

AVL68XX

Software Porting Guide

v2.0



www.Availink.c

Availink Inc.
Scotia Centre
P. O. Box 268GT
George Town, Grand Cayman,
Cayman Islands

Availink (China) Inc.
Xueyuan International Tower, Ste
1608
No. 1 Zhichun Road
Haidian District, Beijing, 100083
P. R. China

Availink (US) Inc.
20201 Century Blvd., STE 160
Germantown MD 20874 USA

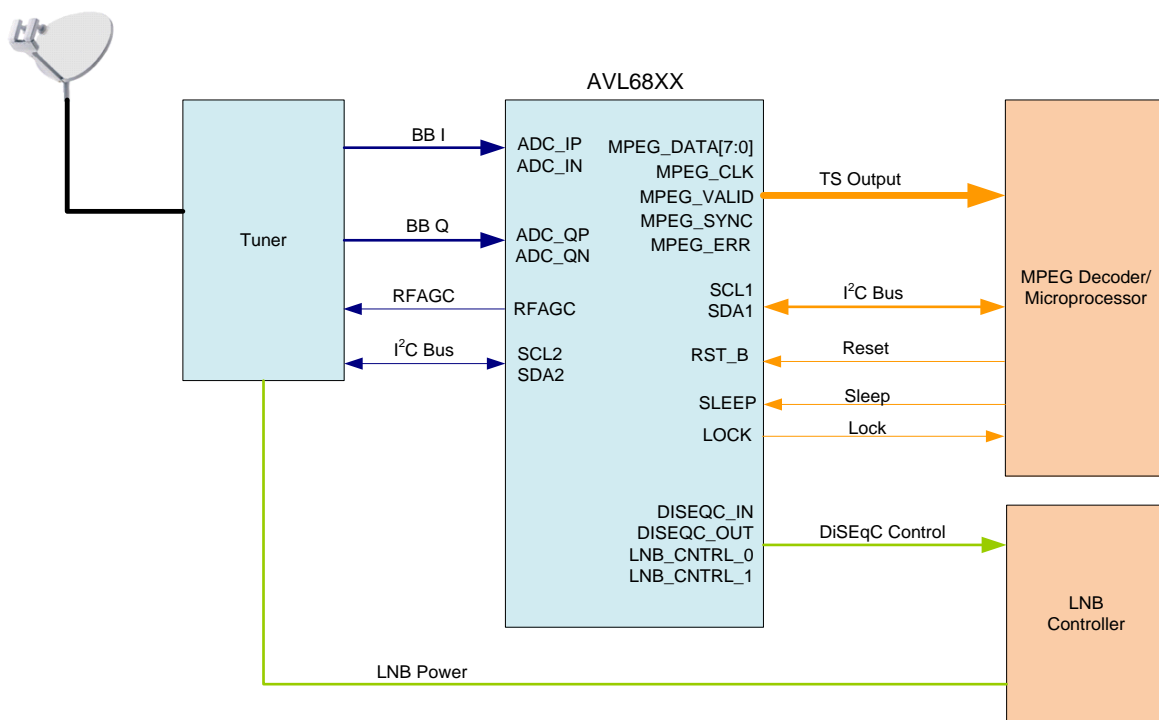


Table of Contents

1. Overview.....	3
2. DEMOD Work Flow	4
2.1 Demod work flow.....	4
2.2 SDK Lock Function Interface	5
2.3 DVBT2 ChannelScan AND DVBSx BlindScan Work Flow	6
3. Porting Method.....	8
3.1 BSP Layer	8
3.2 Application Layer	11
4. Q&A.....	15
Appendix I. Function Description in Example Program	17

1. Overview

The document is only making a simple description about AVL68XX SDK porting according to “Example” program. AVL68XX is a series of multi-mode supported (DVBC/DVBTX/ ISDBT/DVBSX) of digital demodulation chips.



According to the above diagram shows, the AVL68XX demodulation chip receives the IQ baseband signal generated by the tuner. IQ baseband signal as input signal and demodulated by the demodulation chip, then the chip outputs standard MPEG TS (Transport stream) from the demodulation.

