

AFE4432 Driver

1.0

Texas Instruments

1 Data Structure Index	1
1.1 Data Structures	1
2 File Index	3
2.1 File List	3
3 Data Structure Documentation	5
3.1 AFE_RegMap Struct Reference	5
3.2 AFE_RegMap::GLOBAL Struct Reference	5
3.3 AFE_RegMap::PPM Struct Reference	8
3.4 sigParameter Struct Reference	9
4 File Documentation	11
4.1 AFE_Functions_PPG.c File Reference	11
4.1.1 Detailed Description	13
4.1.2 Function Documentation	13
4.1.2.1 AFE_clearPPM()	13
4.1.2.2 AFE_compute_CF()	13
4.1.2.3 AFE_config_AMBCancellation()	14
4.1.2.4 AFE_config_clockMode()	14
4.1.2.5 AFE_config_phaseTimingScheme()	14
4.1.2.6 AFE_config_regMapInit()	15
4.1.2.7 AFE_configTIA()	15
4.1.2.8 AFE_disableTimingEngine()	16
4.1.2.9 AFE_enableTimingEngine()	16
4.1.2.10 AFE_initializeAFE()	16
4.1.2.11 AFE_set_AUTO_AMB_INSERT()	17
4.1.2.12 AFE_set_CF_TIA()	17
4.1.2.13 AFE_set_ENABLE_DRE()	18
4.1.2.14 AFE_set_FIFO_DATA_CTRL()	18
4.1.2.15 AFE_set_FILTER_BW()	18
4.1.2.16 AFE_set_ILED_DRVx()	19
4.1.2.17 AFE_set_ILED_FS()	19
4.1.2.18 AFE_set_IN_TIA()	20
4.1.2.19 AFE_set_INT_MUX_ADC_RDY()	20
4.1.2.20 AFE_set_LED_DC_EN_TIA()	21
4.1.2.21 AFE_set_LED_DRVx_TXN()	21
4.1.2.22 AFE_set_NUMAV()	21
4.1.2.23 AFE_set_PDNAFE()	22
4.1.2.24 AFE_set_phaseType()	22
4.1.2.25 AFE_set_PRPCT()	23
4.1.2.26 AFE_set_REG_SCALE_DRE()	23
4.1.2.27 AFE_set_REG_TWLED()	23

4.1.2.28 AFE_set_RF_TIA()	24
4.1.2.29 AFE_set_SW_RESET()	24
4.2 AFE_Functions_PPG.h File Reference	25
4.2.1 Detailed Description	27
4.2.2 Function Documentation	27
4.2.2.1 AFE_clearPPM()	27
4.2.2.2 AFE_compute_CF()	28
4.2.2.3 AFE_config_AMBCancellation()	28
4.2.2.4 AFE_config_clockMode()	29
4.2.2.5 AFE_config_LEDCancellation()	29
4.2.2.6 AFE_config_phaseTimingScheme()	29
4.2.2.7 AFE_config_regMapInit()	30
4.2.2.8 AFE_configAFE_forHRM()	30
4.2.2.9 AFE_configAFE_forSpO2()	30
4.2.2.10 AFE_configTIA()	31
4.2.2.11 AFE_disableTimingEngine()	31
4.2.2.12 AFE_enableTimingEngine()	32
4.2.2.13 AFE_initializeAFE()	32
4.2.2.14 AFE_set_AUTO_AMB_INSERT()	32
4.2.2.15 AFE_set_CF_TIA()	33
4.2.2.16 AFE_set_ENABLE_DRE()	33
4.2.2.17 AFE_set_FIFO_DATA_CTRL()	34
4.2.2.18 AFE_set_FILTER_BW()	34
4.2.2.19 AFE_set_ILED_DRVx()	35
4.2.2.20 AFE_set_ILED_FS()	35
4.2.2.21 AFE_set_IN_TIA()	35
4.2.2.22 AFE_set_INT_MUX_ADC_RDY()	36
4.2.2.23 AFE_set_LED_DC_EN_TIA()	36
4.2.2.24 AFE_set_LED_DRVx_TXN()	37
4.2.2.25 AFE_set_NUMAV()	37
4.2.2.26 AFE_set_PDNAFE()	37
4.2.2.27 AFE_set_phaseType()	38
4.2.2.28 AFE_set_PRPCT()	38
4.2.2.29 AFE_set_REG_SCALE_DRE()	39
4.2.2.30 AFE_set_REG_TWLED()	39
4.2.2.31 AFE_set_RF_TIA()	39
4.2.2.32 AFE_set_SW_RESET()	41
4.3 AFE_HostControl.c File Reference	41
4.3.1 Detailed Description	42
4.3.2 Function Documentation	42
4.3.2.1 AFE_readReg()	42
4.3.2.2 AFE_writeReg()	42

4.4 AFE_HostControl.h File Reference	43
4.4.1 Detailed Description	43
4.4.2 Function Documentation	43
4.4.2.1 AFE_readReg()	43
4.4.2.2 AFE_writeReg()	44
4.5 AFE_RegMap.c File Reference	44
4.5.1 Detailed Description	45
4.5.2 Function Documentation	45
4.5.2.1 AFE_assignRegMap()	45
4.5.2.2 AFE_modifyRegGlobal()	45
4.5.2.3 AFE_modifyRegPPM()	46
4.5.2.4 AFE_switchPage()	46
4.6 AFE_RegMap.h File Reference	46
4.6.1 Detailed Description	48
4.6.2 Function Documentation	48
4.6.2.1 AFE_assignRegMap()	48
4.6.2.2 AFE_modifyRegGlobal()	48
4.6.2.3 AFE_modifyRegPPM()	49
4.6.2.4 AFE_switchPage()	49
4.7 HRM_Configuration_Example.c File Reference	49
4.7.1 Detailed Description	50
4.7.2 Function Documentation	50
4.7.2.1 AFE_configAFE_forHRM()	50
4.8 SpO2_Configuration_Example.c File Reference	50
4.8.1 Detailed Description	51
4.8.2 Function Documentation	51
4.8.2.1 AFE_configAFE_forSpO2()	51
Index	53

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

AFE_RegMap	5
AFE_RegMap::GLOBAL	5
AFE_RegMap::PPM	8
sigParameter	9

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

AFE_Functions_PPG.c	
Function definitions for AFE's signal chain	11
AFE_Functions_PPG.h	
Function declarations for AFE's signal chain	25
AFE_HostControl.c	
Function definition for AFE register access This contains the definition for AFE's register read/write functions. User needs to update definition of these function in this file with appropriate function call for I2C/SPI Read/Write function based on their platform	41
AFE_HostControl.h	
Function declaration for AFE register access	43
AFE_RegMap.c	
Register map definition of AFE4432	44
AFE_RegMap.h	
Register map declaration of AFE4432	46
HRM_Configuration_Example.c	
An example configuration for HRM (Heart Rate Monitoring) based on AFE4432EVM	49
SpO2_Configuration_Example.c	
An example configuration for SpO2 based on AFE4432EVM	50

Chapter 3

Data Structure Documentation

3.1 AFE_RegMap Struct Reference

Data Structures

- struct [GLOBAL](#)
- struct [PPM](#)

Data Fields

- struct [AFE_RegMap::GLOBAL](#) **GLOBAL**
- struct [AFE_RegMap::PPM](#) **PPM**

The documentation for this struct was generated from the following file:

- [AFE_RegMap.h](#)

3.2 AFE_RegMap::GLOBAL Struct Reference

Data Fields

- [sigParameter](#) **SPI_REG_READ**
- [sigParameter](#) **SW_RESET**
- [sigParameter](#) **RW_CONT**
- [sigParameter](#) **PAGE_SEL**
- [sigParameter](#) **TM_COUNT_RST**
- [sigParameter](#) **FIFO_EN**
- [sigParameter](#) **EN_LED_SAT_DET**
- [sigParameter](#) **MASK_REVERSE**
- [sigParameter](#) **HIGH_PRF_MODE**
- [sigParameter](#) **MAX_AMB_REJ**
- [sigParameter](#) **DIS_POST_AMB_MAX_AMB_REJ**
- [sigParameter](#) **POL_IOFFDAC_LED**
- [sigParameter](#) **POL_IOFFDAC_AMB**

- [sigParameter](#) SWAP_DAC
- [sigParameter](#) EN_LED_OFFDAC_CHOP
- [sigParameter](#) EN_TIA_RST
- [sigParameter](#) IOFFDAC_PD1
- [sigParameter](#) IOFFDAC_PD2
- [sigParameter](#) REG_NUM_TIA_MAX
- [sigParameter](#) IOFFDAC_PD3
- [sigParameter](#) PD_DISCONNECT_TIA1
- [sigParameter](#) PD_DISCONNECT_TIA2
- [sigParameter](#) INT_MUX_ADC_RDY
- [sigParameter](#) INT_MUX_GPIO2
- [sigParameter](#) INPUT_PRF_RST_ON_GPIO2
- [sigParameter](#) PROG_INT2_STC
- [sigParameter](#) PROG_INT2_ENDC
- [sigParameter](#) SEL1_CLK_TE
- [sigParameter](#) OSCL_DIS
- [sigParameter](#) PDN_OSCL_IN_DEEP_SLEEP
- [sigParameter](#) EN_PRF_RESET
- [sigParameter](#) SEL1_CLK_PRF
- [sigParameter](#) DIV_CLK_EXT
- [sigParameter](#) EN_INT_IN_SINGLE_SHOT
- [sigParameter](#) EN_CLK_MODE_MIX
- [sigParameter](#) SET_OSCH_4M_1
- [sigParameter](#) PRPCT
- [sigParameter](#) REG_NUMPHASE
- [sigParameter](#) PRF_COUNTER_ENABLE
- [sigParameter](#) TIMER_ENABLE
- [sigParameter](#) PDNAFE
- [sigParameter](#) EN_AMB_DAC_LSB
- [sigParameter](#) ILED_FS
- [sigParameter](#) SET_OSCH_4M_3
- [sigParameter](#) SET_OSCH_4M_2
- [sigParameter](#) EN_LED_OFFDAC_TIA1
- [sigParameter](#) EN_LED_OFFDAC_TIA2
- [sigParameter](#) DESIGN_ID
- [sigParameter](#) SDOUT_TRISTATE
- [sigParameter](#) EN_GPIO2_OUT
- [sigParameter](#) EN_GPIO2_INT_ON_ADC_RDY
- [sigParameter](#) EN_GPIO2_IN
- [sigParameter](#) REG_WM_FIFO
- [sigParameter](#) FIFO_OFFSET_TO_FORCE
- [sigParameter](#) FORCE_FIFO_OFFSET
- [sigParameter](#) MASK_FIFO_RDY
- [sigParameter](#) AUTO_MASK_FIFO_RDY
- [sigParameter](#) LED_OFFDAC_MARKER_ON_LSB
- [sigParameter](#) EN_LED_OFFDAC_MARKER
- [sigParameter](#) FRAME_SYNC_ON_LSB
- [sigParameter](#) EN_FRAME_SYNC
- [sigParameter](#) REG_POINTER_DIFF
- [sigParameter](#) FIFO_OVERFLOW
- [sigParameter](#) EN_OSCL_CLKOUT
- [sigParameter](#) SPLIT_CLK_FOR_TE_PRF
- [sigParameter](#) IFS_AMB_OFFDAC_TIA1
- [sigParameter](#) IFS_AMB_OFFDAC_TIA2
- [sigParameter](#) PDN_BG_IN_DEEP_SLEEP

