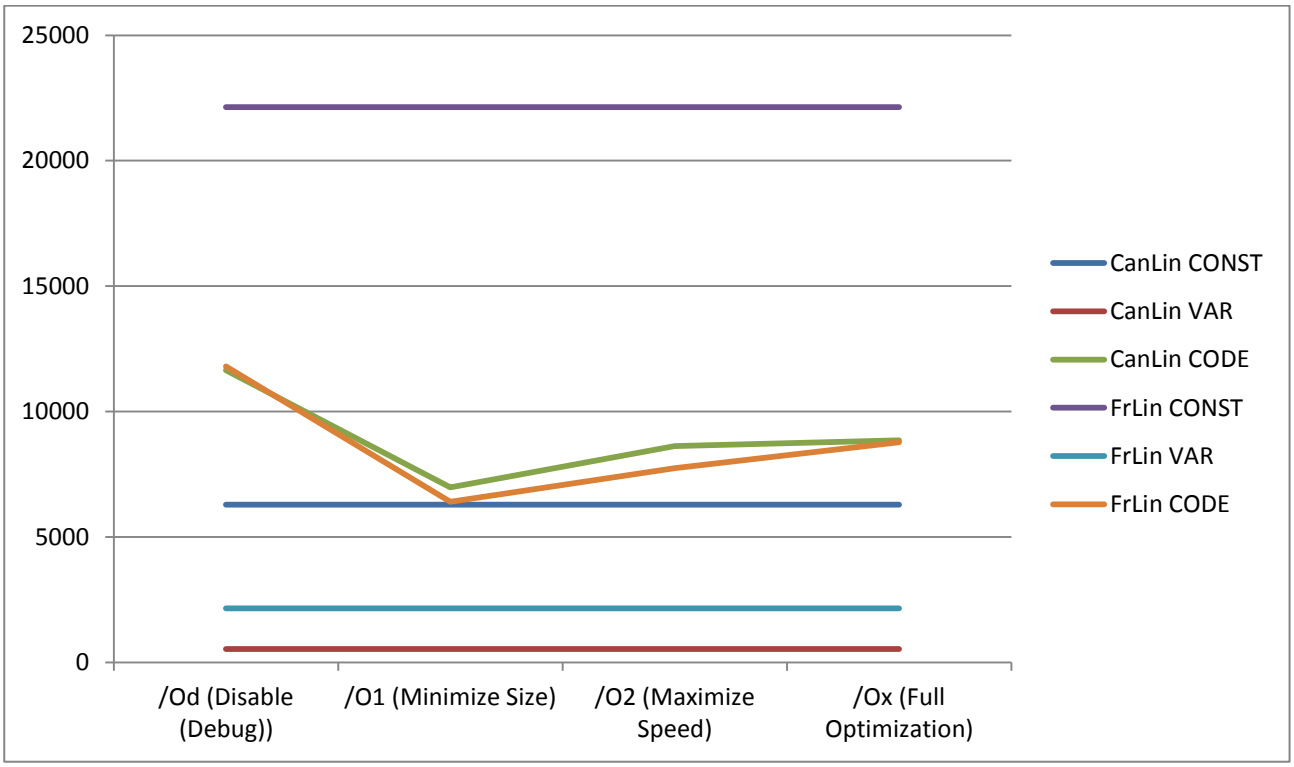
2.1 Architecture Overview



-

3 Functional Description

-



-

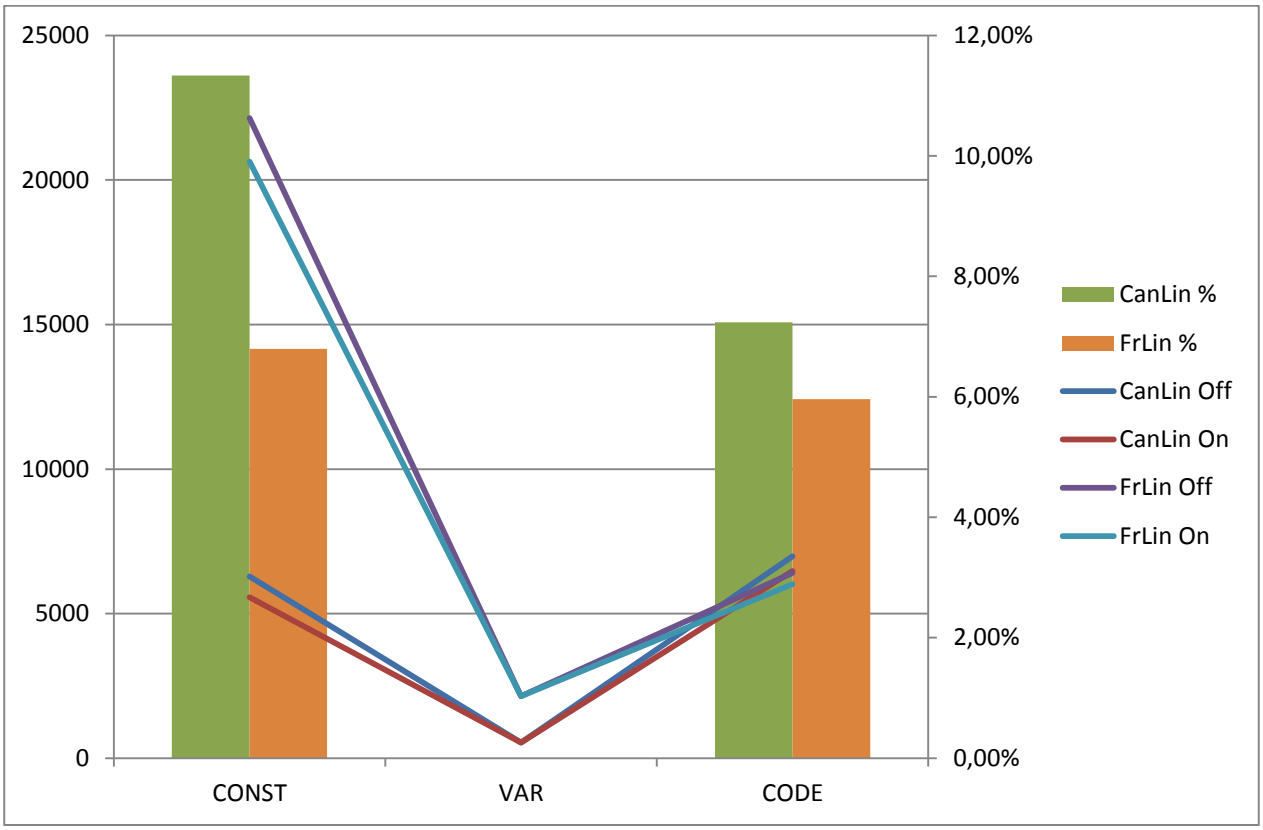
3.1 CONFIG-CLASS of Data

-
- -
- -

3.2 CONFIG-CLASS PRE-COMPILE Optimizations