2. DEMOD Work Flow

2.1 Demod work flow

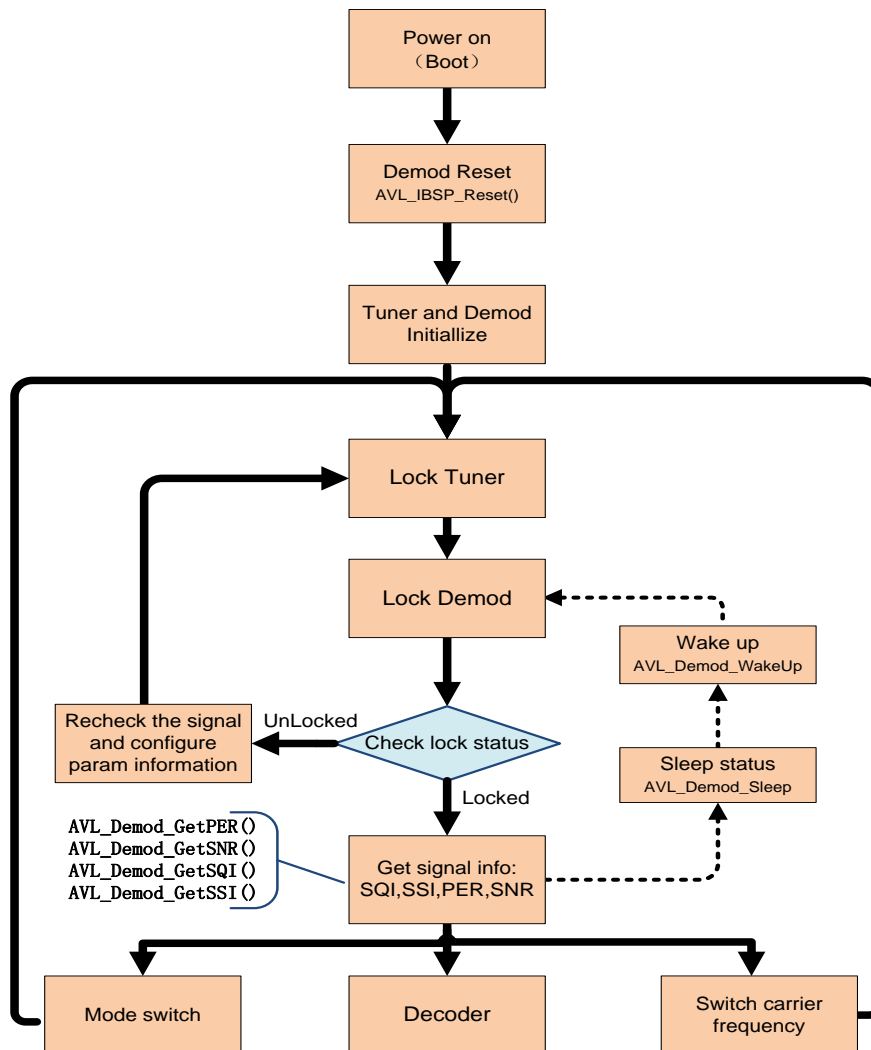


Figure 2-1 Demod Work Flow



2.2 SDK Lock Function Interface

For “lock demod” module in figure2-1, SDK interface called as follows:

- *DVBC Mode :*
AVL_Demod_DVBCAutoLock(AVL_uint32 uiChipNo);
- *ISDBT Mode :*
AVL_Demod_ISDBTAutoLock(AVL_uint32 uiChipNo);
- *DVBTX Mode mainly includes “signal lock ” and “Channel scan” two functions:*

Signal lock function:

```
AVL_Demod_DVBTAutoLock(AVL_DVBTxBandWidth  
    eBandWidth,AVL_DVBTX_PROFILE eDVB2Profile,  
    AVL_uchar ucDVB2PLPID, AVL_uint32 uiChipNo);  
AVL_Demod_DVBTAutoLock(AVL_DVBTxBandWidth eBandWidth,  
    AVL_uchar ucDVBTLayer, AVL_uint32 uiChipNo);
```

ChannelScan function:

Please according to figure 2-2 to understand “ChannelScan” function, for the detailed usage , please referenced to function “AVL_DVBTxChannelScan_example()” located in file “AVL68XX_PortingExample.c” .

- *DVBSX Mode mainly includes “signal lock ” and “Channel scan” two functions:*

Signal lock function:

```
AVL_Demod_DVBSxAutoLock(AVL_uint32 uiSymbolRateSps, AVL_uint32 uiChipNo);
```

BlindScan function:

Please according to figure 2-3 to understand “BlindScan” function, for the detailed usage , please referenced to function “AVL_BlindScanProcess()” located in file “AVL68XX_PortingExample.c” .