- [sigParameter](#) TX_AMP_ALWAYS_ACTIVE
- [sigParameter](#) DEGLITCH_AMB_DAC_MODE
- [sigParameter](#) FILTER_BW_FINE_SET1
- [sigParameter](#) FILTER_BW_PRE_SET1
- [sigParameter](#) OVERRIDE_BW_PRE
- [sigParameter](#) FILTER_BW_FINE_SET2
- [sigParameter](#) FILTER_BW_PRE_SET2
- [sigParameter](#) LOW_THRESHOLD_CODE1
- [sigParameter](#) HIGH_THRESHOLD_CODE1
- [sigParameter](#) LOW_THRESHOLD_CODE2
- [sigParameter](#) HIGH_THRESHOLD_CODE2
- [sigParameter](#) REG_SAMPLE_DELAY_SET1
- [sigParameter](#) REG_SAMPLE_DELAY_SET2
- [sigParameter](#) REG_SAMPLE_DELAY_SET3
- [sigParameter](#) REG_SAMPLE_DELAY_SET4
- [sigParameter](#) EN_PHASE_INT_GPIO
- [sigParameter](#) REG_TSEP_CONV_START
- [sigParameter](#) REG_TSAMP_SEP
- [sigParameter](#) Always1
- [sigParameter](#) EARLY_SAMP_FALL
- [sigParameter](#) REG_TSEP
- [sigParameter](#) REG_TDEEP_SLEEP_PWRUP
- [sigParameter](#) REG_TACTIVE_PWRUP
- [sigParameter](#) REG_TACTIVE_DATA_RDY
- [sigParameter](#) REG_TW_DATA_RDY
- [sigParameter](#) REG_TACTIVE_PWDN
- [sigParameter](#) REG_TDEEP_SLEEP_PWDN
- [sigParameter](#) REG_STEP_COUNT
- [sigParameter](#) REG_TW_FILTER_PRE
- [sigParameter](#) RF_ANA_AACM_END
- [sigParameter](#) REG_DELAY_ANA_ACQ
- [sigParameter](#) REG_TW_ANA_ACQ
- [sigParameter](#) REG_TSEP_ANA_ACQ_LED
- [sigParameter](#) TW_ANA_ACQ_OVERRIDE
- [sigParameter](#) TSEP_ANA_ACQ_LED_OVERRIDE
- [sigParameter](#) EMBED_ANA_AACM_IN_FIFO
- [sigParameter](#) RF_ANA_AACM_START_TIA1
- [sigParameter](#) RF_ANA_AACM_START_TIA2
- [sigParameter](#) CHANNEL_OFFSET_TIA1
- [sigParameter](#) CHANNEL_OFFSET_TIA2
- [sigParameter](#) REG_RECONV_THR_LED_DC
- [sigParameter](#) IOFFDAC_LED_DC_READ
- [sigParameter](#) POL_LED_DC_READ
- [sigParameter](#) LED_DC_LOOP_NUM_READ
- [sigParameter](#) FREEZE_LED_DC_LOOP0
- [sigParameter](#) FREEZE_LED_DC_LOOP1
- [sigParameter](#) FREEZE_LED_DC_LOOP2
- [sigParameter](#) FREEZE_LED_DC_LOOP3
- [sigParameter](#) FREEZE_LED_DC_LOOP4
- [sigParameter](#) FREEZE_LED_DC_LOOP5
- [sigParameter](#) FREEZE_LED_DC_LOOP6
- [sigParameter](#) FREEZE_LED_DC_LOOP7
- [sigParameter](#) REG_SCALE_DRE
- [sigParameter](#) USE_MSB_FOR_DRE
- [sigParameter](#) GAIN_CALIB_LED_DC_142K

- [sigParameter](#) GAIN_CALIB_LED_DC_166K
- [sigParameter](#) GAIN_CALIB_LED_DC_200K
- [sigParameter](#) GAIN_CALIB_LED_DC_250K
- [sigParameter](#) THR_DET_EN
- [sigParameter](#) THR_SEL_LOGIC
- [sigParameter](#) REG_THR_DET_PHASE
- [sigParameter](#) COMB_THR_DET_EN
- [sigParameter](#) THR_PPG_FLAG
- [sigParameter](#) DIS_DEEP_SLEEP
- [sigParameter](#) EN_ALWAYS_ACTIVE

The documentation for this struct was generated from the following file:

- [AFE_RegMap.h](#)

3.3 AFE_RegMap::PPM Struct Reference

Data Fields

- [sigParameter](#) LED_DRV1_TXN
- [sigParameter](#) LED_DRV2_TXN
- [sigParameter](#) THR_SEL_TIA_NUM
- [sigParameter](#) THR_SEL
- [sigParameter](#) THR_SEL_DATA_CTRL
- [sigParameter](#) PHASE_INT_GPIO2
- [sigParameter](#) ILED_DRV1
- [sigParameter](#) ILED_DRV2
- [sigParameter](#) REG_TWLED
- [sigParameter](#) RF_TIA1
- [sigParameter](#) CF_TIA1
- [sigParameter](#) IOFFDAC_LED_TIA1
- [sigParameter](#) LED_DC_EN_TIA1
- [sigParameter](#) IN_TIA1
- [sigParameter](#) RF_TIA2
- [sigParameter](#) CF_TIA2
- [sigParameter](#) IOFFDAC_LED_TIA2
- [sigParameter](#) LED_DC_EN_TIA2
- [sigParameter](#) IN_TIA2
- [sigParameter](#) UPDATE_BASELINE_AMB
- [sigParameter](#) USE_ANA_AACM
- [sigParameter](#) FILTER_SET_SEL
- [sigParameter](#) REG_NUM_TIA
- [sigParameter](#) AUTO_AMB_INSERT
- [sigParameter](#) ENABLE_DRE
- [sigParameter](#) REG_MASK_FACTOR
- [sigParameter](#) REG_NUMAV
- [sigParameter](#) FIFO_DATA_CTRL
- [sigParameter](#) REG_DEC_FACTOR
- [sigParameter](#) SEL_SAMPLE_DELAY_SET

The documentation for this struct was generated from the following file:

- [AFE_RegMap.h](#)

3.4 sigParameter Struct Reference

Data Fields

- `uint8_t address`
- `uint8_t msb`
- `uint8_t lsb`

The documentation for this struct was generated from the following file:

- [AFE_RegMap.h](#)

Chapter 4

File Documentation

4.1 AFE_Functions_PPG.c File Reference

Function definitions for AFE's signal chain.

```
#include "AFE_Functions_PPG.h"
```

Functions

- void [AFE_config_regMapInit](#) ()
Initialise AFE's Register Map.
- void [AFE_set_SW_RESET](#) ()
Applies software reset to AFE.
- void [AFE_set_PDNAFE](#) ()
Does software power down of AFE.
- void [AFE_config_phaseTimingScheme](#) (enum phaseTimingScheme phTmgScheme)
Configures the Phase Timing Scheme of the AFE.
- void [AFE_config_clockMode](#) (enum clockMode clkmode)
Configures the clocking mode of the AFE.
- void [AFE_set_PRPCT](#) (uint16_t prpcount)
Configures the PRPCT (i.e. sampling frequency) of the AFE.
- void [AFE_enableTimingEngine](#) ()
Enables the timing engine and PRF counter PRF.
- void [AFE_disableTimingEngine](#) ()
Disables the timing engine and PRF counter of AFE.
- void [AFE_set_ILED_FS](#) (enum ILED_FS FS_Value_inmA)
Sets the full scale of LED current.
- void [AFE_set_FILTER_BW](#) (enum REG_TWLED ledOnForSet1_inuS, enum REG_TWLED ledOnForSet2_inuS)
Sets the bandwidth of noise reduction filter based on LED ON times. In case of different LED ON times, user need to set the bandwidths based on datasheet's guidelines.
- void [AFE_clearPPM](#) ()
Clears the per phase registers. Recommended to call this function as SW/HW reset.
- void [AFE_set_LED_DRVx_TXN](#) (uint8_t PhNo, enum LED_DRV_TXN txnDrv1, enum LED_DRV_TXN txnDrv2)

- Assigns the LED (TXN switches) to be used for the given Phase.*

 - void **AFE_set_ILED_DRVx** (uint8_t PhNo, uint8_t iled1, uint8_t iled2)
- Configures the LED current for the given Phase.*

 - void **AFE_set_REG_TWLED** (uint8_t PhNo, enum REG_TWLED ledOn_inuS)

Configures the LED ON time for the given Phase if different LED ON time needs to be used, configure noise reduction filter, CF of TIA based on datasheet.
- void **AFE_set_NUMAV** (uint8_t PhNo, uint8_t ADC_samplesToAverage)

Configures the ADC averaging for the given Phase.
- void **AFE_set_AUTO_AMB_INSERT** (uint8_t PhNo, enum AUTO_AMB_INSERT autoAMB)

Configures the automatic insertion of AMB for the given Phase.
- void **AFE_set_FIFO_DATA_CTRL** (uint8_t PhNo, enum FIFO_DATA_CTRL fifoData)

Configures the FIFO data control for the given Phase.
- void **AFE_set_IN_TIA** (uint8_t PhNo, enum TIA_NO tiaSelected, enum IN_TIA pd1, enum IN_TIA pd2, enum IN_TIA pd3)

Configures the PD used for the given Phase.
- void **AFE_set_RF_TIA** (uint8_t PhNo, enum TIA_NO tiaSelected, enum RF_TIA rfSelected)

Configures the RF for the given Phase.
- void **AFE_set_CF_TIA** (uint8_t PhNo, enum TIA_NO tiaSelected, enum CF_TIA cfSelected)

Configures the CF for the given Phase.
- void **AFE_set_LED_DC_EN_TIA** (uint8_t PhNo, enum TIA_NO tiaSelected, bool state)

Enables or disables LED DC Cancellation for the given Phase.
- void **AFE_set_ENABLE_DRE** (uint8_t PhNo, bool state)

Enables or disables DRE for the given Phase LED DC Cancellations needs to be enabled for the DRE.
- void **AFE_configTIA** (uint8_t PhNo, enum TIA_NO tiaSelected, enum IN_TIA pdSelected, enum RF_TIA rfSelected, enum REG_TWLED ledOn_inuS, enum LED_DC_CancellationScheme LED_DC_cnclSelected)

Configures the TIA for the given Phase based on given arguments.
- void **AFE_config_AMBCancellation** (uint8_t PhNo, enum AMB_CancellationScheme aacmType)

Configures the AMB cancellation scheme for the given Phase.
- void **AFE_set_INT_MUX_ADC_RDY** (enum INT_MUX_ADC_RDY intOnADC_RDY)

Configures the interrupt on ADC_RDY pin.
- int **AFE_compute_CF** (enum REG_TWLED ledOn_inuS, enum RF_TIA rfSelected)

Computes the CF based on RF and LED ON time.
- void **AFE_set_phaseType** (uint8_t PhNo, enum phaseType selectedPhaseType, enum LED_DRV_TXN txnDrv1, enum LED_DRV_TXN txnDrv2)

Configures the Phase type along with LED to be used if decalred as LED.
- void **AFE_initializeAFE** ()

Initializes the AFE.
- void **AFE_set_REG_SCALE_DRE** (enum RF_TIA rfSelected)

Configures the scaling factor for the DRE based on given RF.