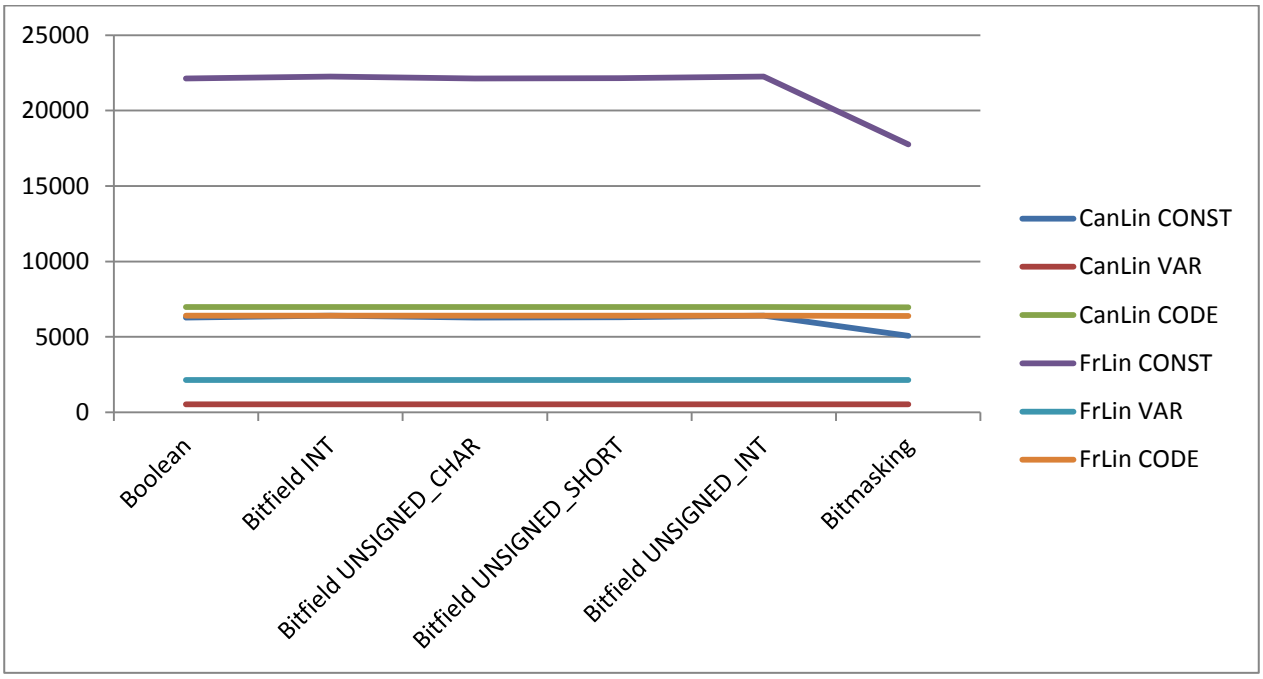
3.2.1 Optimize Const Data to Defines

- -

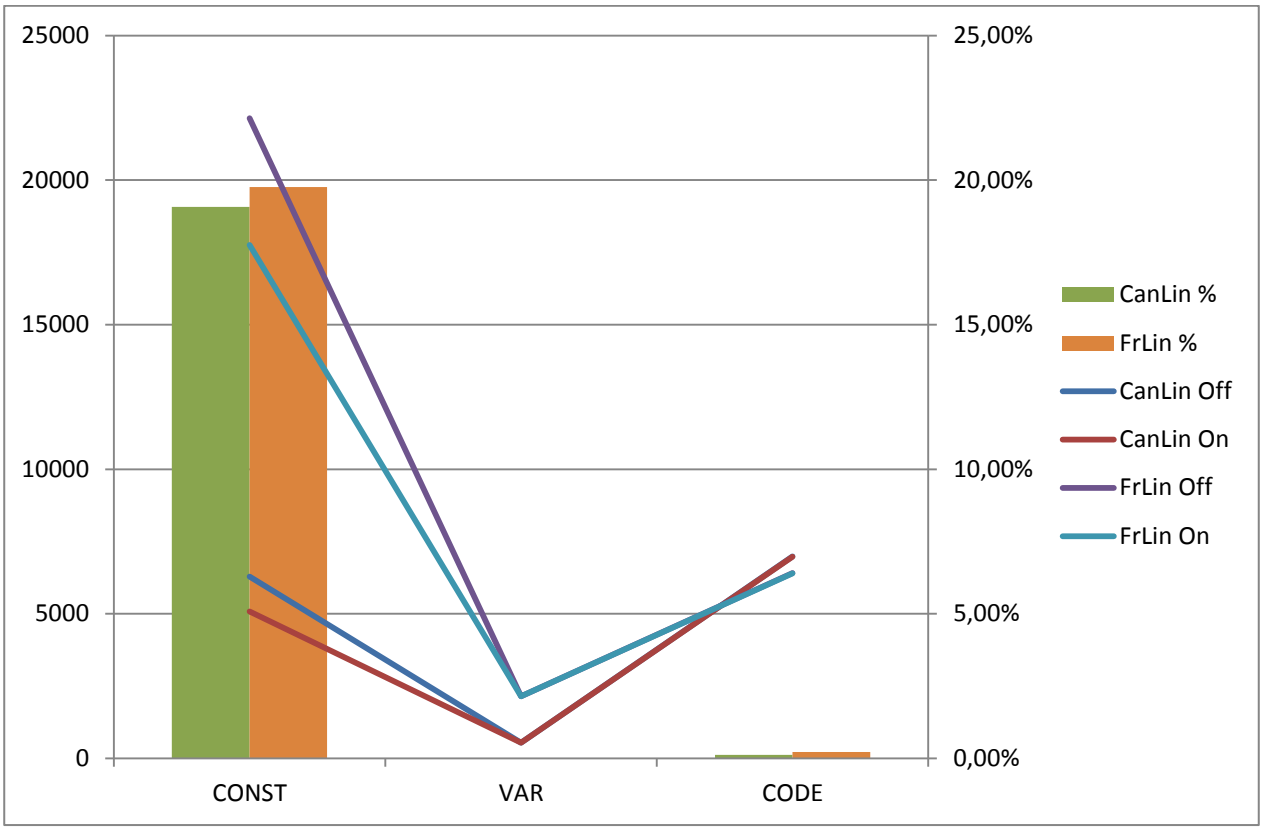


-

3.2.2 Optimize Bool Data in Structs



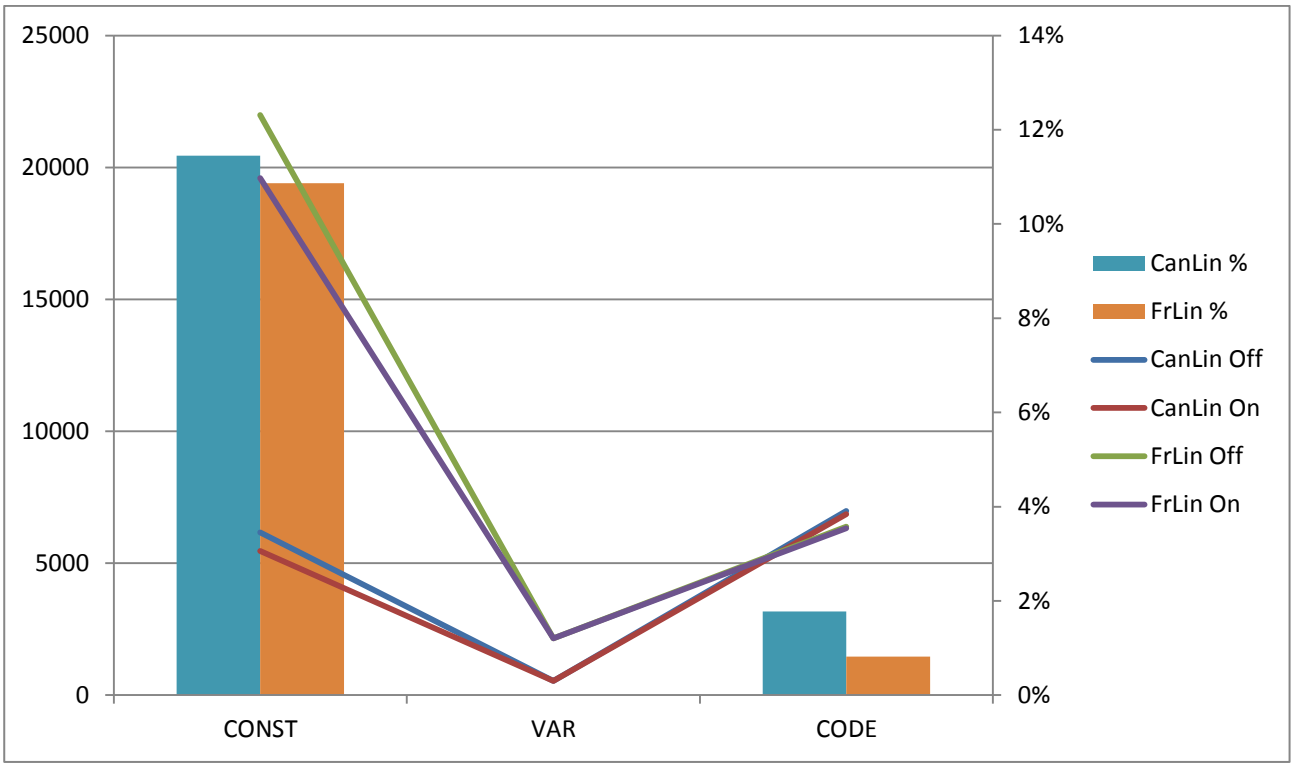
-



-

