SiLabs API Wrapper Software Release Note

Silicon Laboratories, Inc.

Broadcast Video Products

V2.8.0

June 12, 2018

Version 1.0

1 Table of Contents

2 Overview

This document describes the features and errata of the current software release for the SiLabs API wrapper.
The SiLabs API wrapper is used on top of SiLabs Video demodulator code to allow a single application code to manage any demodulator from SiLabs.

For implementation details, refer to the documents provide together with the source code, in the ‘DOC’ folder.

3 Related SiLabs Demodulators

Part Number	DTV standards										
	DVB-T	DVB-T2	DVB-T2 Lite	ISDB-T	DVB-C	MCNS	DVB-C2	DVB-S	DVB-S2	DVB-S2X	DSS
Si2111	●										
Si2113					●						
Si2115	●				●						
Si2160A	●				●	●	●	●	●		●
Si2160B	●				●	●	●	●	●	●	●
Si21602	●				●	●	●	●	●		●
Si21602B	●				●	●	●	●	●	●	●
Si2161D	●										
Si2162	●	●	●		●	●	●				
Si2162B	●	●	●		●	●	●				
Si21622	●	●	●		●	●	●				
Si21622B	●	●	●		●	●	●				
Si2163D					●						
Si2163E					●	●					
Si2163F					●	●	●				
Si2164	●	●			●	●	●	●	●		●
Si2164B	●	●			●	●	●	●	●	●	●
Si21642	●	●			●	●	●	●	●		●
Si21642B	●	●			●	●	●	●	●	●	●
Si21647	●	●			●	●	●	●	●	●	●
Si2165D	●				●						
Si21652B	●				●						
Si2166								●	●		●
Si2166B								●	●		●
Si2166C								●	●		●
Si21662								●	●		●
Si21662B								●	●		●
Si2166B								●	●		●
Si2167	●							●	●		●
Si2167B	●				●			●	●		●
Si21672B	●				●			●	●		●
Si2168	●	●			●						
Si2168B	●	●			●	●					
Si2168C	●	●			●	●					

Part Number	DTV standards										
	DVB-T	DVB-T2	DVB-T2 Lite	ISDB-T	DVB-C	MCNS	DVB-C2	DVB-S	DVB-S2	DVB-S2X	DSS
Si21682B	●	●			●	●					
Si21682C	●	●			●	●					
Si2169	●	●			●	●		●	●		●
Si2169B	●	●			●	●		●	●		●
Si2169C											
Si21692B	●	●			●			●	●		●
Si21692C	●	●			●			●	●		●
Si2180	●			●	●	●					
Si21802	●			●	●	●					
Si21804	●			●	●	●					
Si2181	●			●	●	●		●	●	●	●
Si21812	●			●	●	●		●	●	●	●
Si21817	●			●	●	●		●	●	●	●
Si2182	●	●	●	●	●	●		●	●	●	●
Si21822	●	●	●	●	●	●		●	●	●	●
Si2183	●	●	●	●	●	●	●	●	●	●	●
Si21832	●	●	●	●	●	●	●	●	●	●	●
Si2185	●				●						

4 Related SiLabs tuners

Part	DTV	ATV	Details
Si2124	●		Via SiLabs_TER_Tuner API
Si2141	●		Via SiLabs_TER_Tuner API
Si2144	●		Via SiLabs_TER_Tuner API
Si2146	●		Via SiLabs_TER_Tuner API
Si2147	●		Via SiLabs_TER_Tuner API
Si2148	●		Via SiLabs_TER_Tuner API
Si2148B	●		Via SiLabs_TER_Tuner API
Si2151	●		Via SiLabs_TER_Tuner API
Si2156	●		Via SiLabs_TER_Tuner API
Si2157	●		Via SiLabs_TER_Tuner API
Si2158	●		Via SiLabs_TER_Tuner API
Si2158B	●		Via SiLabs_TER_Tuner API
Si2173	●		Via SiLabs_TER_Tuner API
Si2176	●		Via SiLabs_TER_Tuner API

Si2177	●		*Via SiLabs_TER_Tuner API*
Si2178	●		*Via SiLabs_TER_Tuner API*
Si2178B	●		*Via SiLabs_TER_Tuner API*
Si2190	●		*Via SiLabs_TER_Tuner API*
Si2190B	●		*Via SiLabs_TER_Tuner API*
Si2191	●		*Via SiLabs_TER_Tuner API*
Si2191B	●		*Via SiLabs_TER_Tuner API*
Si2196	●		*Via SiLabs_TER_Tuner API*

Related non-SiLabs TER tuners

Part	*Supplier*	
DTT759x	*Thomson*	*Only with 'legacy' chips*
CUSTOMTER	*'Any'*	*Via SiLabs_TER_Tuner API*

5 Related SAT tuners

Part	*Supplier*	
AV2012	*Airoha*	*Via SiLabs_SAT_Tuner API*
AV2018	*Airoha*	*Via SiLabs_SAT_Tuner API*
MAX2112	*Maxxim*	*Via SiLabs_SAT_Tuner API*
RDA16110	*RDA*	*Via SiLabs_SAT_Tuner API*
RDA16110D	*RDA*	*Via SiLabs_SAT_Tuner API*
RDA16116SW	*RDA*	*Via SiLabs_SAT_Tuner API*
RDA5812	*RDA*	*Via SiLabs_SAT_Tuner API*
RDA5815	*RDA*	*Via SiLabs_SAT_Tuner API*
RDA5815S	*RDA*	*Via SiLabs_SAT_Tuner API*
RDA5815M	*RDA*	*Via SiLabs_SAT_Tuner API*
RDA5816	*RDA*	*Via SiLabs_SAT_Tuner API*
RDA5816S	*RDA*	*Via SiLabs_SAT_Tuner API*
RDA5816SD	*RDA*	*Via SiLabs_SAT_Tuner API*
NXP20142	*NXP*	*Via SiLabs_SAT_Tuner API*
CUSTOMSAT	*'Any'*	*Via SiLabs_SAT_Tuner API*

6 Related LNB controllers

Part	*Supplier*
LNBH21	*ST Microelectronics*
LNBH25	*ST Microelectronics*
LNBH26	*ST Microelectronics*
LNBH29	*ST Microelectronics*
A8203	*Allegro (dual)*
A8292	*Allegro*

SiLabs API Wrapper V2.8.0 Software

6.1 *Description*

This software release is intended for all products listed in the above paragraphs.
The features and errata of the V2.8.0 software are indicated in the sections below.

6.2 *Features*

- DTV front-end – satellite/terrestrial/cable
  - DVB-T2/C2/T/C/C/MCNS
  - ISDB-T
  - DVB-S2/S2X/S/DSS

- ATV front_end – terrestrial/cable
  - PAL B/G, D/K, I
  - SECAM L, L'
  - NTSC M

6.2.1 As from V2.8.0 (2018/06/12)

<improvement>[CNR]
Correction of CNR reported by demodulator in DVB-T and ISDB-T to better match gaussian CNR level of test equipment.
<improvement>[SAT/Unicable/Swap]
Adding swap_detection_done flag in SILABS_FE_Context (only for SAT and Unicable).

6.2.2 As from V2.7.9 (2018/01/04)

<new_feature>[TER/FW_load]
Adding SiLabs_API_Store_TER_TUNER_FW, to allow loading the TER tuner FW from a file
<compatibility>[SAT/A8297]
Defining A8297_COMPATIBLE items in c file (now defined as extern in header file)
<compatibility>[compiler/warnings]
In SiLabs_API_SAT_PLS_Init: moving trace line after all declarations.
<compatibility>[SSI/SQI]
In SiLabs_API_Demod_status_selection: always calling Si2183_L1_SSI_SQI. The drawback is that it will raise an API error with legacy chips. This avoids having a difficult to read/support code in the future, when no legacy chips will be used.
<improvement>[SAT/comments]
In SiLabs_API_SAT_read_diseqc_reply comments: removed one unwanted line.
<cleanup>[SAT/Unicable]
Removing unused 'Unicable install'
<improvement>[TERACOM/BER]
In SiLabs_API_Demod_status_selection: Removing dynamic BER settings. This is not required anymore to pass Nordig(i.e. Teracom)/DBook tests due to test specification changes
<new_feature>[TER/DTVTune]
Adding SiLabs_API_TER_Tuner_DTV_Tune (For test purposes). In DTV mode, tuning should be done via the L2, not directly from L3. This will be used FOR TESTING in Japan on cable networks used for ISDB-T reception.
<new_feature>[TER/Tuner]
Adding SiLabs_API_TER_Tuner_SetProperty and SiLabs_API_TER_Tuner_GetProperty, for test purpose
<new_feature>[SAT/Unicable]

Adding SiLabs_API_SAT_Random_Delay_Init and SiLabs_API_SAT_Random_Delay_Shift to compute the delays required by DiSEqC 1.1 in the DiSEqC collision detection algorithm

6.2.3 As from V2.7.8 (2017/03/06)

<correction>[return_value]
In SiLabs_API_SAT_Unicable_Swap_Detect:
always returning front_end->Si2183_FE->unicable_spectrum_inversion.
<correction>[SAT/UnicableII]
In SiLabs_API_SAT_Unicable_Config: setting front_end->Si2183_FE->unicable_mode (new member of Si2183_L2_Context) to allow dynamic selection of the SAT scan bandwidth.<compatibility>[Legacy/ROMID/0]
<correction>[Si2165D/C_N] In SiLabs_API_Demod_status_selection: avoid rounding to 100 in c_n_100 (issue for Si2165D only).
<compatibility>[Legacy/ROMID/0] In SiLabs_API_Demod_status_selection:
only calling Si2183_L1_DD_SSI_SQI for ROM IDs > 0, to avoid an error message in the traces (no impact on the final result).
<compatibility>[compiler/warnings]
In SiLabs_API_Set_Index_and_Tag: tag defined as const char*.
<compatibility>[compiler/warnings]
In SiLabs_API_TS_Mode: ts_mode defined as unsigned int.
<compatibility>[compiler/warnings]
In SiLabs_API_SAT_Unicable_Swap_Detect: lock_end_ms defined as signed int.
<compatibility>[compiler/warnings]
In SiLabs_API_SAT_Gold_Sequence_Init: k defined as signed int.
<compatibility>[compiler/warnings]
In Silabs_API_Test: target, cmd and sub_cmd defined as const char*.
<compatibility>[compiler/shadowing]
In Silabs_API_Test: removing FE_Status and custom_status, now defined at a higher level.

6.2.4 As from V2.7.7 (2017/01/16)

<improvement>[status/uncors]
In SiLabs_API_lock_to_carrier: resetting the uncorrs count if lock succeeds.
<improvements>[standby/tuners]
Adding SiLabs_API_SAT_Tuner_Standby and SiLabs_API_TER_Tuner_Standby (only required for DUALS/TRIPLE/QUAD when all frontends are going to standby)
<improvement>[SAT/Band_polar]
In SiLabs_API_lock_to_carrier: storing polarization and band in L3 context for proper display even in Unicable mode.
<new_feature>[SSI/RSSI_offset]
Adding TER_RSSI_offset and SAT_RSSI_offset to context, to allow taking into account possible offset on the RF paths (in 1 dB steps)
In SiLabs_API_Demod_status_selection: Adding TER_RSSI_offset/SAT_RSSI_offset to tuner RSSI
In Silabs_API_Test: Adding options 'ter_rssi_offset' and 'sat_rssi_offset' to control the RSSI offsets
<new_feature>[TER/Active_Loop_Through]
Adding SiLabs_API_TER_Tuner_Loop_Through to allow controlling the active loop through state with TER tuners supporting this feature.
<new_feature>[SAT/Unicable]
Adding SiLabs_API_SAT_Unicable_Position to allow selecting 'position A/B' using the L3 API (only for Unicable mode).
<new_feature>[SAT/Unicable[
Adding SiLabs_API_SAT_Unicable_Swap_Detect to detect the value which should be used as unicable_spectrum_inversion in the frontend settings.
NB: It can also be used to detect the value of spectrum_inversion (for non-Unicable mode), but this is normally easy to detect via a normal lock on a DVB-S signal.
console:
<new_feature>[demo/option]
Adding 'status' option (similar to pressing <return> but can be used in a call to Silabs_demoloop("status");)

<new_feature>[SAT/Unicable]
Adding Silabs_UserInput_Unicable_Config to allow setting all Unicable options from the console