Variables

- uint8_t **maxNoTIA** = 1
- enum IFS_AMB_OFFDAC_TIA **AMB_DAC_FS_selected** = AMB_DAC_FS_16uA
- **AFE_RegMap dev1**
- char **CF_Array1** [6] = {CF_20pF, CF_17p5pF, CF_10pF, CF_10pF, CF_5pF, CF_2p5pF}
- char **CF_Array2** [4] = {CF_22p5pF, CF_17p5pF, CF_7p5pF, CF_2p5pF}
- char **CF_Array3** [3] = {CF_20pF, CF_10pF, CF_5pF}
- enum REG_TWLED LED_ON_TIME1 **LED_ON_TIME2**
- enum phaseTimingScheme **selectedPhaseTimingScheme** = STAGGER
- uint32_t **errorFlags** = 0
- uint32_t **enableBitsDRE** = 0

4.1.1 Detailed Description

Function definitions for AFE's signal chain.

This contains definition low level functions.
 User can add/modify these low level function based on their requirement.
 AFE parameter can be modified using "AFE_modifyRegGlobal" or "AFE_modifyRegPPM" function as shown in follow

Author

Prabin Yadav (yadav@ti.com)

4.1.2 Function Documentation

4.1.2.1 AFE_clearPPM()

```
void AFE_clearPPM ( )
```

Clears the per phase registers. Recommended to call this function as SW/HW reset.

Parameters

None	
------	--

Returns

None.

4.1.2.2 AFE_compute_CF()

```
int AFE_compute_CF (
    enum REG_TWLED ledOnForSet1_inuS,
    enum RF_TIA rfSelected )
```

Computes the CF based on RF and LED ON time.

Parameters

<i>ledOnForSet1_inuS</i>	LED ON time interms of fCLK_TE , Available Options: LED_ON_16uS, LED_ON_24uS, LED_ON_31uS, LED_ON_39uS, LED_ON_47uS, LED_ON_63uS, LED_ON_70uS, LED_ON_78uS, LED_ON_94uS, LED_ON_102uS, LED_ON_117uS
<i>rfSelected</i>	RF Value , Available Options: RF_3p7KOhm, RF_5KOhm, RF_10KOhm, RF_25KOhm, RF_33p3KOhm, RF_50KOhm, RF_71p5KOhm, RF_100KOhm, RF_142KOhm, RF_166KOhm, RF_200KOhm, RF_250KOhm, RF_500KOhm, RF_1MOhm

Returns

None.

4.1.2.3 AFE_config_AMBCancellation()

```
void AFE_config_AMBCancellation (
    uint8_t PhNo,
    enum AMB_CancellationScheme )
```

Configures the AMB cancellation scheme for the given Phase.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>AMB_CancellationScheme</i>	Available Options: AMB_Disabled, AMB_estimateAndCancel, AMB_cancel AMB_estimateAndCancel : AFE estimates ambient signal before just before given phase and cancels it AMB_cancel : Ambient signal is cancelled based on estimates from previous phase

Returns

None.

4.1.2.4 AFE_config_clockMode()

```
void AFE_config_clockMode (
    enum clockMode clkmode )
```

Configures the clocking mode of the AFE.

Parameters

<i>phTmgScheme</i>	Available Options: CLK_MODE_INT, CLK_MODE_EXT, CLK_MODE_SS, CLK_MODE_MIX
--------------------	--

Returns

None.

4.1.2.5 AFE_config_phaseTimingScheme()

```
void AFE_config_phaseTimingScheme (
    enum phaseTimingScheme phTmgScheme )
```

Configures the Phase Timing Scheme of the AFE.

Parameters

<i>phTmgScheme</i>	Available Options: STAGGER, HIGH_PRF_MODE, MAX_AMB_REJ, DIS_POST_AMB_MAX_AMB_REJ
--------------------	--

Returns

None.

4.1.2.6 AFE_config_regMapInit()

```
void AFE_config_regMapInit ( )
```

Initialise AFE's Register Map.

Parameters

<i>None</i>	
-------------	--

Returns

None.

4.1.2.7 AFE_configTIA()

```
void AFE_configTIA (
    uint8_t PhNo,
    enum TIA_NO tiaSelected,
    enum IN_TIA pdSelected,
    enum RF_TIA rfSelected,
    enum REG_TWLED ledOn_inuS,
    enum LED_DC_CancellationScheme LED_DC_cnclSelected )
```

Configures the TIA for the given Phase based on given arguments.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>tiaSelected</i>	TIA Number, Available Options: TIA1, TIA2, ,
<i>pdSelected</i>	PDs to be used, Available Options: PD_DISCONNECT, PD1, PD2, PD3
<i>rfSelected</i>	RF Value , Available Options: RF_3p7KOhm, RF_5KOhm, RF_10KOhm, RF_25KOhm, RF_33p3KOhm, RF_50KOhm, RF_71p5KOhm, RF_100KOhm, RF_142KOhm, RF_166KOhm, RF_200KOhm, RF_250KOhm, RF_500KOhm, RF_1MOhm
<i>ledOn_inuS</i>	LED ON time interms of fCLK_TE , Available Options: LED_ON_16uS, LED_ON_24uS, LED_ON_31uS, LED_ON_39uS, LED_ON_47uS, LED_ON_63uS, LED_ON_70uS, LED_ON_78uS, LED_ON_94uS, LED_ON_102uS,
Texas Instruments	LED_ON_117uS
<i>LED_DC_cnclSelected</i>	Available Options: LED_cancelDisabled, LED_cancelWithDRE, LED_cancelWithoutDRE

Returns

None.

4.1.2.8 AFE_disableTimingEngine()

```
void AFE_disableTimingEngine ( )
```

Disables the timing engine and PRF counter of AFE.

Parameters

None	
------	--

Returns

None.

4.1.2.9 AFE_enableTimingEngine()

```
void AFE_enableTimingEngine ( )
```

Enables the timing engine and PRF counter PRF.

Parameters

None	
------	--

Returns

None.

4.1.2.10 AFE_initializeAFE()

```
void AFE_initializeAFE ( )
```

Initializes the AFE.

Parameters

None	
------	--

Returns

None.

4.1.2.11 AFE_set_AUTO_AMB_INSERT()

```
void AFE_set_AUTO_AMB_INSERT (
    uint8_t PhNo,
    enum AUTO_AMB_INSERT autoAMB )
```

Configures the automatic insertion of AMB for the given Phase.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>autoAMB</i>	Available Options: NONE, PRE_AMB, PRE_AND_POST_AMB, POST_AMB

Returns

None.

4.1.2.12 AFE_set_CF_TIA()

```
void AFE_set_CF_TIA (
    uint8_t PhNo,
    enum TIA_NO tiaSelected,
    enum CF_TIA cfSelected )
```

Configures the CF for the given Phase.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>tiaSelected</i>	TIA Number, Available Options: TIA1, TIA2, ,
<i>cfSelected</i>	CF Value , Available Options: CF_2p5pF, CF_5pF, CF_7p5pF, CF_10pF, CF_17p5pF, CF_20pF, CF_22p5pF, CF_25pF

Returns

None.

4.1.2.13 AFE_set_ENABLE_DRE()

```
void AFE_set_ENABLE_DRE (
    uint8_t PhNo,
    bool state )
```

Enables or disables DRE for the given Phase LED DC Cancellations needs to be enabled for the DRE.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>state</i>	Available Options: true, false

Returns

None.

4.1.2.14 AFE_set_FIFO_DATA_CTRL()

```
void AFE_set_FIFO_DATA_CTRL (
    uint8_t PhNo,
    enum FIFO_DATA_CTRL fifoData )
```

Configures the FIFO data control for the given Phase.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>fifoData</i>	Available Options: NO_DATA, DEFINED_PHASE, LED_AMB, LED_AVG_OF_AMBS, PHASE_AND_AUTO_AMB_INSERTS

Returns

None.

4.1.2.15 AFE_set_FILTER_BW()

```
void AFE_set_FILTER_BW (
    enum REG_TWLED ledOnForSet1_inuS,
    enum REG_TWLED ledOnForSet2_inuS )
```

Sets the bandwidth of noise reduction filter based on LED ON times Incase of different LED ON times, user need to set the bandwidths based on datasheets's guidlines.

Parameters

<i>ledOnForSet1_inuS</i>	First LED ON time of system, Available Options: LED_ON_16uS, LED_ON_24uS, LED_ON_31uS, LED_ON_39uS, LED_ON_47uS, LED_ON_63uS, LED_ON_70uS, LED_ON_78uS, LED_ON_94uS, LED_ON_102uS, LED_ON_117uS
<i>ledOnForSet2_inuS</i>	Second LED ON time of system, Available Options: LED_ON_16uS, LED_ON_24uS, LED_ON_31uS, LED_ON_39uS, LED_ON_47uS, LED_ON_63uS, LED_ON_70uS, LED_ON_78uS, LED_ON_94uS, LED_ON_102uS, LED_ON_117uS

Returns

None.

4.1.2.16 AFE_set_ILED_DRVx()

```
void AFE_set_ILED_DRVx (
    uint8_t PhNo,
    uint8_t iled1,
    uint8_t iled2 )
```

Configures the LED current for the given Phase.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>iled1</i>	LED current interms of code for LED_DRV1 , Available Options: 0 to 255 codes
<i>iled2</i>	LED current interms of code for LED_DRV2 , Available Options: 0 to 255 codes

Returns

None.

