

# Compression Module for LZMA

## Technical Reference

SysService\_CmprLzma

Version 1.0

Authors	Thomas Müller
Status	Released

## Document Information

### History

Author	Date	Version	Remarks
Thomas Müller	2016-11-28	1.0	Initial version

### Reference Documents

No.	Source	Title	Version
[1]	SEGGER	emCompress – LZMA compression system	2.20b
[2]	ISO	Road vehicles — Unified diagnostic services (UDS) — Part 1: Specification and requirements	ISO 14229-1:2005(E)



#### Caution

We have configured the programs in accordance with your specifications in the questionnaire. Whereas the programs do support other configurations than the one specified in your questionnaire, Vector's release of the programs delivered to your company is expressly restricted to the configuration you have specified in the questionnaire.

## Contents

<b>1</b>	<b>Introduction.....</b>	<b>5</b>
<b>2</b>	<b>Embedded C Code.....</b>	<b>6</b>
2.1	Files.....	6
2.2	Configuration .....	7
2.3	Decompression API .....	8
2.4	Memory usage .....	10
<b>3</b>	<b>Compression Tools .....</b>	<b>11</b>
3.1	Command Line .....	11
<b>4</b>	<b>Abbreviations.....</b>	<b>12</b>
<b>5</b>	<b>Contact.....</b>	<b>13</b>

## Tables

Table 2-1	Files.....	6
Table 2-2	Configuration options.....	7
Table 2-3	CmprLzmaInit .....	8
Table 2-4	CmprLzmaDeinit.....	8
Table 2-5	CmprLzmaDecompress .....	9
Table 2-6	Memory usage of decompression module .....	10
Table 3-1	Possible options of compression command line tool .....	11
Table 4-1	Abbreviations.....	12

## 1 Introduction

The compression module is mainly intended for

- > the usage in automotive ECUs.
- > the usage in the Flashbootloader.

The compression module consists of:

- > Embedded C code for decompression
- > Command line tool for compression

The compression method is based on the Lempel-Ziv-Markow algorithm compression method.

The decompression module can be configured using a configuration file. Possible parameters are described in chapter 2.2.

## 2 Embedded C Code

The decompression module consists of several files which contain code for a decompression wrapper and a LZMA library. To use the decompression module in a reasonable way the API described in chapter 2.3 should be used. It is not recommended to bypass the Decompression Wrapper and call functions of the LZMA library directly. The compression is available in the compression tools (see chapter 3.1) and not available as source code.

### 2.1 Files

File Name	Description
cmpr_lzma.c	Decompression wrapper source file
cmpr_lzma.h	Decompression wrapper header file
_cmpr_lzma_cfg.h	Template file for configuration of decompression see chapter 2.2
COMPRESS_LZMA_Decode.c	Decompression algorithm source file
COMPRESS_LZMA_DECODE.h	Decompression algorithm header file
COMPRESS_LZMA_DECODE_Conf.h	Configuration wrapper header file
COMPRESS_LZMA_DECODE_ConfDefaults.h	Default configuration for not using _cmpr_lzma_cfg.h.
COMPRESS_LZMA_DECODE_Int.h	Common decompression header file contains type definitions of used structures.
COMPRESS_LZMA_Int.h	Shared header file for compression and decompression source files.

Table 2-1 Files

Some of the compression module files are delivered as templates, which can be adapted to fit specific needs. Those files are marked with a preceding underscore which has to be removed before usage.

## 2.2 Configuration

The configuration options can be found in the template file `_cmpr_lzma_cfg.h`. They offer the adaption of the LZMA decompression algorithm and allow specifying a consecutively called watchdog trigger routine.