6.2.5 As from V2.7.6 (2016/10/05)

<improvement>[EVB/macro]
In DTV_DUAL_TER_SAT_51A: changing clock settings to use the SAT clock.
In TER_SAT_EVB_BR_DUAL: replacing CUSTOMTER_CODE with 0
<new_feature>[SAT/RDA16110SW]
Adding SiLabs_API_SAT_Tuner_SelectRF to allow controlling the RF switch in RDA5816SD/RDA16110SW

6.2.6 As from V2.7.5 (2016/09/14)

<improvement>[Standards/definitions]
Adding SILABS_SLEEP(100) and SILABS_OFF(200) cases in Silabs_standardCode and Silabs_Standard_Text
<new_feature>[BER/monitoring]
Duplicating status->ber (float) and status->ber_mant/status->ber_exp information in status->ber_count/status->ber_window. This closely matches some middleware expectations.
<new_feature>[ISDB-T/monitoring]
Adding SiLabs_API_TER_ISDBT_Layer_Info
Adding layer-specific ISDB-T status fields
In SiLabs_API_TER_ISDBT_Monitoring_mode, adding a 'loop mode' option (0xABC) to update the status information for all layers.
In SiLabs_API_Demod_status_selection, updating ISDB-T Layers status in 'loop mode', when they are in use.
NB: When using all 3 layers, at least 3 calls to SiLabs_API_Demod_status_selection are required to have all status fields updated.
In SiLabs_API_Text_status_selection: printing all 3 ISDB-T layer status
<compatibility>[ISDB-T/status]
In Custom_coderateCode: simplifying the code to deal with ISDB-T code rates.
<compatibility>[traces/on/off]
In SiLabs_API_FE_status_selection: returning 'res' with value matching the lock state
<compatibility>[No_TER] Only declaring i when needed, to avoid compilation errors when not compiling for TER
<correction>[QUAD/pointer] In SiLabs_Channel_Bonding: pointer correction for 'unused' part.

6.2.7 As from V2.7.4 (2016/08/23)

<new_part>[LNB/TPS65233] Adding compatibility with Texas Instruments TPS65233 SAT LNB controller
<improvement>[SAT/Unicable] Redefining SiLabs_API_SAT_Unicable_Config to add unicable_spectrum_inversion

6.2.8 As from V2.7.3 (2016/06/30)

<new_feature>[DVB-S2/roll_off] Adding roll_off in CUSTOM_Status_Struct
<new_feature>[DVB-S2/roll_off]
In SiLabs_API_Demod_status_selection: updating roll_off in DVB-DS2
<new_feature>[SAT/Unicable]
Adding SiLabs_API_SAT_Unicable_Config to configure Unicable from L3.
In SiLabs_API_SAT_Unicable_Install: adding a trace indicating that using SiLabs_API_SAT_Unicable_Config is preferred to enable Unicable.
In SiLabs_API_SAT_Unicable_Uninstall: not changing front_end->unicable->installed anymore
The new behavior is:
SiLabs_API_SAT_Unicable_Config allows selecting the unicable mode (unused/1/2) in the Unicable context
SiLabs_API_SAT_Unicable_Install and SiLabs_API_SAT_Unicable_Uninstall control front_end->lnb_type to select Unicable or normal tuning at L3 level.
<improvement>[Compatibility/No_Traces]
In SiLabs_API_FE_status_selection: moving declaration of 'res' to allow compiling without SiTRACES
<improvement>[Traces]

In SiLabs_API_FE_status_selection: Calling SiLabs_API_Text_status_selection with the same status_selection as the one used to refresh the statuses
In SiLabs_API_Text_status_selection: Tracing only items selected by status_selection
In SiLabs_API_Set_Index_and_Tag: Also setting the tag for LNB controllers
<improvement>[SAT/LNB]
In SiLabs_API_lock_to_carrier: in SAT, enabling/disabling i2c access to the LNB chips before calling SiLabs_API_SAT_voltage
This is useful in case the LNB controllers are not on the main I2c bus.
<improvement>[comments]
In SiLabs_API_Channel_Seek_Init comments: correcting freq min/max text for local blindscan SiLabs_API_Channel_Seek_Init (front_end, freq-4850000, freq+3150000,8000000, 8000000, 3500000, 7500000, 0, 0, 0, 0);'
<improvement>[DVB-S2/status]
In SiLabs_API_Get_Stream_Info: translating isi_constellation and isi_code_rate values to match L3 definitions for constellation and code_rate.

6.2.9 As from V2.7.2 (2016/04/18)

<correction>[S2X/flag] In SiLabs_API_Demod_status_selection: setting status->s2x correctly. The flag was inverted in most cases, and 32APSK cases were missing.
<new_feature>[DVB-S2/PLS/Init] Adding SiLabs_API_SAT_PLS_Init to allow using other non-standard sequences
<new_feature>[DVB-S2/PLS/ISI] In Silabs_API_Test: adding options to test DVB-S2/ISI and DVB-S2/PLS
<new_part>[LNB/A8304] Adding compatibility with Allegro A8297 SAT LNB controller
<improvement>[LNB/LNBH25/LNBH26] Transient overcurrent detection with LNBH25/LNBH26 can make it difficult to control LNBs.
Since this occurs when changing the voltage, the following counter measures are applied:
In SiLabs_API_SAT_voltage_and_tone: swapping calls to SiLabs_API_SAT_tone and SiLabs_API_SAT_voltage.
- If a transient overload happens, the tone will not be sent during the first second, so let's send the tone first.

In SiLabs_API_SAT_voltage: calling the status function for LNB controllers allowing this feature.
- If a transient overload happens, the traces will indicate it, such that the appropriate measures can be taken (the detection threshold is configurable using external components).

<improvement>[LOG_Function] Silabs_Log10_10000 improved to work better between 1 and 2
<new_part>[LNB/A8304] Adding compatibility with Allegro A8297 SAT LNB controller
Since this part is a mixed dual where reading the control byte is not possible, specific pointers to the L3 front-ends are added, to enable matching the context bytes in the A8297 driver.
<cleanup>[Status] Removing unused RFlevel and plp_type values from CUSTOM_Status_Struct
<improvement>[EVB/macro] Adding DTV_DUAL_TER_SAT_A8297 macro
<improvement>[EVB/macro] Adding DTV_SINGLE_TER_SAT_Rev2_0_691 macro
<improvement>[EVB/macro] Adding TER_SAT_EVB_BR_SINGLE macro
<improvement>[EVB/macro] Adding TER_SAT_EVB_BR_DUAL macro

6.2.10 As from V2.7.1 (2016/02/03)

<new_feature>[DVB-T2/C2/MPLP/Group_id] Adding SiLabs_API_Get_PLP_Group_Id, to allow retrieving the group_id in DVB-T2 or DVB-S2 with multiple PLPs.
<improvement>[Status/DVB-T/coderate]
In SiLabs_API_Demod_status_selection: coping status->coderate_hp/status->coderate_lp to status->coderate depending on status->stream, in case the application only uses status->coderate.
In SiLabs_API_Text_status_selection: Using status->coderate also when in DVB-T.
<improvement>[Comments]
In SiLabs_API_Demod_status_selection: better comment when setting initial status->num_plp value.

6.2.11 As from V2.7.0 (2016/01/13)

<new_feature>[Config/driveTS] Adding SiLabs_API_TS_Strength_Shape function to allow configuring the TS drive from the configuration macro.

This is useful when different platforms don't use the same TS drive settings.
<improvement>[Status/DVB-T-T2_only]
In Silabs_constelCode: simplifying constellation switch to always return the expected value.
In Custom_constelCode: simplifying constellation switch to always return the expected value.
The previous version only returned the proper values for DVB-T when compiled with DVB-C compatibility (QAM16 and QAM64) and DVB-S compatibility (QPSK).
<improvement>[Status/ISDB-T]
In Silabs_constelCode: Adding DQPSK case (for ISDB-T).
In Custom_constelCode: Adding DQPSK case (for ISDB-T).
In Silabs_Constel_Text: Adding DQPSK case (for ISDB-T).
In CUSTOM_Constel_Enum: Adding SILABS_DQPSK (for ISDB-T).

6.2.12 As from V2.6.9 (2015/12/17)

<improvement>[Unicable/I2c] In SiLabs_API_HW_Connect: removing 'connect' for unicable->i2c (not used).
<improvement>[Config/TS] In SiLabs_API_HW_Connect: improved comments. Now with the possibility to keep parameters untouched using values different from 0 or 1.
<improvement>[TS/GPIF] In SiLabs_API_TS_Mode: Improved GPIF control, to avoid testing GPIF and FIFO_SLAVE modes with single and dual EVBs. (Only available with the Cypress USB interface)

6.2.13 As from V2.6.8 (2015/12/03).

<improvement>[T2/C2/MPLP] In SiLabs_API_Select_PLP: now testing modulation against dd_status.modulation. This allows using SiLabs_API_Select_PLP in 'AUTO_DETECT/AUTO_T_T2' mode.
<new_Feature>[Config/TS] Adding SiLabs_API_TS_Config function to allow configuring the TS from the configuration macro. This is useful when different platforms don't use the same TS settings.

6.2.14 As from V2.6.7 (2015/11/25)

<improvement>[dual/triple/quad/Broadcast_i2c] In SiLabs_API_Demods_Broadcast_I2C: setting Silabs_multiple_front_end_init_done when done, to avoid calling the kickstart function when using the broadcast_I2C mode.
<improvement>[Legacy/SAT_tuner_init] In SiLabs_API_SAT_Tuner_Init: setting SAT_tuner_init_done.
NB: this function is normally not used. It's kept for compatibility with some customer MW.
<improvement>[Legacy/TER_tuner_init] In SiLabs_API_TER_Tuner_Init: setting TER_tuner_init_done.
NB: this function is normally not used. It's kept for compatibility with some customer MW.

6.2.15 As from V2.6.6 (2015/11/19)

<correction>[SSI/C/Not_locked/legacy] In SiLabs_API_SSI_SQI: correction contel selection to use 256QAM when not locked.
NB: This function is only used for 'legacy' products not supporting Si2183_L1_DD_SSI_SQI (FW computed SSI). No impact on current products.
<improvement>[RSSI/SAT] In SiLabs_API_Demod_status_selection: checking return value of SAT_TUNER_RSSI_FROM_IFAGC against -1000 (instead of -1), since -1 can be a valid value.
NB: This requires an update to the SAT tuner wrapper to V0.2.4, where SAT_TUNER_RSSI_FROM_IFAGC returns -1000 when not supported by the current SAT tuner.
<improvement>[dual/triple/quad/Broadcast_i2c] In SiLabs_API_Demods_Broadcast_I2C: setting Silabs_multiple_front_end_init_done when done, to avoid calling the kickstart function when using the broadcast-i2c mode.

6.2.16 As from V2.6.5 (2015/11/06)

<correction>[SSI/C/C2/S/S2/Not_locked] In SiLabs_API_Demod_status_selection:
Always calling Si2183_L1_DD_SSI_SQI when checking FE_QUALITY, even when not locked. When not locked, SQI will always be 0 but SSI will provide a useful info anyway.
Previous versions called SiLabs_API_SSI_SQI for C/C2/S/S2 in this case, and the returned SSI value is different from the value returned by Si2183_L1_DD_SSI_SQI.