4.1.2.17 AFE_set_ILED_FS()

```
void AFE_set_ILED_FS (
    enum ILED_FS FS_Value_inmA )
```

Sets the full scale of LED current.

Parameters

<i>FS_Value_inmA</i>	Available Options: ILED_FS_25mA, ILED_FS_50mA, ILED_FS_100mA, ILED_FS_125mA, ILED_FS_167mA
----------------------	--

Returns

None.

4.1.2.18 AFE_set_IN_TIA()

```
void AFE_set_IN_TIA (
    uint8_t PhNo,
    enum TIA_NO tiaSelected,
    enum IN_TIA pd1,
    enum IN_TIA pd2,
    enum IN_TIA pd3 )
```

Configures the PD used for the given Phase.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>tiaSelected</i>	TIA Number, Available Options: TIA1, TIA2, ,
<i>pd1</i>	Available Options: PD1, PD_DISCONNECT
<i>pd2</i>	Available Options: PD2, PD_DISCONNECT
<i>pd3</i>	Available Options: PD3, PD_DISCONNECT

Returns

None.

4.1.2.19 AFE_set_INT_MUX_ADC_RDY()

```
void AFE_set_INT_MUX_ADC_RDY (
    enum INT_MUX_ADC_RDY intOnADC_RDY )
```

Configures the interrupt on ADC_RDY pin.

Parameters

<i>intOnADC_RDY</i>	Interrupt signal , Available Options: DATA_RDY, THR_DET_RDY, FIFO_RDY, INT_OUT1, PRF_RST
---------------------	--

Returns

None.

4.1.2.20 AFE_set_LED_DC_EN_TIA()

```
void AFE_set_LED_DC_EN_TIA (
    uint8_t PhNo,
    enum TIA_NO tiaSelected,
    bool state )
```

Enables or disables LED DC Cancellation for the given Phase.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>tiaSelected</i>	TIA Number, Available Options: TIA1, TIA2, ,
<i>state</i>	Available Options: true, false

Returns

None.

4.1.2.21 AFE_set_LED_DRVx_TXN()

```
void AFE_set_LED_DRVx_TXN (
    uint8_t PhNo,
    enum LED_DRV_TXN txnDrv1,
    enum LED_DRV_TXN txnDrv2 )
```

Assigns the LED (TXN switches) to be used for the given Phase.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>txnDrv1</i>	TXN Switch connection for LED_DRV1 , Available Options: TXN1, TXN2, ... TXN8
<i>txnDrv2</i>	TXN Switch connection for LED_DRV2 , Available Options: TXN1, TXN2, ... TXN8

Returns

None.

4.1.2.22 AFE_set_NUMAV()

```
void AFE_set_NUMAV (
    uint8_t PhNo,
    uint8_t ADC_samplesToAverage )
```

Configures the ADC averaging for the given Phase.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>ADC_samplesToAverage</i>	Number of samples to be averaged , Available Options: 1,2,3,4,8

Returns

None.

4.1.2.23 AFE_set_PDNAFE()

```
void AFE_set_PDNAFE ( )
```

Does software power down of AFE.

Parameters

<i>None</i>	
-------------	--

Returns

None.

4.1.2.24 AFE_set_phaseType()

```
void AFE_set_phaseType (
    uint8_t PhNo,
    enum phaseType selectedPhaseType,
    enum LED_DRV_TXN txnDrv1,
    enum LED_DRV_TXN txnDrv2 )
```

Configures the Phase type along with LED to be used if decalred as LED.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>selectedPhaseType</i>	Type of Phase , Available Options: explicitAMB, explicitLED, LED_WithPreAMB, LED_WithPreAndPostAMB, LED_WithPostAMB
<i>txnDrv1</i>	TXN Switch connection for LED_DRV1 , Available Options: TXN1, TXN2, ... TXN8
<i>txnDrv2</i>	TXN Switch connection for LED_DRV2 , Available Options: TXN1, TXN2, ... TXN8

Returns

None.

4.1.2.25 AFE_set_PRPCT()

```
void AFE_set_PRPCT (
    uint16_t prpcount )
```

Configures the PRPCT (i.e. sampling frequency) of the AFE.

Parameters

<i>prpcount</i>	calculated as $fCLK_PRF / PRF$
-----------------	---------------------------------

Returns

None.

4.1.2.26 AFE_set_REG_SCALE_DRE()

```
void AFE_set_REG_SCALE_DRE (
    enum RF_TIA rfSelected )
```

Configures the scaling factor for the DRE based on given RF.

Parameters

<i>rfSelected</i>	RF Value , Available Options: RF_3p7KOhm, RF_5KOhm, RF_10KOhm, RF_25KOhm, RF_33p3KOhm, RF_50KOhm, RF_71p5KOhm, RF_100KOhm, RF_142KOhm, RF_166KOhm, RF_200KOhm, RF_250KOhm, RF_500KOhm, RF_1MOhm Among all the active TIA and Phases with the DRE enabled, pick the largest value of RF to be used with this function
-------------------	--

Returns

None.

4.1.2.27 AFE_set_REG_TWLED()

```
void AFE_set_REG_TWLED (
    uint8_t PhNo,
    enum REG_TWLED ledOn_inuS )
```

Configures the LED ON time for the given Phase if different LED ON time needs to be used, configure noise reduction filter, CF of TIA based on datasheet.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>ledOn_inuS</i>	LED ON time interms of fCLK_TE , Available Options: LED_ON_16uS, LED_ON_24uS, LED_ON_31uS, LED_ON_39uS, LED_ON_47uS, LED_ON_63uS, LED_ON_70uS, LED_ON_78uS, LED_ON_94uS, LED_ON_102uS, LED_ON_117uS

Returns

None.

4.1.2.28 AFE_set_RF_TIA()

```
void AFE_set_RF_TIA (
    uint8_t PhNo,
    enum TIA_NO tiaSelected,
    enum RF_TIA rfSelected )
```

Configures the RF for the given Phase.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>tiaSelected</i>	TIA Number, Available Options: TIA1, TIA2, ,
<i>rfSelected</i>	RF Value , Available Options: RF_3p7KOhm, RF_5KOhm, RF_10KOhm, RF_25KOhm, RF_33p3KOhm, RF_50KOhm, RF_71p5KOhm, RF_100KOhm, RF_142KOhm, RF_166KOhm, RF_200KOhm, RF_250KOhm, RF_500KOhm, RF_1MOhm

Returns

None.

4.1.2.29 AFE_set_SW_RESET()

```
void AFE_set_SW_RESET ( )
```

Applies software reset to AFE.

Parameters

<i>None</i>	
-------------	--

Returns

None.

4.2 AFE_Functions_PPG.h File Reference

Function declarations for AFE's signal chain.

```
#include "AFE_RegMap.h"
```

Enumerations

- enum **AMB_CancellationScheme** { **AMB_Disabled** , **AMB_estimateAndCancel** , **AMB_cancel** }
- enum **LED_DC_CancellationScheme** { **LED_cancelDisabled** , **LED_cancelWithDRE** , **LED_cancelWithoutDRE** }
- enum **phaseType** { **explicitAMB** , **explicitLED** , **LED_WithPreAMB** , **LED_WithPreAndPostAMB** , **LED_WithPostAMB** }
- enum **phaseTimingScheme** { **STAGGER** , **HIGH_PRF_MODE** , **MAX_AMB_REJ** , **DIS_POST_AMB** , **MAX_AMB_REJ** }
- enum **clockMode** { **CLK_MODE_INT** , **CLK_MODE_EXT** , **CLK_MODE_SS** , **CLK_MODE_MIX** }
- enum **RF_TIA** { **RF_3p7KOhm** , **RF_5KOhm** , **RF_10KOhm** , **RF_25KOhm** , **RF_33p3KOhm** , **RF_50KOhm** , **RF_71p5KOhm** , **RF_100KOhm** , **RF_142KOhm** , **RF_166KOhm** , **RF_200KOhm** , **RF_250KOhm** , **RF_500KOhm** , **RF_1MOhm** }
- enum **CF_TIA** { **CF_2p5pF** , **CF_5pF** , **CF_7p5pF** , **CF_10pF** , **CF_17p5pF** , **CF_20pF** , **CF_22p5pF** , **CF_25pF** }
- enum **ILED_FS** { **ILED_FS_25mA** = 0 , **ILED_FS_50mA** = 1 , **ILED_FS_100mA** = 2 , **ILED_FS_125mA** = 3 , **ILED_FS_167mA** = 4 }
- enum **FILTER_BW** { **FILT_BW_2p5KHz** = 5 , **FILT_BW_5KHz** = 6 , **FILT_BW_7p5KHz** = 7 , **FILT_BW_10KHz** = 0 , **FILT_BW_20KHz** = 12 , **FILT_BW_30KHz** = 8 , **FILT_BW_50KHz** = 16 , **FILT_BW_25KHz** = 14 , **FILT_BW_32p5KHz** = 9 , **FILT_BW_15KHz** = 2 , **FILT_BW_35KHz** = 10 }
- enum **REG_TWLED** { **LED_ON_16uS** = 3 , **LED_ON_24uS** = 5 , **LED_ON_31uS** = 7 , **LED_ON_39uS** = 9 , **LED_ON_47uS** = 11 , **LED_ON_63uS** = 15 , **LED_ON_70uS** = 17 , **LED_ON_78uS** = 19 , **LED_ON_94uS** = 23 , **LED_ON_102uS** = 25 , **LED_ON_117uS** = 29 }
- enum **IFS_AMB_OFFDAC_TIA** { **AMB_DAC_FS_16uA** = 0 , **AMB_DAC_FS_32uA** = 1 , **AMB_DAC_FS_64uA** = 3 , **AMB_DAC_FS_128uA** = 5 , **AMB_DAC_FS_255uA** = 7 }
- enum **LED_DRV_TXN** { **AMB_PH** = 0 , **TXN1** = 1 , **TXN2** = 2 , **TXN3** = 4 , **TXN4** = 8 }
- enum **AUTO_AMB_INSERT** { **NONE** , **PRE_AMB** , **PRE_AND_POST_AMB** , **POST_AMB** }
- enum **FIFO_DATA_CTRL** { **NO_DATA** , **DEFINED_PHASE** , **LED_AMB** , **LED_AVG_OF_AMBS** , **PHASE_AND_AUTO_AMB_INSERTS** }
- enum **TIA_NO** { **TIA1** , **TIA2** }
- enum **IN_TIA** { **PD_DISCONNECT** = 0 , **PD1** = 1 , **PD2** = 2 , **PD3** = 4 }

- enum **INT_MUX_ADC_RDY** {
DATA_RDY , **THR_DET_RDY** , **FIFO_RDY** , **INT_OUT1** ,
PRF_RST }
- enum **REG_RECONV_THR_LED_DC** {
RECONV_THR_0p9V = 192 , **RECONV_THR_0p8V** = 171 , **RECONV_THR_0p7V** = 149 , **RECONV_THR_0p6V** = 128 ,
RECONV_THR_0p5V = 107 }