2.3 DVBT2 ChannelScan AND DVBSx BlindScan Work Flow

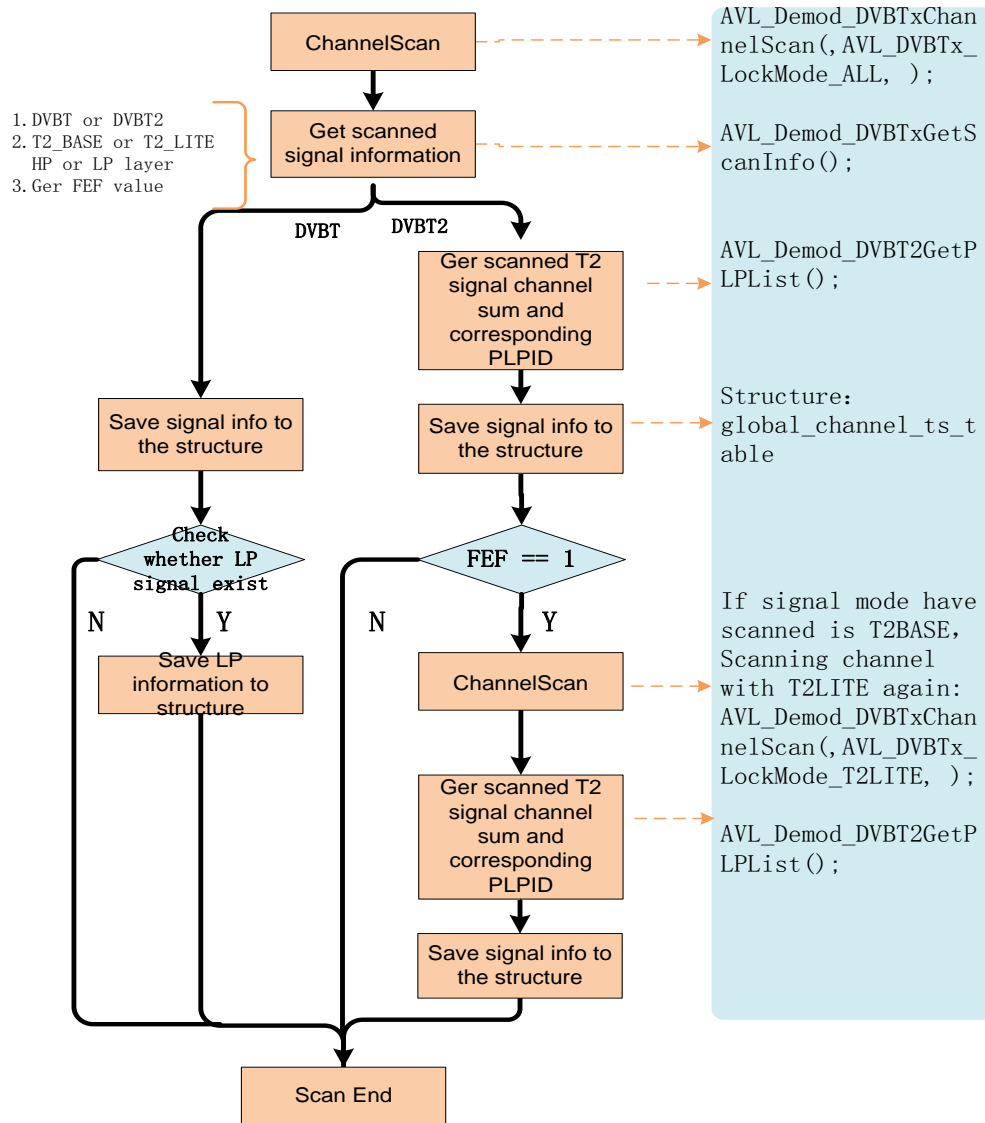


Figure 2-2 DVBTx ChannelScan Work Flow

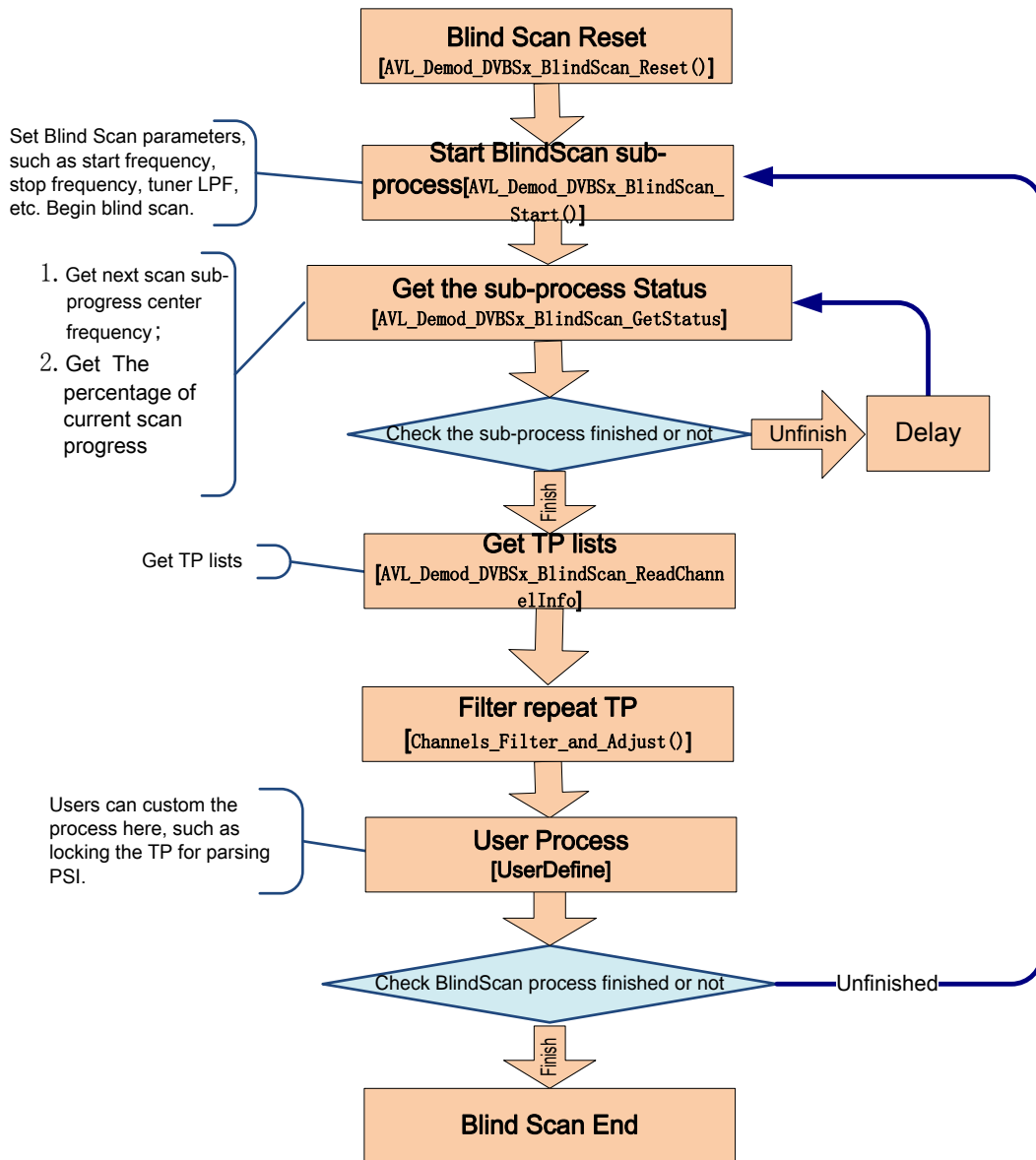


Figure 2-3 DVBSX BlindScan Work Flow

3. Porting Method

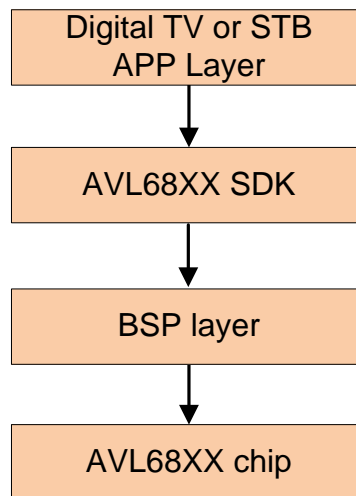


Figure 3-1 software layers

As shown in figure 3-1, on the software, the main work is to modify the digital television receiver application layer and BSP layer.

3.1 *BSP Layer*

Firstly, BSP layer is composed of three files *user_defined_data_type.h*, *user_defined_function.c* and *user_defined_function.h*.

The files storage path: *AVL68XX_SDK\AVL68XX_SDK\sdk_src*.

- *user_defined_data_type.h*

Redefine the data type in the “*bspdatadef.h*” according to the platform you are using.



- user_defined_function.c

Realize the function in the template folder.

Delay function

AVL_IBSP_Delay (AVL_uint32 uiDelay_ms)

Notes: Unit is milliseconds.

