



pC/LZFG Reference

V1.10a

Haftungsausschluß

Der Autor übernimmt keinerlei Haftung für durch diesen Code entstandene oder entstehende Schäden an Hard- und Software. Er versichert lediglich, daß er den Code vielfältigen Test´s auf unterschiedlicher Hardware unterzogen hat, um seinerseits keine Fehler bestehen zu wissen. Sollten dennoch Fehler auftauchen oder Vorschläge zur Verbesserung des Codes an den Autor weitergegeben werden, so ist dieser bestrebt, Fehler schnellstmöglich auszumerzen oder Vorschläge einzuarbeiten.

liability exclusion:

The author takes over no liability for through this code originated or emerging damages to hardware and software. He assures merely that he subjected the code of diverse tests on different hardware, about for his part no mistakes to know exists. Mistakes nevertheless should appear or suggestions are passed on at the author to the improvement of the code, so this is striving, mistakes fastest to wipe out or to incorporate suggestions.

Most decompressors require RAM again amply, up to 256kB, reached not even 20 percent the data over the Huffman-table to decompress or the pack council about for example.

The LZFG-decompressor module requires exactly 4kB RAM once, that become dynamically allocate in term. Furthermore the code is very compact and the packer reaches amazing 40 ..55 percent with the pile of an application. As transfer parameters, the here introduced decompressor requires the address of the packed record, the length of this record as well as two pointers for address decompressed record, becomes dynamically allocated internally, and length of the same. It is tested within the moduls whether it is about a compatible LZFG-datastream and whether the length and the CRC16 of the decompressed data are identically the values deposited in the LZFG-datastream.

User-Functions:

LZFG-Decompressor:	
LZFG_Decomp	decompress the datafield

Error-Codes:

Name	Decimal_Value	Description
LZFG_SUCCESS	0	decompress successfully
LZFG_noLZFG	130	Data are not in the necessary format
LZFG_lenERR	131	length error
LZFG_crcERR	132	CRC error
LZFG_memERR	133	error in allocation of the window or destination memory

pC/LZFG - Decompressor

LZFG_Decomp

```
U08 LZFG_Decomp(U08 OS_HUGE *source, U32 source_l, U08 OS_HUGE **dest_pp, U32 *dest_l)
```

Decompress the handed over record examination of the expansions CRC-16 and length of the original after the LZFG-Mode inclusive.

Parameters

*source	pointer to compressed data
source_l	length of compressed data
**dest_pp	pointer to get the position of decompressed data
*dest_l	pointer to get the length of decompressed data

Return Value

LZFG_SUCCESS	decompress successfully
LZFG_noLZFG	Data are not in the necessary format
LZFG_lenERR	Longitudinal mistakes
LZFG_crcERR	CRC error
LZFG_memERR	Mistakes in allocation of the window or destination memory

Example

```
void OS_FAR Task2(void *data)
{
    U08 OS_HUGE *source_p;
    U08 OS_HUGE *decomp_p;

    U08 state;
    U32 rx_lenght;
    U32 decomp_lenght;

    .
    .
    while(1)
    {
        .
        state=OS_MemAlloc(&source_p, 46000);
        if(state == OS_NO_ERR)
        {
            rx_lenght=UART1_receive(source_p, .....
            .
            state=LZFG_Decomp(source_p, rx_lenght, &decomp_p, &decomp_lenght);
            if(state == LZFG_SUCCESS)
            {
                .
            }
        }
    }
}
```

```
        state=FLSH_Update(decomp_p, decomp_lenght ..
        .
        .
        state=OS_MemFree((OS_MEM OS_HUGE *) decomp_p);
        .
    }
    state=OS_MemFree((OS_MEM OS_HUGE *) source_p);
}
}
```

Comments

Comments