<correction>[RSSI/SAT/legacy] In SiLabs_API_Demod_status_selection:
Checking that SAT_TUNER_RSSI_FROM_IFAGC supports the current SAT tuner before calling it.
If not, keep the status->RSSI value obtained from SiLabs_SAT_Tuner_Status.
NB: SAT_TUNER_RSSI_FROM_IFAGC has been introduced with V2.1.0. With V2.1.0 up to V2.6.4 the SAT RSSI provided by the SAT tuner status (a feature not available with all SAT tuners) was overwritten by SAT_TUNER_RSSI_FROM_IFAGC.
<new_feature>[demod_loop] In Silabs_demoloop: it's now possible to provide a string as an argument to demod_loop, in order to execute easily small tests.
<compatibility>[Linux/adapter_nr] In SiLabs_API_SAT_Select_LNB_Chip: Calling L0_SetAddress to set adapter_nr as add[15:8] (only useful if LINUX_I2C_Capability)

6.2.17 As from V2.6.4 (2015/10/12)

<correction>[S2X/Stream] In SiLabs_API_Select_Stream: stream_id type changed to 'signed int' to allow selection of the 'auto' mode using -1.
<correction>[S2X/flag] In SiLabs_API_Demod_status_selection: Adding missing 'break' lines in constellation/code_rate code used to set status->s2x.
<improvement>[EVB/macro] Adding Si2124 in possible tuners with DTV_SINGLE_TER_SAT_Rev2_0 macro
<compatibility>[TER/No_DVBT] In Silabs_UserInput_SeekNext: adding tags to allow compiling without DVBT

6.2.18 As from V2.6.3 (2015/10/06)

<correction>[RSSI/SAT/legacy] In SiLabs_API_Demod_status_selection:
Using status->RSSI instead of status->rssi in calls to SiLabs_API_SSI_SQI.
Calling SiLabs_API_SSI_SQI in SAT only when DD_SSI_SQI is not supported.
Only useful under the following combined conditions:

- Using floats is allowed
- Demodulator not supporting DD_SSI_SQI for all standards (legacy demodulators only)
- SAT reception
- SAT tuner with SAT_TUNER_RSSI_FROM_IFAGC capability (leading to status->RSSI being different from status->rssi)

<new_feature>[DVB-S2X/status] In SiLabs_API_Demod_status_selection: storing some DVB-S2X specific values in the status.
<improvement>[LINUX/ST_SDK2]
Changing type from CUSTOM_Standard_Enum to signed int for standard in severla function.
This is because the ST SDK2 enums used for 'standard' use more than 8 bits, so won't fit into a 'char'.
In SiLabs_API_Text_status_selection: not using divisions, using only int values for freq and symbol rate, since this is not allowed with ST SDK2.
<improvement>[SPI/setup] In Silabs_UserInput_SPI_Setup: not asking for sub fields if SPI is not used

<improvement>[LINUX/ST_SDK2] Adding some structures to make porting more convenient:
SiLabs_Lock_Struct
SiLabs_Seek_Init_Struct
SiLabs_Seek_Result_Struct
SiLabs_Params_Struct

6.2.19 As from V2.6.2 (2015/08/14)

<correction>[FE_status_selection/flags] In SiLabs_API_FE_status_selection: correcting test on flags to make it work as expected.
<new_part>[LNB/A8304] Adding compatibility with Allegro A8304 SAT LNB controller
<compatibility>[Xtal/Cap/SUPERSET] Adding SiLabs_API_XTAL_Capacitance to configure the XTAL capacitance value when using a XTAL as the clock source. The start_clk.tune_cap default value is set in Si2183_L2_SW_Init. Using SiLabs_API_XTAL_Capacitance is useful if different values need to be used for different platforms (i.e. when using Xtals with different internal capacitance). Only implemented with Si2183 (SUPERSET).
<compatibility>[NO_FLOATS] In SiLabs_API_Text_status_selection: changing BW and SR print-out to avoid issues if no floats are allowed.

<compatibility>[NO_FLOATS] In SiLabs_API_SSI_SQI_no_float: removing traces used during function development, to avoid trace issue if no floats are allowed.
<compatibility>[Si2165D/ber] In SiLabs_API_Demod_status_selection: setting status->ber by default at '1'. (This is only used with Si2165D)
<compatibility>[No TER] In SiLabs_SW_config_from_macro: using TER_Tuner_count only if compiled for TERRESTRIAL
<improvement>[DVB-T2/FFT_mode_1k] Adding code for FFT_MODE_1K in Silabs_fftCode/Custom_fftCode/Silabs_FFT_Text
<improvement>[Si1256D/DVB_T/QPSK] In Silabs_constelCode, Custom_constelCode and : adding DVB-T qpsk case.
<improvement>[No_DVB-T/BW] In Silabs_UserInput_bw_Hz: setting BW for 'TERRESTRIAL' instead of "DVB-T' to work with ISDB-T
Adding macros:
DTV_DUAL_TER_SAT_51A

6.2.20 As from V2.6.1 (2015/07/02)

<compatibility>[SILABS_SUPERSET/ISDB-T] Adding tags to allow compilation with ISDB-T only in several functions:
Silabs_fftCode / Silabs_giCode / Custom_fftCode / Custom_giCode
<compatibility>[SILABS_SUPERSET/SAT_ONLY] Adding tags in SiLabs_API_Broadcast_I2C to allow compiling without TER_TUNER_SILABS
<compatibility>[SILABS_SUPERSET/SAT_ONLY] Adding tags in SiLabs_API_bytes_trace to allow compiling without TERRESTRIAL_FRONT_END

6.2.21 As from V2.6.0 (2015/06/15)

<new_feature>[TER_Tuner/GPIOS] Adding SiLabs_API_TER_Tuner_GPIOs. Requires having the TER tuner init done, and the i2c pass-through to be controlled.
<correction>[NO_MATHS/LOG] In Silabs_Log10: Correction of an issue for values between 100 and 199, which all returned as '2'. Only used when 'NO_MATHS' is declared.
<compatibility>[Linux/Ubuntu] In SiLabs_API_SW_Init and SiLabs_API_Set_Index_and_Tag: minor changes to avoid compiling issues with Linux.
<compatibility>[VisualStudio] In Silabs_API_TS_Tone_Cancel: moving lines to have all declarations before assignments. Adding a trick to avoid 'set but not used' warning.

6.2.22 As from V2.5.9 (2015/06/07)

<new_feature>[DVB-S2X/Gold_Sequences] Adding SiLabs_API_SAT_Gold_Sequence_Init to compute a Gold Sequence initialisation value for a given Gold Sequence index.
<improvement>[EVB/macro] Adding Si216x_8x_EVB_RM_Rev1_0 macro
<compatibility>[Tizen/int&char] explicitly declaring all 'int as 'signed int' and all 'char' as 'signed char'.
This is because Tizen interprets 'int' as 'unsigned int' and 'char' as 'unsigned char'.
All other OSs interpret 'int' as 'signed int' and 'char as 'signed char', so this change doesn't affect other compilers.

To compare versions above V2.5.8 with older versions:

- Do not compare whitespace characters
- Either filter 'signed' or replace 'signed int' with 'signed' and 'signed char' with 'char' in the newer code first. (Take care to use 3 spaces in the string to be replaced).

<correction>[export/tags] Correcting tag in SiLabs_API_Get_AC_DATA for proper export without DVB-C2.

6.2.23 As from V2.5.8 (2015/06/01)

<new_feature>[Broadcast_I2c/Multiples/Si2183]
Adding 'broadcast_demods' and 'broadcast_i2c' options to demo loop
Adding SiLabs_API_Demods_Broadcast_I2C to load FW in demodulators using the 'broadcast i2c' mode.

Adding SiLabs_API_Broadcast_I2C to load FW in TER tuners then demodulators using the 'broadcast i2c' mode.
In SiLabs_API_TER_Broadcast_I2C: Filling a table with the demodulator's TER_Tuner_init_done flags, and setting them all to 1 if TER tuner FW downolad is ok.
Adding 'filtering' to SiLabs_API_Get_AC_DATA
<new_feature>[ISDB-T/AC_data]
Adding 'Get_AC_data' to demo loop
In SiLabs_API_Get_AC_DATA: second release of this function after testing and adding the filtering flag.
<new_feature>[SAT/Unicable_II]
Support for Unicable II
Adding 'unicable_install_II' option in demo_loop
(requires compiling with UNICABLE_II_COMPATIBLE, in addition to UNICABLE_COMPATIBLE)
In SiLabs_API_SW_Init: Adding SiLabs_API_SAT_read_diseqc_reply pointer in call to SiLabs_Unicable_API_Init (If UNICABLE_II_COMPATIBLE is defined)
<new_feature>[DVB-C2/status]
Adding DVB-C2 system information values in CUSTOM_Status_Struct (c2_system_id, c2_start_freq_hz, c2_system_bw_hz, num_data_slice)
<new_feature>[DVB-C2/Seek]

- In SiLabs_API_Demod_status_selection: Storing DVB-C2 system information in status
- In SiLabs_API_Text_status_selection: Removing duplicated lines in text status
- In Silabs_UserInput_bw_Hz: supporting DVB-C2 (6 or 8 MHz)
- In Silabs_UserInput_Lock: asking for BW for DVB-C2
- In Silabs_UserInput_SeekInit: asking for Seek_Step for DVB-C2
- In Silabs_UserInput_SeekNext: supporting DVB-C2 (browsing through Data Slices and PLPs)
- In SiLabs_Scan_Table_Carrier_Text: tracing BW in DVB-C2

<correction>[DVB-C2]
In SiLabs_API_Select_PLP:
Selecting DVB-T2 PLP only in DVB-T2
Correcting order of fields when selecting DVB-C2 PLPs.
<correction>[typo/Text_Status]
In SiLabs_API_Text_status_selection:
Replacing 'isdbt_system_id by t2_system_id when in DVB-T2
<correction>[typo/traces]
In SiLabs_API_Select_Stream:
Replacing 'PLP' by 'ISI stream' in error trace after calling Si2183_L1_DVBS2_STREAM_SELECT
<improvement>[traces] In Silabs_UserInput_SeekNext: tracing freq during handshakes
<improvement>[traces/typo] In SiLabs_API_Demod_status_selection: correction typo in trace (incorrect function name)
<improvement>[traces/status]
In SiLabs_API_Demod_status_selection: returning status_selection, to be used to fill the text status.
(the status_selection bits may be changed depending on the lock state)
In SiLabs_API_FE_status_selection: calling SiLabs_API_Text_status_selection and tracing the resulting string (only when SiTRACES are declared).
This is useful to check the front_end status in traces, we already asked several customer to add similar code, so now it's native.

6.2.24 As from V2.5.7 (2015/05/18)

<new_feature>[ISDB-T/AC_data] Adding SiLabs_API_Get_AC_DATA function to retrieve ISDB-T AC data.
<improvement>[kickstart] In SiLabs_API_Demods_Kickstart: directly setting demods to their final clock input for all demodulators (previous version only matched Si2183).

6.2.25 As from V2.5.6 (2015/04/02)

<improvement>[DVB_S2/status]

- In SiLabs_API_Demod_status_selection: storing status->num_is and status->isi_id when locked in DVB_S2, setting status->s2x to 1 if locked in a DVB-S2X MODCODE combination

- In SiLabs_API_Text_status_selection: filling text with status->num_is and status->isi_id values when locked in DVB_S2

<improvement>[DVB_S2/NO_S2X] In Custom_constelCode: moving S2 constellations to have them kept in the code when DEMOD_DVB_S2X is not defined.
<improvement>[SLEEP/ANALOG] In SiLabs_API_lock_to_carrier: returning 1 after calling SiLabs_API_switch_to_standard in ANALOG or SLEEP modes.
<improvement>[SOC_EVB/Si2165D] In SiLabs_API_Demod_status_selection for Si2165D: setting ber_mant/ber_exp based on status->ber, to allow proper BER display using SiLabs_API_Text_status_selection.
<improvement>[GPIF/Cypress] In SiLabs_API_TS_Mode: not using the result of the check on '-gpif', since it doesn't return the proper value.
<new_feature>[Cypress/streaming] In SiLabs_API_TS_Mode: allowing SILABS_TS_STREAMING option, to configure TS streaming independently from the demod settings
<new_feature>[SAT/TAG] In Silabs_API_Test: adding 'sat_tag' option

console
<new_feature>[Auto_Config] In SiLabs_macro_selection: adding auto configuration based on Cypress-contained macro

6.2.26 As from V2.5.5 (2015/02/05)

<new_feature>[SILABS_SUPERSET] Adding tags to allow compilation for TER-only/SAT-only/TER+SAT based on the superset code.