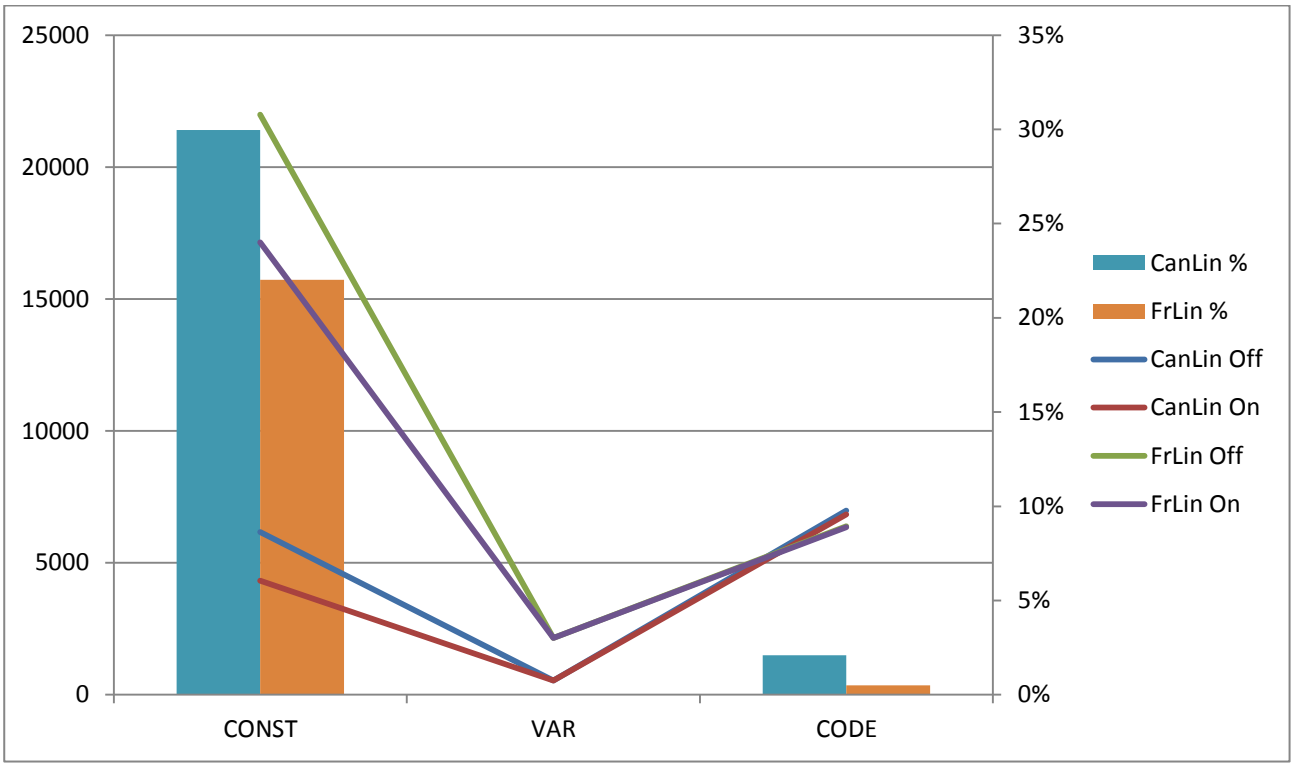
3.2.3 Data Deduplication and Reduction

3.2.3.1 Equal Data



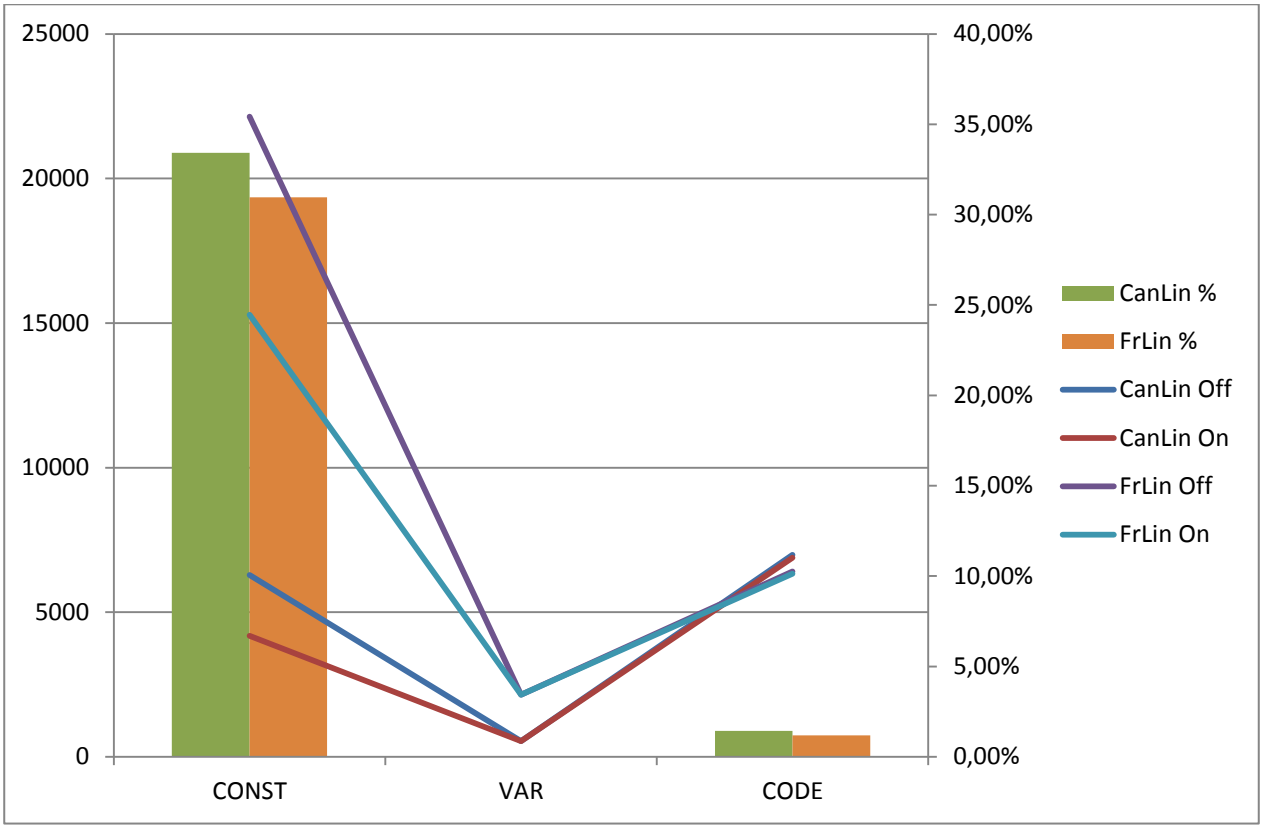
-

3.2.3.2 Unary and Binary Operations



-

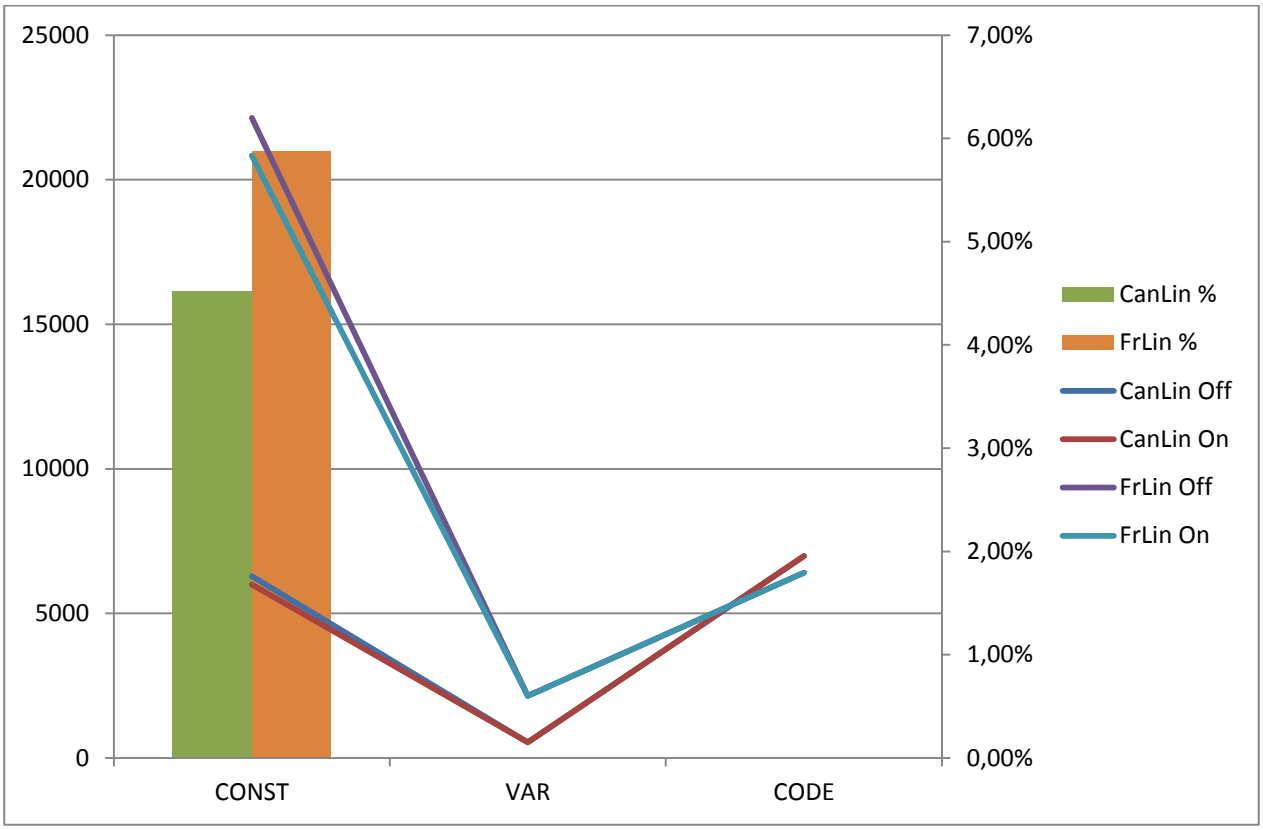