Configuration Option	Description
Watchdog	
COMPRESS_LZMA_DECODE_WATCHDOG_TICK(I)	Watchdog routine which will be called by the decompression module. The statement which can be configured by the customer's needs is already pre-configured. The parameter passed to the macro is in general a loop index. <i>Note: The macro is sometimes called with value 0.</i>
Data types	
CMPR_WATCHDOG_CYCLE_COUNT	Specifies how often the watchdog trigger routine is called in loops. The value must be a power of two.
COMPRESS_LZMA_CONFIG_DECODE_LC_MAX	Maximum number of literal context (LC) bits the decoder will accept [0, 8].
COMPRESS_LZMA_CONFIG_DECODE_LP_MAX	Maximum number of literal position (LP) bits the decoder will accept [0, 4].
COMPRESS_LZMA_CONFIG_DECODE_PB_MAX	Maximum number of position bits (PB) bits the decoder will accept [0, 4].
COMPRESS_LZMA_CONFIG_DECODE_WINDOW_SIZE_MAX	Specifies the maximum size of LZMA window.

Table 2-2 Configuration options

## 2.3 Decompression API

Prototype	
<code>tFblResult CmprLzmaInit ( void )</code>	
Parameter	
-	-
Return code	
<code>kFblOk</code>	Decompression module successfully initialized
<code>kFblFailed</code>	Decompression module initialization failed
Functional Description	
Initializes the status and element variables for the decompression.	
Particularities and Limitations	
> Has to be executed before the first call of <code>CmprLzmaDecompress</code>	
Call context	
> Task context	

Table 2-3 CmprLzmaInit

Prototype	
<code>tFblResult CmprLzmaDeinit ( void )</code>	
Parameter	
-	-
Return code	
<code>kFblOk</code>	Decompression module successfully deinitilized
<code>kFblFailed</code>	Decompression module failed deinitialization
Functional Description	
Initializes the status and element variables for the decompression.	
Call context	
> Task context	

Table 2-4 CmprLzmaDeinit



Prototype				
tFblResult CmprLzmaDecompress( V_MEMRAM1 tProcParam V_MEMRAM2 V_MEMRAM3 * procParam )				
Parameter				
tProcParam	Type	Member	Input	Output
	vuint8 *	dataBuffer	Buffer containing compressed data	-
	vuint16	dataLength	Length of compressed data	Number of processed input bytes
	vuint8 *	dataOutBuffer	Buffer for decompression output	Decompressed data
	vuint16	dataOutLength	Length of decompressed data (size of outputBuffer)	Number of decompressed output bytes
Return code				
kFblOk		kFblOk Decompression of buffer successful		
kFblFailed		kFblFailed Failure in buffer decompression		
Functional Description				
Decompress data given in input buffer. Decompressed data is stored in provided output buffer.				
Processing will abort as soon as output buffer is filled. Decompression can continue at this point, in case a new output buffer is provided. The caller has to ensure that the newly given input buffer points to the first previously unprocessed byte.				
Particularities and Limitations				
<div>&gt; CmprLzmaInit has to be executed before first call of function</div> <div>&gt; Members of tProcParam have to be fully initialized</div>				
Call context				
<div>&gt; Task context</div>				

Table 2-5 CmprLzmaDecompress

## 2.4 Memory usage

The following table shows the estimated usage of RAM on decompression. Each size is measured in kilobytes with a zero window size, for a typical 32-bit byte-addressed machine.

