



Eeprom maintenance

Eeprom maintenance

EEPROM driver purpose

- To make it possible to store parameters non-volatile in a mobile station
- To save the cost of an external EEPROM device

Eeprom maintenance

Basic architecture

EEPROM is logically and divided in 3 different areas:

- EEP static
- EEP dynamic
- EEP default

due to to different needs and implementation

Eeprom maintenance

Default Static Part

Parameters that are set in the production, and they are the same for all MS's.

These parameters will never be changed.

Static Part

Parameters that are set in the production, and they are not the same for all MS's.

E.g.:

- RF calibration parms.
- Audio parms.

These parameters can be adjusted by the EEPROM editor.

Dynamic Part

These parameters can be changed during runtime in normal operation mode.

E.g.:

If the user changes a menu-setting.

These parameters can also be adjusted by the EEPROM editor.

Eeprom maintenance

Files involved

Eep.h : types and
structures definition

Eep.c : types and
structures initialization

Eeprom maintenance

Structure members alignment

- All parameters should be encapsulated in sub-types of minimum one sub-type level. Sub-types can be nested without restrictions.
- To ensure that the target compiler and the PC compiler will be aligning the structures identical, it's necessary to force correct alignment of the different sub-structures.
- On E-GoldLite all subtypes should be made so the next subtype starts on 16 bit aligned address.

Eeprom maintenance

Structure example

```
typedef struct {  
    Parm1_type parm1_name;  
    ...  
    ParmN_type parmN_name;  
} sub_type_1_name;
```

```
typedef struct {  
    sub_type_1_name sub_parm1;  
    ...  
    sub_type_N_name sub_parmN;  
} eep_static_type or eep_dynamic_type;
```

Eeprom maintenance

Modifiing eep_static_type

- “eep_static_type” has to be backward compatible at any time to ensure that calibrated data is not destroyed when downloading a new SW version.

Eeprom maintenance

Modifiing eep_static_type structure

- Add the new sub-types in the end of eep_static_type in eep.h
- Increment EEP_VERSION and set EEP_REVISION to 0 in eep.h
- Add initialization values for the new parameters in EEP_static in eep.c
- Generate a new CFG file ("projectname_eepxxxxx.cfg") using the "Make CFG" tool. (xxxxxx: EEP version number)
- Make a new DEP file ("projectname_eepxxxxx.yyy.dep") and update the EEP version and revision parameter values in the DEP file. (xxxxxx: EEP version number, yyy: EEP revision number)
- Add the new parameters with default values to the DEP file (only non-calibration parameters shall be added).

Eeprom maintenance

Modifiing eep_static_type values

- Increment EEP_REVISION in eep.h
- Change initialization values for the parameters in EEP_static in eep.c
- Make a new DEP file ("projectname_eepxxxxx.yyy.dep") and update the EEP version and revision parameter values in the DEP file. (xxxxx: EEP version number, yyy: EEP revision number).
- Add the new parameters with default values to the DEP file (only non-calibration parameters shall be added)

Eeprom maintenance

Modifiing eep_dynamic_type

- “eep_dynamic_type” **doesn’t** have to be backward compatible, so existing sub-types can be changed or removed, and new sub-types can be inserted at any location without restrictions.

Eeprom maintenance

Modifiing eep_dynamic_type structure

- Make the structure-changes in the new sub-types of eep_dynamic_type in eep.h
- Increment EEP_VERSION and set EEP_REVISION to 0 in eep.h
- Add/remove/change initialization values for the parameter-changes in EEP_dynamic in eep.c
- Add/remove/change initialization values for the parameter-changes in EEP_dynamic_default_filter in eep.c. Values different from 0 will cause parameters to be affected by the EE_set_factory_default() function
- Generate segmentation data:
 - Generate a new factory segmentation table using the "Segmentation tool"
 - Copy the new segmentation data to "EEP_dynamic_factory[]" in eep.c
 - Update "EEP_DYNAMIC_FACTORY_SIZE" in eep.h
- Generate a new CFG file ("projectname_eepxxxxx.cfg") using the "Make CFG" tool. (xxxxx: EEP version number).

Eeprom maintenance

Modifiing eep_dynamic_type values

- Change initialization values for the parameters in EEP_dynamic in eep.c
- Generate segmentation data:
 - ☐ Generate a new factory segmentation table using the “Segmentation tool”
 - ☐ Copy the new seg data to “EEP_dynamic_factory[]” in eep.c
 - ☐ Update “EEP_DYNAMIC_FACTORY_SIZE” in eep.h.

Eeprom maintenance

Modifiing eep_default

- The EEP_default structure is only used as an ordinary const structure and is used as a common place for storing fixed parameters for the different HW-drivers
- The sub-types can be changed/removed/added without restrictions. Default values can be changed without restrictions
- EEP_default parameters are not included in the CFG and DEP files
- EEP version and revision numbers are not affected by changes in EEP_default.

Eeprom maintenance