3.2.4 Data Streaming



-

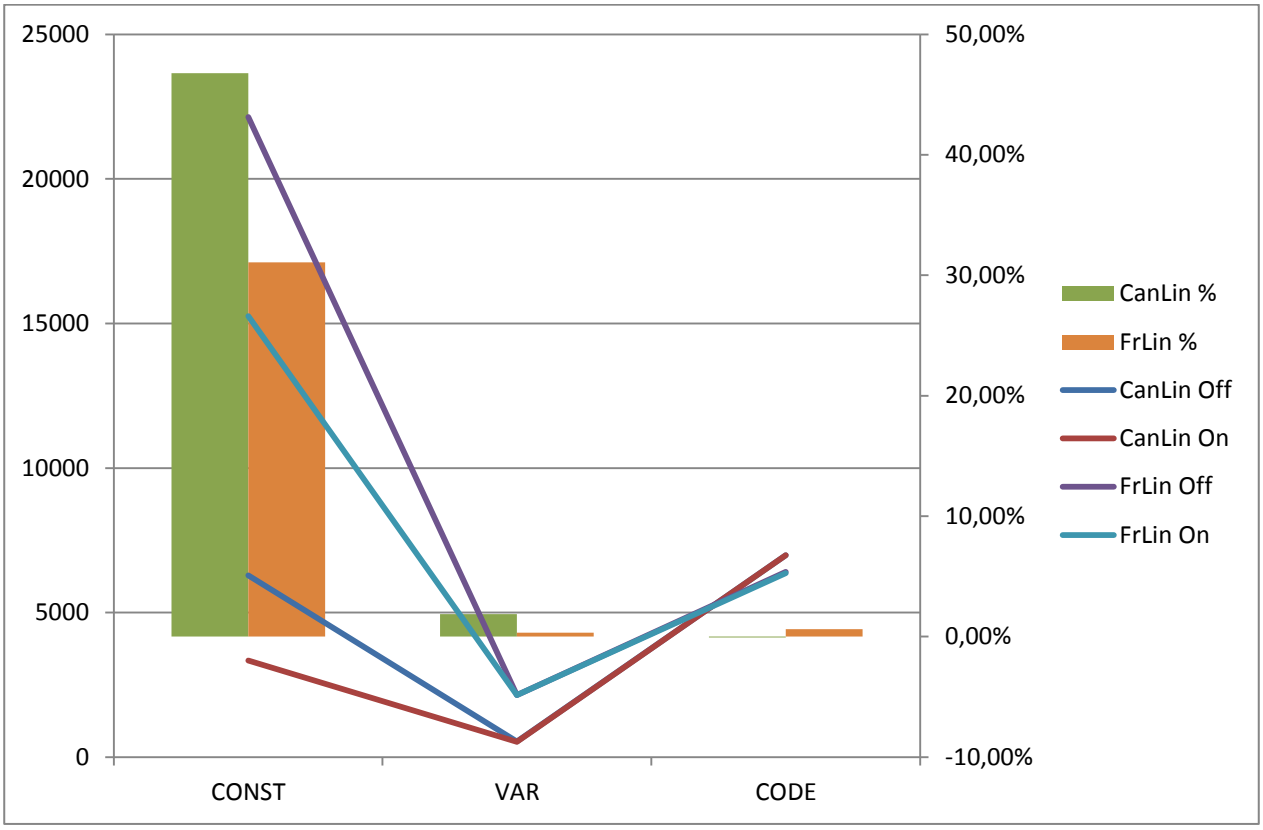
3.3 CONFIG-CLASS Independent Optimizations

3.3.1 Sort Struct Elements



-

3.3.2 Optimize Data Types



-

3.4 SELECTABLE Optimizations

3.4.1 Merge of VAR and CONST Based Data

Example

3.5 Freedom from Interference

> **Index checking:**

> **Index saturation:**

4 Integration

4.1 Dynamic Files

File Name	Description
	> > > > > - -
	> > > > > > - -
	> > > > > -
	> > > > > > - -

File Name	Description
	> > > > > - -
	> > > > > > - -

-

4.2 IMPLEMENTATION-CONFIG-VARIANT dependent Data

- - - - -

Expert Knowledge

-

IMPLEMENTATION-CONFIG-VARIANT	Description
- -	> - - > - -
- -	> - - > - > - -
- - -	> - - > - > - -