LP	PB	LC=0	LC=1	LC=2	LC=3	LC=4	LC=5	LC=6	LC=7	LC=8
0	0	3.8	5.3	8.3	14.3	26.3	50.3	98.3	194.3	386.3
0	1	4.0	5.5	8.5	14.5	26.5	50.5	98.5	194.5	386.5
0	2	4.3	5.8	8.8	14.8	26.8	50.8	98.8	194.8	386.8
0	3	4.9	6.4	9.4	15.4	27.4	51.4	99.4	195.4	387.4
0	4	6.2	7.7	10.7	16.7	28.7	52.7	100.7	196.7	388.7
1	0	5.3	8.3	14.3	26.3	50.3	98.3	194.3	386.3	770.3
1	1	5.5	8.5	14.5	26.5	50.5	98.5	194.5	386.5	770.5
1	2	5.8	8.8	14.8	26.8	50.8	98.8	194.8	386.8	770.8
1	3	6.4	9.4	15.4	27.4	51.4	99.4	195.4	387.4	771.4
1	4	7.7	10.7	16.7	28.7	52.7	100.7	196.7	388.7	772.7
2	0	8.3	14.3	26.3	50.3	98.3	194.3	386.3	770.3	1538.3
2	1	8.5	14.5	26.5	50.5	98.5	194.5	386.5	770.5	1538.5
2	2	8.8	14.8	26.8	50.8	98.8	194.8	386.8	770.8	1538.8
2	3	9.4	15.4	27.4	51.4	99.4	195.4	387.4	771.4	1539.4
2	4	10.7	16.7	28.7	52.7	100.7	196.7	388.7	772.7	1540.7
3	0	14.3	26.3	50.3	98.3	194.3	386.3	770.3	1538.3	3074.3
3	1	14.5	26.5	50.5	98.5	194.5	386.5	770.5	1538.5	3074.5
3	2	14.8	26.8	50.8	98.8	194.8	386.8	770.8	1538.8	3074.8
3	3	15.4	27.4	51.4	99.4	195.4	387.4	771.4	1539.4	3075.4
3	4	16.7	28.7	52.7	100.7	196.7	388.7	772.7	1540.7	3076.7
4	0	26.3	50.3	98.3	194.3	386.3	770.3	1538.3	3074.3	6146.3
4	1	26.5	50.5	98.5	194.5	386.5	770.5	1538.5	3074.5	6146.5
4	2	26.8	50.8	98.8	194.8	386.8	770.8	1538.8	3074.8	6146.8
4	3	27.4	51.4	99.4	195.4	387.4	771.4	1539.4	3075.4	6147.4
4	4	28.7	52.7	100.7	196.7	388.7	772.7	1540.7	3076.7	6148.7

Table 2-6 Memory usage of decompression module

## 3 Compression Tools

To generate the compressed data for download a compression command line tool is provided.

### 3.1 Command Line

The command line tool requires a bin-file as input. It does not support the compression of the data inside an Intel Hex or a Motorola S-Record file. To compress files the tool `COMPRESS_LZMA_Util.exe` has been called in command window with arguments described below.

The command line tool is used as follows:

```
COMPRESS_LZMA_Util.exe <options> <infile> <outfile>
```

Option	Description
-O $number$	Set the optimization level. Acceptable values for $number$ are 0 (faster, good compression) to 9 (slower, best compression).
-lc= $number$	Set the number of literal context bits. Acceptable values for $number$ are 0 to 8 inclusive. Default $number$ is 0.
-lp= $number$	Set the number of literal position bits. Acceptable values for $number$ are 0 through 8 inclusive. Default $number$ is 0.
-pb= $number$	Set the number of position bits. Acceptable values for $number$ are 0 through 4 inclusive. Default $number$ is 0.
-ws= $number$	Set the window size used for compression. The window size is the naming convention used by emCompress and emCompress-LZMA to describe the maximum match distance and, therefore, the maximum number of octets that must be stored to satisfy references made by the decompressor. The LZMA SDK refers to this as the “dictionary size” and both terms should be considered equivalent. Increasing the window size will usually increase compression ratios and reduce the size of the compressed bitstream at the expense of requiring extra RAM during decompression. Default $number$ is 131072.

Table 3-1 Possible options of compression command line tool



#### Example

```
COMPRESS_LZMA_Util.exe -O9 -lc=2 -lp=3 -pb=2 -ws=4096  
test.bin test.out
```

“test.bin” is compressed and the decompressed data is written to the “test.out” file.

## 4 Abbreviations

Abbreviation	Description
API	Application Programming Interface

Table 4-1 Abbreviations

## 5 Contact

Visit our website for more information on

- > News
- > Products
- > Demo software
- > Support
- > Training data
- > Addresses

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