Functions

- void **AFE_config_regMapInit** ()
Initialise AFE's Register Map.
- void **AFE_set_SW_RESET** ()
Applies software reset to AFE.
- void **AFE_set_PDNAFE** ()
Does software power down of AFE.
- void **AFE_config_phaseTimingScheme** (enum phaseTimingScheme phTmgScheme)
Configures the Phase Timing Scheme of the AFE.
- void **AFE_config_clockMode** (enum clockMode clkmode)
Configures the clocking mode of the AFE.
- void **AFE_set_PRPCT** (uint16_t prpcount)
Configures the PRPCT (i.e. sampling frequency) of the AFE.
- void **AFE_enableTimingEngine** ()
Enables the timing engine and PRF counter PRF.
- void **AFE_disableTimingEngine** ()
Disables the timing engine and PRF counter of AFE.
- void **AFE_set_ILED_FS** (enum ILED_FS FS_Value_inmA)
Sets the full scale of LED current.
- void **AFE_set_FILTER_BW** (enum REG_TWLED ledOnForSet1_inuS, enum REG_TWLED ledOnForSet2_inuS)
Sets the bandwidth of noise reduction filter based on LED ON times Incase of different LED ON times, user need to set the bandwidths based on datasheets's guidelines.
- void **AFE_clearPPM** ()
Clears the per phase registers. Recommended to call this function as SW/HW reset.
- void **AFE_set_LED_DRVx_TXN** (uint8_t PhNo, enum LED_DRV_TXN txnDrv1, enum LED_DRV_TXN txnDrv2)
Assigns the LED (TXN switches) to be used for the given Phase.
- void **AFE_set_phaseType** (uint8_t PhNo, enum phaseType selectedPhaseType, enum LED_DRV_TXN txnDrv1, enum LED_DRV_TXN txnDrv2)
Configures the Phase type along with LED to be used if decalred as LED.
- void **AFE_set_ILED_DRVx** (uint8_t PhNo, uint8_t iled1, uint8_t iled2)
Configures the LED current for the given Phase.
- void **AFE_set_REG_TWLED** (uint8_t PhNo, enum REG_TWLED ledOn_inuS)
Configures the LED ON time for the given Phase if different LED ON time needs to be used, configure noise reduction filter, CF of TIA based on datasheet.
- void **AFE_set_NUMAV** (uint8_t PhNo, uint8_t ADC_samplesToAverage)
Configures the ADC averaging for the given Phase.
- void **AFE_set_AUTO_AMB_INSERT** (uint8_t PhNo, enum AUTO_AMB_INSERT autoAMB)
Configures the automatic insertion of AMB for the given Phase.
- void **AFE_set_FIFO_DATA_CTRL** (uint8_t PhNo, enum FIFO_DATA_CTRL fifoData)
Configures the FIFO data control for the given Phase.

- void [AFE_set_IN_TIA](#) (uint8_t PhNo, enum TIA_NO tiaSelected, enum IN_TIA pd1, enum IN_TIA pd2, enum IN_TIA pd3)
Configures the PD used for the given Phase.
- void [AFE_set_RF_TIA](#) (uint8_t PhNo, enum TIA_NO tiaSelected, enum RF_TIA rfSelected)
Configures the RF for the given Phase.
- void [AFE_set_CF_TIA](#) (uint8_t PhNo, enum TIA_NO tiaSelected, enum CF_TIA cfSelected)
Configures the CF for the given Phase.
- void [AFE_set_LED_DC_EN_TIA](#) (uint8_t PhNo, enum TIA_NO tiaSelected, bool state)
Enables or disables LED DC Cancellation for the given Phase.
- void [AFE_set_ENABLE_DRE](#) (uint8_t PhNo, bool state)
Enables or disables DRE for the given Phase LED DC Cancellations needs to be enabled for the DRE.
- void [AFE_configTIA](#) (uint8_t PhNo, enum TIA_NO tiaSelected, enum IN_TIA pdSelected, enum RF_TIA rfSelected, enum REG_TWLED ledOn_inuS, enum LED_DC_CancellationScheme LED_DC_cnclSelected)
Configures the TIA for the given Phase based on given arguments.
- void [AFE_config_AMBCancellation](#) (uint8_t PhNo, enum AMB_CancellationScheme)
Configures the AMB cancellation scheme for the given Phase.
- void [AFE_config_LEDCancellation](#) (uint8_t PhNo, enum LED_DC_CancellationScheme)
Configures the LED DC Cancellation scheme for the given Phase.
- void [AFE_set_INT_MUX_ADC_RDY](#) (enum INT_MUX_ADC_RDY intOnADC_RDY)
Configures the interrupt on ADC_RDY pin.
- int [AFE_compute_CF](#) (enum REG_TWLED ledOnForSet1_inuS, enum RF_TIA rfSelected)
Computes the CF based on RF and LED ON time.
- void [AFE_initializeAFE](#) ()
Initializes the AFE.
- void [AFE_set_REG_SCALE_DRE](#) (enum RF_TIA rfSelected)
Configures the scaling factor for the DRE based on given RF.
- void [AFE_configAFE_forHRM](#) ()
Configuration example for HRM (Heart Rate Monitoring)
- void [AFE_configAFE_forSpO2](#) ()
Configuration example for SpO2.

4.2.1 Detailed Description

Function declarations for AFE's signal chain.

This contains the prototypes for the AFE's PPG signal chain driver and eventually any macros, constants, or global variables you will need.

Author

Prabin Yadav (yadav@ti.com)

4.2.2 Function Documentation

4.2.2.1 AFE_clearPPM()

```
void AFE_clearPPM ( )
```

Clears the per phase registers. Recommended to call this function as SW/HW reset.

Parameters

<i>None</i>	
-------------	--

Returns

None.

4.2.2.2 AFE_compute_CF()

```
int AFE_compute_CF (
    enum REG_TWLED ledOnForSet1_inuS,
    enum RF_TIA rfSelected )
```

Computes the CF based on RF and LED ON time.

Parameters

<i>ledOnForSet1_inuS</i>	LED ON time interms of fCLK_TE , Available Options: LED_ON_16uS, LED_ON_24uS, LED_ON_31uS, LED_ON_39uS, LED_ON_47uS, LED_ON_63uS, LED_ON_70uS, LED_ON_78uS, LED_ON_94uS, LED_ON_102uS, LED_ON_117uS
<i>rfSelected</i>	RF Value , Available Options: RF_3p7KOhm, RF_5KOhm, RF_10KOhm, RF_25KOhm, RF_33p3KOhm, RF_50KOhm, RF_71p5KOhm, RF_100KOhm, RF_142KOhm, RF_166KOhm, RF_200KOhm, RF_250KOhm, RF_500KOhm, RF_1MOhm

Returns

None.

4.2.2.3 AFE_config_AMBCancellation()

```
void AFE_config_AMBCancellation (
    uint8_t PhNo,
    enum AMB_CancellationScheme )
```

Configures the AMB cancellation scheme for the given Phase.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>AMB_CancellationScheme</i>	Available Options: AMB_Disabled, AMB_estimateAndCancel, AMB_cancel AMB_estimateAndCancel : AFE estimates ambient signal before just before given phase and cancels it AMB_cancel : Ambient signal is cancelled based on estimates from previous phase

Returns

None.

4.2.2.4 AFE_config_clockMode()

```
void AFE_config_clockMode (
    enum clockMode clkmode )
```

Configures the clocking mode of the AFE.

Parameters

<i>phTmgScheme</i>	Available Options: CLK_MODE_INT, CLK_MODE_EXT, CLK_MODE_SS, CLK_MODE_MIX
--------------------	--

Returns

None.

4.2.2.5 AFE_config_LEDCancellation()

```
void AFE_config_LEDCancellation (
    uint8_t PhNo,
    enum LED_DC_CancellationScheme )
```

Configures the LED DC Cancellation scheme for the given Phase.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>LED_DC_CancellationScheme</i>	Available Options: LED_cancelDisabled, LED_cancelWithDRE, LED_cancelWithoutDRE

Returns

None.

4.2.2.6 AFE_config_phaseTimingScheme()

```
void AFE_config_phaseTimingScheme (
    enum phaseTimingScheme phTmgScheme )
```

Configures the Phase Timing Scheme of the AFE.

Parameters

<i>phTmgScheme</i>	Available Options: STAGGER, HIGH_PRF_MODE, MAX_AMB_REJ, DIS_POST_AMB_MAX_AMB_REJ
--------------------	--

Returns

None.

4.2.2.7 AFE_config_regMapInit()

```
void AFE_config_regMapInit ( )
```

Initialise AFE's Register Map.

Parameters

<i>None</i>	
-------------	--

Returns

None.

4.2.2.8 AFE_configAFE_forHRM()

```
void AFE_configAFE_forHRM ( )
```

Configuration example for HRM (Heart Rate Monitoring)

Parameters

<i>None</i>	
-------------	--

Returns

None.

4.2.2.9 AFE_configAFE_forSpO2()

```
void AFE_configAFE_forSpO2 ( )
```

Configuration example for SpO2.

Parameters

None	
------	--

Returns

None.

4.2.2.10 AFE_configTIA()

```
void AFE_configTIA (
    uint8_t PhNo,
    enum TIA_NO tiaSelected,
    enum IN_TIA pdSelected,
    enum RF_TIA rfSelected,
    enum REG_TWLED ledOn_inuS,
    enum LED_DC_CancellationScheme LED_DC_cnclSelected )
```

Configures the TIA for the given Phase based on given arguments.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>tiaSelected</i>	TIA Number, Available Options: TIA1, TIA2, ,
<i>pdSelected</i>	PDs to be used, Available Options: PD_DISCONNECT, PD1, PD2, PD3
<i>rfSelected</i>	RF Value , Available Options: RF_3p7KOhm, RF_5KOhm, RF_10KOhm, RF_25KOhm, RF_33p3KOhm, RF_50KOhm, RF_71p5KOhm, RF_100KOhm, RF_142KOhm, RF_166KOhm, RF_200KOhm, RF_250KOhm, RF_500KOhm, RF_1MOhm
<i>ledOn_inuS</i>	LED ON time interms of fCLK_TE , Available Options: LED_ON_16uS, LED_ON_24uS, LED_ON_31uS, LED_ON_39uS, LED_ON_47uS, LED_ON_63uS, LED_ON_70uS, LED_ON_78uS, LED_ON_94uS, LED_ON_102uS, LED_ON_117uS
<i>LED_DC_cnclSelected</i>	Available Options: LED_cancelDisabled, LED_cancelWithDRE, LED_cancelWithoutDRE

Returns

None.