6.2.27 As from V2.5.4 (2015/01/22)

<new_feature>[SPI/Config] In SiLabs_API_SPI_Setup: setting front_end->Si2183_FE->demod->spi_download = 1 if send_option not 0
<new_feature>[Cypress/Ports] Adding SiLabs_API_Cypress_Ports

Console
<new_feature>[SPI/Config] Adding Silabs_UserInput_SPI_Setup
Adding 'SPI_Config' option in demo_loop
<new_feature>[Cypress/Ports] Adding Silabs_UserInput_Cypress_Ports
Adding 'Cypress_Ports' option in demo_loop

6.2.28 As from V2.5.3 (2015/01/16)

<new_feature>[Cypress/TS_SLAVE] In SiLabs_API_TS_Mode: Adding code to support SILABS_TS_SLAVE_FIFO (parallel TS retrieved using Cypress chip)
<improvement>{sw_options/LNBH29] In SiLabs_API_SAT_Possible_LNB_Chips: Adding text for LNBH29.
<new_feature>[Cypress/process] In Silabs_API_Test: Adding acess to L0_Cypress_Process
<new_part>[LNB/A8302] Adding compatibility with Allegro A8302 SAT LNB controller
<new_feature>[LNB/index] Adding SiLabs_API_SAT_LNB_Chip_Index, a function used to select the portion of an LNB controller is use. (set to 0 or 1 depending on the case). Compatible with LNBH26 and A8302.
<new_feature>[SAT_TUNER/sub] Adding SiLabs_API_SAT_Tuner_Sub, a funcitonused to select the sub-portion of a dual SAT tuner
<new_feature>[Cypress/TS_SLAVE] Adding SILABS_TS_SLAVE_FIFO = 5, in CUSTOM_TS_Mode_Enum;
<new_part>[LNB/A8302] Adding SAT_Select_LNB_Chip_lnb_index in SILABS_FE_Context, to store the index. This flag indicates which part in a dual LNB controller is in use.

Console:
<new_feature>[TER_Tuner/Config]
Adding Silabs_UserInput_TER_Tuner_ClockConfig
Adding Silabs_UserInput_SAT_Tuner_Sub
Adding SiLabs_UserInput_SAT_LNB_Chip_Index
Adding 'TER_Tuner_ClockConfig' option in demo_loop

Adding Silabs_UserInput_TER_Tuner_AGC_Input, Silabs_UserInput_TER_Tuner_IF_Output, Silabs_UserInput_TER_Tuner_ClockConfig, Silabs_UserInput_SAT_Tuner_Sub, SiLabs_UserInput_SAT_LNB_Chip_Index and Silabs_UserInput_TER_Tuner_ClockConfig calls in manual configuration loop
<improvement>[no_TER] In Silabs_UserInput_bw_Hz: not compiling code is not a terrestrial frontend

6.2.29 As from V2.5.2 (not published)

<correction>[ATV/TEXT status] In SiLabs_API_TER_Tuner_ATV_Text_status: enabling/disabling i2c to execute TER tuner status
<improvement>[SAT/LNB] In SiLabs_API_SAT_voltage: calling the L1_xxxx_InitAfterReset function for LNBH25, LNBH26 and LNBH29.
(other LNB chips don't need this, since all registers are written for each call)
<new_feature>[TS_spurs]
Adding SiLabs_API_Get_TS_Dividers to retrieve the TS clock dividers (only supported by Si2183 with recent FWs (above 5_0b13))
Adding Silabs_API_TS_Tone_Cancel to activate the Tone cancellation in the TER tuner (only supported by Si2190B initially)
<improvement>[Duals/SiLabs_EVBs] In SiLabs_API_SW_Init: surrounding ts_mux related code by USB_Capability, since it's only valid on some SiLabs EVBs
<improvement>[comments] In SiLabs_TS_Crossbar_TS1_TS2: comment correction

Console:
<improvement>[seek/future] in Silabs_UserInput_SeekNext: comparing t2_version >= SILABS_T2_VERSION_1_3_1 to be compliant with possible future T2 versions.

6.2.30 As from V2.5.1 (2014/11/21)

<correction> [status/BER] In SiLabs_API_Demod_status_selection: Swapping dd_ber_resol exp and mant for Si2164/83 for exp=7 and mant=1 (in place of exp=1 mant=7).
<new_feature>[TER_Tuner/Config]
Adding SiLabs_API_TER_Tuner_FEF_Input to allow configuration of the TER tuner FEF input. This needs to be added to the configuration macros.
The default value is '1' to select GPIO1 on the TER tuner side.
<improvement>[TERACOM/BER] In SiLabs_API_Demod_status_selection: Changing BER settings when locked in DVB-T (1;6) vs other standards (1;7). This is to improve measurement accuracy for BER criteria.
<improvement> [status/return value] In SiLabs_API_Demod_status_selection: returning 1 when status function meets no problem.
<improvement>[cleanup] In SiLabs_API_TER_Tuner_ATV_Tune: removing invert_spectrum.
<improvement/compatibility>
In SiLabs_API_SAT_Possible_LNB_Chips: Setting i to avoid warning when not used.
In SiLabs_API_Select_PLP: Setting plp_id and plp_mode to avoid warning when not used.
In SiLabs_API_TER_Tuner_ATV_Tune: Setting all variables to avoid warnings when not used.
In Silabs_API_Test: using standard to avoid warning when not used.
<correction> [constel/DVB-S2X] In Silabs_constelCode and Custom_constelCode: Adding DVB-S2-X specific constellations.

Config Macros:
Adding calls to SiLabs_API_TER_Tuner_FEF_Input to all macros for TER configurations (by default FE input set on GPIO1).

Console code:
<improvement>[cleanup] removing unused variables:
In Silabs_UserInput_Test: removing valid_target.
In Silabs_demoloop: removing dval, num_data_slice, num_plp, symbol_rate_bps, constellation.

Crossbar Code:
<improvement>[cleanup]

In SiLabs_TS_Crossbar_TS_Status: removing front_end (unused).
In SiLabs_TS_Crossbar_TS1_TS2: removing several unused variables.

6.2.31 As from V2.5.0 (2014/09/04)

<improvement>[status/ISDBT> Adding partial_flag in _CUSTOM_Status_Struct
In SiLabs_API_Demod_status_selection: storing partial flag information in status->partial_flag.
<new_feature>[Test_Pipe/init_ok] In Silabs_API_Test: adding 'init_ok' to know if demod init is done
<new feature>[TER_Tuner/Config]
adding SiLabs_API_TER_Tuner_Block_VCO2 and SiLabs_API_TER_Tuner_Block_VCO3 to allow configuration of the TER_Tuner block_VCO2_code and block_VCO3_code.
<new_feature>[I2C/Tuners_Direct] In SiLabs_API_XXX_Tuner_I2C_Enable/SiLabs_API_XXX_Tuner_I2C_Enable: using a special value (100) to allow having direct connection to tuners (without demod pass-through).
API CONFIG in such case:
SiLabs_API_TER_tuner_I2C_connection(front_end, 100);
SiLabs_API_SAT_tuner_I2C_connection(front_end, 100);
<new_feature>[Test_Pipe/LNBH26] in Silabs_API_Test: adding 'lnbh26' 'a_b' '0/1' option to select which LNB controller is used (LNBH26 is a dual).
<new_feature>[CONFIG/tracing] Adding all configuration fields in SILABS_FE_Context (to enable configuration checking after init).
NB: This allows removing some previous code used to avoid compilation warnings, since all fields are not used.
Adding SiLabs_API_Config_Infos. This function is useful to check the configuration parameters based on the related function name. (Use "Full" for the function name to get the entire configuration).

Config macros
Renaming macros:

Si21682_EVB_Rev1_0_Si2164	becomes	Si21682_EVB_Rev1_0_41A_64A
Si21682_EVB_Rev1_0_Si21652B	becomes	Si21682_EVB_Rev1_0_41A_67B
Si21682_EVB_Rev1_0_Si2183	becomes	Si21682_EVB_Rev1_0_41A_83A
Si21662_EVB_Rev1_0_Si2167B	becomes	Si21662_EVB_Rev1_0_67B

Console code
<improvement>[TS/GPIF] In Silabs_UserInput_TS: Adding control of the mux used to select which TS is sent to GPIF
<correction>[T2_lock_mode] In Silabs_UserInput_Lock, using T2_lock_mode in the call to SiLabs_API_lock_to_carrier.
<improvement>[config/TER_only/SAT_only]: In main: if tuner i2c address is 0x00, skip config for TER or SAT respectively.

6.2.32 As from V2.4.9 (2014/08/22)

<improvement/Duals> Adding demod_die in _CUSTOM_Status_Struct
In SiLabs_API_Demod_status_selection: storing demod die information in status->demod_die.
In SiLabs_API_Text_status_selection: adding die information to status text.
<typo/T2_lock_mode> In SiLabs_API_lock_to_carrier comments: T2_lock_mode is independent of num_lp.
Parameter: T2_lock_mode the DVB-T2 lock mode (0='ANY', 1='T2-Base', 2='T2-Lite')
<typo/ISDB-T> In SiLabs_API_lock_to_carrier comments: bandwidth_Hz is also used for ISDB-T
Parameter: bandwidth_Hz the channel bandwidth in Hz (only for DVB-T, DVB-T2, ISDB-T)
<correction/Duals> In SiLabs_API_Demods_Kickstart: directly setting demods to their final clock input.

NB: For duals (Si216x2), the clock source should not change between TER and SAT, and the clock should be 'always-on': In calls to SiLabs_API_TER_Clock/SiLabs_API_SAT_Clock: use identical settings and force 'clock_control = 1'.

Console code:
<correction/T2_lock_mode> In Silabs_UserInput_Lock, using T2_lock_mode in the call to SiLabs_API_lock_to_carrier.

6.2.33 As from V2.4.8 (2014/08/19)

<new_part/A8293/LNB_Supply> Compatibility with Allegro's A8293 (needs A8293_COMPATIBLE)

6.2.34 As from V2.4.7 (2014/08/13)

<improvement/Settings> In SiLabs_API_Demod_status_selection: Saving status->IFagc depending on the AGC in use. Previous versions assumed that AGC1 for SAT and AGC2 for TER. This restriction doesn't apply anymore.
<improvement/compatibility> Changing all '//' comments to '/* */'
In SiLabs_API_SAT_Tuner_status: setting sat_tuner by default to avoid compiler warnings.
In SiLabs_API_TER_Tuner_ClockConfig: setting tuner_ter by default to avoid compiler warnings.
In SiLabs_API_TS_Mode: moving SiTRACE after all variables are declared.
In SiLabs_API_SAT_AutoDetectCheck: moving SiTRACE after all variables are declared.
<improvement/NO_FLOAT> Adding SiLabs_API_SSI_SQI_no_float and Silabs_Log10_10000 to enable SSI and SQI computing when not done by FW and when using floats is not allowed.
<correction/BER>
In SiLabs_API_Demod_status_selection:
storing ber_exp+1 in status->ber_exp.
storing per_exp+1 in status->per_exp.
storing fer_exp+1 in status->fer_exp.
In rate_f_mant_exp: exp treated to match the changes in SiLabs_API_Demod_status_selection.
<correction/RSSI/Si2167B> In SiLabs_API_Demod_status_selection: Calling SAT_TUNER_RSSI_FROM_IFAGC for Si2167B in SAT.

Console code:
<improvement/compatibility>
In Silabs_UserInput_SeekNext: setting hierarchy to 0 by default to avoid compiler warning.
In Silabs_UserInputTSCrossbar: setting variables default values to avoid compiler warnings.

TS_CROSSBAR:
In SiLabs_TS_Crossbar_TS1_TS2: setting ts_1_source and ts_2_source to NULL to avoid compiler warnings

Header:
<compatibility/Tizen> SILABS_QAMAUTO: replacing '-1' by '100. This is because Tizen uses 'unsigned char' when 'char foo;' is defined. All other OSs use 'signed char' by default and have no issue with this.

6.2.35 As from V2.4.6 (2014/07/18)

<improvement>[TERACOM/BER] Changing BER settings when locked in DVB-T (5;8) vs other standards (1;7). ONLY for Si2183.
<improvement>[DUALS/XTAL] In SiLabs_API_Demods_Kickstart: Using TUNE_CAP_15P6 and CLK_MODE_XTAL to make sure demodulators are not pulling the clock input pin low.

CONFIG_MACROS
Adding macros for EVBs with Airoha tuners
For dual/triple/quad: Setting clock_control field to '1' to avoid glitches on other front-ends when changing the clock source.

TS_Crossbar (only for duals with crossbar capability)

<improvement/Crossbar> [SW_Init] In SiLabs_TS_Crossbar_SW_Init: removing call to status function, to keep the SW init function only managing pointers and structures, with no i2c traffic and no need for prior HW init.
<improvement/Crossbar> [Default values] In SiLabs_TS_Crossbar_TS_Status: Setting ts_1_source and ts_2_source by default, to have them initialized in all cases. There was an issue with the previous version when an invalid configuration was requested.