- - -

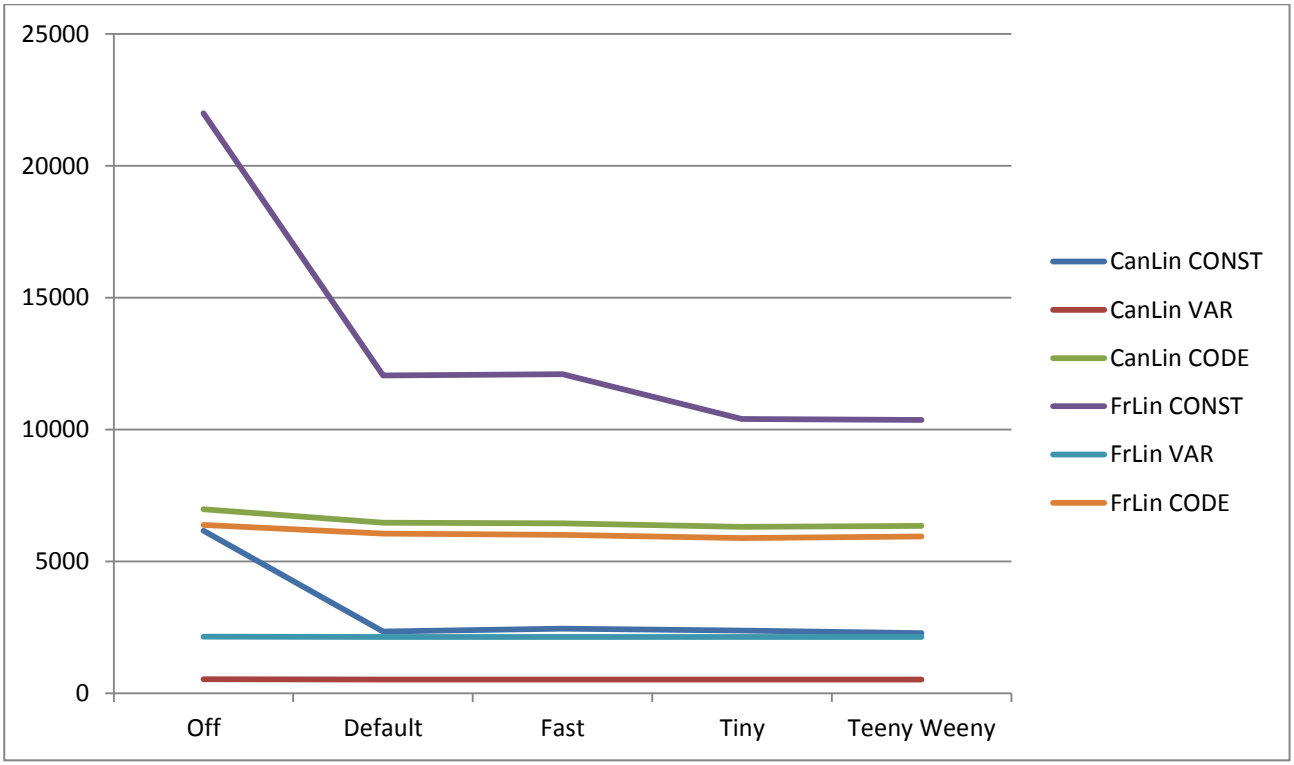
4.3 Optimization Levels

-

-

Optimization	Description
-	

-



-

Optimization Level / Parameter				'
	DEDUPLICATE_CONST_DATA_WITH_CAST	DEDUPLICATE_CONST_DATA_WITH_CAST	DEDUPLICATE_CONST_DATA_WITH_CAST	DEDUPLICATE_CONST_DATA_WITH_CAST
	TRUE	TRUE	TRUE	TRUE
	TRUE	TRUE	TRUE	TRUE
	BITMASKING	BOOLEAN	BOOLEAN	BITMASKING
	TRUE	TRUE	TRUE	TRUE
	2	0	0	2
	2	0	0	2
	2	0	0	2
	2	0	0	2
	2	0	0	2
	FALSE	FALSE	FALSE	FALSE
	FALSE	FALSE	TRUE	TRUE

-

4.4 MISRA, PRQA and Compiler Warnings

4.4.1 General

Note

Deviation ID	MD_CSL_3199

	``
	#define MSN_PROCESS_DATA STD_OFF
	#define MSN_USE_DUMMY_STATEMENT STD_ON
	
	void Msn_foo(uint8 a)
	{
	#if (MSN_PROCESS_DATA == STD_ON)
	  /* some code which uses the parameter a */
	#endif
	#if (MSN_USE_DUMMY_STATEMENT == STD_ON)
	# if (MSN_PROCESS_DATA == STD_OFF)
	  a=a;
	# endif
	#endif
	}
	``

-

Deviation ID	MD_CSL_750_759

	/* symbolic data access for A2L */
	typedef struct sMsn_FooDataStructType
	{
	  boolean indexA;
	  boolean indexB;
	} Msn_FooDataStructType;
	
	/* union data type to have array and symbolic data access */
	typedef union uMsn_FooDataType
	{
	  boolean raw[2]; /**< this element is used for array based data access from the embedded code */
	  Msn_FooDataStructType str; /**< this element is used for symbolic based data access from A2L */
	} Msn_FooDataUType;
	