4.2.2.11 AFE_disableTimingEngine()

```
void AFE_disableTimingEngine ( )
```

Disables the timing engine and PRF counter of AFE.

Parameters

<i>None</i>	
-------------	--

Returns

None.

4.2.2.12 AFE_enableTimingEngine()

```
void AFE_enableTimingEngine ( )
```

Enables the timing engine and PRF counter PRF.

Parameters

<i>None</i>	
-------------	--

Returns

None.

4.2.2.13 AFE_initializeAFE()

```
void AFE_initializeAFE ( )
```

Initializes the AFE.

Parameters

<i>None</i>	
-------------	--

Returns

None.

4.2.2.14 AFE_set_AUTO_AMB_INSERT()

```
void AFE_set_AUTO_AMB_INSERT (
    uint8_t PhNo,
    enum AUTO_AMB_INSERT autoAMB )
```

Configures the automatic insertion of AMB for the given Phase.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>autoAMB</i>	Available Options: NONE, PRE_AMB, PRE_AND_POST_AMB, POST_AMB

Returns

None.

4.2.2.15 AFE_set_CF_TIA()

```
void AFE_set_CF_TIA (
    uint8_t PhNo,
    enum TIA_NO tiaSelected,
    enum CF_TIA cfSelected )
```

Configures the CF for the given Phase.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>tiaSelected</i>	TIA Number, Available Options: TIA1, TIA2, ,
<i>cfSelected</i>	CF Value , Available Options: CF_2p5pF, CF_5pF, CF_7p5pF, CF_10pF, CF_17p5pF, CF_20pF, CF_22p5pF, CF_25pF

Returns

None.

4.2.2.16 AFE_set_ENABLE_DRE()

```
void AFE_set_ENABLE_DRE (
    uint8_t PhNo,
    bool state )
```

Enables or disables DRE for the given Phase LED DC Cancellations needs to be enabled for the DRE.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>state</i>	Available Options: true, false

Returns

None.

4.2.2.17 AFE_set_FIFO_DATA_CTRL()

```
void AFE_set_FIFO_DATA_CTRL (
    uint8_t PhNo,
    enum FIFO_DATA_CTRL fifoData )
```

Configures the FIFO data control for the given Phase.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>fifoData</i>	Available Options: NO_DATA, DEFINED_PHASE, LED_AMB, LED_AVG_OF_AMBS, PHASE_AND_AUTO_AMB_INSERTS

Returns

None.

4.2.2.18 AFE_set_FILTER_BW()

```
void AFE_set_FILTER_BW (
    enum REG_TWLED ledOnForSet1_inuS,
    enum REG_TWLED ledOnForSet2_inuS )
```

Sets the bandwidth of noise reduction filter based on LED ON times Incase of different LED ON times, user need to set the bandwidths based on datasheets's guidelines.

Parameters

<i>ledOnForSet1_inuS</i>	First LED ON time of system, Available Options: LED_ON_16uS, LED_ON_24uS, LED_ON_31uS, LED_ON_39uS, LED_ON_47uS, LED_ON_63uS, LED_ON_70uS, LED_ON_78uS, LED_ON_94uS, LED_ON_102uS, LED_ON_117uS
<i>ledOnForSet2_inuS</i>	Second LED ON time of system, Available Options: LED_ON_16uS, LED_ON_24uS, LED_ON_31uS, LED_ON_39uS, LED_ON_47uS, LED_ON_63uS, LED_ON_70uS, LED_ON_78uS, LED_ON_94uS, LED_ON_102uS, LED_ON_117uS

Returns

None.

4.2.2.19 AFE_set_ILED_DRVx()

```
void AFE_set_ILED_DRVx (
    uint8_t PhNo,
    uint8_t iled1,
    uint8_t iled2 )
```

Configures the LED current for the given Phase.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>iled1</i>	LED current interms of code for LED_DRV1 , Available Options: 0 to 255 codes
<i>iled2</i>	LED current interms of code for LED_DRV2 , Available Options: 0 to 255 codes

Returns

None.

4.2.2.20 AFE_set_ILED_FS()

```
void AFE_set_ILED_FS (
    enum ILED_FS FS_Value_inmA )
```

Sets the full scale of LED current.

Parameters

<i>FS_Value_inmA</i>	Available Options: ILED_FS_25mA, ILED_FS_50mA, ILED_FS_100mA, ILED_FS_125mA, ILED_FS_167mA
----------------------	--

Returns

None.

4.2.2.21 AFE_set_IN_TIA()

```
void AFE_set_IN_TIA (
    uint8_t PhNo,
    enum TIA_NO tiaSelected,
    enum IN_TIA pd1,
    enum IN_TIA pd2,
    enum IN_TIA pd3 )
```

Configures the PD used for the given Phase.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>tiaSelected</i>	TIA Number, Available Options: TIA1, TIA2, ,
<i>pd1</i>	Available Options: PD1, PD_DISCONNECT
<i>pd2</i>	Available Options: PD2, PD_DISCONNECT
<i>pd3</i>	Available Options: PD3, PD_DISCONNECT

Returns

None.

4.2.2.22 AFE_set_INT_MUX_ADC_RDY()

```
void AFE_set_INT_MUX_ADC_RDY (
    enum INT_MUX_ADC_RDY intOnADC_RDY )
```

Configures the interrupt on ADC_RDY pin.

Parameters

<i>intOnADC_RDY</i>	Interrupt signal , Available Options: DATA_RDY, THR_DET_RDY, FIFO_RDY, INT_OUT1, PRF_RST
---------------------	--

Returns

None.

4.2.2.23 AFE_set_LED_DC_EN_TIA()

```
void AFE_set_LED_DC_EN_TIA (
    uint8_t PhNo,
    enum TIA_NO tiaSelected,
    bool state )
```

Enables or disables LED DC Cancellation for the given Phase.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>tiaSelected</i>	TIA Number, Available Options: TIA1, TIA2, ,
<i>state</i>	Available Options: true, false

Returns

None.

4.2.2.24 AFE_set_LED_DRVx_TXN()

```
void AFE_set_LED_DRVx_TXN (
    uint8_t PhNo,
    enum LED_DRV_TXN txnDrv1,
    enum LED_DRV_TXN txnDrv2 )
```

Assigns the LED (TXN switches) to be used for the given Phase.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>txnDrv1</i>	TXN Switch connection for LED_DRV1 , Available Options: TXN1, TXN2, ... TXN8
<i>txnDrv2</i>	TXN Switch connection for LED_DRV2 , Available Options: TXN1, TXN2, ... TXN8

Returns

None.

4.2.2.25 AFE_set_NUMAV()

```
void AFE_set_NUMAV (
    uint8_t PhNo,
    uint8_t ADC_samplesToAverage )
```

Configures the ADC averaging for the given Phase.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>ADC_samplesToAverage</i>	Number of samples to be averaged , Available Options: 1,2,3,4,8

Returns

None.

4.2.2.26 AFE_set_PDNAFE()

```
void AFE_set_PDNAFE ( )
```

Does software power down of AFE.

Parameters

<i>None</i>	
-------------	--

Returns

None.

4.2.2.27 AFE_set_phaseType()

```
void AFE_set_phaseType (
    uint8_t PhNo,
    enum phaseType selectedPhaseType,
    enum LED_DRV_TXN txnDrv1,
    enum LED_DRV_TXN txnDrv2 )
```

Configures the Phase type along with LED to be used if decalred as LED.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>selectedPhaseType</i>	Type of Phase , Available Options: explicitAMB, explicitLED, LED_WithPreAMB, LED_WithPreAndPostAMB, LED_WithPostAMB
<i>txnDrv1</i>	TXN Switch connection for LED_DRV1 , Available Options: TXN1, TXN2, ... TXN8
<i>txnDrv2</i>	TXN Switch connection for LED_DRV2 , Available Options: TXN1, TXN2, ... TXN8

Returns

None.

4.2.2.28 AFE_set_PRPCT()

```
void AFE_set_PRPCT (
    uint16_t prpcount )
```

Configures the PRPCT (i.e. sampling frequency) of the AFE.

Parameters

<i>prpcount</i>	calculated as $fCLK_PRF / PRF$
-----------------	---------------------------------

Returns

None.

4.2.2.29 AFE_set_REG_SCALE_DRE()

```
void AFE_set_REG_SCALE_DRE (
    enum RF_TIA rfSelected )
```

Configures the scaling factor for the DRE based on given RF.

Parameters

<i>rfSelected</i>	RF Value , Available Options: RF_3p7KOhm, RF_5KOhm, RF_10KOhm, RF_25KOhm, RF_33p3KOhm, RF_50KOhm, RF_71p5KOhm, RF_100KOhm, RF_142KOhm, RF_166KOhm, RF_200KOhm, RF_250KOhm, RF_500KOhm, RF_1MOhm Among all the active TIA and Phases with the DRE enabled, pick the largest value of RF to be used with this function
-------------------	--

Returns

None.

4.2.2.30 AFE_set_REG_TWLED()

```
void AFE_set_REG_TWLED (
    uint8_t PhNo,
    enum REG_TWLED ledOn_inuS )
```

Configures the LED ON time for the given Phase if different LED ON time needs to be used, configure noise reduction filter, CF of TIA based on datasheet.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>ledOn_inuS</i>	LED ON time interms of fCLK_TE , Available Options: LED_ON_16uS, LED_ON_24uS, LED_ON_31uS, LED_ON_39uS, LED_ON_47uS, LED_ON_63uS, LED_ON_70uS, LED_ON_78uS, LED_ON_94uS, LED_ON_102uS, LED_ON_117uS

Returns

None.

4.2.2.31 AFE_set_RF_TIA()

```
void AFE_set_RF_TIA (
    uint8_t PhNo,
```

```
enum TIA_NO tiaSelected,  
enum RF_TIA rfSelected )
```

Configures the RF for the given Phase.

Parameters

<i>PhNo</i>	Phase Number , Available Options: Phase1, Phase2, ... Phase16
<i>tiaSelected</i>	TIA Number, Available Options: TIA1, TIA2, ,
<i>rfSelected</i>	RF Value , Available Options: RF_3p7KOhm, RF_5KOhm, RF_10KOhm, RF_25KOhm, RF_33p3KOhm, RF_50KOhm, RF_71p5KOhm, RF_100KOhm, RF_142KOhm, RF_166KOhm, RF_200KOhm, RF_250KOhm, RF_500KOhm, RF_1MOhm

Returns

None.

4.2.2.32 AFE_set_SW_RESET()

```
void AFE_set_SW_RESET ( )
```

Applies software reset to AFE.

Parameters

<i>None</i>	
-------------	--

Returns

None.