<redefinition>[New_TS_Pads] SiLabs_TS_Crossbar_Serial_Config redefined to add support for the new TS pads.
<improvement>[warning] In SiLabs_TS_Crossbar_TS1_TS2: moving setup of ts1_source and ts2_source to avoid compilation warnings.
<new_feature>[New_TS_Pads] In SiLabs_TS_Crossbar_TS1_TS2: Setting DD_TS_SLR_SERIAL property when in serial mode

6.2.36 As from V2.4.5 (2014/07/04)

Console code:

<correction>[RENAMING] renaming Silabs_UserInput_TER_Address for consistency with the rest of the code.
In 'Silabs_demoloop': adding 'TER_address'
<new_feature>[TER_Tuner/Config]
Adding Silabs_UserInput_TER_Tuner_AGC_Input and Silabs_UserInput_TER_Tuner_IF_Output
Adding 'TER_Tuner_AGC_Input' and 'TER_Tuner_IF_Output' options in demo_loop

6.2.37 As from V2.4.4 (2014/06/06)

Console code:

<new_feature> [DVB-T2/MPLP] In Silabs_UserInput_SeekNext: DVB-T2 MPLP parsing OK with MPLP in both T2-base and T2-lite

6.2.38 As from V2.4.3 (2014/06/02)

Adding SiLabs_Scan_Check_And_Add_Carrier, used during SeekNext to check signal quality, turn TS on and store the channel information.
In Silabs_UserInput_SeekNext: checking all hierarchy and PLP information when required, and calling SiLabs_Scan_Check_And_Add_Carrier when locked.
This shows the recommended behavior when dealing with MPLP and hierarchical channels.
In the final application, SI/PSI parsing should be done in a function similar to SiLabs_Scan_Check_And_Add_Carrier.

6.2.39 As from V2.4.2 (2014/05/28)

<correction> [Si2164A/Si2169B/Si2168B/DVBT2_C/N] In SiLabs_API_Demod_status_selection:
Correctly setting status->c_n to dvbt2_status.cnr/4.0 for the related parts.
<correction>[TS_Crossbar] TS crossbar feature now working (with duals with this capability) as from FW 4_ab4.

Console code:

Adding SiLabs_Scan_Check_And_Add_Carrier, used during SeekNext to check signal quality, turn TS on and store the channel information.
In Silabs_UserInput_SeekNext: checking all hierarchy and PLP information when required, and calling SiLabs_Scan_Check_And_Add_Carrier when locked.
This shows the recommended behavior when dealing with MPLP and hierarchical channels.
In the final application, SI/PSI parsing should be done in a function similar to SiLabs_Scan_Check_And_Add_Carrier.
There may still be some details to improve for DVB-T2 MPLP.

TS_Crossbar Code:

<redefinition> SiLabs_TS_Crossbar_TS_Status redefined to do the status on both demods, to be able to set the TS_1/TS_2 modes
<correction> In SiLabs_TS_Crossbar_TS1_TS2:

Corrected test to avoid sending twice the same command when a single source is used for both TS_1 and TS_2.
Propagating TS drive strength/shape values in secondary TS, not only in primary TS.

6.2.40 As from V2.4.1 (2014/05/26)

<improvement>[Si2167B] In Silabs_coderateCode/Custom_coderateCode: tags changed to allow export for Si21652B
<new_Feature> Adding TS crossbar feature. NB: This is only valid for dual demodulators including the TS crossbar feature.
<new_feature> [ISDB-T/LAYER] Adding SiLabs_API_TER_ISDBT_Monitoring_mode property to select the layer used for BER, CBER, PER, and UNCOR monitoring in ISDB-T.
<improvement> [TUNER_STATUS] In SiLabs_API_FE_status_selection: only tracing RSSI and freq if tuner has been statused. Not returning an error if the standard is unknown.

Adding SiLabs_API_TER_Tuner function to retrieve the ter tuner context pointer
Adding SiLabs_API_TER_Tuner_Dual_Driving_Xtal to handle the case where the HW is using a SiLabs Dual TER tuner and the TER tuner in fe[0] is driving the xtal.
Adding SiLabs_API_Demod_Dual_Driving_Xtal to handle the case where the HW is using a SiLabs Dual Demod and the demod in fe[0] is driving a xtal.

Adding TS crossbar capability (in 2 separate files). This is only available with dual demodulators, and shouldn't be included in applications not able to use it.

6.2.41 As from V2.4.0 (2014/05/02)

<new_feature> [tag/level] adding tag and level support for Si2167B and derivatives

6.2.42 As from V2.3.9 (2014/04/22)

<new_parts> Status updated for ISDB-T support.
Adding ISDB-T values in CUSTOM_Status_Struct (isdbt_system_id, nb_seg_a, nb_seg_b, nb_seg_c)
Console code completed for ISDB-T support
Adding configuration macros for SiLabs socket EVB to differentiate between Si2178 and Si2178B.

6.2.43 As from V2.3.8 (2014/04/16)

<new_feature> Using STRING_APPEND_SAFE macro (defined in Si_I2C V3.4.5), for Linux compatibility.

6.2.44 As from V2.3.7 (2014/04/12)

<new_parts> [Si2183] Adding Si2183 support
<new_standard> [ISDB-T] Adding support for ISDB-T
<new_feature> [DVB-S2X] Adding Constellation and code rate functions for DVB-S2X.
<new_feature> [TAG/LEVEL] Adding definitions for TAG and level
Adding SiLabs_API_Set_Index_and_Tag

6.2.45 As from V2.3.6 (2014/03/28)

<correction> [LNBH29] In SiLabs_API_SAT_Select_LNB_Chip: using lnb_code 29 to select LNBH29

<new_feature> [STATUS/SELECTION] Adding SiLabs_API_Demod_status_selection / SiLabs_API_FE_status_selection / SiLabs_API_Text_status_selection.
The behavior when calling SiLabs_API_Demod_status / SiLabs_API_FE_status_ / SiLabs_API_Text_status is unchanged, since these call the new functions with the value '0x00' which means 'status all items'
These can be used to status only a portion of the CUSTOM_Status_Struct, depending on a status_selection bit field, using the following bit flags:
FE_LOCK_STATE : demod_lock, fec_lock, uncorrs, TS_bitrate_kHz, TS_clock_kHz
FE_LEVELS : RSSI, RFagc, IFagc
FE_RATES : BER, PER, FER (depending on standard)
FE_SPECIFIC : symbol_rate, stream, constellation, c/n, freq_offset, timing_offset, code_rate, t2_version, num_plp, plp_id, ds_id, cell_id, etc (generally one function called per standard).
FE_QUALITY : SSI, SQI
FE_FREQ : freq
<new_feature> [SW_CONFIG] Adding SiLabs_API_Frontend_Chip / SiLabs_API_TER_tuner_I2C_connection / SiLabs_API_SAT_tuner_I2C_connection
These will be used instead of direct access to the L3 context values.
They also allow easier access from the top level, and allow configuring the GUI using script files.
<improvement> [T2/C2/MPLP/SEEK] In SiLabs_API_Channel_Seek_Next: if locked, updating value of front_end->standard.
This removes the need to call SiLabs_API_Demod_status to update this value, which is used when retrieving the plp_ids and ds_ids.
<improvement> [portability] In SiLabs_API_SSI_SQI: moving code after all declarations, because this creates compilation errors with some compilers.
<improvement> [portability/NO_FLOATS_ALLOWED] In status functions, store information as rate_mant/rate_exp for ber/per/fer, and use these instead of the double fields.
<improvement> [renaming] SiLabs_API_TER_FEF_CONFIG renamed as SiLabs_API_TER_FEF_Config, for consistency with other configuration functions.
<improvement> [traces] Adding dedicated trace messages to help trace wrapper function calls:
'API CALL CONFIG' for SW configuration functions, formatted as in configurations macros.
These will be useful to check the SW configuration in the traces, and create the corresponding configuration macros.
'API CALL SEEK' for scan-related functions
'API CALL INIT' for init-related functions
'API CALL LOCK' for lock-related functions
'API CALL STATUS' for statusing functions
<improvement> [DVB-T2] In SiLabs_API_Demod_status: statusing status->t2_system_id
<improvement> [Si2164/ANALOG] In Silabs_standardCode: Adding 'ANALOG' value for Si2164.

Console code:
Removing call to SiLabs_API_Auto_Detect_Demods in ‘main’, as this creates issues with some customer HW when using a Xtal on the demodulator.

6.2.46 As from V2.3.5 (2014/01/07)

<new_feature> [Si2164/SPI) Adding SiLabs_API_SPI_Setup (Only available with Si2164 derivatives as from today)
CAUTION1: In any case, this requires updating the following item to support SPI download:
- SiLabs_L0 source code. The SPI support functions also need to be ported to your platform(s)

CAUTION2: When used with SiLabs EVBs, this requires updating the following items to support SPI download using the Cypress chip:
- Cypress FW
- Cypress DLL

Adding SiLabs_API_SPI_Setup
<improvement> [code_checkers] In text-oriented functions using sprintf:
Replacing sprintf by snprintf with a max size at 1000, as this is safer.
The only constraint is that text strings need to be declared with a minimum size of 1000 bytes.
This should be enough to pass through code checkers.

6.2.47 As from V2.3.4 (2013/11/22)

<new_feature> [TER TUNER/Multi-frontend] Adding SiLabs_API_TER_Tuner_ClockConfig, to easily configure the TER tuner clock:
int SiLabs_API_TER_Tuner_ClockConfig (SILABS_FE_Context *front_end, int xtal, int xout);
xtal = 1: a Xtal is connected to and driven by the TER tuner.
xtal = 0: a clock signal is connected to the TER tuner, which doesn't drive a Xtal.
xout = 1: the clock is going out of the TER tuner.
xout = 1: no clock is going out of the TER tuner.

Adding SW_INIT_Dual_Si2191_Si216x2_Si2164 macro (for QUAD EVB, using new SiLabs_API_TER_Tuner_ClockConfig function)

6.2.48 As from V2.3.3 (2013/11/18)

<new feature> [handshake] Adding SiLabs_API_Handshake_Setup, to easily control the handshake parameters from the wrapper level
<new feature> [Si2164] Adding SiLabs_API_TER_T2_lock_mode, to select the T2 lock mode. It can be used to select the T2 lock mode during channel Seek.
This avoids the need to add a parameter to Seek_Init

console:

In Silabs_UserInput_SeekInit: calling Silabs_UserInput_T2_lock_mode if required.
In Silabs_demoloop: Adding T2_lock_mode option

6.2.49 As from V2.3.2 (2013/11/11)

<new feature> [Si21x8 tuners] Adding SiLabs_API_TER_Broadcast_I2C, useful to enable the broadcast i2c feature (only available with Si21x8B tuners)
<improvement> [AUTO_T_T2] In SiLabs_API_Demod_status: setting front_end->standard to match status->standard when locked. This is useful for SiLabs_API_Get_PLP_ID_and_TYPE when in AUTO_T_T2 and locked on a T2 signal:
if front_end->standard is left as 'SILABS_DVB_T' the function returns 0 while it needs to call Si216x_L1_DVBT2_PLP_INFO
<new feature> [MCNS] In SiLabs_API_Demod_status: adding MCNS support

Console:
Adding Silabs_UserInput_T2_lock_mode
In Silabs_UserInput_Lock: calling Silabs_UserInput_T2_lock_mode if required.
In SiLabs_Scan_Table_Carrier_Text: displaying T2 mode

6.2.50 As from V2.3.1 (2013/10/03)

<correction> In SiLabs_API_TS_Mode:
Correcting the 'SILABS_TS_TRISTATE' case to use the 'TRISTATE' mode
Adding SILABS_TS_OFF in CUSTOM_TS_Mode_Enum structure
Adding 'SILABS_TS_OFF' case for Si2164/Si2167B/Si2169A
<correction> In SiLabs_API_DEMOD_Status:
Adding the SILABS_MCNS case
<new feature> Adding 'int clock_control' to SiLabs_API_TER_Clock and SiLabs_API_SAT_Clock prototypes.
This is used for multi-frontends applications when a tuner's clock is forwarded to another frontend.
In this case it needs to be 'ALWAYS_ON'.
To keep the previous behavior, use '2' (i.e. the 'MANAGED' mode)
Adding the corresponding code in:
SiLabs_API_TER_Clock_Options/SiLabs_API_TER_Clock
SiLabs_API_SAT_Clock_Options/SiLabs_API_SAT_Clock
<new_feature> Adding t2_version monitoring and related functions

6.2.51 As from V2.3.0 (2013/09/24)

Reverting changes to constellation type in SiLabs_API_lock_to_carrier function, as this forbids using the value of '-1' as SILABS_QAMAUTO.