I2C Write/ Read function

AVL_IBSP_I2C_Write (AVL_uint16 uiSlaveAddr, AVL_puchar pucBuff, AVL_puint16 puiSize)

uiSlaveAddr: Device address;

pucBuff: A Pointer to a buffer which includes the data to be written;

Notes: including the register address to be written and date in the buffer.

puiSize: The byte number we need to write. Notes: excluding the I2C address.

AVL_IBSP_I2C_Read (AVL_uint16 uiSlaveAddr, AVL_puchar pucBuff, AVL_puint16 puiSize)

uiSlaveAddr: Device address;

pucBuff: A pointer to a buffer in which to store the read data;

Notes: The function only perform a direct I2C read operation without first writing the register address. If you want to operate a special register, you should write the regsiteer address by I2C write function prior to performing the read operation.

puiSize: The byte number we need to read.

Notes: the function default supports **a 7 bits I2C address**, if you are using the platform address for 8 bits, the address must be shifted one bit to the left in the function.

I2C communication protocol :

A - Acknowledge(SDA LOW)

N - Not Acknowledge(SDK HIGH)



Sr - Repeated Start Condition

S - START Condition

P - STOP Condition

R – R/W bit, 1 = Read

AVL_IDSP_I2C_Write

S	ucSlaveAddr	R/ Wb	A	pucBuff[0]	A	pucBuff[1]	A	...	pucBuff[*puiSize-1]	P
	7 bits	0	1-bit	8 bits	1-bit	8 bits	1-bit		8 bits	

AVL_IDSP_I2C_Read

S	ucSlaveAddr	R/ Wb	A	pucBuff[0]	A	pucBuff[1]	A	...	pucBuff[*puiSize-1]	N	P
	7 bits	1	1-bit	8 bits	1-bit	8 bits	1-bit		8 bits		

BSP Initialize

AVL_IBSP_Initialize(void)

The function should be realized by user according to the platform. It can be set to empty function and return directly if user don't need.

Initialize Semaphore

AVL_IBSP_InitSemaphore (AVL_psemaphore pSemaphore)

Initializes a semaphore object. It can be realized by mutex.

Wait Semaphore

AVL_IBSP_WaitSemaphore (AVL_psemaphore pSemaphore)

The semaphore query function. If the semaphore is held by others, then the function should be blocked until the semaphore is available. It can be realized by mutex.



Release semaphore

AVL_IBSP_ReleaseSemaphore (AVL_psemaphore pSemaphore)

This function releases the semaphore. It can be realized by mutex.

Notes: Currently, including 3 semaphore in the AVL68XX SDK, therefore the system should assign at 3 semaphores at least.

- **user_defined_function.h**
It is the header file of file "*user_defined_function.c*", you don't need update it if you not change the interface function in file "*user_defined_function.c*".

3.2 Application Layer

The porting of application layer function mainly includes;

1. Tuner parameters configuration;
2. Demod parameters configuration;
3. Tuner lock;
4. Demod lock;
5. Get lock status;
6. Get signal information, such as PER, SNR, SSI, SQI;
7. Signal mode switch.

● Tuner parameters configuration

Tuner parameters configuration is in the structure of "AVL_Tuner", the structure is mainly include the following parameters:

- 1) **usTunerI2CAddr**: tuner I2C address, please refer to the Tuner datasheet ;
- 2) **ucTunerLocked**: if Tuner PLL configure is ok, the value is 1, vice-vers;
- 3) **eDTVMMode**: enum type, indicates the signal mode you are using;
- 4) **uiRFFrequencyHz**: the RF frequency of input to tuner. The unit is Hz;
- 5) **uiIFHz**: the frequency outputting from Tuner, The unit is Hz.

Notes: ① the parameter is only valid for terrestrial /cable tuner



② the value of this parameter should keep same with the IF value located in file "AVL_Demod_config.c";

For example , when signal mode is DVBT/DVBT2, configuring uilFHz equal to 36000000Hz, the IF value should also configure to 36000000Hz located in the structure of "AVL_DVBTxConfigChip0" in file "AVL_Demod_config.c";

- 6) uiBandwidthHz: the signal bandwidth outputting from Tuner. The unit is Hz;
Notes: the parameter is only valid for terrestrial /cable tuner;
- 7) uiLPFHz: cut-off frequency of LPF, the unit is Hz;
Notes: the parameter is only valid for satellite tuner;
- 8) vpMorePara: usually, setting it to NULL;
- 9) fpInitializeFunc、fpLockFunc、fpGetLockStatusFunc、fpGetRFStrength: the function pointer about the tuner operation.