	/* this variable array uses the union data type */
	Msn_FooDataUType msn_FooData;

-

Deviation ID	MD_CSL_0779

	#if (MSN_DEFRXSIGGRPINFOENDIDXOFDEFRXPDUINFO == STD_ON)
	{
	  Msn_DefRxSigGrpInfoEndIdxOfDefRxPduInfoType idxRxSigGrpInfo =
	  Msn_GetDefRxSigGrpInfoStartIdxOfDefRxPduInfo(idxRxPduInfo);
	  /* some code */
	
	#endif

-

Deviation ID	MD_CSL_2018

```
#define MSN_PROCESS_DATA          STD_ON
#define MSN_CASE_SMALL            5
#define MSN_CASE_MEDIUM           8
#define MSN_CASE_LARGE            12
#define MSN_CASE_SMALL_USED       FALSE
#define MSN_CASE_MEDIUM_USED      TRUE
#define MSN_CASE_LARGE            FALSE

/* this array is reduced to a constant define
const uint8 msn_FooData [2] =
{
  MSN_CASE_MEDIUM,
  MSN_CASE_MEDIUM
};
*/
#define Msn_GetFooData (Index)      MSN_CASE_MEDIUM

void Msn_foo(uint8 a)
{
#if (MSN_PROCESS_DATA == STD_ON)
  switch(Msn_GetFooData(a))
  {
#if (MSN_CASE_SMALL_USED == STD_ON)
    case MSN_CASE_SMALL:
      /* some MSN_CASE_SMALL code */
      break;
#endif
#if (MSN_CASE_MEDIUM_USED == STD_ON)
    case MSN_CASE_MEDIUM:
      /* some MSN_CASE_MEDIUM code */
      break;
#endif
#if (MSN_CASE_LARGE_USED == STD_ON)
    case MSN_CASE_LARGE:
      /* some MSN_CASE_LARGE code */
      break;
#endif
    default:
      /* some default handling like calling Det */
  }
#endif
}
```

-

Deviation ID	MD_CSL_3355_3356_3358_3359_3325

	#define MSN_PROCESS_DATA STD_ON
	
	/* this array is reduced to a define
	const boolean msn_FooData [2] =
	{
	TRUE,
	TRUE
	};
	*/
	#define Msn_IsFooData (Index) TRUE
	
	void Msn_foo(uint8 a)
	{
	#if (MSN_PROCESS_DATA == STD_ON)
	if(Msn_IsFooData(a))
	{
	/* some code */
	}
	#endif
	}

-

Deviation ID	MD_CSL_3453
	-

	#define MSN_PROCESS_DATA STD_ON
	
	/* this array is accessed by a generated data access macro */
	const boolean msn_FooData [2] =
	{
	TRUE,
	TRUE
	};
	
	#define Msn_IsFooData (Index) msn_FooData[Index]
	
	void Msn_foo(uint8 a)
	{
	#if (MSN_PROCESS_DATA == STD_ON)
	if(Msn_IsFooData(a))
	{
	/* some code */
	}
	#endif
	}

-

Deviation ID	MD_CSL_310
	- - -

	#define Msn_GetFoo(Index) 1U
	#define Msn_HasFoo () (Msn_ConfigDataPtr->FooPtrOfPCConfig != NULL_PTR)
	#define Msn_Foo ((Msn_FooPtrType)(&(Msn_PCConfig)))
	
	CONST(Msn_PCConfigsType, MSN_CONST) Msn_PCConfig = {
	{ /* Index: 0 Keys: [Config_LeftFront] */
	Msn_Foo /**< the pointer to Msn_Foo */ /* PRQA S 0310 */ /* MD_CSL_310 */
	, 5U /**< the number of elements in Msn_Foo */
	},
	{ /* Index: 1 Keys: [Config_RightFront] */
	NULL_PTR /**< the pointer to Msn_Foo */
	, 0U /**< the number of elements in Msn_Foo */
	}
	};

-

4.4.2 Bitfields

-

BitFieldDataType Literal	Description
	• •
	• •
	• •
	• •

-

4.4.3 <MSN>_Has Macros in the SELECTABLE Use Case

Example

```
                                                Msn_GetFooData()
                                                 Msn_HasFooData()


#define MSN_USE_INIT_POINTER           STD_ON

#define Msn_HasFooData()               TRUE

void Msn_foo(uint8 a)
{
#if (MSN_USE_INIT_POINTER == STD_ON)
  if(Msn_HasFooData())
#endif
  {
    /* some code and process Msn_GetFooData() */
  }
```

5 Configuration

5.1 Configuration Variants

5.2 Configuration with a GCE

Note

Container Name	
Path	
Multiplicity	
Description	

-

Attribute Name	Value Type	Description

Attribute Name	Value Type	Description
		- -
		- -
		- -

Attribute Name	Value Type	Description
		-
		-

Attribute Name	Value Type	Description
		- -
		- -
		- -
		- -

Attribute Name	Value Type	Description
		- -
		- -
		- -

Attribute Name	Value Type	Description

-

6 Glossary and Abbreviations

6.1 Glossary

Term	Description
-	- -

-

6.2 Abbreviations

Abbreviation	Description
	-

-

7 Contact