Using 'unsigned char constellation' broke the DVB-C AUTO qam capability.

SiLabs_API_lock_to_carrier prototype is now:

```
int   SiLabs_API_lock_to_carrier (SILABS_FE_Context *front_end,
                                  unsigned char standard,
                                           int freq,
                                           int bandwidth_Hz,
                                  unsigned char stream,
                                  unsigned int  symbol_rate_bps,
                                           char constellation,
                                  unsigned char polarization,
                                  unsigned char band,
                                           int data_slice_id,
                                           int plp_id,
                                  unsigned char T2_lock_mode);
```

6.2.52 As from V2.2.9 (2013/09/24)

In Custom_giCode / Silabs_giCode / Silabs_GI_Text:

Adding 1/64 GI code handling (for DVB-C2)

Adding lnb_chip_address to SILABS_FE_Context

Adding SiLabs_API_TER_FEF_Options and SiLabs_API_TER_FEF_CONFIG functions, to allow different FEF configuration depending on the frontend. This is required when using dual demodulators, where there are restrictions on MP_x and GPIOx pin usage.

Changing SiLabs_API_SAT_Select_LNB_Chip function definition to add teh lnb chip address. This is required for multi-frontend SAT applications.
Changing SiLabs_API_switch_to_standard and SiLabs_API_set_standard function definitions to use 'unsigned char' instead of 'int' for standard.

Changing SiLabs_API_lock_to_carrier function definitions to use 'unsigned char' instead of 'int' for standard/stream/constellation/polarization/band/T2_lock_Mode. This avoids casting the related values to (unsigned int) within the functions. data_slice_id and plp_id are kept as 'int', as they may take a value of '-1' at wrapper level to select the corresponding 'auto' modes.

In SiLabs_API_Demod_status:

Adding one SiTRACE right after DD_STATUS, to trace the demod address (useful in multi-front-ends), the lock state and the standard.

In SiLabs_API_Demod_status / SiLabs_API_SAT_Tuner_status / SiLabs_API_SAT_Tuner_Tune / SiLabs_API_TER_Tuner_Init / SiLabs_API_TER_Tuner_Text_status / SiLabs_API_TER_Tuner_ATV_Tune / SiLabs_API_TER_Tuner_Block_VCO:

Changing I2C Enable/Disable calls to use the TER and SAT indirect i2c enable/disable calls, to allow tuner rssi statusing if INDIRECT_I2C_CONNECTION is used

In SiLabs_API_SAT_Tuner_I2C_Enable and SiLabs_API_TER_Tuner_I2C_Enable:

Replacing 'count' by 'fe_count', as 'count' may be a reserved word in some implementations.

In SiLabs_API_SAT_Tuner_I2C_Enable and SiLabs_API_SAT_Tuner_I2C_Disable:

Correcting code to properly connect the required i2c pass-through (previously only working for the SAT tuner on frontend 0 only).

In SiLabs_API_FE_status:

Directly tracing freq and tuner rssi before calling SiLabs_API_Demod_status.

In SiLabs_API_Text_status:

Adding config_code to text status. This is useful to knwo which frontend is statused in multi-frontend applications

In SiLabs_API_SSI_SQI:

Correcting SiTRACEs to display entire messages (last parameter wasn't displayed).

In SiLabs_API_Select_PLP:

Adding DVB-C2

Console code:

Limiting string lengths below 500 in sprintf and printf, to avoid warnings when compiled with '-pedantic'.

In 'main':

Using unsigned int for values entered as hexa values.during manual configuration.

Manual configuration now including FEF configuration and LNBH controller address

Macros code:

Adding Si216x2_EVB_Rev1_x_Si2164 macro, for Dual EVB based using Si2164 source code

6.2.53 As from V2.2.8

Adding t2_base_lite in CUSTOM_Status_Struct

Adding Silabs_T2_Base_Lite_Text function.

In SiLabs_API_Text_status:

Adding T2 base/lite text for T2

Added MCNS in frequency display. MCNS text status didn't fill entirely due to this.

Reduced code for frequency diplay.

In SiLabs_API_TER_Clock / SiLabs_API_TER_AGC : adding tags to remove code for non-TER parts

In SiLabs_API_SAT_Clock / SiLabs_API_SAT_AGC / SiLabs_API_SAT_Spectrum : adding tags to remove code for non-SAT parts

Console code:

In Silabs_UserInput_tune: Compatibility with NCNS

Adding 'simu' and 'close' options in demo_loop

Separating the init macro management from the console code.
To enable configuration using macros, add the SiLabs_API_L3_Config_Macros code, and define CONFIG_MACROS at project level.

Added files:
SiLabs_API_L3_Config_Macros.c
SiLabs_API_L3_Config_Macros.h
These files can be used to prepare configuration macros for various platforms.
The functions in this code were previously in the console code.
Moving them to separate files allows using these from other applications.

6.2.54 As from V2.2.7

In SiLabs_API_Get_PLP_ID_and_TYPE: comparing standards value to SILABS_DVB_T2.
(previously using Si2164_DD_MODE_PROP_MODULATION_DVBT2, which is incorrect at wrapper level).
In SiLabs_API_TER_Clock: correction of Si2165 text related to clock source pin numbers
In SiLabs_API_TER_AGC: correction of code used for Si2165

6.2.55 As from V2.2.6

SiLabs_API_L3_Wrapper.c
In SILABS_FE_Context structure: Adding config_code, used to store the i2c addresses of the TER tuner (bits[23:16]), the SAT tuner (bits[15:8]) the demod (bits[7:0]).
This is used to know which path is controlled in multi-frontend applications, even when not tracing L0 bytes.
In SiLabs_API_Channel_Seek_Next: Adding T2_base_lite flag
In SiLabs_API_Channel_Seek_Next: Adding T2_base_lite flag (indicates whether the locked signal is T2-Base or T2-Lite)
In SiLabs_API_SAT_Select_LNB_Chip: Returning front_end->lnb_chip if OK, 0 otherwise. This compiles correctly for non-SAT products.
In SiLabs_API_lock_to_carrier: Adding T2_lock_mode flag (selects whether to lock on the T2-Base or T2-Lite signal (o='any'))
In SiLabs_API_Tune: Compatibility with Si2169B
In SiLabs_API_Get_PLP_ID_and_TYPE: Adding C2 compatibility (for Si2164)
Adding SiLabs_API_Get_DS_ID_Num_PLP_Freq function, for DVB-C2 Dataslice handling
Adding SiLabs_API_Auto_Detect_Demods, for demodulators auto-detection.
In SiLabs_API_TER_Tuner_Text_status: Compatibility with SiLabs_TER_Tuner wrapper
In SiLabs_API_TER_Tuner_ATV_Text_status: Compatibility with SiLabs_TER_Tuner wrapper
In SiLabs_API_TER_Tuner_DTV_Text_status: Compatibility with SiLabs_TER_Tuner wrapper
In SiLabs_API_TER_Tuner_ATV_Tune: Compatibility with SiLabs_TER_Tuner wrapper
In SiLabs_API_TER_Tuner_Block_VCO: Compatibility with SiLabs_TER_Tuner wrapper
WARNING: The latest TER tuners are NOT supported if not using the SiLabs_TER_Tuner wrapper
In SiLabs_API_SSI_SQI: Adding C2 SSI SQI

SiLabs_API_L3_Console.c
In Silabs_UserInput_tune: Compatibility with DVB-C2
In Silabs_UserInput_Lock: Adding T2 Base/Lite flag for T2 in call to SiLabs_API_lock_to_carrier
In Silabs_UserInput_SeekNext: Compatibility with DVB-C2 (early version)
In SiLabs_Scan_Table_Carrier_Text: Compatibility with DVB-C2
Adding T2 Base/Lite flag in channel tables and in function calls
In Silabs_demoloop:
Adding 'detect' option

Adding 'cell_id' option
In Silabs_help: Adding C2 help

6.2.56 As from V2.2.5

In SiLabs_API_SSI_SQI: added DVB-C capability
In SiLabs_API_Demod_status: calling SiLabs_API_SSI_SQI whenever SSI/SQI values haven't been set earlier.
(the latest demodulators will have the SSI/SQI feature implemented in FW, and SiLabs_API_Demod_status will use the FW function if this is the case)

6.2.57 As from V2.2.4

In SiLabs_API_SAT_Tuner_status: compatibilty with SILABS_SAT_TUNER_API

6.2.58 As from V2.2.3

Compatibility with several LNBH controllers in the same application.
Adding SiLabs_API_SAT_Possible_LNB_Chips and SiLabs_API_SAT_Select_LNB_Chip to allow easy selection of the LNB controller
Si2167B compatibility with INDIRECT_I2C_CONNECTION
Si2167B compatibility with TEr and SAT configuration

6.2.59 As from V2.2.2

SILABS_SAT_TUNER_API compatibility (the only way to work with Si2164):
Adding SiLabs_API_Select_SAT_Tuner function, useful to select the SAT tuner for each demodulator
Adding SiLabs_API_SAT_Address function, useful to set the I2C address of any TER tuner
Adding SiLabs_API_SAT_Clock and SiLabs_API_SAT_AGC functions, to configure the clock paths (source, input, freq) and AGC.
NB: This only works if matching functions are added to the demodulator code.
NB: In this first version, these functions only support SI2164, to keep the 'legacy' device codes untouched

In console code:
Adding SiLabs_SW_config_from_macro
Adding SiLabs_SW_config_selection

In console main:
SW init via several ways for ‘single/dual/triple/quad’:

1- init from a batch file, using the first argument as the configuration selector
   - example: if executable is named ‘SiLabs.exe’ and a configuration named ‘my_design’ is known, calling from a .bat file with ‘Silabs.exe my_design’ will automatically load this configuration.
2- init from a user-selected configuration
   - example: in the case described above, if the executable is launched by double-clicking, the user will be asked to select one configuration among the existing ones.
3- init using manually-entered settings
   example: if no configuration is selected, the user will be asked to enter all the necessary information.

6.2.60 As from V2.2.1

SiLabs_API_Wrapper for V2.2.1:
Adding INDIRECT_I2C_CONNECTION control, allowing tuner i2c connection via any demodulator.
This is used for applications with multiple demodulators

Adding SiLabs_API_SAT_Tuner_I2C_Enable, SiLabs_API_SAT_Tuner_I2C_Disable, SiLabs_API_TER_Tuner_I2C_Enable, SiLabs_API_TER_Tuner_I2C_Disable
These functions are used for INDIRECT_I2C_CONNECTION control
SILABS_TER_TUNER_API compatibility (the only way to work with Si2164):
Adding SiLabs_API_Select_TER_Tuner function, useful to select the TER tuner for each demodulator
Adding SiLabs_API_TER_Address function, useful to set the I2C address of any TER tuner
Adding SiLabs_API_TER_Clock and SiLabs_API_SAT_Clock functions, to configure the clock paths (source, input, freq).
NB: This only works if matching functions are added to the demodulator code.
NB: In this first version, these functions only support SI2164, to keep the 'legacy' device codes untouched
In SiLabs_API_TER_Tuner_ATV_Text_status: not implemented for SILABS_TER_TUNER_API (not sure it was ever used)

6.2.61 As from V2.2.0

Added Si2168B and Si2169B in list of related SiLabs demodulators
In SiLabs_API_SAT_Unicable_Install:
For Si2167: Using ds_sequence_mode 'manual' to select 'no_gap' sequences in Unicable mode
In SiLabs_API_SAT_Unicable_Uninstall:
For Si2167: Using ds_sequence_mode 'auto' to select 'gap' sequences in Normal mode

6.2.62 As from V2.1.9

Using '0x216...' for Si2164 derivatives, to differentiate 68A/69A from 68B/69B
Compatibility with 'generic' TER_TUNER_SILABS API, for easier integration of future SiLabs tuners
In SiLabs_API_SAT_Unicable_Install:
Using new DD_DISEQC_PARAM property to select 'no_gap' sequences in Unicable mode
In SiLabs_API_SAT_Unicable_Uninstall:
Using new DD_DISEQC_PARAM property to select 'gap' sequences in Normal mode
In console code:
In Silabs_UserInput_standard:
Adding 'C2'