● Demod parameters configuration

Demod parameters configuration is located in file "AVL_Demod_Config.c"

The common configuration is in structure named as "Avl_CommonConfig", the structure is mainly include the following parameters:

- 1) usI2CAddr: Demod I2C address, depends on the voltage state of AVL68XX chip of CS_0 pin (pin 15), when pin15 voltage is 0V, the demod I2C address is 0x0C; when pin15 voltage is 3.3V, the demod I2C address is 0x0D.
Notes: the address is 7 bit.
- 2) eDemodXtal: Enum-type, indicates the oscillator frequency the demod using;
- 3) eTSMODE: Enum-type, you can configure the TS output mode to serial or parallel;
- 4) eClockEdge: Enum-type, you can configure to "Rising" or "Falling";
- 5) eClockMode: Enum-type, indicates whether configure TS CLK to "continuous" mode or not. When configure to "continuous" mode, TS CLK frequency will keep continuous output and CLK frequency is not change when changed the signal mode of configuration.



The paramaters configuration related with DVBTx mode is in structure named as "AVL_DVBTxConfig", the structure is mainly include the following parameters:

- 1) eDVBTxInputPath: Enum-type, indicates the signal input to Demod by I path or Q path;
- 2) uiDVBTxIFFreqHz: IF frequency inputting to Demod;
- 3) eDVBTxAGCPola: Enum-type, you can configure AGC polarity to "NORMAL" or "INVERTED". When set to "NORMAL", the Tuner output signal will increases with the increase of the AGC.

The paramaters configuration related with DVBSx mode is in structure named as "AVL_DVBSxConfig", the structure is mainly include the following parameters:

- 1) eDVBSxAGCPola: Enum-type, you can configure AGC polarity to "NORMAL" or "INVERTED";
- 2) e22KWaveForm: Enum-type, you can configure 22K wave form to "Normal" or "Envelope".

The paramaters configuration related with ISDBT mode is in structure named as "AVL_ISDBTConfig", the structure is mainly include the following parameters:

- 1) eISDBTInputPath: Enum-type, indicates the signal input to Demod by I path or Q path;
- 2) eISDBTBandwidth: Enum-type, indicates the signal bandwidth;
- 3) uiISDBTIFFreqHz: IF frequency inputting to Demod;
- 4) eISDBTAGCPola: Enum-type, indicates AGC polarity("normal" or "inverted")

The paramaters configuration related with DVBC mode is in structure named as "AVL_DVBCCConfig", the structure is mainly include the following parameters:

- 1) eDVBCInputPath: Enum-type, indicates the signal input to demod by I path or Q path;
- 2) uiDVBCIFFreqHz: IF frequency inputting to demod;



- 3) uiDVBCSymbolRateSps: DVBC symbol rate;
- 4) eDVBCAGCPola: Enum-type, you can configure AGC polarity to "NORMAL" or "INVERTED";
- 5) eDVBCStandard: Enum-type, DVBC standard: J83A or j83b.

● Tuner Lock

We will provide the tuner driver, usually mainly include three interface function: tuner initialize; tuner lock; get tuner lock status. In our SDK, we call those functions by function pointer.

Notes: if you need to change another tuner or upgrade Tuner driver, please contact us.

● Demod Lock

You can call function "AVL_Demod_SetMode()" to switch signal mode, then locking the signal by calling appropriate function. For example, when switch signal mode to DVBT, you can lock the signal by calling function "AVL_Demod_DVBTAutoLock()".

● Get Demod Lock Status

You can get demod lock status by calling function "AVL_Demod_GetLockStatus()".

● Get Signal Information

You can get signal PER, SQI, SSI, SNR by calling function

"AVL_Demod_GetPER()", "AVL_Demod_GetSQI()",
"AVL_Demod_GetSSI()", "AVL_Demod_GetSNR()".

● Mode Switch

You can switch signal to other mode by calling function "AVL_Demod_SetMode()", you need call the lock function to relock the signal after mode switch.



4. Q&A

If the problems appeared in the porting process, please check following items:

1> Check if I2C communication is correct

- a) Check demod i2c address whether is correct, demod i2c address is depend on the voltage status of pin15, if AVL68xx CS_0 pin (pin 15) voltage is 0V, the address is 0X14; if AVL68xx CS_0 pin (pin 15) voltage is 3.3V, the address is 0X15.
Notes: the address is 7 bit.
- b) Getting chip family id by calling function *"GetFamilyID_Demod()"*, if i2c communication is correct , you can get the value is 0x68624955; User can also debug the issue by writing a data to a register first, then read the data from the same register and check if it is same with the value written to the register.
- c) Through the oscilloscope to measure the i2c waveform, check whether the reading and writing i2c waveform is correct; check whether there is ACK signal.