4.3 AFE_HostControl.c File Reference

Function definition for AFE register access This contains the definition for AFE's register read/write functions. User needs to update definition of these function in this file with appropriate function call for I2C/SPI Read/Write function based on their platform.

```
#include "AFE_HostControl.h"
```

Functions

- `uint32_t AFE_readReg (uint8_t registerAddress)`
Reads the specified register address from the AFE. User needs to update definition of these function in "AFE_↔_HostControl.c" file with appropriate function call for I2C/SPI Read/Write function based on their platform.
- `void AFE_writeReg (uint8_t registerAddress, uint32_t value)`
Writes the specified register of AFE with the given value. User needs to update definition of these function in "AFE_↔_HostControl.c" file with appropriate function call for I2C/SPI Read/Write function based on their platform.

4.3.1 Detailed Description

Function definition for AFE register access This contains the definition for AFE's register read/write functions. User needs to update definition of these function in this file with appropriate function call for I2C/SPI Read/Write function based on their platform.

Author

Prabin Yadav (yadav@ti.com)

4.3.2 Function Documentation

4.3.2.1 AFE_readReg()

```
uint32_t AFE_readReg (
    uint8_t registerAddress )
```

Reads the specified register address from the AFE. User needs to update definition of these function in "AFE_HostControl.c" file with appropriate function call for I2C/SPI Read/Write function based on their platform.

Parameters

<i>registerAddress</i>	Register address in uint8_t format.
------------------------	-------------------------------------

Returns

24bit content of specified register in uint32_t format.

4.3.2.2 AFE_writeReg()

```
void AFE_writeReg (
    uint8_t registerAddress,
    uint32_t value )
```

Writes the specified register of AFE with the given value. User needs to update definition of these function in "AFE_HostControl.c" file with appropriate function call for I2C/SPI Read/Write function based on their platform.

Parameters

<i>registerAddress</i>	Register address in uint8_t format.
<i>value</i>	Register content in uint32_t format.

Returns

None.

4.4 AFE_HostControl.h File Reference

Function declaration for AFE register access.

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <stdbool.h>
#include "EVM_Hardware.h"
```

Functions

- uint32_t [AFE_readReg](#) (uint8_t registerAddress)
Reads the specefied register address from the AFE. User needs to update definition of these function in "AFE_↔ HostControl.c" file with appropriate function call for I2C/SPI Read/Write function based on their platform.
- void [AFE_writeReg](#) (uint8_t registerAddress, uint32_t value)
Writes the specified register of AFE with the given value. User needs to update definition of these function in "AFE_↔ _HostControl.c" file with appropriate function call for I2C/SPI Read/Write function based on their platform.

4.4.1 Detailed Description

Function declaration for AFE register access.

This contains the declaration for AFE's reagister read/write functions.
User needs to update definition of these function in "AFE_HostControl.c" file
with appropriate function call for I2C/SPI Read/Write function based on their platform

Author

Prabin Yadav (yadav@ti.com)

4.4.2 Function Documentation

4.4.2.1 AFE_readReg()

```
uint32_t AFE_readReg (
    uint8_t registerAddress )
```

Reads the specefied register address from the AFE. User needs to update definition of these function in "AFE_↔ HostControl.c" file with appropriate function call for I2C/SPI Read/Write function based on their platform.

Parameters

<i>registerAddress</i>	Register address in uint8_t format.
------------------------	-------------------------------------

Returns

24bit content of specified register in uint32_t format.

4.4.2.2 AFE_writeReg()

```
void AFE_writeReg (
    uint8_t registerAddress,
    uint32_t value )
```

Writes the specified register of AFE with the given value. User needs to update definition of these function in "AFE_HostControl.c" file with appropriate function call for I2C/SPI Read/Write function based on their platform.

Parameters

<i>registerAddress</i>	Register address in uint8_t format.
<i>value</i>	Register content in uint32_t format.

Returns

None.

4.5 AFE_RegMap.c File Reference

Register map definition of AFE4432.

```
#include "AFE_RegMap.h"
#include <stdio.h>
```

Functions

- void [AFE_assignRegMap](#) ([AFE_RegMap](#) *AFE)
Assigns the register map to the AFE.
- void [AFE_modifyRegGlobal](#) ([sigParameter](#) *param, uint32_t value)
Modifies the global registers in Page0.
- void [AFE_modifyRegPPM](#) (uint8_t PhaseNo, [sigParameter](#) *param, uint32_t value)
Modifies the per-phase registers in Page1. Registers in 0x20 to 0x9F in Page1.
- void [AFE_switchPage](#) (int pageNo)
Switches the Page.

Variables

- volatile uint16_t **lastRegAddress**
- volatile uint32_t **lastRegValue**

4.5.1 Detailed Description

Register map definitaion of AFE4432.

```
This file contains the definition for the AFE4432's parameter.  
Each AFE parameter's attributes ( address, page, msb bit and lsb bit) is defined here.
```

Author

Prabin Yadav (yadav@ti.com)

4.5.2 Function Documentation

4.5.2.1 AFE_assignRegMap()

```
void AFE_assignRegMap (  
    AFE_RegMap * AFE )
```

Assigns the reggister map to the AFE.

Parameters

None	
------	--

Returns

None.

4.5.2.2 AFE_modifyRegGlobal()

```
void AFE_modifyRegGlobal (  
    sigParameter * param,  
    uint32_t value )
```

Modifies the global registers in Page0.

Parameters

<i>AFE_Parameter</i>	
----------------------	--

Algorithm of the method is as follows

- Masks the bits for the new value to be written
- Creates a new register value to be written to h/w based on created mask and data to be written
- Initiates I2C write to device

4.5.2.3 AFE_modifyRegPPM()

```
void AFE_modifyRegPPM (
    uint8_t PhaseNo,
    sigParameter * param,
    uint32_t value )
```

Modifies the per-phase registers in Page1. Registers in 0x20 to 0x9F in Page1.

Parameters

<i>PhaseNo</i>	
----------------	--

4.5.2.4 AFE_switchPage()

```
void AFE_switchPage (
    int pageNo )
```

Switches the Page.

Parameters

<i>PageNo</i>	0: For Page0, 1: For Page1
---------------	----------------------------

Returns

None.

4.6 AFE_RegMap.h File Reference

Register map declaration of AFE4432.

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <stdbool.h>
#include "AFE_HostControl.h"
```

Data Structures

- struct [sigParameter](#)
- struct [AFE_RegMap](#)
- struct [AFE_RegMap::GLOBAL](#)
- struct [AFE_RegMap::PPM](#)

Macros

- #define **Phase1** 32
- #define **Phase2** 37
- #define **Phase3** 42
- #define **Phase4** 47
- #define **Phase5** 52
- #define **Phase6** 57
- #define **Phase7** 62
- #define **Phase8** 67
- #define **Phase9** 72
- #define **Phase10** 77
- #define **Phase11** 82
- #define **Phase12** 87
- #define **Enable** 1
- #define **Disable** 0

Typedefs

- typedef struct [sigParameter](#) **sigParameter**
- typedef struct [AFE_RegMap](#) **AFE_RegMap**

Functions

- void [AFE_assignRegMap](#) ([AFE_RegMap](#) *AFE)
Assigns the reggister map to the AFE.
- void [AFE_modifyRegGlobal](#) ([sigParameter](#) *param, uint32_t value)
Modifies the global registers in Page0.
- void [AFE_modifyRegPPM](#) (uint8_t PhaseNo, [sigParameter](#) *param, uint32_t value)
Modifies the per-phase registers in Page1. Registers in 0x20 to 0x9F in Page1.
- void [AFE_switchPage](#) (int pageNo)
Switches the Page.

4.6.1 Detailed Description

Register map declaration of AFE4432.

This file contains the declaration for the AFE4432's parameter.
Each AFE parameter has address, page, msb bit and lsb bit as attributes.

Author

Prabin Yadav (yadav@ti.com)

4.6.2 Function Documentation

4.6.2.1 AFE_assignRegMap()

```
void AFE_assignRegMap (
    AFE_RegMap * AFE )
```

Assigns the reggister map to the AFE.

Parameters

None	
------	--

Returns

None.

4.6.2.2 AFE_modifyRegGlobal()

```
void AFE_modifyRegGlobal (
    sigParameter * param,
    uint32_t value )
```

Modifies the global registers in Page0.

Parameters

AFE_Parameter	
---------------	--

Algorithm of the method is as follows

- Masks the bits for the new value to be written

- Creates a new register value to be written to h/w based on created mask and data to be written
- Initiates I2C write to device

4.6.2.3 AFE_modifyRegPPM()

```
void AFE_modifyRegPPM (
    uint8_t PhaseNo,
    sigParameter * param,
    uint32_t value )
```

Modifies the per-phase registers in Page1. Registers in 0x20 to 0x9F in Page1.

Parameters

<i>PhaseNo</i>	
----------------	--

4.6.2.4 AFE_switchPage()

```
void AFE_switchPage (
    int pageNo )
```

Switches the Page.

Parameters

<i>PageNo</i>	0: For Page0, 1: For Page1
---------------	----------------------------

Returns

None.

4.7 HRM_Configuration_Example.c File Reference

An example configuration for HRM (Heart Rate Monitoring) based on AFE4432EVM.

```
#include "AFE_Functions_PPG.h"
```

Functions

- void [AFE_configAFE_forHRM \(\)](#)
Configuration example for HRM (Heart Rate Monitoring)

Variables

- struct [AFE_RegMap](#) **dev1**
- uint32_t **errorFlags**

4.7.1 Detailed Description

An example configuration for HRM (Heart Rate Monitoring) based on AFE4432EVM.

```
This example shows how to use the driver to configure the AFE based on given hardware (i.e. AFE4432EVM).
User can take this as a reference and modify based on their requirements such as LED-PD associations, clock
```

Author

Prabin Yadav (yadav@ti.com)

4.7.2 Function Documentation

4.7.2.1 AFE_configAFE_forHRM()

```
void AFE_configAFE_forHRM ( )
```

Configuration example for HRM (Heart Rate Monitoring)

Parameters

None	
------	--

Returns

None.

4.8 SpO2_Configuration_Example.c File Reference

An example configuration for SpO2 based on AFE4432EVM.

```
#include "AFE_Functions_PPG.h"
```

Functions

- void [AFE_configAFE_forSpO2](#) ()
Configuration example for SpO2.

Variables

- struct [AFE_RegMap](#) **dev1**
- uint32_t **errorFlags**

4.8.1 Detailed Description

An example configuration for SpO2 based on AFE4432EVM.

This example shows how to use the driver to configure the AFE based on given hardware (i.e. AFE4432EVM). User can take this as a reference and modify based on their requirements such as LED-PD associations, clock

Author

Prabin Yadav (yadav@ti.com)

4.8.2 Function Documentation

4.8.2.1 AFE_configAFE_forSpO2()

```
void AFE_configAFE_forSpO2 ( )
```

Configuration example for SpO2.