6.2.63 As from V2.1.8

In SiLabs_API_SAT_voltage, for LNBH25 (as this part requires an init):
if (front_end->lnb_chip_init_done == 0) { front_end->lnb_chip_init_done = L1_LNBH25_InitAfterReset(front_end->lnbh25); }
Adding SiLabs_API_SAT_Unicable_Uninstall, to allow easily switching between NORMAL and UNICABLE modes
In console code:
Replaced 'DEMOD_Si21xx' compilation flags by 'Si21xx_COMPATIBLE', to have unused flags removed during export
In Silabs_UserInput_SeekNext: 'beep' on each new carrier
In SiLabs_Scan_Loop: displaying number of carriers found during the last loop instead of the total number of carriers.
In Silabs_demoloop: adding 'unicable_install' and 'unicable_uninstall' options

6.2.64 As from V2.1.7

Adding SiLabs_API_SAT_voltage and SiLabs_API_SAT_tone, to allow managing the voltage separately from the tone.
This is mostly interesting for Unicable, where the tone is not used to select the band.
It's used in the Unicable code as from 2013/03/14 (SVN3657) to save time when sending a Unicable message over the DiSEqC bus

In SiLabs_API_SAT_prepare_diseqc_sequence:
Adding Si2164/Si2167B/Si2169
In SiLabs_API_SAT_trigger_diseqc_sequence:
Adding Si2164/Si2167B/Si2169

6.2.65 As from V2.1.6

Adding SiLabs_API_SAT_prepare_diseqc_sequence and SiLabs_API_SAT_trigger_diseqc_sequence, to allow preparing the DiSEqC message and sending it in two steps. This is required for Unicable with some demodulator (such as Si2167A), as otherwise the preparation takes too much time to stay within the Unicable Td specification (4 to 22 ms).
In SiLabs_API_Demod_status:
Added comments to differentiate the various status blocks
In SiLabs_API_SW_Init:
Adding initialization of two new functions for Unicable:
SiLabs_API_SAT_prepare_diseqc_sequence
SiLabs_API_SAT_trigger_diseqc_sequence
In SiLabs_API_SAT_voltage_and_tone:
tracing lnb_chip value

In console code:
In Silabs_UserInput_data_slice_id:
Comments improved (removed 'plp id')
In Silabs_UserInput_VoltageTone:
Maintaining consistency between Unicable and normal mode settings (for polarization and band)
In Silabs_UserInput_SeekNext:
'bell' on a lock only if traces are activated
In Silabs_demoloop:
Better Linux compatibility (using "%s" in printf)
BW forced to 8MHz for 'blindlock'
'kbit' not possible if not on WIN32

6.2.66 As from V2.1.5

In Custom_constelCode and Silabs_Constel_Text: adding QAM1024 and QAM4096 (for DVB C2)
In SiLabs_API_Demod_status:
More DVB-C2 statuses

6.2.67 As from V2.1.4

In SiLabs_API_Demod_status:
Si2164: first statuses for DVB-C2 added
Setting BER, PER for Si2185

6.2.68 As from V2.1.3

Compatibility with Si2191
In Silabs_UserInput_demod:
Compatibility with Si2185
In Silabs_API_Test:
Init of num_data_slice, to avoid compilation warning when not used
Adding initial version of DVB-T2 signalling code

6.2.69 As from V2.1.2

Compatibility with Si2164:

data_slice_id added as a parameter for 'lock_to_carrier'
*num_data_slice' addes as a parameter for 'Seek_Next'
Adding SiLabs_API_SSI_SQI function (for S/S2 reception only)
In SiLabs_API_Demod_status:
Correcting status->uncorrs for all API controlled demodulators: '(uncor_msb<<8) + uncor_lsb' instead of '(uncor_msb<<16) + uncor_lsb'
In SiLabs_API_SAT_Unicable_Tune:
Removing one printf
In SiLabs_API_lock_to_carrier:
Correcting voltage levels for SAT polarization selection:
13V is for 'Vertical', 18V is for 'Horizontal'
Removing copy of front_end polarization and band values to Unicable structure
(these may use different values)
In SiLabs_API_Channel_Seek_Next:
Removing Unicable polarization and band setup (the reference valeus are those in the front_end structure)
In SiLabs_API_SAT_voltage_and_tone:
Correcting voltage levels for SAT polarization selection:
13V is for 'Vertical', 18V is for 'Horizontal'
In Silabs_API_Test:
Initializing all variables to avoid warnings when not used
Adding access to test pipe for Si2165D
Adding sat_scan_unicable option

Console code:

Adding Silabs_UserInput_data_slice_id (for DVB-C2)
Adding Silabs_UserInput_Position function (for Unicable SAT)
Moving input functions before Silabs_UserInput_SeekInit (as they are used by SeekInit)
In Silabs_UserInput_plp_id:
dvbt_t2_plp_id renamed as plp_id (valid for T2 and C2)
In Silabs_UserInput_Lock:
Adding data_slice_id for C2
Asking for polarization and band only for SAT
In Silabs_UserInput_SeekInit:
Asking for SAT voltage and tone for SAT standards
In Silabs_demoloop:
Adding 'monitor' option
Adding 'cell_id' option
Adding 'T+' and 'T-' options (for Unicable SAT)
Setting Unicable values according to the user selection
For 'blindlock' option:
displaying execution time
In Silabs_UserInput_SeekNext:
Adding num_data_slice for C2

6.2.70 As from V2.1.1

SiLabs_API_L3_Wrapper.c
In SiLabs_API_Demod_status:
Setting status->cell_id by default at 0.
Updating status->cell_id for Si2165, Si2167 and Si2167B
In SiLabs_API_lock_to_carrier:
Tracing input parameters with the corresponding names

SiLabs_API_L3_Console.c
Adding Silabs_UserInput_demod, to allow several demods in a single application
In SiLabs_Scan_Table_Carrier_Text:
Tracing Ku frequencies for SAT
In main:
Calling Silabs_UserInput_demod to select demod

6.2.71 As from V2.1.0

In SiLabs_API_Demod_status:
Calling SAT_TUNER_RSSI_FROM_IFAGC if it exists
In SiLabs_API_SAT_Tuner_status:
Removed duplicate call to SiLabs_API_Tuner_I2C_Enable

6.2.72 As from V2.0.9

In SiLabs_I2C_UserInput_read & SiLabs_I2C_UserInput_write:
prototypes changed to 'void', to avoid shadowing the global 'i2c'.
In Silabs_UserInput_bw_Hz:
Converting bandwidth_MHz (float) to bandwidth_Hz (int) for comparing user entry with possible values.
('==' operator does not work well on float values, and may provide various results depending on the platform)

In SiLabs_API_L3_Wrapper.c:
In SiLabs_API_Demod_status:
For Si2165D:
status->spectral_inversion = Si2165_L1_DVB_T_get_spectral_inversion (front_end->Si2165_FE->demod);
For Si2167:
status->spectral_inversion = Si2167_L1_DVB_T_get_spectral_inversion (front_end->Si2167_FE->demod);
For Si2169:
removing duplicate status->num_plp = ... line
In SiLabs_API_Text_status:
Comparing 'float' ratios with int values using (int) cast
In SiLabs_API_Channel_Seek_Next:
num_plp = 0; (the previous code, without the '', 'erased' the pointer...)
In Silabs_API_Test:
Adding easy access to VDAPPS functions (for internal use)

In SiLabs_API_L3_Wrapper.h:
Removing commas on last lines of type declarations, to avoid some ISO-C compiler warnings

6.2.73 As from V2.0.8

In SiLabs_API_L3_Console.c:
In Silabs_UserInput_SeekNext:
DVB-T: Storing 2 carriers in table when locked on a Hierarchical DVB-T signal
(similar to what is done on T2 signals for MPLP: storing one carrier per 'DATA' plp_id)
MPLP management changed for C2 compatibility (which also has MPLP capability).
The criterion is now 'num_plp>1', to allow DVB-T2/C2 compatibility

in SiLabs_API_L3_Wrapper.c:

Adding SiLabs_API_Get_DVBT_Hierarchy function, to retrieve the hiearchy information from the wrapper
In SiLabs_API_TER_Tuner_status & SiLabs_API_SAT_Tuner_status:
Moving lines for compatibility with VisualStudio (all variables need to be declared before any one is used).
Without this it can be quite complex to debug, as the compiler message is not really explicit.
In SiLabs_API_bytes_trace:
Corrected when trackWrite flag was sent twice, instead of setting trackWrite & trackRead.
In SiLabs_API_SAT_voltage_and_tone:
Tracing voltage and tone values

In SiLabs_API_L3_Wrapper.h:
int SiLabs_API_Get_DVBT_Hierarchy (SILABS_FE_Context *front_end, int *hierarchy);

6.2.74 As from V2.0.7

Compatibility with LNBH29
In SiLabs_API_SW_Init:
Using a compilation flag to set the LNBH controller chip address if not defined at project level.
It is written to allow the LNBH_I2C_ADDRESS flag to be defined at project level.
If not defined at project level, it defaults to '#define LNBH_I2C_ADDRESS 0x10'
In SiLabs_API_SAT_voltage_and_tone:
Displaying 'in Unicable Mode' trace only when in Unicable mode.
In SiLabs_API_Channel_Seek_Init: improved function comments

6.2.75 As from V2.0.6

In SiLabs_API_Demod_status:
For Si2165D:
status->IFagc = Si2165_L1_Demod_get_IFagc (front_end->Si2165_FE->demod);
(previously returning aci_agc_cmd)
For Si2169:
Calling Si2169_L1_DVBT2_TX_ID
status->cell_id = front_end->Si2169_FE->demod->rsp->dvbt2_tx_id.cell_id;
In console code:
Adding SiLabs_API_Tune function, to allow DVB-C blindlock
In Silabs_demoloop:
Adding ‘tune ‘option
Adding ‘blindlock’ option

6.2.76 As from V2.0.5

LNBH init correction:
In SiLabs_API_SW_Init, front_end->lnb_chip_init_done = 0; to force the flag at '0'.
(Some compilers may set it randomly, so it needs to be forced to '0' for compatibility reasons.)

6.2.77 As from V2.0.4

ADDED FEATURE: Added SiLabs_API_Channel_Lock_Abort function, to allow aborting a call to SiLabs_API_lock_to_carrier.
In SiLabs_API_Demod_status:
Added Si2167 compatibility with TER tuners not from SiLabs

6.2.78 As from V2.0.3

In SiLabs_API_TS_Mode:
For Si2167B and Si2169: settings ts parallel clock and data shape to 7 for GPIF mode, and back to 2 for parallel mode

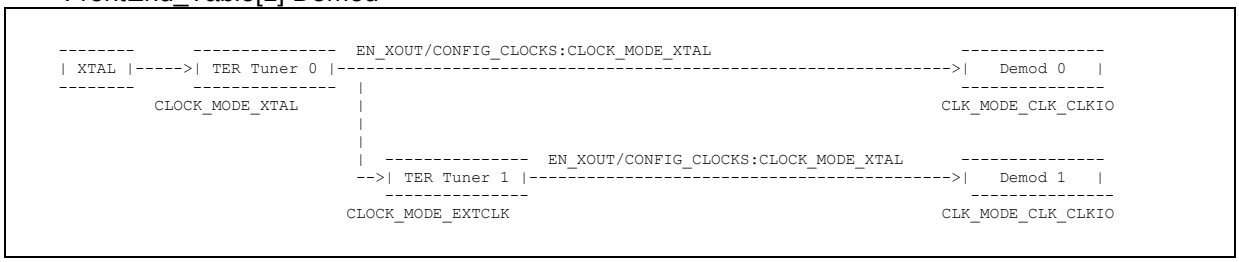
6.2.79 As from V2.0.2

In SiLabs_API_Select_PLP:
For Si2169: if (plp_mode == Si2169_DVBT2_PLP_SELECT_CMD_PLP_ID_SEL_MODE_AUTO) { Si2169_L1_DD_RESTART(front_end->Si2169_FE->demod); system_wait(300); }

6.2.80 As from V2.0.1

Added lnb_chip_init_done in SILABS_FE_Context
In SiLabs_API_SAT_voltage_and_tone:
if (front_end->lnb_chip_init_done == 0) { front_end->lnb_chip_init_done = L1_LNBH25_InitAfterReset(front_end->lnbh25); }
This is because the LNBH25 requires an init of all registers

main function compatible with dual TER front-end applications where a single xtal is shared between 2 front-end:
XTAL on FrontEnd_Table[0] TER tuner, provided to:
FrontEnd_Table[0] Demod
FrontEnd_Table[1] TER tuner
clock from FrontEnd_Table[1] TER tuner, provided to:
FrontEnd_Table[1] Demod