2> Demod Unlock

- a) Check whether boot is ok.
- b) Check whether tuner lock is ok.
- c) Check the crystal frequency, whether crystal frequency configuration in the software is consistent with the hardware.
- d) Check parameters configuration, such as frequency, symbol rate, etc.
- e) Check signal source, signal strength, make sure the signal is good enough.

3> Demod can lock, but can't play the vedio

- a) Check Mpeg output configuration, make sure it's consistent with Decoder, including serial/parallel output, clock rising edge/falling edge sample, the output pin D0/D7, the output format TS/TSP etc.
- b) Check whether make "TS output" enable.

If confirmed that all of the above is correct, but the system still can't run normally, please contact with the Availink FAE staff.



4> I/O interface

- a) User can call function "AVL_Demod_GetGPIOValue()" to set the pin37/pin38 as input interface and get the input level value.
- b) User can call function "AVL_Demod_SetGPIOValue()" to set the pin37/pin38 as output interface and output the high/low level.

5> No signal detection feature

If signal mode is DVBTx or ISDBT, the user can call the function "AVL_Demod_DVBTxSignalDetection()" and "ISDBT_GetSignalDetection()" to check whether the signal is exist. The function can be used with "check lock status", making quit exit "check lock status" when the signal is not exist.

The detailed usage reference sample function "AVL_Check_LockStatus()" located in file "AVL68XX_PortingExample.c".



Appendix I. Function Description in Example Program

- ***A8293_Control(AVL_uint32 LNB_LEVEL)***

Function: Controlling 8293 chip make it output 13v or 18v voltage.

- ***AVL_Check_LockStatus(AVL_uchar *pLockFlag)***

Param: pLockFlag: pointer type. Lock status flag.

Function: Get lock status.

- ***DVB_Sx_tuner_Lock(AVL_uint32 Freq_Khz,AVL_uint32 Symbol_Khz)***

Param: Freq_Khz: signal carrier frequency, unit is KHz;

Symbol_Khz: signal symbol rate, unit is KHz;

Function: Tuner lock to DVBSx mode.

- ***DVB_C_tuner_Lock(AVL_uint32 Freq_Khz,AVL_uint32 BandWidth_Khz)***

Param: Freq_Khz: signal carrier frequency, unit is KHz;

BandWidth_Khz: signal bandwidth, unit is KHz;

Function: Tuner lock to DVBC mode.

- ***DVB_Tx_tuner_Lock(AVL_uint32 Freq_Khz,AVL_uint16 BandWidth_Khz)***

Param: Freq_Khz: signal carrier frequency, unit is KHz;

BandWidth_Khz: signal bandwidth, unit is KHz;

Function: Tuner lock to DVBTx mode.

- ***ISDBT_tuner_Lock(AVL_uint32 Freq_Khz,AVL_uint16 BandWidth_Khz)***



Param: Freq_Khz: signal carrier frequency, unit is KHz;
BandWidth_Khz: signal bandwidth, unit is KHz;
Function: Tuner lock to ISDBT mode.

- ***AVL_Init(void)***

Param: Initialize the demod and tuner.

- ***AVL_LockChannel_DVBSx(AVL_uint32 Freq_Khz, AVL_uint32 Symbol_Khz)***

Param: Freq_Khz: signal carrier frequency, unit is KHz;
Symbol_Khz: symbol rate, unit is KHz;
Function: Demod lock to DVBSx mode.

- ***AVL_LockChannel_ISDBT(AVL_uint32 Freq_Khz, AVL_uint16 BandWidth_Khz)***

Param: Freq_Khz: signal carrier frequency, unit is KHz;
BandWidth_Khz: signal bandwidth, unit is KHz;
Function: Demod lock to ISDBT mode.

- ***AVL_LockChannel_T(AVL_uint32 Freq_Khz, AVL_uint16 BandWidth_Khz, AVL_int32 DVBT_layer_info)***

Param: Freq_Khz: signal carrier frequency, unit is KHz;
BandWidth_Khz: signal bandwidth, unit is KHz;
DVBT_layer_info: DVBT layer information(0:LP, 1:HP).
Function: Demod lock to DVBT mode.