Parameters

<i>None</i>	
-------------	--

Returns

None.

Index

AFE_assignRegMap
 AFE_RegMap.c, [45](#)
 AFE_RegMap.h, [48](#)
AFE_clearPPM
 AFE_Functions_PPG.c, [13](#)
 AFE_Functions_PPG.h, [27](#)
AFE_compute_CF
 AFE_Functions_PPG.c, [13](#)
 AFE_Functions_PPG.h, [28](#)
AFE_config_AMBCancellation
 AFE_Functions_PPG.c, [14](#)
 AFE_Functions_PPG.h, [28](#)
AFE_config_clockMode
 AFE_Functions_PPG.c, [14](#)
 AFE_Functions_PPG.h, [29](#)
AFE_config_LEDCancellation
 AFE_Functions_PPG.h, [29](#)
AFE_config_phaseTimingScheme
 AFE_Functions_PPG.c, [14](#)
 AFE_Functions_PPG.h, [29](#)
AFE_config_regMapInit
 AFE_Functions_PPG.c, [15](#)
 AFE_Functions_PPG.h, [30](#)
AFE_configAFE_forHRM
 AFE_Functions_PPG.h, [30](#)
 HRM_Configuration_Example.c, [50](#)
AFE_configAFE_forSpO2
 AFE_Functions_PPG.h, [30](#)
 SpO2_Configuration_Example.c, [51](#)
AFE_configTIA
 AFE_Functions_PPG.c, [15](#)
 AFE_Functions_PPG.h, [31](#)
AFE_disableTimingEngine
 AFE_Functions_PPG.c, [16](#)
 AFE_Functions_PPG.h, [31](#)
AFE_enableTimingEngine
 AFE_Functions_PPG.c, [16](#)
 AFE_Functions_PPG.h, [32](#)
AFE_Functions_PPG.c
 AFE_clearPPM, [13](#)
 AFE_compute_CF, [13](#)
 AFE_config_AMBCancellation, [14](#)
 AFE_config_clockMode, [14](#)
 AFE_config_phaseTimingScheme, [14](#)
 AFE_config_regMapInit, [15](#)
 AFE_configTIA, [15](#)
 AFE_disableTimingEngine, [16](#)
 AFE_enableTimingEngine, [16](#)
 AFE_initializeAFE, [16](#)
 AFE_set_AUTO_AMB_INSERT, [17](#)
 AFE_set_CF_TIA, [17](#)
 AFE_set_ENABLE_DRE, [17](#)
 AFE_set_FIFO_DATA_CTRL, [18](#)
 AFE_set_FILTER_BW, [18](#)
 AFE_set_ILED_DRVx, [19](#)
 AFE_set_ILED_FS, [19](#)
 AFE_set_IN_TIA, [20](#)
 AFE_set_INT_MUX_ADC_RDY, [20](#)
 AFE_set_LED_DC_EN_TIA, [20](#)
 AFE_set_LED_DRVx_TXN, [21](#)
 AFE_set_NUMAV, [21](#)
 AFE_set_PDNAFE, [22](#)
 AFE_set_phaseType, [22](#)
 AFE_set_PRPCT, [23](#)
 AFE_set_REG_SCALE_DRE, [23](#)
 AFE_set_REG_TWLED, [23](#)
 AFE_set_RF_TIA, [24](#)
 AFE_set_SW_RESET, [24](#)
AFE_Functions_PPG.h
 AFE_clearPPM, [27](#)
 AFE_compute_CF, [28](#)
 AFE_config_AMBCancellation, [28](#)
 AFE_config_clockMode, [29](#)
 AFE_config_LEDCancellation, [29](#)
 AFE_config_phaseTimingScheme, [29](#)
 AFE_config_regMapInit, [30](#)
 AFE_configAFE_forHRM, [30](#)
 AFE_configAFE_forSpO2, [30](#)
 AFE_configTIA, [31](#)
 AFE_disableTimingEngine, [31](#)
 AFE_enableTimingEngine, [32](#)
 AFE_initializeAFE, [32](#)
 AFE_set_AUTO_AMB_INSERT, [32](#)
 AFE_set_CF_TIA, [33](#)
 AFE_set_ENABLE_DRE, [33](#)
 AFE_set_FIFO_DATA_CTRL, [34](#)
 AFE_set_FILTER_BW, [34](#)
 AFE_set_ILED_DRVx, [34](#)
 AFE_set_ILED_FS, [35](#)
 AFE_set_IN_TIA, [35](#)
 AFE_set_INT_MUX_ADC_RDY, [36](#)
 AFE_set_LED_DC_EN_TIA, [36](#)
 AFE_set_LED_DRVx_TXN, [37](#)
 AFE_set_NUMAV, [37](#)
 AFE_set_PDNAFE, [37](#)
 AFE_set_phaseType, [38](#)
 AFE_set_PRPCT, [38](#)
 AFE_set_REG_SCALE_DRE, [39](#)

- AFE_set_REG_TWLED, 39
- AFE_set_RF_TIA, 39
- AFE_set_SW_RESET, 41
- AFE_HostControl.c
 - AFE_readReg, 42
 - AFE_writeReg, 42
- AFE_HostControl.h
 - AFE_readReg, 43
 - AFE_writeReg, 44
- AFE_initializeAFE
 - AFE_Functions_PPG.c, 16
 - AFE_Functions_PPG.h, 32
- AFE_modifyRegGlobal
 - AFE_RegMap.c, 45
 - AFE_RegMap.h, 48
- AFE_modifyRegPPM
 - AFE_RegMap.c, 46
 - AFE_RegMap.h, 49
- AFE_readReg
 - AFE_HostControl.c, 42
 - AFE_HostControl.h, 43
- AFE_RegMap, 5
- AFE_RegMap.c
 - AFE_assignRegMap, 45
 - AFE_modifyRegGlobal, 45
 - AFE_modifyRegPPM, 46
 - AFE_switchPage, 46
- AFE_RegMap.h
 - AFE_assignRegMap, 48
 - AFE_modifyRegGlobal, 48
 - AFE_modifyRegPPM, 49
 - AFE_switchPage, 49
- AFE_RegMap::GLOBAL, 5
- AFE_RegMap::PPM, 8
- AFE_set_AUTO_AMB_INSERT
 - AFE_Functions_PPG.c, 17
 - AFE_Functions_PPG.h, 32
- AFE_set_CF_TIA
 - AFE_Functions_PPG.c, 17
 - AFE_Functions_PPG.h, 33
- AFE_set_ENABLE_DRE
 - AFE_Functions_PPG.c, 17
 - AFE_Functions_PPG.h, 33
- AFE_set_FIFO_DATA_CTRL
 - AFE_Functions_PPG.c, 18
 - AFE_Functions_PPG.h, 34
- AFE_set_FILTER_BW
 - AFE_Functions_PPG.c, 18
 - AFE_Functions_PPG.h, 34
- AFE_set_ILED_DRVx
 - AFE_Functions_PPG.c, 19
 - AFE_Functions_PPG.h, 34
- AFE_set_ILED_FS
 - AFE_Functions_PPG.c, 19
 - AFE_Functions_PPG.h, 35
- AFE_set_IN_TIA
 - AFE_Functions_PPG.c, 20
 - AFE_Functions_PPG.h, 35

- AFE_set_INT_MUX_ADC_RDY
 - AFE_Functions_PPG.c, 20
 - AFE_Functions_PPG.h, 36
- AFE_set_LED_DC_EN_TIA
 - AFE_Functions_PPG.c, 20
 - AFE_Functions_PPG.h, 36
- AFE_set_LED_DRVx_TXN
 - AFE_Functions_PPG.c, 21
 - AFE_Functions_PPG.h, 37
- AFE_set_NUMAV
 - AFE_Functions_PPG.c, 21
 - AFE_Functions_PPG.h, 37
- AFE_set_PDNAFE
 - AFE_Functions_PPG.c, 22
 - AFE_Functions_PPG.h, 37
- AFE_set_phaseType
 - AFE_Functions_PPG.c, 22
 - AFE_Functions_PPG.h, 38
- AFE_set_PRPCT
 - AFE_Functions_PPG.c, 23
 - AFE_Functions_PPG.h, 38
- AFE_set_REG_SCALE_DRE
 - AFE_Functions_PPG.c, 23
 - AFE_Functions_PPG.h, 39
- AFE_set_REG_TWLED
 - AFE_Functions_PPG.c, 23
 - AFE_Functions_PPG.h, 39
- AFE_set_RF_TIA
 - AFE_Functions_PPG.c, 24
 - AFE_Functions_PPG.h, 39
- AFE_set_SW_RESET
 - AFE_Functions_PPG.c, 24
 - AFE_Functions_PPG.h, 41
- AFE_switchPage
 - AFE_RegMap.c, 46
 - AFE_RegMap.h, 49
- AFE_writeReg
 - AFE_HostControl.c, 42
 - AFE_HostControl.h, 44

C:/Old Laptop/C Drive/A0229988/Desktop/AEF4432/Driver Development/AFE4432_Driver_V1/AFE4432 Driver V1 Source Files/AFE_Functions_PPG.c, 11

C:/Old Laptop/C Drive/A0229988/Desktop/AEF4432/Driver Development/AFE4432_Driver_V1/AFE4432 Driver V1 Source Files/AFE_Functions_PPG.h, 25

C:/Old Laptop/C Drive/A0229988/Desktop/AEF4432/Driver Development/AFE4432_Driver_V1/AFE4432 Driver V1 Source Files/AFE_HostControl.c, 41

C:/Old Laptop/C Drive/A0229988/Desktop/AEF4432/Driver Development/AFE4432_Driver_V1/AFE4432 Driver V1 Source Files/AFE_HostControl.h, 43

C:/Old Laptop/C Drive/A0229988/Desktop/AEF4432/Driver Development/AFE4432_Driver_V1/AFE4432 Driver V1 Source Files/AFE_RegMap.c, 44

C:/Old Laptop/C Drive/A0229988/Desktop/AEF4432/Driver
Development/AFE4432_Driver_V1/AFE4432
Driver V1 Source Files/AFE_RegMap.h, [46](#)

C:/Old Laptop/C Drive/A0229988/Desktop/AEF4432/Driver
Development/AFE4432_Driver_V1/AFE4432
Driver V1 Source Files/HRM_Configuration_Example.c,
[49](#)

C:/Old Laptop/C Drive/A0229988/Desktop/AEF4432/Driver
Development/AFE4432_Driver_V1/AFE4432
Driver V1 Source Files/SpO2_Configuration_Example.c,
[50](#)

HRM_Configuration_Example.c
AFE_configAFE_forHRM, [50](#)

sigParameter, [9](#)

SpO2_Configuration_Example.c
AFE_configAFE_forSpO2, [51](#)