This is provided as an example for Si2158:

```
if (fe==0) {
  front_end->Si216x_FE->tuner_ter->cmd->power_up.clock_mode      =
Si2158_POWER_UP_CMD_CLOCK_MODE_XTAL;
  front_end->Si216x_FE->tuner_ter->cmd->power_up.en_xout         =
Si2158_POWER_UP_CMD_EN_XOUT_EN_XOUT;
  front_end->Si216x_FE->tuner_ter->cmd->config_clocks.clock_mode =
Si2158_CONFIG_CLOCKS_CMD_CLOCK_MODE_XTAL;
}
if (fe==1) {
  front_end->Si216x_FE->tuner_ter->cmd->power_up.clock_mode      =
Si2158_POWER_UP_CMD_CLOCK_MODE_EXTCLK;
  front_end->Si216x_FE->tuner_ter->cmd->power_up.en_xout         =
Si2158_POWER_UP_CMD_EN_XOUT_EN_XOUT;
  front_end->Si216x_FE->tuner_ter->cmd->config_clocks.clock_mode =
Si2158_CONFIG_CLOCKS_CMD_CLOCK_MODE_EXTCLK;
}
```

It also requires adaptations in the corresponding Si21x8_L2_API.c/Si21x8_PowerUpWithPatch function:
The clock_mode and en_xout values needs to be set based on the above values:
(example for Si2158):

```
if ((return_code = Si2158_L1_POWER_UP (api,
                    Si2158_POWER_UP_CMD_SUBCODE_CODE,
```

api->cmd->power_up.clock_mode,
api->cmd->power_up.en_xout,

6.2.81 As from V2.0.0

In SiLabs_API_Text_status:
sprintf(formatted_status, "%s TS bitrate %d kbps\n", formatted_status, status->TS_bitrate_kHz);
sprintf(formatted_status, "%s TS clock %d kHz\n" , formatted_status, status->TS_clock_kHz);
SiLabs_API_Reset_Uncorrs compatibility with Si2165
SiLabs_API_Demod_reset compatibility with Si2165 and Si2167
In console code:
Defining FRONT_END_COUNT if not defined at project level.
This allows setting FRONT_END_COUNT at project level only for multi front-end applications

6.2.82 As from V1.9.9

Adding NO_SAT tags to allow using Si2169 code without SAT features

6.2.83 As from V1.9.8

Compatibility with TER_TUNER_Si2190
Compatibility with TER_TUNER_CUSTOMTER
In SiLabs_API_SAT_AutoDetectCheck:
Adapting Si2169 code to return the current SAT standard when locked, 0 otherwise.
In Silabs_API_Test:
adding wrapper/sat_auto_detect option
Compatibility with SAT_TUNER_RDA5816S
In console code:
last_plp_id stored during init as '-1' to indicate auto mode

6.2.84 As from V1.9.7

Adding LNB control in the API, to allow driving LNBH25 or LNBH21 easily
In SiLabs_API_Demod_status: If Si2169, setting plp_id based on rsp.dvbt2_status.plp_id
Handling MCNS in Silabs_UserInput_qam, Silabs_UserInput_standard, Silabs_UserInput_Lock
Removing TER_TUNER_MENU from Silabs_menu
Removing TER_TUNER_LOOP from Silabs_demoloop

6.2.85 As from V1.9.6

Passing pointer to LNB function when calling SiLabs_Unicable_API_Init, following the new definition of SiLabs_Unicable_API_Init
In main:
setting i2c to L0_FastI2C(); by default

6.2.86 As from V1.9.5

Adding SILABS_MCNS, SILABS_DVB-C2 and SILABS_SLEEP possibilities
Adding MCNS statusing
In SiLabs_API_Demod_status: no demod status in SLEEP mode

6.2.87 As from V1.9.4

In SiLabs_API_SAT_Get_AGC:
Corrected value returned for Si2169 SAT AGC
In Silabs_API_Test:
Added Unicable test pipe access

6.2.88 As from V1.9.3

adding TS_bitrate_kHz and TS_clock_kHz in demod status

6.2.89 As from V1.9.2

Si2146 ATV and DTV STATUS removed (not in the Si2146 API anymore)

6.2.90 As from V1.9.1

added/moved tags to allow Si2166B export
In SiLabs_API_Demod_status:
setting SSi and SQI at 0 by default.
compatibility with rssi from CUSTOMTER and CUSTOMSAT tuners

6.2.91 As from V1.9.0

In SiLabs_API_SAT_Tuner_status: moving tags to allow export for Si2168
Wrapper code compatible with Si2167B: checked to be able to lock a Si2169 board when using the Si2167B code with the proper FW.
In Silabs_demoloop:
Removing duplicate SATauto options

6.2.92 As from V1.8.9

Tracing Wrapper source code info during init and in SiLabs_API_Infos
SiLabs_API_SatAutoDetectCheck renamed as SiLabs_API_SAT_AutoDetectCheck for consistency
Adding Test Pipe feature (only if SILABS_API_TEST_PIPE is defined at project level), using new Silabs_API_Test function
Adding PLP management (for DVB_T2 only).
In SiLabs_API_Demod_status:
updating spectral_inversion for Si2169 in DVB-T and DBVB-T2
In SiLabs_API_TS_Mode:
Stopping GPIF clock if using the Cypress USB interface and not using GPIF mode
In SiLabs_API_Demod_status and SiLabs_API_Text_status:
Not storing current standard as front_end->standard, to avoid creating problems with standard switching.
Using status->standard in all switches.
In SiLabs_API_TER_Tuner_status and SiLabs_API_SAT_Tuner_status:
Enabling i2c passthru before statusing tuners

In console code:
Treating symbol_rate as unsigned int (for proper display of high SAT SR when scanning)

Adding Test Pipe feature (only if SILABS_API_TEST_PIPE is defined at project level), using new Silabs_UserInput_Test function
In Silabs_UserInput_SeekNext:
Checking num_plp if locked in DVB-T2, check PLP infos for each PLP ID and store one 'channel' per PLP ID
In SiLabs_Scan_Loop:
Adding scan duration display
Adding console options:
AutoSAT : SiLabs_API_SAT_AutoDetect(front_end, 1)
NoAutoSAT : SiLabs_API_SAT_AutoDetect(front_end, 0)
autocheck : Display auto detect settings
install : SiLabs_API_SAT_Unicable_Install
positionA : unicable->satellite_position = UNICABLE_POSITION_A
positionB : unicable->satellite_position = UNICABLE_POSITION_B

6.2.93 As from V1.8.8

In SiLabs_API_Demod_status:
setting more statuses by default to indicate a no-lock:
status->c_n = 0;

status->freq_offset = 0;
status->timing_offset = 0;
status->code_rate = -1;
status->SSI = 0;
status->SQI = 0;

For Si2169: returning '0' immediately in case a standard-specific status returns with an error.
In Silabs_demoloop:
Adding 'ATV', 'up' and 'down' options
Adding functions to manage easily 'up' and 'down' zapping.

6.2.94 As from V1.8.7

Compatibility with Si2167B (coming soon)

6.2.95 As from V1.8.6

Compatibility with Si2148/Si2158
Compatibility with export for non 'SATELLITE_FRONT_END'

6.2.96 As from V1.8.5

Adding auto-detect functions:
int SiLabs_API_SAT_AutoDetect (SILABS_FE_Context *front_end, int on_off);
int SiLabs_API_TER_AutoDetect (SILABS_FE_Context *front_end, int on_off);
In Silabs_UserInput_SeekInit: set default bw as 8MHz (for DVB-C)
In Silabs_UserInput_SeekNext: tracing detected standard as well, to test auto-detect
In SiLabs_Scan_Table_Carrier_Text: tracing freq un Ku band if Unicable

Adding console options:
AutoTER : SiLabs_API_TER_AutoDetect(front_end, 1)
NoAutoTER : SiLabs_API_TER_AutoDetect(front_end, 0)
autocheck : Display auto detect settings
voltage_and_tone
AutoSAT : SiLabs_API_SAT_AutoDetect(front_end, 1)
NoAutoSAT : SiLabs_API_SAT_AutoDetect(front_end, 0)
install : SiLabs_API_SAT_Unicable_Install
horizontal: unicable->polarization = SILABS_POLARIZATION_HORIZONTAL
vertical : unicable->polarization = SILABS_POLARIZATION_VERTICAL
low : unicable->band = UNICABLE_LOW_BAND
high : unicable->band = UNICABLE_HIGH_BAND
diseqc : tracing Diseqc data (toggle)
hardtune : selecting Unicable values
unicable : SiLabs_Unicable_API_Tune_Infos
tones : SiLabs_Unicable_API_All_Tones
tones_off : SiLabs_Unicable_API_All_Off

6.2.97 As from V1.8.4

In SiLabs_API_SAT_voltage_and_tone: disegBuffer value correction
Adding UNICABLE functions (compiled if #define UNICABLE_COMPATIBLE).

6.2.98 As from V1.8.3

Compatibility with Si2178

6.2.99 As from V1.8.2

In SiLabs_API_Demod_status:
setting ber, fer and per by default at '-1' to indicate unavailability if not set later on.
In SiLabs_API_Text_status:

ber and per displayed as '--------' when not available

6.2.100 As from V1.8.1

In SiLabs_API_Demod_status:
(Si2169) Comments correction indicating that the rate checks are done on the exponent
Comments correction indicating that the rate checks are done on the exponent
In SiLabs_API_Text_status:
spectral inversion added to text status

6.2.101 As from V1.8.0

Compatibility with DTT759x (Terrestrial can tuner)
Checking exponent for rate in SiLabs_API_Demod_status, to return -1 if not available

6.2.102 As from V1.7.9

voltage_and_tone working with Si2167
In Silabs_demoloop:
Adding 'slow' and 'fast' options to allow easy testing of several i2c speeds

6.2.103 As from V1.7.8

voltage_and_tone working with LNBH21
SAT and DVB-C blindscan working for Si2169
compatibility with NO_TER 'dummy' TER tuner (for lab use)

6.2.104 As from V1.7.7

BER monitored for Si2169 in DVB-T2 and DVB-S2 as well as for all DTV standards (previously not in FW so it was skipped)

6.2.105 As from V1.7.6

2 lines added to allow exporting for demods with no 'STANDBY' or 'CLOCK_ON' feature
TERRESTRIAL_FRONT_END tag replacing DEMOD_DVB_T to allow exporting for Si2163/Si2113
Some SiTRACE calls surrounded by curly brackets to allow compiling without SITRACES

6.2.106 As from V1.7.5

Silabs_UserInput_qam changed to take text values for constellations

6.2.107 As from V1.7.3

some lines moved for greater compatibility with Visual Studio

6.2.108 As from V1.7.2

Si2169 agc values retrieved in SiLabs_API_Demod_status

6.2.109 As from V1.7.0

adding WrapperI2C context to allow easy i2c read/write
added SiLabs_API_ReadString/SiLabs_API_WriteString functions
For SAT: added voltage/tone and DiSEqC functions

6.2.110 As from V1.6.9

For Si2169: status->stream based on demod->prop->dvbt_hierarchy.stream;

6.2.111 As from V1.6.7

Compatibility with NXP20142 SAT tuner
API change: using Si2169 DD_SSI_SQI instead of Si2169_DVBT_SSI_SQI (also available in DVB-T2)

6.2.112 As from V1.6.6

Adding missing BER status for Si2169

6.2.113 As from V1.6.5

Using SATELLITE_FRONT_END and TERRESTRIAL_FRONT_END compilation flags, as it makes it easier to handle C-only or T-only exports

6.2.114 As from V1.6.3

compatibility with SAT-only exports in Silabs_UserInput_bw_Hz

6.2.115 As from V1.6.1

Compatibility with TER tuner cans (not using API mode)
SiLabs_API_TER_Tuner_ATV_Tune compatible with Si2165

6.2.116 As from V1.6.0

Added Si2185 support
In SiLabs_API_Demod_status:
Set to 0 all info used to relock (bandwidth_Hz, symbol_rate, stream, constellation)
In SiLabs_API_switch_to_standard:
For Si2169: Checking dd_status.modulation if switch_to_standard fails
In SiLabs_API_lock_to_carrier:
Returning 0 if switch_to_standard fails

6.2.117 As from V1.5.6

Silabs_UserInput_SeekInit uses 8000000 as default seekBWHz

6.2.118 As from V1.5.1

power_of_n corrected to return the proper value

6.3 *Errata*

None