- ***AVL_LockChannel_T2(AVL_uint32 Freq_Khz, AVL_uint16 BandWidth_Khz, AVL_uchar T2_Profile, AVL_int32 PLP_ID)***

Param: Freq_Khz: signal carrier frequency, unit is KHz;



BandWidth_Khz: Signal bandwidth, unit is KHz;
T2_Profile: Setting DVBT2 Profile (0: Base, 1: Lite);
PLP_ID: Setting PLP ID.

Function: Demod lock to DVBT2 mode.

- ***AVL_LockChannel_DVBC(AVL_uint32 Freq_Khz, AVL_uint16 BandWidth_Khz)***

Param: Freq_Khz: signal carrier frequency, unit is KHz;
BandWidth_Khz: signal symbol rate, unit is Kbps;
Function: Demod lock to DVBC mode.

- ***Channels_Filter_and_Adjust(struct AVL_ChannellInfo *Ch_list_valid, AVL_uchar *TP_No_valid, struct AVL_ChannellInfo *Ch_list_Temp, AVL_uchar TP_No_Temp)***

Param: Ch_list_valid: structure pointer, saving the signal info have scanned;
TP_No_valid: structure pointer, saving the signal num have scanned;
Ch_list_Temp: structure pointer, saving the signal info have scanned in a scan sub-progress;
TP_No_Temp: structure pointer, saving the total number of signals have scanned in a scan sub-progress;

Function: Save the valid signal info have scanned(frequency and SR) to "AVL_ChannellInfo" structure.

- ***AVL_Blindscan_init(void)***

Function: Initialize the Blind scan. Clear the buffer saved the signal info.

- ***AVL_BlindScanProcess(AVL_uint16 centerFreq_Mhz, AVL_uint16 *pNextCenterFreq_Mhz, AVL_ChannellInfo *pChannelList, AVL_uchar *Find_TP_num, AVL_int32 uiChipNo)***



Param: centerFreq_Mhz: center frequency of the current scan Process,
unit is MHz
pnextCenterFreq_Mhz: center frequency of the next scan Process,
unit is MHz
pChannelList: structure pointer, point to a memory space used to save
the valid signal info.
Find_TP_num: the signal num of have scanned.
uiChipNo: chip number.

Function: a sub-progress of blind scan and print the signal info have scanned.

- ***BlindScanExample()***

Function: Blind Scan Example, scan range from 950MHz to 2150MHz.

- ***AVL_SX_SetToneOut(AVL_uchar ucTone)***

Function: Send "Tone burst" signal.

- ***AVL_SX_22K_Control(AVL_uchar OnOff)***

Function: Make DisEqc turn on or turn off 22KHz CLK.

- ***AVL_SX_DiseqcSendCmd(AVL_puchar pCmd, AVL_uchar CmdSize)***

Function: Send Diseqc command.

- ***DiseqcExamples(void)***

Function: DisEqc control example, including some control flow to DisEqc , such
as send Tone burst signal, turn on/off 22KHz CLK.



- ***AVL_SetWorkMode(AVL_DemodMode eDemodWorkMode)***

Parma: eDemodWorkMode: Switch to a specific signal mode.

Function: Signal mode switch.

- ***AVL_ScanChannel_Tx(AVL_uint32 Freq_Khz, AVL_uint16 BandWidth_Khz)***

Parma: Param: Freq_Khz: signal carrier frequency, unit is KHz;

BandWidth_Khz: signal bandwidth, unit is KHz;

Function: Scan channel for DVBTX signal, and save signal info to a global array named as "*global_channel_ts_table*", global variable named as "*g_nChannel_ts_total*" used to save the total of TP number you have scanned.

Notes: The function is only used to "scan" signal, if the user want to lock the signal, please re-call function "AVL_LockChannel_T2" or "AVL_LockChannel_T".

- ***get_SSI_info(void)***

Function: Get signal SSI.

- ***get_SQI_info(void)***

Function: Get signal SQI.

- ***get_SNR_info(void)***

Function: Get signal SNR.

- ***get_PER_info(void)***



Function: Get signal PER.

- ***DVB_Tx_locksignal_example(AVL_uint32 Freq_Khz,AVL_uint16 BandWidth_Khz)***

Parma: Freq_Khz: signal carrier frequency, unit is KHz;

BandWidth_Khz: signal bandwidth, unit is KHz;

Function:

Example program of DVBTX lock signal based on the global array “*global_channel_ts_table*” and the global variable “*g_nChannel_ts_total*”, the interface function “*AVL_LockChannel_T*” is used to lock to a DVB-T signal, the interface function “*AVL_LockChannel_T2*” is used to lock to a DVB-T2 signal.

- ***AVL_PrintVersion(void)***

Function: Print SDK and FW version you are using.